Chapter: Dimensionality reduction

Solved exercises

October 23, 2019

Contents

1 Notations

- If $A \in \mathbb{R}^{n \times m}$, then A_i (or a_i) is the i$^{\text{th}}$ column and $(A^\top)_i$ is the i$^{\text{th}}$ row as a column vector.
- If we have a data matrix X, we dispose the observations as columns, each row being an attribute. MUST REWRITE THE QUESTIONS IN EXERCISES IF IT'S OTHERWISE

2 Prerequisites

2.1 Linear Algebra in $\mathbb{R}^n$ Prerequisites

Solution: check the specific chapter in DP book or google the answer

2.1.1 Basics, Linear ..., Orthogonal ...

1. (CMU, NBalcan, fall 2017, ex.1.1) Properties of invertible matrix

   An $n \times n$ matrix A is said to be invertible, provided there exists another $n \times n$ matrix B such that

   $$AB = BA = I$$

   Please indicate Truw or False to the properties listed below. No further explanations are needed.

   (a) If A is invertible, so is $A^\top$, and $(A^\top)^{-1} = (A^{-1})^\top$.

   (b) If A is invertible and c is a nonzero scalar, then cA is invertible and $(cA)^{-1} = c^{-1}A^{-1}$.

   (c) If A and B both are invertible then so is AB and $(AB)^{-1} = A^{-1}B^{-1}$.

   (d) A is invertible if and only if $\text{rank}(A = n)$.

   (e) A is invertible if and only if $\det(A) \geq 0$.

   (f) A is invertible if and only if all its row vectors are linearly independent, and all its column vectors are linearly independent.

   (g) A has pseudo inverse if and only if it has linearly independent rows.

   Solution: T T F T F T F

2. (LA4ML. Review Packet 2, ex.1) What is the inverse of a diagonal matrix, $D = \text{diag}\{\sigma_1, \ldots, \sigma_r\}$?

   Solution: $D^{-1} = \text{diag}\{\frac{1}{sigma_1}, \frac{1}{\sigma_2}, \ldots, \frac{1}{\sigma_r}\}$

3. (LA4ML. Review Packet 2, ex.2) What is the effect of multiplying a matrix, X, by a diagonal matrix on the right (as in XD)? But on the left?

   Solution:

   XD - scales the columns of X by corresponding diagonal element of D

   DT - scales the rows of X

4. (LA4ML. Review Packet 2, ex.3) Combining the previous two problems, what happens when we multiply a data matrix, X, by D^{-1} on the right if $D = \text{diag}\{\sigma_1, \ldots, \sigma_r\}$ (as in XD^{-1})? But on the left?

   Solution: It would divide each element in column j by σ_j (This is what standardization/normalization looks like)

5. (LA4ML. Review Packet 2, ex.4) For a general matrix $A \in \mathbb{R}^{m \times n}$ describe what the following products will provide. Also give the size of the result (i.e., "$n \times 1$ vector" or "scalar"). Hint: If you cannot see these effects in the general sense, try using a simple 3×3 matrix A as an example first.

   (a) Ae_j
   (b) $e_i^\top A$
   (c) $e_i^\top A e_j$
   (d) Ae
   (e) $e^\top A$
   (f) $\frac{1}{m} e^\top A$

   Solution:

   (a) $(m \times 1)$ - j-th column of A
   (b) $(1 \times n)$ - i-th row of A
   (c) (scalar) - A_{ij}
   (d) $(m \times 1)$ - row sums of A
   (e) $(1 \times n)$ - column sums of A
   (f) $(1 \times n)$ - column averages of A

6. Compute the **determinant** and the **trace** of the following matrices: $M_1 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $M_2 = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$.

7. Given $u = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ and $v = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$ compute the **dot product** (i.e., the Euclidian inner product; $\cdot$), the norm induced by this inner product (i.e., 2-norm or **Euclidian norm**), the **distance** induced by this norm (i.e., 2-norm distance or **Euclidian distance**), the **outer product** ($\otimes$) between u and v.

8. (LA4ML. Review Packet 2. ex. 11a) Determine the **unit vector** that points in the same direction as the following vector: $\begin{bmatrix}1\\1\end{bmatrix}$.

9. (DP) Determine whether the given matrix is in **row echelon form** (REF). If it is, state whether it is also in **reduced row echelon form** (RREF):

$M_1 = \begin{bmatrix}1 & 0 & 1\\0 & 0 & 3\\0 & 1 & 0\end{bmatrix}$, $M_2 = \begin{bmatrix}7 & 0 & 1 & 0\\0 & 1 & -1 & 4\\0 & 0 & 0 & 0\end{bmatrix}$, $M_3 = \begin{bmatrix}0 & 1 & 3 & 0\\0 & 0 & 0 & 1\end{bmatrix}$, $M_4 = \begin{bmatrix}0 & 0 & 0\\0 & 0 & 0\\0 & 0 & 0\end{bmatrix}$, $M_5 = \begin{bmatrix}1 & 0 & 3 & -4 & 0\\0 & 0 & 0 & 0 & 0\\0 & 1 & 5 & 0 & 1\end{bmatrix}$, $M_6 = \begin{bmatrix}0 & 0 & 1\\0 & 1 & 0\\1 & 0 & 0\end{bmatrix}$, $M_7 = \begin{bmatrix}1 & 2 & 3\\1 & 0 & 0\\0 & 1 & 1\\0 & 0 & 1\end{bmatrix}$, $M_8 = \begin{bmatrix}2 & 1 & 3 & 5\\0 & 0 & 1 & -1\\0 & 0 & 0 & 3\\0 & 0 & 0 & 0\end{bmatrix}$.

Observation: The word echelon comes from the Latin word scala, meaning "ladder" or "stairs" The French word for "ladder" echelle, is also derived from this Latin base. A matrix in echelon form exhibits a staircase pattern.

10. (DP) Use elementary row operations to reduce the given matrix to **row echelon form** and, then, to **reduced row echelon form**.

$M_1 = \begin{bmatrix}0 & 0 & 1\\0 & 1 & 1\\1 & 1 & 1\end{bmatrix}$, $M_2 = \begin{bmatrix}4 & 3\\2 & 1\end{bmatrix}$, $M_3 = \begin{bmatrix}3 & 5\\5 & -2\\2 & 4\end{bmatrix}$, $M_4 = \begin{bmatrix}2 & -4 & -2 & 6\\3 & 1 & 6 & 6\end{bmatrix}$, $M_5 = \begin{bmatrix}3 & -2 & -1\\2 & -1 & -1\\4 & -3 & -1\end{bmatrix}$, $M_6 = \begin{bmatrix}-2 & -4 & 7\\-3 & -6 & 10\\1 & 2 & -3\end{bmatrix}$.

11. (DP) Solve the given system of equations using either **Gaussian or Gauss-Jordan elimination**, write the linear system in **parametric form** and specify the **free and pivot variables**:

$(S_1): \begin{cases} x_1 + 2x_2 - 3x_3 = 9\\ 2x_1 - x_2 + x_3 = 0\\ 4x_1 - x_2 + x_3 = 4 \end{cases}$, $(S_2): \begin{cases} x - y + z = 0\\ -x + 3y + z = 5\\ 3x + y + 7z = 2 \end{cases}$, $(S_3): \begin{cases} x_1 - 3x_2 - 2x_3 = 0\\ -x_1 + 2x_2 + x_3 = 0\\ 2x_1 + 4x_2 + 6x_3 = 0 \end{cases}$,

$(S_4): \begin{cases} 2w + 3x - y + 4z = 1\\ 3w - x + z = 1\\ 3w - 4x + y - z = 2 \end{cases}$, $(S_5): \begin{cases} 2r + s = 3\\ 4r + s = 7\\ 2r + 5s = -1 \end{cases}$

12. (DP)

    (a) Compute an **LU decomposition** (in two forms: **reduced (also called economy sized or thin)** and **full**) with $l_{ii} = 1$ of the following matrices and mention if it is **unique**: $M_1 = \begin{bmatrix} 1 & 2 \\ -3 & -1 \end{bmatrix}$, $M_2 = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 8 & 7 & 9 \end{bmatrix}$, $M_3 = \begin{bmatrix} 1 & 0 & 1 & -2 \\ 0 & 3 & 3 & 1 \\ 0 & 0 & 0 & 5 \end{bmatrix}$, $M_4 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 1 & 3 & 0 \\ -2 & 1 & 5 \end{bmatrix} = M_3^\top$, $M_5 = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$

    (b) Can you compute an LU decomposition of the following matrix? If not, compute a $P^\top LU$ **decomposition** of it: $M = \begin{bmatrix} 0 & 1 & 4 \\ -1 & 2 & 1 \\ 1 & 3 & 3 \end{bmatrix}$.

13. (DP)

    (a) Determine if the vector v is a **linear combination** of the remaining vectors: $v = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $u_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, $u_2 = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$

    (b) Determine whether v is in the **span** of the remaining vectors: $v = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}$, $u_1 = \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix}$, $u_2 = \begin{bmatrix} -1 \\ 1 \\ -3 \end{bmatrix}$.

14. (LA4ML. Chapter 3. Exercises. ex.1) **Six views of a matrix multiplication**

    Let $A \in \mathbb{R}^{m \times k}, B \in \mathbb{R}^{k \times n}, C \in \mathbb{R}^{m \times n}$ be matrices such that $AB = C$.

    (a) Express the first column of C as a linear combination of the columns of A.

    (b) Express the first column of C as a matrix-vector product.

    Observation: One can view the matrix vector multiplication $Ax = A(x)$ as a function/transformation (A) applied to the vector (x). Also: $AX = [A(x_1)|\ldots|A(x_n)]$.

(c) Express C as a sum of outer products.

(d) Express the first row of C as a linear combination of the rows of B.

(e) Express the first row of C as a matrix-vector product.

(f) Express the element C_{ij} as an inner product of row or column vectors from A and B.

Hint: If you cannot answer to a question, try using simple matrices (3×3, 2×2, 3×2, etc.) as an example first.

Observations that should be read after knowing come notions as span, column space, row space, dimension, rank.

Observation: $Ax = b \Rightarrow b \in \text{col}(A)$ and

$A^\top x = b$ (or $x^\top A = b^\top$) $\Rightarrow b \in \text{row}(A)$

Observation 2: The observation above in terms of our problem notations: $AB_i = C_i \Rightarrow b \in \text{col}(A)$ and

$B^\top (A^\top)_i = (C^\top)_i$ (or $((A^\top)_i)^\top B = ((C^\top)_i)^\top) \Rightarrow b \in \text{row}(A)$.

Observation 3: $AB = C \Rightarrow \text{col}(A) \subseteq \text{col}(B)$ (not equal!) and

$AB = C \Rightarrow \text{row}(A) \subseteq \text{row}(C)$ (not equal!)

Observation 4 (very important from a Machine Learning perspective; **for a concrete example, go to MF_prereq_solved.tex and apply what you learn from this exercise there**): In an informal, but very **intuitive**, way the matrix multiplication $AB = C$ can be read:

- by **columns**: the i$^{\text{th}}$ column of A is the sum of the columns of B (we can call the columns of B ***canonical*** columns), each multiplied by the numbers on the i$^{\text{th}}$ column of C (we can call the columns of C ***coordinate*** columns)
- by **rows**: the i$^{\text{th}}$ row of A is the sum of the rows of C (we can call the rows of C ***canonical*** rows), each multiplied by the numbers on the i$^{\text{th}}$ row of B (we can call the rows of B ***coordinate*** rows)

Observation 5:

Let $A \in \mathbb{R}^{m \times n}$. Prove that $\dim(\text{col}(A)) = \dim(\text{row}(A))$ (i.e., $\text{rank}(A) = \text{rank}(A^\top)$).

Proof:

Let $\dim(\text{col}(A)) = c \Rightarrow \exists B \in \mathbb{R}^{m \times c}, C \in \mathbb{R}^{c \times n} : A = BC \Rightarrow \dim(\text{row}(A)) \leq c$.

Let $\dim(\text{row}(A)) = r \Rightarrow \exists D \in \mathbb{R}^{n\times r}, E \in \mathbb{R}^{r\times m} : A^\top = DE \Rightarrow \dim(\text{row}(A^\top)) \leq r$. But $\dim(\text{row}(A^\top)) = \dim(\text{col}(A)) = c \Rightarrow c \leq r$.

From above, we have $\dim(\text{row}(A)) \leq c$. But $\dim(\text{row}(A)) = r \Rightarrow r \leq c$.

From $c \leq r$ and $r \leq c$, we have $c = r$. So, $\dim(\text{col}(A)) = \dim(\text{row}(A))$.

15. (LA4ML. Review Packet 2. ex.10)

    Suppose I want to compute the matrix product $A = UDV^\top$ where U is $n \times r$, D is an $r \times r$ diagonal matrix, $D = \text{diag}\{\sigma_1, \ldots, \sigma_r\}$, and $V^\top$ is $r \times p$. (*Side note: we will quite often want to compute such a matrix product - this is the form of the singular value decomposition (SVD)! The following exercise is not just for fun - what you end up with in the second part of the exercise is exactly how we will want to write the SVD to best understand how it works.*)

    (a) Using what you know about multiplication by diagonal matrices, if we view the matrix U as a collection of columns,

    $$U = (u_1 | \ldots | u_r)$$

    then how would I write the same partition of the matrix UD?

    $$UD = (?| \ldots |?)$$

    Keep in mind that when multiplying matrices/vectors by scalars, it is always preferable to write the scalar first (σu rather than uσ).

    (b) Now, using the above representation for UD, what happens when I multiply it by the matrix $V^\top$, viewed as a collection of rows,

    $$V^\top = \begin{bmatrix} (v_1)^\top \\ \vdots \\ (v_r)^\top \end{bmatrix} ?$$

    (Hint: your answer should be a sum. Each term in the sum should be an outer product.)

    $$UDV^\top = ?$$

16. Prove that $\text{tr}(ABC) = \sum_{i=1}^{n} ((A^\top)_i)^\top BC_i$, where $A \in \mathbb{R}^{n\times m}$, $B \in \mathbb{R}^{m\times p}$, $C \in \mathbb{R}^{p\times n}$.

17. (DP) Determine whether the following sets of vectors are **linearly independent**. If, for any of these, the answer can be determined by inspection (i.e., without calculation), state why. For any sets that are **linearly dependent**, find a dependence relationship among the vectors.

(a) $\begin{bmatrix}10\\0\\0\\1\end{bmatrix}, \begin{bmatrix}0\\1\\0\\2\end{bmatrix}, \begin{bmatrix}0\\0\\1\\3\end{bmatrix}$

(b) $\begin{bmatrix}1\\10\\1\\-30\end{bmatrix}, \begin{bmatrix}1\\10\\0\\10\end{bmatrix}, \begin{bmatrix}2\\20\\0\\10\end{bmatrix}, \begin{bmatrix}-4\\-40\\-1\\10\end{bmatrix}$

(c) $\begin{bmatrix}1\\4\\1\end{bmatrix}, \begin{bmatrix}-1\\2\\1\end{bmatrix}$

(d) $\begin{bmatrix}1\\1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\1\end{bmatrix}, \begin{bmatrix}1\\0\\1\end{bmatrix},$

(e) $\begin{bmatrix}1\\2\\0\end{bmatrix}, \begin{bmatrix}1\\1\\-1\end{bmatrix}, \begin{bmatrix}1\\4\\2\end{bmatrix}$

(f) $\begin{bmatrix}-2\\3\\7\end{bmatrix}, \begin{bmatrix}4\\-1\\5\end{bmatrix}, \begin{bmatrix}3\\1\\3\end{bmatrix}, \begin{bmatrix}5\\0\\2\end{bmatrix}$

18. (DP) Show that the given vectors form an **orthogonal set**. Determine whether the set is orthonormal. If it is not, normalize the vectors to form an **orthonormal set**.

(a) $\begin{bmatrix}\frac{3}{5}\\\frac{4}{5}\end{bmatrix}, \begin{bmatrix}-\frac{4}{5}\\\frac{3}{5}\end{bmatrix}$

(b) $\begin{bmatrix}\frac{1}{3}\\\frac{2}{3}\\\frac{2}{3}\end{bmatrix}, \begin{bmatrix}\frac{2}{3}\\-\frac{1}{3}\\0\end{bmatrix}, \begin{bmatrix}1\\2\\-\frac{5}{2}\end{bmatrix}$

19. (DP) Determine whether the given **matrix is orthogonal**. If it is, find its inverse.

(a) $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$

(b) $\begin{bmatrix} \frac{1}{3} & \frac{1}{2} & \frac{1}{5} \\ \frac{1}{3} & -\frac{1}{2} & \frac{1}{5} \\ -\frac{1}{3} & 0 & \frac{2}{5} \end{bmatrix}$

Observation: Orthogonal matrix is an unfortunate bit of terminology. "Orthonormal matrix" would clearly be a better term, but it is not standard. Moreover, there is no term for a nonsquare with orthonormal columns.

20. (DP) Determine whether the following vectors form a **basis** of a (sub)space. If the answer is positive, mention the corresponding **(sub)space**.

(a) $\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$

(b) $\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

(c) $\begin{bmatrix} 1 \\ -1 \\ 3 \end{bmatrix}, \begin{bmatrix} -1 \\ 5 \\ 1 \end{bmatrix}$

(d) $\begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \end{bmatrix}$.

21. (DP) Find the **coordinate vector** of $v = \begin{bmatrix} 1 \\ 6 \\ 2 \end{bmatrix}$ with respect to the basis $\mathcal{B} = \left\{ \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \right\}$.

22. (Gareth Williams) Find the **transition matrix** P from the given basis $\mathcal{B}$ to the basis $\mathcal{B}'$ of $\mathbb{R}^2$ (**change of basis**). Use the matrix to find the coordinate vector of u relative to $\mathcal{B}'$.

(a) $\mathcal{B} = \left\{ \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right\}, \mathcal{B}' = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}, [u]_{\mathcal{B}} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

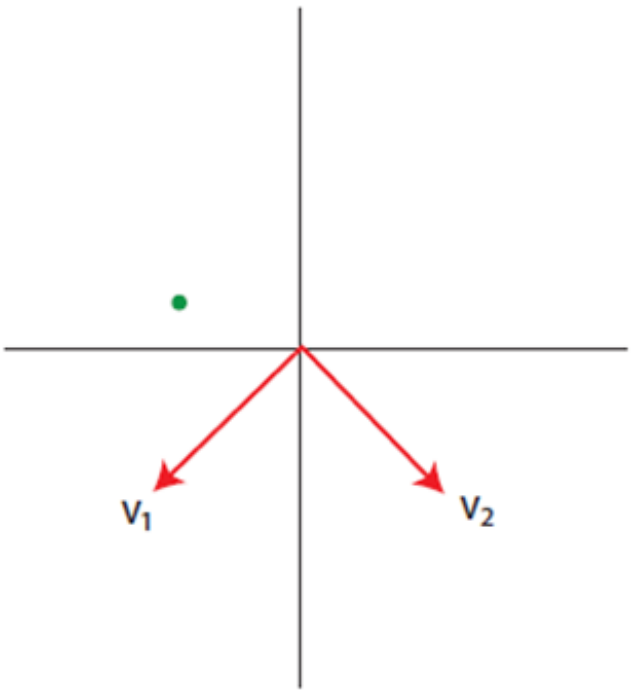
(b) $\mathcal{B} = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}, \mathcal{B}' = \left\{ \begin{bmatrix} 5 \\ 3 \end{bmatrix}, \begin{bmatrix} 3 \\ 2 \end{bmatrix} \right\}, [u]_{\mathcal{B}} = \begin{bmatrix} -2 \\ 5 \end{bmatrix}$

(c) $\mathcal{B} = \left\{ \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 3 \\ 0 \end{bmatrix} \right\}, \mathcal{B}' = \left\{ \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 3 \\ 2 \end{bmatrix} \right\}, [u]_{\mathcal{B}} = \begin{bmatrix} -3 \\ 2 \end{bmatrix}$

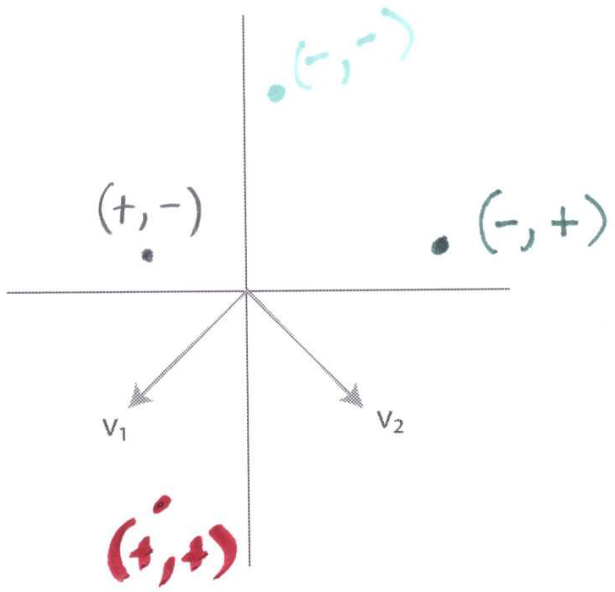
23. (LA4ML) What are the coordinates of the vector $x = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$ in the (orthogonal) basis $\left\{ \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\}$? **Draw** a picture to make sure your answer lines up with intuition.

24. (LA4ML) Compute the coordinates of the point $a = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$ in the (non-orthogonal) basis $\left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\}$. **Draw** a picture which shows the point a and the new basis vectors on a coordinate plane.

25. (LA4ML. Introduction to Vector Space Models - Worksheet. Part Two. ex.2) In the following **picture** what would be the signs (+/-) of the coordinates of the green point in the basis $\{v_1, v_2\}$? Pick another point at random and answer the same question for that point.



Solution:



26. (DP) Find a **basis for the span** of the given vectors: $\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}$.

27. (DP) Find a **basis for $\mathbb{R}^3$ that contains** the vector $\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$.

28. (DP)

    (a) Show that $\mathbb{R}^2 = \text{span}\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}\right)$.

    (b) Show that $\mathbb{R}^3 = \text{span}\left(\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}\right)$.

29. (DP) Find the **orthogonal complement** $W^\top$ of W and give a basis for $W^\top$.

    (a) $W = \left\{ \begin{bmatrix} x \\ y \\ z \end{bmatrix} : x + y - z = 0 \right\}$

(b) $W = \left\{ \begin{bmatrix} x \\ y \\ z \end{bmatrix} : x = t, y = -t, z = 3t \right\}$

(c) $W = \text{span}\left(\begin{bmatrix} 1 \\ -1 \\ 3 \\ -2 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -2 \\ 1 \end{bmatrix} \right)$

(d) $W = \left\{ \begin{bmatrix} x \\ 2x - y \\ 3x + 4y \end{bmatrix} : \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}^2 \right\}$

Observation: The exercise also wanted to highlight the many ways in which one can specify a linear (sub)space: **via span, via linear hyperplane equation, and in a linear-transformation fashion**.

Observation 2: An **intuitive** example of the latter way (i.e., linear-transformation fashion) where we substituted x, y, z etc. with intuitive names (here: the price of a vegetable/fruit):

$$\begin{bmatrix} \text{orange} \\ \text{cucumber} \\ \text{potato} \\ \text{apple} \\ \text{onion} \end{bmatrix} \to \begin{bmatrix} \frac{1}{2}\text{orange} + \frac{1}{2}\text{apple} \\ \frac{1}{3}\text{cucumber} + \frac{1}{3}\text{potato} + \frac{1}{3}\text{onion} \end{bmatrix} \overset{\text{not.}}{=} \begin{bmatrix} \text{mean price of a fruit} \\ \text{mean price of a vegetable} \end{bmatrix}$$

In a ML context, *orange*, *cucumber*, *potato* etc. are the names of the **attributes** of our data.

30. (DP) Find the **matrix of the orthogonal projection** onto the subspace W. Then use this matrix to find the **orthogonal projection** of v onto W ($\text{proj}_W(v)$) and the **component of v orthogonal to W** ($\text{perp}_W(v)$).

    Observation:

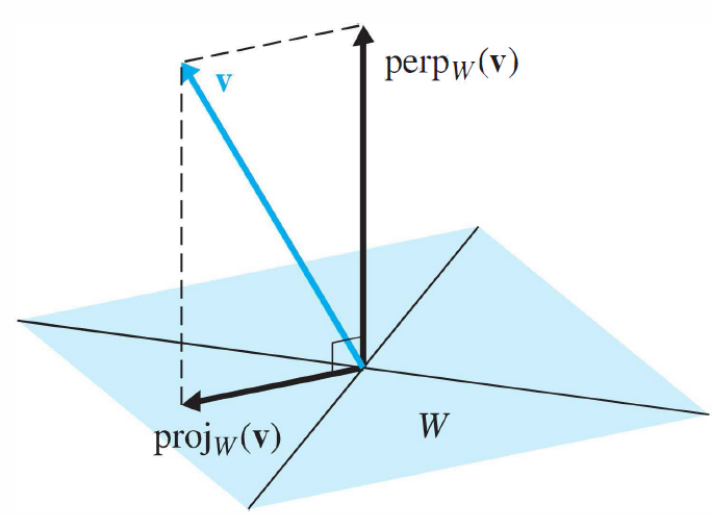


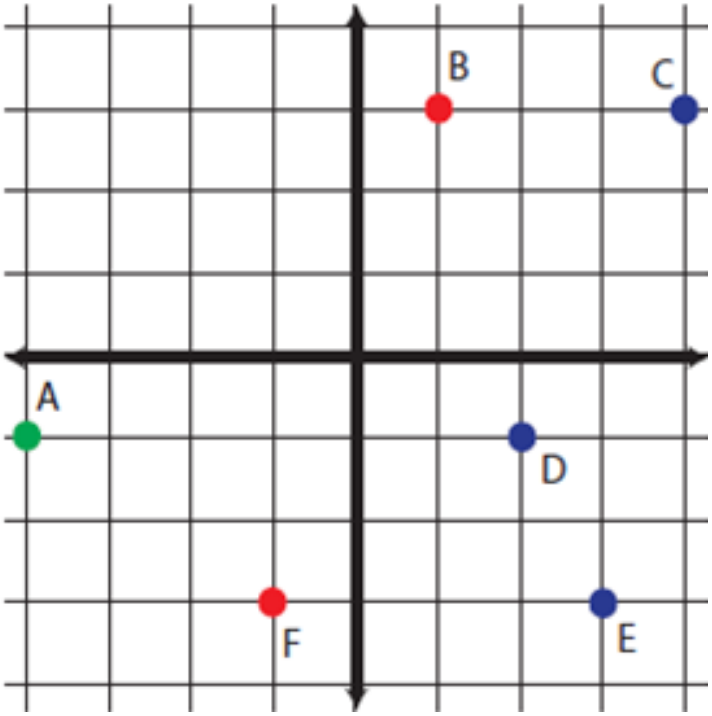
Figure 5.9
$\mathbf{v} = \text{proj}_W(\mathbf{v}) + \text{perp}_W(\mathbf{v})$

(a) $W = \text{span}\left(\begin{bmatrix}1\\1\end{bmatrix}\right), v = \begin{bmatrix}3\\4\end{bmatrix}$

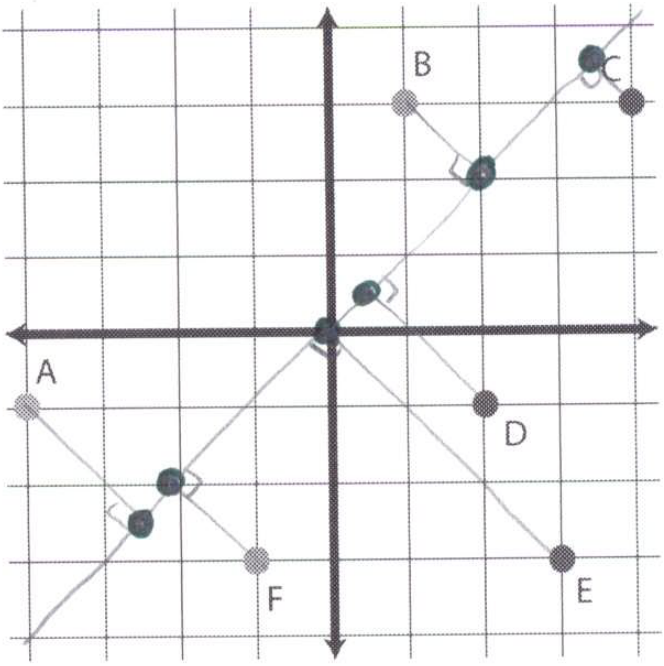
(b) $W = \text{span}\left(\begin{bmatrix}1\\1\\0\end{bmatrix}, \begin{bmatrix}-1\\1\\1\end{bmatrix}\right), v = \begin{bmatrix}3\\-1\\2\end{bmatrix}$

Observation: the matrix of the orthogonal projection onto W does not depend on the choice of basis. Take $\left(\begin{bmatrix}1\\1\\0\end{bmatrix}, \begin{bmatrix}1\\3\\1\end{bmatrix}\right)$ as a basis for W and repeat the calculations to show that the resulting projection matrix is the same.

31. (LA4ML. Orthogonality Worksheet. Part Two. ex.3) **Draw** the orthogonal projection of the points onto the subspace span($\left(\begin{bmatrix}1\\1\end{bmatrix}\right)$.



Solution:



32. (DP) Find the **orthogonal decomposition** of v with respect to W: $v = \begin{bmatrix} 4 \\ -2 \\ 3 \end{bmatrix}, W = \text{span}\left(\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \right)$.

33. (DP) Apply the **Gram-Schmidt Process** to the following basis to obtain an orthogonal (or even orthonormal) basis: $\mathcal{B} = \left\{ \begin{bmatrix} 2 \\ 1 \\ -2 \end{bmatrix}, \begin{bmatrix} 8 \\ 7 \\ -2 \end{bmatrix}, \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix} \right\}$.

34. (DP) Compute a **QR decomposition** (in two forms: **reduced (also called economy sized or thin)** and **full**) of the following matrices and mention if it is **unique**:

$$M_1 = \begin{bmatrix} 2 & 8 & 2 \\ 1 & 7 & -1 \\ -2 & -2 & 1 \end{bmatrix}, \; M_2 = \begin{bmatrix} 1 & 3 \\ 2 & 4 \\ -1 & -1 \\ 0 & 1 \end{bmatrix}, \; M_3 = \begin{bmatrix} 1 & 2 & -1 & 0 \\ 3 & 4 & -1 & 1 \end{bmatrix} = M_2^\top,$$

$$M_4 = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

35. (DP) Fill in the missing entries of Q to make Q an orthogonal matrix:

$$Q = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & * \\ 0 & \frac{1}{\sqrt{3}} & * \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & * \end{bmatrix}$$

36. (DP) The given vectors form a basis (a subspace of $\mathbb{R}^n$). Obtain an **orthogonal basis** and an **orthonormal** basis.

(a) $\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$

(b) $\begin{bmatrix} 1 \\ 1 \\ -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \\ 1 \\ 5 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1 \end{bmatrix}$

37. (DP + Gareth Will) Let $A = \begin{bmatrix} 1 & 1 & 3 & 1 & 6 \\ 2 & -1 & 0 & 1 & -1 \\ -3 & 2 & 1 & -2 & 1 \\ 4 & 1 & 6 & 1 & 3 \end{bmatrix}$. The **four funda-**

mental subspaces of A are column space of A (col(A)), row space of A (row(A)), null space of A (null(A)), null space of $A^\top$ (null($A^\top$)).

(a) i. Find k such that null(A) is a subspace of $\mathbb{R}^k$.
ii. Find k such that col(A) is a subspace of $\mathbb{R}^k$.

(b) Find bases for the four fundamental subspaces of A.

(c) i. Verify that every vector in row(A) is orthogonal to every vector in null(A).
ii. Verify that every vector in col(A) is orthogonal to every vector in null($A^\top$).

(d) Determine the **rank** and **nullity** of A.
Observation: By definition, rank(A) = dim(col(A)). It can be proven that rank(A) = dim(row(A)) (see the observations in the problem with **Six views of a matrix multiplication**)

2.1.2 Eigen...

38. (DP)

(a) Show that $x = \begin{bmatrix}1\\1\end{bmatrix}$ is an **eigenvector** of $A = \begin{bmatrix}3 & 1\\1 & 3\end{bmatrix}$ and find the corresponding **eigenvalue**.

(b) Show that 5 is an eigenvalue of $A = \begin{bmatrix}1 & 2\\4 & 3\end{bmatrix}$ and determine all eigenvectors corresponding to this eigenvalue. Find a basis for its **eigenspace**.
Observation: The **Rayleigh Quotient** may be helpful.

39. (after many sources; the most important ex. for PCA) For any of the following matrices answer to the questions:

(a) Compute the **eigenvalues** (Write down the **characteristic polynomial** and solve the **characteristic equation**). Are the eigenvalues in $\mathbb{R}$?

(b) Compute the **eigenspaces** (E_λ).

(c) Take one vector from each eigenspace and form a set. Verify that the set is **linearly independent**.

(d) Only for matrices that have only real eigenvalues and eigenvectors, take one vector from each eigenspace and form a set. Verify that the set is **orthogonal**.

Observation: If a set of vectors is orthogonal, then the set is also linearly independent. (**orthogonality ⇒ linear independence**)

(e) Compute the **algebraic and geometric multiplicities** of the eigenvalues.

(f) Is the matrix **diagonalizable**? If yes, compute an **eigendecomposition (or spectral decomposition)** of the matrix, **compute the matrix raised to the power of 2019 (M_i^{2019})**, and write the **spectral decomposition in the outer product form**. Is the **decomposition unique**?

(You do not have to calculate inverses, just write $\ldots^{-1}$.)

Complex eigenvalues:

$M_1 = \begin{bmatrix} 3 & -2 \\ 4 & -1 \end{bmatrix}$; Hint: $\lambda_i \in \{1-2i, 1+2i\}$. Source: `http://www.math.utk.edu/~freire/complex-eig2005.pdf`

$M_2 = \begin{bmatrix} -2 & -2 & -9 \\ -1 & 1 & -3 \\ 1 & 1 & 4 \end{bmatrix}$; Hint: $\lambda_i \in \{1, 1+i, 1-i\}$. Source: `http://bellomo.faculty.unlv.edu/Math365/Notes/Ch4-Sect06.pdf`

Algebraic multiplicity ≠ Geometric multiplicity:

$M_3 = \begin{bmatrix} 8 & -9 \\ 4 & -4 \end{bmatrix}$; Hint: $\lambda_i \in \{2\}$. Source: `https://people.math.carleton.ca/~kcheung/math/notes/MATH1107/wk10`

$M_4 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 4 \end{bmatrix}$; Hint: $\lambda_i \in \{1, 4\}$. Source: after `https://math.stackexchange.com/questions/132552/showing-a-matrix-is-not-diagonalizable`

Symmetric and non-symmetric matrices with 1, 2, 3 different eigenvalues:

$M_5 = \begin{bmatrix} 2 & -3 \\ -3 & 10 \end{bmatrix}$; Hint: $\lambda_i \in \{1, 11\}$. Source: DP

$M_6 = \begin{bmatrix} 4 & 0 & 2 \\ 0 & 4 & 0 \\ 2 & 0 & 4 \end{bmatrix}$; Hint: $\lambda_i \in \{2, 4, 6\}$. Source: Gareth Will

$M_7 = \begin{bmatrix} 1 & 2 \\ 3 & 2 \end{bmatrix}$; Hint: $\lambda_i \in \{-1, 4\}$. Source: https://people.math.carleton.ca/~kcheung/math/notes/MATH1107/wk10

$M_8 = \begin{bmatrix} 4 & 1 \\ 4 & 0 \end{bmatrix}$; Hint: $\lambda_i \in \{4\}$. Source: DP

$M_9 = \begin{bmatrix} 1 & 7 & -7 \\ -1 & 3 & -1 \\ -1 & -5 & 7 \end{bmatrix}$; Hint: $\lambda_i \in \{1, 2, 8\}$. Source: Gareth Will

$M_{10} = \begin{bmatrix} 2 & 2 & 2 \\ 0 & 2 & 0 \\ 0 & 1 & 3 \end{bmatrix}$; Hint: $\lambda_i \in \{2, 3\}$. Source: https://people.math.carleton.ca/~kcheung/math/notes/MATH1107/wk10

$M_{11} = \begin{bmatrix} 2 & 1 & -1 \\ -1 & 0 & 2 \\ -1 & -2 & 4 \end{bmatrix}$; Hint: $\lambda_i \in \{2\}$. Source: https://www.jirka.org/diffyqs/htmlver/diffyqsse24.html

$M_{12} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$

$M_{13} = \begin{bmatrix} 7 & 0 & 5 \\ 0 & 2 & 0 \\ 5 & 0 & 7 \end{bmatrix}$; Hint: $\lambda_i \in \{2, 12\}$. Source: Gareth Will

$M_{14} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

0 (zero) eigenvalue:

$M_{15} = \begin{bmatrix} 1 & 4 \\ 2 & 8 \end{bmatrix}$

40. (Radford, 2009f, final exam, ex.3.a) A symmetric 3×3 matrix, A, has the following eigenvectors (which are not necessarily of length one) and corresponding eigenvalues:

$$e_1 = [3, 0, 4]^\top, \lambda_1 = 100$$

$$e_2 = [0, 1, 0]^\top, \lambda_2 = 80$$

$$e_3 = [-4, 0, 3]^\top, \lambda_3 = 25$$

Write the matrix A, giving the actual numerical values of its elements.

41. (after DP)

(a) Use the **power method** to approximate the dominant eigenvalue and dominant eigenvector of A. Use the given initial vector x_0, the specified number of iterations k, and three-decimal-place accuracy.

i. $A = \begin{bmatrix} 9 & 4 & 8 \\ 4 & 15 & -4 \\ 8 & -4 & 9 \end{bmatrix}$, $x_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$, $k = 5$, with 2-norm scaling.

Observation: A is a **real symmetric matrix**. Real symmetric matrices are encountered in the PCA algorithm.

ii. $A = \begin{bmatrix} 0.3 & 0.5 \\ 0.7 & 0.5 \end{bmatrix}$, $x_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $k = 6$, without scaling.

Observation: A is a **positive stochastic matrix**. Positive stochastic matrices are encountered in the PageRank algorithm.

Observation: There are **extensions of the power method** algorithm to find the second, third, ... eigenvalues (and eigenvectors) sequentially via **deflation**. In order to find the first k eigenvalues (and eigenvectors) in a non-sequential manner one can use another immediate generalization of the power method algorithm: simultaneous iteration, also called, subspace iteration or **block power iteration**. In fact, the latter algorithm is equivalent to the famous **QR algorithm** which computes the eigenvalues and eigenvectors of a matrix.

(b) Regarding the above two matrices, compute the **percentage of the dominant eigenvalue from the sum of eigenvalues without computing the other eigenvalues**.

Observation: This could be helpful when computing the percentage of the explained variance within the PCA algorithm (without knowing all the eigenvalues).

42. (CMU, NBalcan, fall 2017, ex.1.2) Let A be a real $n \times n$ matrix with $A^\top = A$. We say that A is

(a) positive definite provided that

$$x^\top Ax > 0, \forall \in \mathbb{R}^n \text{ and } x \neq 0$$

(b) positive semidefinite provided that

$$x^\top Ax \geq 0, \forall x \in \mathbb{R}^n \text{ and } x \neq 0$$

Covariance matrix is a typical example of positive semidefinite matrix.

Assume A has all real entries and is positive semidefinite. What can you state about the eigenvalues of A?

Solution: All eigenvalues of A are non-negative.

2.1.3 Singular...

43. Find the **singular values** of $A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$.

Observation:

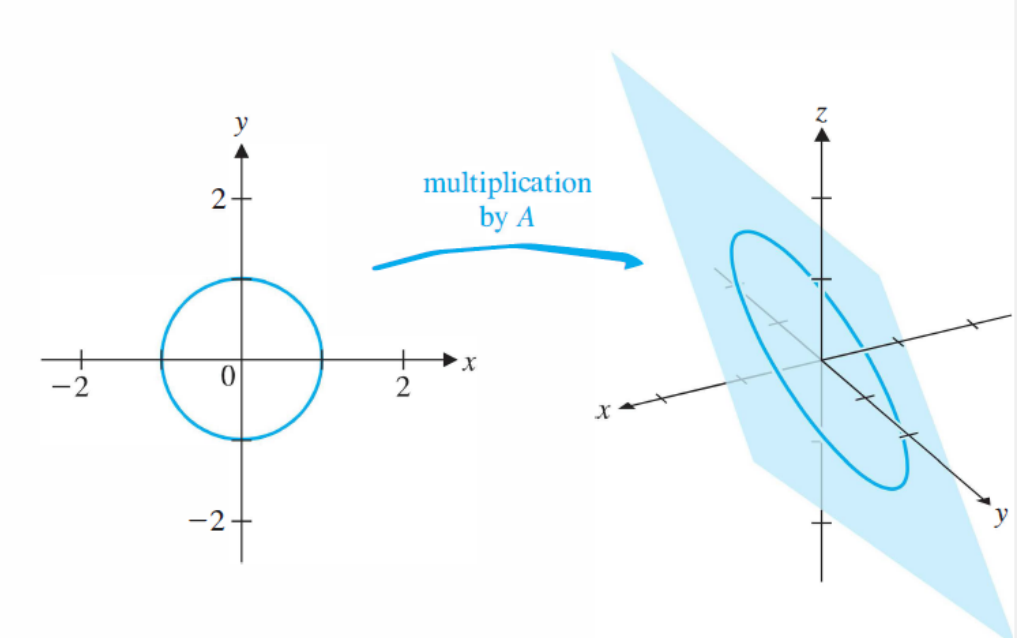


Figure 7.18
The matrix A transforms the unit circle in $\mathbb{R}^2$ into an ellipse in $\mathbb{R}^3$

44. (DP)

(a) Find a **singular value decomposition (SVD)** (in two forms: **reduced (also called economy sized or thin)** and **full**) for the following matrices and mention if it is **unique**.

i. $A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

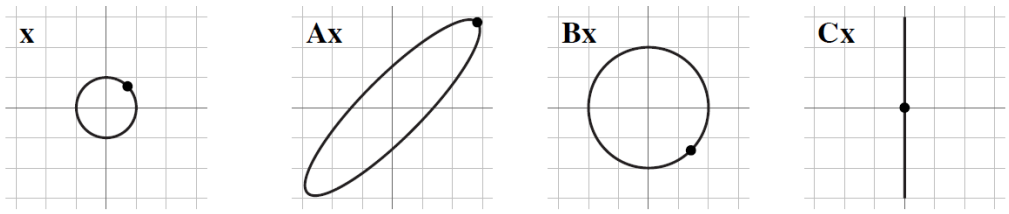
ii. $A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

iii. $A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$

(b) For each the matrices A above, mention:

- the **singular values**
- the **left singular vectors**
- the **right singular vectors**
- the **outer product form of the SVD**
- rank(A), nullity(A)
- an orthonormal basis for col(A)
- an orthonormal basis for row(A)
- an orthonormal basis for null(A)
- an orthonormal basis for null($A^\top$).

45. (Cornell University, CS 3220, Final Exam, ex.3 - svd/cs3220-2009sp-final.pdf) Consider the 2×2 matrices A, B, and C, which transform the unit circle into the following three shapes (the dot shows where a particular point is mapped to by each of the three transformations):



(a) The singular values of these matrices are all integers. What are the singular values of A, B, and C?

(b) Write down the SVD of each of these matrices, choosing from the following matrices for U and V:

$$(i)\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}; (ii)\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}; (iii)\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, (iv)\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

46. Compute the **eigendecomposition and the SVD** of the following matrix: $\begin{bmatrix} 2 & -3 \\ -3 & 10 \end{bmatrix}$. Are they **the same**?

Observation: The matrix is symmetric and positive definite. The same result is also valid for symmetric and positive semi-definite matrices.

Observation 2: If a matrix is symmetric, then $\sigma_i = |\lambda_i|$, where σ_i is a singular value of the matrix and λ_i is an eigenvalue of the matrix.

47. (DP) Let $u_1 = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}$, $u_2 = \begin{bmatrix} 5 \\ -2 \\ 1 \end{bmatrix}$, and $v = \begin{bmatrix} 3 \\ 2 \\ 5 \end{bmatrix}$. Find the **best approximation** to v in the plane $W = \text{span}(u_1, u_2)$ and find the Euclidian distance from v to W.

48. (DP)

    (a) Compute the **pseudoinverse (or Moore-Penrose inverse)** of the matrix A (i.e., A^+) via the **formula available only when A has linearly independent columns**.
    $A = \begin{bmatrix} 2 & 1 \\ 2 & 0 \\ 1 & 1 \end{bmatrix}$.
    Observation: If A had been square and invertible, then $A^{-1} = A^+$. For example, one can verify that $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^+$.
    Observation 2: The pseudoinverse can always (also in this case) be computed using the SVD. See the (e) point.

    (b) Find a **least squares solution** to the inconsistent system $Ax = b$ by constructing and solving the **normal equations**, where $A = \begin{bmatrix} 2 & 1 \\ 2 & 0 \\ 1 & 1 \end{bmatrix}$ and $b = \begin{bmatrix} 2 \\ 3 \\ -1 \end{bmatrix}$.

    (c) A **QR factorization** of A is given. Use it to find a **least squares solution** of $Ax = b$:
    $A = \begin{bmatrix} 2 & 1 \\ 2 & 0 \\ 1 & 1 \end{bmatrix}$, $Q = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{2}{3} & -\frac{2}{3} \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix}$, $R = \begin{bmatrix} 3 & 1 \\ 0 & 1 \end{bmatrix}$, $b = \begin{bmatrix} 2 \\ 3 \\ -1 \end{bmatrix}$.

    (d) Show that the **least squares solution of $Ax = b$ is not unique** and solve the normal equations to find all the least squares solutions,

where $A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$ and $b = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$.

(e) Compute the **pseudoinverse (or Moore-Penrose inverse)** of the matrix A (i.e., A^+) via the **SVD**, where $A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$.

(f) Compute the **minimal length least squares solution** to $Ax = b$, where $A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$ and $b = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$.

2.1.4 Inner product, norm, distance

49. (DP) Let $u = \begin{bmatrix} -1 \\ 4 \\ -5 \end{bmatrix}$ and $v = \begin{bmatrix} 2 \\ -2 \\ 0 \end{bmatrix}$.

(a) Compute the **1-norm** (or the sum norm or the taxicab norm or the Manhattan norm; $\|\cdot\|_1$) of u and the 1-norm of v. Compute the induced **distance** from the 1-norm between u and v ($d_1(u, v)$).

(b) Compute the **2-norm** (or the Euclidian norm; $\|\cdot\|_2$) of u and the 2-norm of v. Compute the induced **distance** from the 2-norm between u and v ($d_2(u, v)$).

(c) Compute the **∞-norm** (or the max norm or the uniform norm; $\|\cdot\|_\infty$) of u and the ∞-norm of v. Compute the induced **distance** from the ∞-norm between u and v ($d_\infty(u, v)$).

Observation: The **p-norm** (or the Minkowski norm) is as follows: $\|v\|_p = (|v_1|^p + \cdots + |v_n|^p)^{\frac{1}{p}}$

50. (DP) Let $A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$. Compute the following **matrix norms** and **condition numbers**:

(a) $\|A\|_{\text{Fro}}$ in two modes and $\text{cond}_{\text{Fro}}(A)$, knowing that the singular values of A are $\sqrt{3}$ and 1.

(b) $\|A\|_1$ and $\text{cond}_1(A)$, knowing that its pseudoinverse is $A^+ = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ \frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \end{bmatrix}$.

(c) $\|A\|_2$ and $\text{cond}_2(A)$, knowing that the singular values of A are $\sqrt{3}$ and 1.

(d) $\|A\|_\infty$ and $\text{cond}_\infty(A)$, knowing that its pseudoinverse is $A^+ = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ \frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \end{bmatrix}$.

2.1.5 Hyperplanes

51. (after MIT, fall 2011 video "Problem Solving: Projection onto Subspaces")

    (a) The following is an equation of a **(linear) hyperplane** which represents a subspace of $\mathbb{R}^3$: $x + y - z = 0$.

        i. Specify a **normal vector to this hyperplane**.

        ii. Find the **orthogonal complement** of the given subspace **in two ways**.
        Observation: $I = P + P_\perp$, where P is the projection matrix onto a subspace and $P_\perp$ is the projection matrix onto the orthogonal complement of that subspace.

        iii. Compute the Euclidian **distance from the point** $v = \begin{bmatrix} 8 \\ 0 \\ 9 \end{bmatrix}$ **to the hyperplane**.

    (b) The following is an equation of a **affine hyperplane** which represents an affine subspace of $\mathbb{R}^3$: $x + y - z = 2$.
    Having the new hyperplane, answer to the three questions of the exercise above.

    Observation: This exercise was included to highlight the link between linear spaces and hyperplanes. Hyperplanes are linear spaces. When talking about hyperplanes the notions like span are not mentioned and this is a good thing. But if the person knows linear algebra, he should also know this and that's why we mention it here.

2.1.6 Revision

52. Give **similarities between the following vector and matrix notations** ($x \in \mathbb{R}^n$, $X \in \mathbb{R}^{m \times n}$, $a, b \in \mathbb{R}^n$, $A, B \in \mathbb{R}^{m \times n}$, $\sigma \in \mathbb{R}$, $u \in \mathbb{R}^n$, $U \in \mathbb{R}^{m \times n}$, $\Sigma \in \mathbb{R}^{n \times n}$):

    (a) $\|x\|_2$ and $\|X\|_{\text{Fro}}$

    (b) $a^\top b$ and $\text{tr}(A^\top B)$

(c) $\|x\|_2^2 = x^\top x$ and $\|X\|_{\text{Fro}}^2 = \text{tr}(X^\top X)$

(d) $x^\top x = 1$ and $X^\top X = I$

(e) σu and $U\Sigma$

53. (CS189 Spring 2018 Hw0, ex.3) Consider the vectors $u = \begin{bmatrix}1\\2\end{bmatrix}$ and $v = \begin{bmatrix}2\\3\end{bmatrix}$. Define the matrix $M = uv^\top$.

(a) Compute the eigenvalues and eigenvectors of matrix M.

(b) Compute the rank and the determinant of the matrix M. What is the dimension of the null space os the matrix M?

(c) Now consider two non-zero vectors p and q in $\mathbb{R}^d$ and the matrix $N = pq^\top$. Repeat the computations for the two parts above for the matrix N. To compute the eigenvectors, you can assume that the d-th coordinate of the vector q is not zero, i.e., $q_d \neq 0$.

Please explain your computations/arguments precisely.

Solution:

There are two ways to approach the problem: (1) Perform numerical computations for parts (a) and (b), as one would do if the matrix M was a general matrix. This has the benefit of immediately making visible the patterns involved. (2) Exploit the special structure of the matrix M and make use of the general solution from part (c).

(a) The matrix M is given by $\begin{bmatrix}2 & 3\\4 & 6\end{bmatrix}$. To find the eigenvalues, we have to find solutions of the equation $\det(M - \lambda I) = 0$. We have

$$\det(M-\lambda I) = 0 \Rightarrow \det\begin{bmatrix}2-\lambda & 3\\4 & 6-\lambda\end{bmatrix} = 0 \Rightarrow (2-\lambda)(6-\lambda) - 12 = 0$$
$$\Rightarrow \lambda^2 - 8\lambda = 0$$
$$\Rightarrow \lambda = 0.8$$

To compute the eigenvector corresponding to an eigenvalue λ, we have to determine a basis of the nullspace of the matrix $M - \lambda I$. In other words, we have to find a maximal set of linearly independent solutions for the equation $(M - \lambda I)x = 0$. In our case, these equations are given by

$$\lambda = 0 : \begin{bmatrix}2 & 3\\4 & 6\end{bmatrix} x = 0$$

$$\lambda = 8 : \begin{bmatrix} -6 & 3 \\ 4 & -2 \end{bmatrix} x = 0.$$

To solve the first linear equation, we have

$$\begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix} \sim \begin{bmatrix} 2 & 3 \\ 2 & 3 \end{bmatrix} \Rightarrow 2x_1 + 3x_2 = 0 \Rightarrow x = \begin{bmatrix} -3 \\ 2 \end{bmatrix}$$

We immediately notice that the resulting x is orthogonal to v. And in hindsight, it makes complete sense. This makes $uv^\top x = u(v^\top x) = 0$.

Similarly, for the second linear eigenvector equation, we have

$$\begin{bmatrix} -6 & 3 \\ 4 & -2 \end{bmatrix} \sim \begin{bmatrix} -2 & 1 \\ -2 & 1 \end{bmatrix} \Rightarrow -2x_1 + x_2 = 0 \Rightarrow x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Hmm... This is clearly just u itself. And in hindsight, it makes complete sense. This makes $uv^\top x = u(v^\top u) = (v^\top u)u$ and so u has to be an eigenvector.

(b) The rank of a matrix is equal to the number of linearly independent columns and hence the rank of the matrix M is 1. Alternatively, we know that the rank isn't zero since the matrix isn't the zero matrix. The rank isn't 2 since the matrix has an eigenvalue of 0 and hence has a nontrivial nullspace and hence is not invertible. So, it must have rank 1 by the process of elimination.

Since the determinant of any matrix is equal to the product of all its eigenvalues (repeated with multiplicity if any have multiplicity), the determinant of the matrix M is 0. Alternatively, since the matrix is not invertible, the determinant must be 0.

Furthermore, the dimension of the nullspace of any matrix is equal to the number of columns minus the rank and hence for M, it is 1. Alternatively, it is the number of linearly indepenent eigenvectors corresponding to eigenvalue 0. Either way, it is 1.

(c) To solve this part, we exploit the special structure of the matrix N. We use the following three important facts:

- Any matrix of the form $pq^\top$ is a rank one matrix, because all its rows are just a multiple of the row vector $q^\top$. Hence it has at most one non-zero eigenvalue.
- For two matrices A and B of compatible size, we have

$$\text{trace}(AB) = \text{trace}(BA)$$

Recalling the special case that we already say, we split the analysis in two parts: (1) $q^\top p \neq 0$ and (2) $q^\top p = 0$.

Case 1: To find the non-zero eigenvalue, follow what we observed earlier.

$$Np = pq^\top p = (q^\top p)p$$

That is for matrix N, the vector p is an eigenvector with corresponding eigenvalue $(q^\top p)$ which is not zero. (Notice that even if $q^\top p = 0$, that p would still be an eigenvector.)

Now, all the other $d-1$ eigenvalues are zero since the matrix is clearly rank 1 by construction. In other words, the nullity of the matrix is $d-1$. Finding $d-1$ linearly independent (LI) eigenvectors corresponding to the zero eigenvalue is equivalent to finding a basis for the nullspace of the matrix N, which in turn is equivalent to finding a set of $d-1$ LI vectors that are orthogonal to the vector q. That is we have to find the solutions to the equation

$$q_1 x_1 + q_2 x_2 + \cdots + q_d x_d = 0 (1)$$

Since q $neq0$ there exists a coordinate i such that $q_i \neq 0$. Without loss of generality, we can assume that $q_d \neq 0$. We now try to find a general solution of the equation above. We have

$$x_d = -(q_1 x_2 + q_2 x_2 + \cdots + q_{d-1} x_{d-1})/q_d$$

As a result a general solution has the form

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{d-1} \\ -q_1/q_d x_1 - q_2/q_d x_2 - \cdots - q_{d-1}/q_d x_{d-1} \end{bmatrix}$$

$$= x_1 \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ -q_1/q_d \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 0 \\ -q_2/q_d \end{bmatrix} + \cdots + x_{d-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ -q_{d-1}/q_d \end{bmatrix}$$

which yields that we have the following set of LI solutions for the equation (1):

$$\mathcal{S} = \left\{ \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ -q_1/q_d \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 0 \\ -q_2/q_d \end{bmatrix}, \ldots, \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ -q_d/q_1 \end{bmatrix} \right\}.$$

To verify that the set $\mathcal{S}$ has linearly independent vectors, we observe that

$$\alpha_1 \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ -q_1/q_d \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 0 \\ -q_2/q_d \end{bmatrix} + \cdots + \alpha_{d-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ -q_{d-1}/q_d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix} \Leftrightarrow$$

$$\Leftrightarrow \alpha_1 = \alpha_2 = \cdots = \alpha_{d-1} = 0$$

Remark: Students are not required to prove all the arguments in extensive detail. However an statement of the form "the eigenvectors are the solution of the equation (1)" without further explanation is not sufficient.

To summarize for the case $p^\top q \neq 0$:

- The eigenvalues of the matrix N are given by $(q^\top p)$, 0 corresponding eigenvectors are p and the set $\mathcal{S}$ mentioned above.
- The rank of the matrix N is 1 and its determinant is 0. The nullity of the matrix is $d - 1$.

Case 2: When $p^\top q = 0$, then the situation becomes quite interesting. Note that as argued before the matrix N can have at most one non-zero eigenvalue, due to its rank being one. We also know that the sum of eigenvalues is equal to the trace of the matrix. Furthermore using the trace trick, trace(N) = trace($pq^\top$) = trace($p^\top q$) = $p^\top q = 0$. Thus the sum of the eigenvalues is zero. Since d-1 eigenvalues are known to be zero, adding the fact that the sum of d eigenvalues is zero, implies that all eigenvalues are zero.

Thus, we try to find d linearly independent eigenvectors corresponding to the 0 eigenvalue.

A set of d - 1 LI eigenvectors can be found as before. We need to determine if there exists a d-th LI eigenvector corresponding to the 0 eigenvalue. Note that p is orthogonal to q and is an eigenvector corresponding to 0. In fact it now lies in the linear span of the set $\mathcal{S}$, defined in equation (2).

To find the d-th eigenvector, we need to look outside the linear span of the set $\mathcal{S}$. In this case, this space is simply linear multiple of the vector q. However, q is not an eigenvector because

$$N_q = pq^\top q = \|q\|^2 p$$

which is not a scalar multiple of q because q is orthogonal to the vector p. As a result, there is no d-th linearly independent eigenvector.

This example illustrates the phenomenon where the geometric multiplicity (dimension of the eigenspace) is less than the algebraic multiplicity of the eigenvalue. Notice that this is because the matrix N is clearly nilpotent and not zero. $NN = pq^\top pq^\top = p(q^\top p)q^\top = p0q^\top = 0$. Clearly a nilpotent matrix can't be diagonalizable since all of its eigenvalues have to zero while the matrix itself is nonzero. So, it has to be lacking a full complement of eigenvectors.

To summarize for the case $p^\top q = 0$:

- The eigenvalues of the matrix N are given by $0, \ldots, 0$ and corresponding eigenvectors are given by the set $\mathcal{S}$ mentioned above. The matrix has only d-1 linearly independent eigenvectors.
- The rank of the matrix N is still 1 and its determinant is 0. The nullity of the matrix is d-1.

Remark: A common mistake is to assume that the eigenvectors are usually orthogonal to each other. That is necessary only for symmetric matrices and need not hold for a general matrix. Also, note that the matrix N is not symmetric unless p = q.

54. (Stanford, Sta306b, 2015s midterm test, ex.13 - exams/New folder) You have a sparse 100, 000 by 100, 000 matrix and want to compute the singular vectors corresponding to the two largest singular values. How would you compute them?

    Solution: Use the power method, and exploit the sparsity if the matrix when carrying out the multiplication of the matrix times a vector.

55. (CS246 Final Exam Solutions, Winter 2011, ex.6.a) Let A be a square matrix of full rank, and the SVD of A is given as: $A = U\Sigma V^\top$, where U and

V are orthogonal matrices. The inverse of A can be computed easily given U, V and Σ. Write down an expression for A^{-1} in their terms. Simplify as much as possible.

Solution: $A^{-1} = V\Sigma^{-1}U^\top$

56. (LA4ML) Mark each statement as **true or false**. Justify your response.

    (a) (LA4ML. Review Packet 2, ex.6) For a set of vectors, $\{v_1, \ldots, v_n\}$, the linear combination $\alpha_1 v_1 + \cdots + \alpha_n v_n$ can be written as a matrix vector product. If it is true, define the matrix and vector which should be multiplied together to achieve this sum.

    Solution: the sum $= V\alpha$

    $V = (V_1|V_2|V_3|\ldots|V_n)$

    $$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}$$

    (b) (LA4ML. Chapter 3. Exercises. ex.5.1) If $Ax = b$ has a solution then b can be written as a linear combination of the columns of A.

    (c) (LA4ML. Chapter 3. Exercises. ex.5.2) If $Ax = b$ has a solution then b is in the span of the columns of A.

    (d) (LA4ML. Chapter 3. Exercises. ex.5.3) If the vectors v_1, v_2, v_3 form a linearly dependent set, then v_1 is in the span(v_2, v_3).

    (e) (LA4ML. Eigenvector and Intro to PCA - Worksheet. ex.2) A rectangular matrix can have eigenvalues/eigenvectors.

    Solution: no. The equation wouldn't even make sense!

    $Ax = \lambda x$

    $(m \times n)(n \times 1) = (m \times 1)$

    x cannot be $n \times 1$ on left and $m \times 1$ on right!!!

    (f) (Radford, Fall 2008, (13), pr. 7.h) Any non-zero vector of length k is an eigenvector of the $k \times k$ identity matrix.

    Solution: True

    (g) (Radford, Fall 2009, (14), pr.1.a) Every non-zero vector of length k is an eigenvector of the matrix $5I$, where I is the $k \times k$ identity matrix.

    Solution: True

2.1.7 Important Proofs

57. (LA4ML. Chapter 2. Exercises. ex.7) **Pythagorean Theorem**

    (a) Show that x and y are orthogonal if and only if

    $$\|x + y\|_2^2 = \|x\|_2^2 + \|y\|_2^2$$

    (Hint: Recall that $\|x\|_2^2 = x^\top x$)

    (b) **A matrix form**
    Show that if x_i and y_i are orthogonal, $\forall i \in \{1, \ldots, n\}$, $X = [x_1 | \ldots | x_n]$, $Y = [y_1 | \ldots | y_n]$, then

    $$\|X + Y\|_{\text{Fro}}^2 = \|X\|_{\text{Fro}}^2 + \|Y\|_{\text{Fro}}^2.$$

58. **Orthogonal Projection Matrix**

    (a) i. Let $a, b \in \mathbb{R}^2$, $W = \text{span}(a)$. Deduce the formula for the orthogonal projection matrix of b on W via the definition of two orthogonal vectors.

    ii. Let $a, b \in \mathbb{R}^2$, $W = \text{span}(a)$. Prove the following result via (matrix) calculus:

    $$\frac{aa^\top b}{a^\top a} = \text{argmin}_{p \in W} \|b - p\|_2$$

    iii. How would be the orthogonal projection matrix simplified when $\|a\| = 1$, i.e., $a^\top a = 1$?

    (b) i. Let $a_1, \ldots, a_k, b \in \mathbb{R}^n$, $k \leq n$, $\{a_1, \ldots, a_k\}$ - linearly independent, $A = (a_1 | \ldots | a_k)$, $W = \text{span}(a_1, \ldots, a_k)$. Deduce the formula for the orthogonal projection matrix of b on W via the definition of orthogonal vectors.

    ii. Let $a_1, \ldots, a_k, b \in \mathbb{R}^n$, $k \leq n$, $\{a_1, \ldots, a_k\}$ - linearly independent, $A = (a_1 | \ldots | a_k)$, $W = \text{span}(a_1, \ldots, a_k)$. Prove the following result via (matrix) calculus:

    $$A(A^\top A)^{-1} A^\top b = \text{argmin}_{p \in W} \|b - p\|_2$$

    iii. How would be the orthogonal projection matrix simplified when $A^\top A = I$?

59. ***Very Important for PCA: The variational/optimization characterization of eigenvalues*** (after `https://www.caam.rice.edu/~caam440/pca.pdf` and `http://www.cs.columbia.edu/~djhsu/AML/lectures/notes-pca.pdf`)

(a) Let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$ and corresponding orthonormal eigenvectors $v_1, \ldots, v_n$. Prove that, for any $k \in \{1, \ldots, n\}$, we have:

$$\lambda_k = \max_{x \perp \text{span}(x_1, \ldots, x_{k-1}), \|x\|=1} x^\top A x$$

and that this is achieved by v_k.

Observation: This can be solved in at least two ways. The preferred one is via Lagrangian multipliers. For the Lagrangian method one can use CMU, 2008f, EXing, HW5, pr.1 (`https://drive.google.com/open?id=19PXns575xXnvXWfGr4PpoQlETpz_yRgV`) and/or `http://oldwww.ma.man.ac.uk/~mkt/MT3732%20(MVA)/Notes/MVA_Section2.pdf`. For the other method one can use `http://www.cs.columbia.edu/~djhsu/AML/lectures/notes-pca.pdf`) or `https://archive.ins.uni-bonn.de/ins.uni-bonn.de/teaching/vorlesungen/WissRech2SS16/exercises/sheet10-complete-solution.pdf`.

Observation 2: For the singular values there is a similar result:

Let $A \in \mathbb{R}^{m \times n}$ be a matrix with singular values $\sigma_1 \geq \cdots \geq \sigma_{\text{rank}(A)}$ and corresponding orthonormal left singular vectors $u_1, \ldots, u_{\text{rank}(A)}$ left singular vectors $v_1, \ldots, v_{\text{rank}(A)}$. For any $k \in \{1, \ldots, \text{rank}(A)\}$, we have:

$$\sigma_k = \max_{\substack{x \perp \text{span}(x_1, \ldots, x_{k-1}), \|x\|=1 \\ y \perp \text{span}(y_1, \ldots, y_{k-1}), \|y\|=1}} x^\top A y$$

and this is achieved by u_k and v_k.

(b) Let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$ and corresponding orthonormal eigenvectors $v_1, \ldots, v_n$. Prove that, for any $k \in \{1, \ldots, n\}$, we have:

$$\lambda_1 + \cdots + \lambda_k = \max_{U \in \mathbb{R}^{n \times k}, U^\top U = I} \text{tr}(U^\top A U)$$

and that this is achieved by an orthogonal projection to the span of $v_1, \ldots, v_k$.

Observation: The solution is **not unique**: if M^* is a solution for this problem, then $M^* R$, where R is orthogonal, is a solution for the problem. Also, if $n = k$, then $U^\top U = I \Rightarrow U U^\top = I$ and $\text{tr}(U^\top A U) = \text{tr}(U U^\top A) = \text{tr}(A)$, so there are infinitely many solutions (all orthogonal matrices are solutions and one soluton is I).

Observation 2: This can be solved in at least two ways. The preferred one is via Lagrangian multipliers. For the Lagrangian method one can use the photo PCA_Benyiamin (a solution started from `https://arxiv.org/pdf/1906.03148.pdf` and `https://arxiv.org/pdf/1903.11240.pdf` and then completed/corrected via a discussion with an author, Benyiamin). For the other method, one can use `https://archive.ins.uni-bonn.de/ins.uni-bonn.de/teaching/vorlesungen/WissRech2SS16/exercises/sheet10-complete-solution.pdf` and/or `http://www.cs.columbia.edu/~djhsu/AML/lectures/notes-pca.pdf`.

Observation 3: For the singular values there is a similar result:

Let $A \in \mathbb{R}^{m\times n}$ be a matrix with singular values $\sigma_1 \geq \cdots \geq \sigma_{\mathrm{rank}(A)}$ and corresponding orthonormal left singular vectors $u_1, \ldots, u_{\mathrm{rank}(A)}$ left singular vectors $v_1, \ldots, v_{\mathrm{rank}(A)}$. For any $k \in \{1, \ldots, \mathrm{rank}(A)\}$, we have:

$$\sigma_1 + \cdots + \sigma_k = \max_{\substack{U\in\mathbb{R}^{m\times k}, U^\top U = I \\ V\in\mathbb{R}^{n\times k}, V^\top V = I}} \mathrm{tr}(U^\top A V)$$

and this is achieved by an orthogonal projection to the span of $u_1, \ldots, u_k$ and an orthogonal projection to the span of $v_1, \ldots, v_k$.

(c) (after `https://archive.ins.uni-bonn.de/ins.uni-bonn.de/teaching/vorlesungen/WissRech2SS16/exercises/sheet10-complete-solution.pdf`)

i. A **greedy** algorithm is an algorithm that tries to solve an optimization problem by making a **locally** optimal choice at each stage (in a **sequencial** manner). A solution obtained by a greedy algorithm is called a greedy solution. Argue that the k values and the corresponding k vectors obtained in 59a form a greedy solution for the maximization problem considered in 59b.

ii. Use the result from 59b to conclude that the output from 59a gives the **optimal** (**global**) solution for the optimization problem in 59b.

60. (`https://www.math.tecnico.ulisboa.pt/~jvideman/ExameANFO-19012017_Sol.pdf` Master in Mathematics and Applications - Técnico, Lisbon Numerical Functional Analysis and Optimization - Fall Semester 2016 Exam (Part II) – January 19th, 2017, ex.3) *This exercise highlights the fact that the proof using Lagrangian multipliers for the above problem may not be sufficient. We consider a simple problem that also has to do with the variational/optimization characterization of eigenvalues and give a complete*

answer. If one wants to have a 100% correct proof, one can re-solve the above problem using the ideas below, but what can be observed is that the ***above*** *proof is preferred and can be found almost everywhere where PCA comes into discussion.*

Consider the following constrained minimization problem

$$\min_{x\in\mathbb{R}^N, x^\top x=1} x^\top Ax$$

where $A \in \mathbb{R}^{N\times N}$ is a symmetric matrix with N distinct eigenvalues $\lambda_1 < \lambda_2 < \cdots < \lambda_N$.

(a) Write down the **KKT** conditions for the above problem. Determine all KKT points (x_*, λ^*). Show that the **constrained qualification LICQ** is valid at the stationary points x_*.

(b) Determine all local and global solutions of the above problem.

Solution:

(a) The constrained minimization problem reads as

$$\min_{x\in\mathbb{R}^N} \text{ subject to } c_1(x) = 0 (8),$$

where $f(x) = x\top Ax$ and $c_1(x) = x^\top x - 1$. The Lagrangean function associated with problem (8) is thus $L(x, \lambda) = f(x) - \lambda c_1(x)$. The KKT conditions hold at (x_*, λ_*) if

$$\nabla_x L(x_*, \lambda^*) = 0, c_1(x_*) = 0,$$

that is, if $2Ax_* - 2\lambda^* x_* = 0$ and $x_*^\top x_* = 1$. The Lagrange multiplier λ^* satisfying the KKT conditions thus corresponds to one of the N distinct eigenvalues λ_j, $j = 1, \ldots, N$, of matrix A, and the stationary points x_* are the (normalized) eigenvectors w_j associated with λ_j's i.e.

$$Ax_* = \lambda^* x_*, \|x_*\|_2 = 1, \text{ where } (x_*, \lambda^*) = (w_j, \lambda_j), j = 1, \ldots, N.$$

The constraint qualification LCQ is valid if $\nabla c_1(x_*) \neq 0$. Now, $\nabla c_1(x_*) = 2x_* =\neq 0$ since $\|x_*\|_2 = 1$; thus LICQ holds at each of the N stationary points.

(b) The local solutions (minimizers) of the above problem must satisfy the second-order necessary conditions:

$$w^\top \nabla_x^2 L(x_*, \lambda^*) w \geq 0, \forall w \in F_2(x_*, \lambda^*),$$

where $F_2(x_*, \lambda^*) = \{w \in \mathbb{R}^N | \nabla c_1(x_*)^\top w = 0\}$. Given that A is symmetric, $\nabla_x^2 L(x_*, \lambda^*) = 2A - 2\lambda^2 I$, and thus

$$w^\top \nabla_x^2 L(x_*, \lambda^*) w = 2(w^\top A w - \lambda^* w^\top w). \quad (9)$$

Now, fix $j \in \{2, \ldots, N\}$ and consider the KKT point $(x_*, \lambda^*) = (w_j, \lambda_j)$, where w_j is the eigenvector associated with λ_j. The matrix A is symmetric, thus its eigenvectors constitute an orthogonal basis in $\mathbb{R}^N$. Thus, the critical cone $F_2(x_*, \lambda^*)$, spanned by vectors w orthogonal to w_j, is composed of the $N-1$ eigenvectors w_k, $k \neq j$ of A, including, in particular, w_1. Hence, testing the condition (9) at $(x_*, \lambda^*) = (w_j, \lambda_j)$, with $w = w_1 \in F_2(x_*, \lambda^*)$, we get

$$w^\top \nabla_x^2 L(x_*, \lambda^*) w = 2(w_1^\top A w_1 - \lambda_j w_1^\top w_1) = 2(\lambda_1 - \lambda_j)\|w\|_2^2 < 0 \forall j \neq 1,$$

since λ_1, the eigenvalue associated with the eigenvector w_1, is the smallest eigenvalue and the eigenvalues are distinct. The second-order necessary condition of optimality is thus violated and we conclude that the $N-1$ stationary points (x_*, λ^*), we have $F_2(x_*, \lambda^*) = \text{span}\{w_2, w_3, \ldots, w_N\}$ and

$$w^\top \nabla_x^2 L(x_*, \lambda^*) w = 2(w_j^\top A w_j - \lambda_1 w_j^\top w_j) = 2(\lambda_j - \lambda_1)\|w_j\|_2^2 > 0 \forall w = w_j, j = 2, 3 \ldots, N.$$

In other words, the second-order sufficient condition is valid and, therefore, $(x_*, \lambda^*) = (w_1, \lambda_1)$ is a local minimizer.
The point $(x_*, \lambda^*) = (w_1, \lambda_1)$ is also a global solution since by the Rayleigh quotient

$$f(x_*) = x_*^\top A x_* = \lambda_* x_*^\top x_* = \lambda_1 \leq x^\top A x = f(x), \forall x \in \mathbb{R}^N, \text{ with } x^\top x = 1,$$

or because, by Weierstrass' theorem, the objective function must attain its global minimum (and maximum) in the feasible set S since $f : S \to \mathbb{R}^N$ is continuous and S is compact. The solution is $(x_*, \lambda^*) = (w_1, \lambda_1)$ is unique because the LICQ condition is valid at each point of the feasible region so that all possible solutions must satisfy the first-order necessary conditions (KKt conditions).

61. *The solution should be straightforward if we use the results from the problem above regarding . For hints for the solution check* ***https://stats.stackexchange.com/questions/130721/what-norm-of-the-reconstruction-error-is*** *and/or* ***https://archive.ins.uni-bonn.de/ins.uni-bonn.de/teaching/vorlesungen/WissRech2SS16/exercises/sheet10-complete-solution.pdf*** *.*

(a) (see `https://stats.stackexchange.com/questions/366634/eckart-young-mirsky-th` `e2%89%a4k-or-rank-k`) Given a matrix $X \in \mathbb{R}^{n \times d}$ and an integer k with $1 \leq k \leq \text{rank}(X)$, prove that the following two optimization problems are equivalent:

$$\text{argmin}_{A \in \mathbb{R}^{n \times d}, \text{rank}(A)=k} \|X - A\|_{\text{Fro}}^2$$

$$\text{argmin}_{A \in \mathbb{R}^{n \times d}, \text{rank}(A) \leq k} \|X - A\|_{\text{Fro}}^2$$

(b) **Eckart-Young-Mirsky (EYM) theorem**
Given a matrix $X \in \mathbb{R}^{n \times d}$ and an integer k with $1 \leq k \leq \text{rank}(X)$, prove that

$$U_k S_k V_k^\top = \text{argmin}_{A \in \mathbb{R}^{n \times d}, \text{rank}(A)=k} \|X - A\|_{\text{Fro}}^2$$

and

$$\sqrt{\sigma_{k+1}^2 + \cdots + \sigma_{\text{rank}(X)}^2} = \min_{A \in \mathbb{R}^{n \times d}, \text{rank}(A)=k} \|X - A\|_{\text{Fro}}^2$$

where $U_k S_k V_k$ is called the **Truncated SVD** of X (only the k column vectors of U and k row vectors of $V^\top$ corresponding to the k largest singular values (in the diagonal of Σ_k) are calculated; the rest of the matrix is discarded. (from https://en.wikipedia.org/wiki/Singular_value_decomposition#Tru and σ_i is the i^{th} singular value of X. By convention, if $k = \text{rank}(A)$ and σ_{k+1} does not exist, then $\sqrt{\sigma_{k+1}^2 + \cdots + \sigma_{\text{rank}(X)}^2} = 0$.

Observation: according to the previous result, the minimization problem could have been written with $\text{rank}(A) \leq k$.

Observation 2: The above Eckart-Young-Mirsky theorem uses the **Frobenius norm** of a matrix. There is also an EYM theorem that uses the **2-norm** of a matrix:

$$U_k S_k V_k^\top = \text{argmin}_{A \in \mathbb{R}^{n \times d}, \text{rank}(A)=k} \|X - A\|_2^2$$

and

$$\sigma_{k+1} = \min_{A \in \mathbb{R}^{n \times d}, \text{rank}(A)=k} \|X - A\|_2^2$$

Also, by convention, if $k = \text{rank}(A)$ and σ_{k+1} does not exist, then $\sigma_{k+1} = 0$.

2.2 Probability and Statistics Prerequisites

62. (CMU, NBalcan, fall 2017, ex.1.2) Let A be a real $n \times n$ matrix with $A^\top = A$. We say that A is

    (a) positive definite provided that

    $$x^\top Ax > 0, \forall \in \mathbb{R}^n \text{ and } x \neq 0$$

    (b) positive semidefinite provided that

    $$x^\top Ax \geq 0, \forall x \in \mathbb{R}^n \text{ and } x \neq 0$$

    Covariance matrix is a typical example of positive semidefinite matrix.

    Let $X = (X_1, X_2, \ldots, X_p)$ be a p-dimensional random vector and Σ be its covariance matrix, which is defined as

    $$\Sigma = E[(X - E[X])(X - E[X])^\top].$$

    We treat a vector as a column vector. In another exercise we have proved that Σ is a positive semidefinite matrix. When is Σ not a positive definite matrix?

    Solution: If there is a random variable X_i such that $X_i = c$ or $X_i = \sum_{j \neq i} a_j X_j$, where $c, a_j \in \mathbb{R}$ are constants, then Σ is not positive definite.

63. (LA4ML - Review Packet 2. ex.13-15 + Book (course), chapter 2, example 2.4.2)

    (a) **Statistical Formulas Using Linear Algebra Notation**

    Almost every statstical formula can be written in a more compact fashion using linear algebra. Most of the elementary formulas involve vector inner products or the Euclidian norm. To begin, we'll introduce the concept of centering the data. **Centering** the data means that the

mean of a variable is substracted from each observation. For example, if we have some variable, x, and 3 observations on that variable:

$$x = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}$$

then obviously, $\bar{x} = 3$. The **centered** version of x would then be

$$x - \bar{x}e = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix} - \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}.$$

We simply substract the mean from every observation so that the new mean of the variable is 0.

Most multivariate textbooks start by saying "all variable vectors in this textbook are assumed to be centered to have mean zero unless otherwise specified". Looking at the most common statistical formulas helps us see why. Try to re-write the following formulas using linear algebra notation, using the vectors x and y to represent centered data:

$$x = \begin{bmatrix} x_1 - \bar{x} \\ \vdots \\ x_n - \bar{x} \end{bmatrix}, y = \begin{bmatrix} v_1 - \bar{y} \\ \vdots \\ y_n - \bar{y} \end{bmatrix}.$$

For this exercise, keep in mind the following linear algebra constructs, which you should be very familiar with by now:

$$\|a\| = \sqrt{a_1^2 + a_2^2 + \cdots + a_n^2}$$

$$a^\top b = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

i. **Sample standard deviation**:

$$s = \frac{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}}{\sqrt{n-1}} = ...$$

ii. **Sample covariance**:

$$\text{covariance}(x, y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) = ...$$

iii. **Correlation coefficient**:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} = \text{.....}$$

Solution:

$$s = \frac{\|x\|}{\sqrt{n-1}}$$

$$cov = \frac{1}{n-1} x^\top y$$

$$r_{xy} = \frac{x^\top y}{\|x\|\|y\|}$$

(b) Write a **matrix formula for the sample covariance matrix**, Σ, using a matrix of centered data, $X = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$, where $\Sigma_{ij} = \text{cov}(x_i, x_j)$.

Solution:

$$X \in \mathbb{R}^{d \times n}$$

$$\Sigma = \frac{1}{n-1} X X^\top$$

(c) Write a **matrix formula for the sample correlation matrix**, C, using a matrix of centered data, $X = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$, where $C_{ij} = r_{ij}$ is Pearson's correlation measure between variables x_i and x_j. To do this, we need more than an inner product, we need to first divide each column by the corresponding standard deviation $s_i = \|x_i\|$.

Solution:

Let $D = \text{diag}\{s_1, s_2, \ldots, s_d\}$ then $D^{-1}X$ is standardized data. Since correlation is merely covariance of standardized data,

$$C = (D^{-1}X)(D^{-1}X)^\top = D^{-1} X X^\top D^{-1}$$

(d) Write a **matrix formula for the sample cross-covariance matrix**, CC, using two matrices of centered data, $X = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \in \mathbb{R}^{d \times n}$, $Y = \begin{bmatrix} y_1 \\ \vdots \\ y_D \end{bmatrix} \in \mathbb{R}^{D \times n}$, where $CC_{ij} = \text{cov}(x_i, y_j)$.

(e) As we've seen in previous examples, many statistical formulas involve the *centered* data, that is, data from which the mean has been substracted so that the new mean is zero. Suppose we have a matrix of data containing observations of individuals' heights, (h) in inches, weights (w), in pounds and wrist sizes (s), in inches:

$$A = \begin{array}{c} \\ person_1 \\ person_2 \\ person_3 \\ person_4 \\ person_5 \end{array} \begin{array}{c} \begin{array}{ccc} h & w & s \end{array} \\ \begin{bmatrix} 60 & 102 & 5.5 \\ 72 & 170 & 7.5 \\ 66 & 110 & 6.0 \\ 69 & 128 & 6.5 \\ 63 & 130 & 70 \end{bmatrix} \end{array}$$

The average values for height, weight, and wrist size are as follows:

$$\bar{h} = 66$$

$$\bar{w} = 128$$

$$\bar{s} = 6.5$$

To center all of the variables in this data set simultaneously, we could compute an outer product using a vector containing the means and a vector of all ones:

$$\begin{bmatrix} 60 & 102 & 5.5 \\ 72 & 170 & 7.5 \\ 66 & 110 & 6.0 \\ 69 & 128 & 6.5 \\ 63 & 130 & 7.0 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 66 & 128 & 6.5 \end{bmatrix} = ...$$

Solution:

$$= \begin{bmatrix} 60 & 102 & 5.5 \\ 72 & 170 & 7.5 \\ 66 & 110 & 6.0 \\ 69 & 128 & 6.5 \\ 63 & 130 & 7.0 \end{bmatrix} - \begin{bmatrix} 66 & 128 & 6.5 \\ 66 & 128 & 6.5 \\ 66 & 128 & 6.5 \\ 66 & 128 & 6.5 \\ 66 & 128 & 6.5 \end{bmatrix}$$

$$= \begin{bmatrix} -6 & -26 & -1 \\ 6 & 42 & 1 \\ 0 & -18 & -0.5 \\ 3 & 0 & 0 \\ -3 & 2 & 0.5 \end{bmatrix}$$

64. Let $X = \begin{bmatrix} 1 & 2 & 1.3 & 2 \\ 1 & 3 & 3.2 & 2.1 \end{bmatrix}$ and $Y = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 0 & 1 & 3 & 2 \\ 1 & 1 & 2 & 2 \end{bmatrix}$ be two data matrices (row = attribute, column = observation).

(a) Compute the **sample mean** for each data matrix.

(b) Compute the **sample covariance matrix** for each data matrix.

(c) Compute the **sample cross-covariance matrix** between X and Y.

65. (a) Let $X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$ be a **random vector**, with:

$$E[X] = E\left[\begin{bmatrix} X_1 \\ X_2 \end{bmatrix}\right] = \begin{bmatrix} E[X_1] \\ E[X_2] \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$E[XX^\top] = E\left[\begin{bmatrix} X_1^2 & X_1X_2 \\ X_2X_1 & X_2^2 \end{bmatrix}\right] = \begin{bmatrix} E[X_1^2] & E[X_1X_2] \\ E[X_2X_1] & E[X_2^2] \end{bmatrix} = \begin{bmatrix} 2 & 0.1 \\ 0.1 & 5 \end{bmatrix}$$

i. Compute the **covariance matrix** of X: $\mathrm{Cov}(X)$.

ii. Compute $\mathrm{E}[AX]$, where $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$.

Observation: there is also the notion of **random matrix**. For example, $XY^\top$ is a random matrix.

(b) Let $X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}$ and $Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}$ be random vectors, with:

$$E[X] = E\left[\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}\right] = \begin{bmatrix} E[X_1] \\ E[X_2] \\ E[X_3] \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$E[Y] = E\left[\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}\right] = \begin{bmatrix} E[Y_1] \\ E[Y_2] \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$E[XY^\top] = E\left[\begin{bmatrix} X_1Y_1 & X_1Y_2 \\ X_2Y_1 & X_2Y_2 \\ X_3Y_1 & X_3Y_2 \end{bmatrix}\right] = \begin{bmatrix} E[X_1Y_1] & E[X_1Y_2] \\ E[X_2Y_1] & E[X_2Y_2] \\ E[X_3Y_1] & E[X_3Y_2] \end{bmatrix} = \begin{bmatrix} 3 & 0.1 \\ 0.2 & 5 \\ 1 & 2 \end{bmatrix}$$

i. Compute the **cross-covariance matrix** for X and Y: $\text{Cov}(X, Y)$.

ii. Compute $\text{Cov}(AX, BY)$, where $A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \end{bmatrix}$, $B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$.

Observation: See also the exercises 69, 77 from Exercise book for ML, by Ciortuz et al., edition 2018-2019, first term. (MIT, 2006 fall, Tommi Jaakkola, HW1, pr. 5a and 5b).

66. (Radford, 2009f, Assignment 1, ex.2) Recall the spectral decomposition theorem: If A is a $k \times k$ symmetric real matrix, it is possible to find a set of k eigenvectors of A that are orthogonal and have length one, and if $e_1, \ldots, e_k$ are any such set of eigenvectors, with eigenvalues $\lambda_1, \ldots, \lambda_k$, then $A = \lambda e_1 e_1^\top + \cdots + \lambda_k e_k e_k^\top$.

(a) Use the Spectral Decomposition theorem to prove that the trace of a symmetric real matrix (the sum of its diagonal elements) is equal to the sum of its eigenvalues (with each eigenvalue appearing in the sum as many times as there are orthogonal eigenvectors that are associated with that eigenvalues).

(b) Let Q be any orthogonal matrix (for which $QQ^\top = I$), and let Σ_X be the covariance matrix of the random vector X. Define $Y = QX$, and let Σ_Y be the covariance matrix of Y. Prove that the trace of Σ_X is equal to the trace of Σ_Y.

Solution:

(a) Let A be a $k \times k$ symmetric real matrix. By the spectral decomposition theorem, we can write it as

$$A = \lambda_1 e_1 e_1^\top + \cdots + \lambda_k e_k e_k^\top$$

where the e_i are orthogonal eigenvectors of length one, and the λ_i are the corresponding eigenvalues.

It is easy to see that the trace of a sum of matrices is equal to the sum of their traces, and that the trace of a scalar times a matrix is equal to the scalar times the trace of the matrix. So,

$$\text{trace}(A) = \lambda_1 \text{trace}(e_1 e_1^\top) + \cdots + \lambda_k \text{trace}(e_k e_k^\top)$$

The diagonal elements of $e_i e_i^\top$ are $e_{i1}^2, \ldots, e_{ik}^2$. The sum of these is one, since the e_i have length one. So

$$\text{trace}(A) = \lambda_1 + \cdots + \lambda_k$$

(b) The covariance matrix of $Y = QX$ is $\Sigma_Y = Q\Sigma_X Q^\top$ (see p. 76 of the text). If ϵ is an eigenvector of Σ_X, with eigenvalue λ, then Qe is an eigenvector of Σ_Y, with eigenvalue λ, since

$$\Sigma_Y(Qe) = Q\Sigma_X Q^\top Qe = Q\Sigma_X e = Q\lambda e = \lambda(Qe)$$

The set of eigenvalues for Σ_Y is therefore the same as for Σ_X. So by the previous subpoint, the trace of Σ_Y is the same as the trace of Σ_X.

67. (CS189 spring 2018 practice midterm, ex.3.a) Assume you have two zero mean random variables $X \in \mathbb{R}^{d_1}$ and $Y\mathbb{R}^{d_2}$. Let their covariance matrices be given by $\Sigma_{XX}, \Sigma_{YY}, \Sigma_{XY}$. Also define the random variables $X' = A^\top X$ and $Y' = B^\top Y$, where $A \in \mathbb{R}^{d_1 \times d_1}$ and $B \in \mathbb{R}^{d_2 \times d_2}$ are both matrices. Let $W_1 \in \mathbb{R}^{d_1 \times d_1}$ and $W_2 \in \mathbb{R}^{d_2 \times d_2}$ represent two unitary matrices. Also, for any matrix M, we let $U(M)$ denote the matrix of its left singular vectors, and $V(M)$ denote the matrix of right singular vectors.

Which of the following is/are correct?

(a) Choosing $A = \Sigma_{XX}^{-1/2}$ whitens X'.

(b) Choosing $B = \Sigma_{XX}^{-1/2}$ whitens Y'.

(c) Choosing $B = W_2\Sigma_{YY}^{-1/2}$ whitens Y'.

(d) Choosing $A = U(\Sigma_{XX})$ decorrelates the entries of X'.

(e) Choosing $A = U(\Sigma_{XY})$ and $B = V(\Sigma_{XY})$ leads to diagonal $\Sigma_{X'Y'}$.

(f) If we want diagonal $\Sigma_{X'Y'}$ and whitened X', Y' using unitary matrices A, B is insufficient.

68. (LA4ML - Book (course) ex.1.e) Let $u = \begin{bmatrix} 1 \\ 2 \\ -4 \\ -2 \end{bmatrix}$ and $v = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$. Suppose these vectors are observations on four independent variables, which have the following covariance matrix:

$$\Sigma = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Determine the **Mahalanobis distance** between u and v.

69. (*see photos freedom1-6.jpg for some ideas for the solution*)

(a) Prove that $Z \sim \mathcal{N}(0, I) \Rightarrow \|PZ\|^2 \sim \chi^2_{\text{rank}(P)}$, where P is an orthogonal projection matrix (i.e., $P^2 = P$, $P = P^\top$).
Observation: if $P = A(A^\top A)^{-1}A^\top$, then $\text{rank}(A) = \text{rank}(P)$, i.e., $\text{rank}(P) =$ the **dimension** of the subspace spanned by the columns of A. From here the **intuition that the degrees of freedom represent the dimension of a linear space**.
Observation 2: by definition of χ^2 distribution, we have: $Z \sim \mathcal{N}(0, I_d) \Rightarrow \|Z\|^2 \sim \chi^2_d$.
Observation 3: $Z \sim \mathcal{N}(0, I) \Rightarrow AZ \sim \mathcal{N}(0, AA^\top)$

(b) In the context of Linear Regression, we have:

$$x_1, \ldots, x_n \in \mathbb{R}^d \text{ - the training input and } X = (x_1 | \ldots | x_n) \in \mathbb{R}^{d \times n}$$

$$y_1, \ldots, y_n \text{ - the training output}$$

$$Y_i = \beta^\top x_i + \epsilon_i, \text{ where } \epsilon_i \sim \mathcal{N}(0,1), \forall i \in \{1, \ldots, n\}$$

$$\epsilon_i, \epsilon_j \text{ - independent}, \forall i, j \in \{1, \ldots, n\}, i \neq j$$

$$Y_i | x_i \sim \mathcal{N}(\beta^\top x_i, 1), \forall i \in \{1, \ldots, n\}$$

$$\bar{y} = \frac{y_1 + \cdots + y_n}{n} \text{ and } \bar{Y} = \frac{Y_1 + \cdots + Y_n}{n}$$

$$\hat{y}_1, \ldots, \hat{y}_n \text{ - the estimated output: } \hat{y}_i = Hy_i = X(X^\top X)^{-1}X^\top y_i$$

$$\text{and } \hat{Y}_i = HY_i, \forall i \in \{1 \ldots, n\}$$

i. Prove that $\begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} | X \sim \mathcal{N}(0, I_n)$.

ii. Prove that
$$\text{TSS} = \sum_{i=1}^{n} (Y_i - \bar{Y}) \sim \chi^2_{n-1}$$
Observation: apart from our context, this formula is almost the formula for the sample variance.

iii. Prove that
$$\text{SSE} = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \sim \chi^2_{n-d-1}$$

iv. Prove that

$$\text{RegSS} = \sum_{i=1}^{n} (\hat{Y} - \bar{Y}) \sim \chi^2_d$$

Observation: There is a well-known decomposition called **ANOVA** (analysis of variance): TSS = RegSS + SSE.

Observation 2: The result can be generalized to the case when $Y_i|x_i \sim \mathcal{N}(\beta^\top x_i, \sigma^2)$. See `http://www2.econ.iastate.edu/classes/econ671/hallam/documents/QUAD_NORM.pdf`, `http://www.utstat.toronto.edu/~brunner/books/LinearModelsInStatistics.pdf` (chapter 5) and other sources.

2.3 Matrix factorizations: the Machine Learning context

Before starting, one should re-read some exercises in the LA_prereq_proposed.tex: *Six views of a matrix multiplication* with all the observations and the exercises which include a matrix factorization: LU, PLU, QR, eigen-decomposition, SVD, EYM THEOREM!.

2.3.1 Generalities

70. (LA4ML - Introduction to Vector Space Models - Worksheet. Part three. ex.1) Interpret the following Nonnegative Factor Output for a small collection of text documents, answering the following questions:

    (a) What meaning (theme/topic) would you give to each of the three factors?

    (b) What is the dominant factor (theme/topic) for each document?

    (c) What is the loading of the word *baseball* on Factor 2?

    (d) What is the coordinate/score of document 5 along Factor 3?

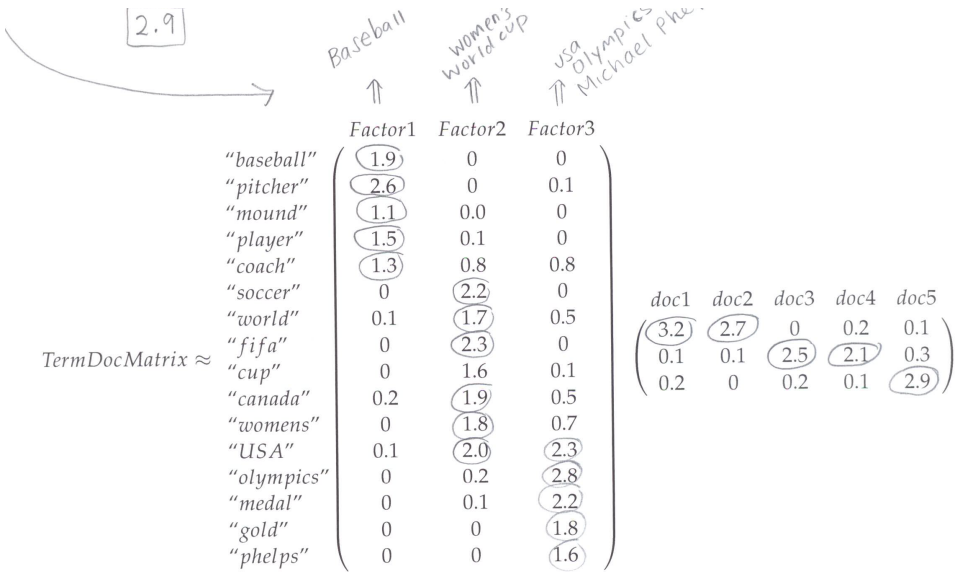
    (e) Interpret the factorization as a **soft clustering** technique applied to attributes, but also to instances.

$$\text{TermDocMatrix} \approx \begin{array}{c} \\ \text{"baseball"} \\ \text{"pitcher"} \\ \text{"mound"} \\ \text{"player"} \\ \text{"coach"} \\ \text{"soccer"} \\ \text{"world"} \\ \text{"fifa"} \\ \text{"cup"} \\ \text{"canada"} \\ \text{"womens"} \\ \text{"USA"} \\ \text{"olympics"} \\ \text{"medal"} \\ \text{"gold"} \\ \text{"phelps"} \end{array} \begin{array}{c} \begin{array}{ccc} \text{Factor1} & \text{Factor2} & \text{Factor3} \end{array} \\ \left[\begin{array}{ccc} 1.9 & 0 & 0 \\ 2.6 & 0 & 0.1 \\ 1.1 & 0 & 0 \\ 1.5 & 0.1 & 0 \\ 1.3 & 0.8 & 0.8 \\ 0 & 2.2 & 0 \\ 0.1 & 1.7 & 0.5 \\ 0 & 2.3 & 0 \\ 0 & 1.6 & 0.1 \\ 0.2 & 1.9 & 0.5 \\ 0 & 1.8 & 0.7 \\ 0.1 & 2.0 & 2.3 \\ 0 & 0.2 & 2.8 \\ 0 & 0.1 & 2.2 \\ 0 & 0 & 1.8 \\ 0 & 0 & 1.6 \end{array}\right] \end{array} \times$$

$$\times \begin{array}{c} \begin{array}{ccccc} \text{doc1} & \text{doc2} & \text{doc3} & \text{doc4} & \text{doc5} \end{array} \\ \left[\begin{array}{ccccc} 3.2 & 2.7 & 0 & 0.2 & 0.1 \\ 0.1 & 0.1 & 2.5 & 2.1 & 0.3 \\ 0.2 & 0 & 0.2 & 0.1 & 2.9 \end{array}\right] \end{array}$$

Observation: Change the text of the exercise after researching the fact that indeed the left matrix can be generally called the loading matrix and the right matrix can be called the score matrix. I think that this terminology is only for Factor Analysis, not for any matrix factorization.

Solution:



(a)

Factor1 = Baseball

Factor2 = women's world cup

Factor3 = USA Olympics Michael Phelps

(b) doc 1-2: Baseball (Factor 1)

doc 3-4: Women's world cup (Factor 2)

doc 5: USA Olympics (Factor 3)

(c) 0 (which means baseball is not relevant to that factor)

(d) 2.9

71. (after `https://www.fbbva.es/wp-content/uploads/2017/05/dat/greenacre_c01_2010.pdf`)

Given the following matrix factorization:

(a) Draw the corresponding **biplot**.

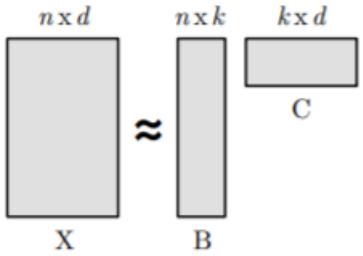
(b) Using only the biplot, compute the dot product between x_1 and y_1.

(c) Calibrate the biplot.

(d) Using only the calibrated biplot, compute the dot product between x_1 and y_1.

$$\begin{bmatrix} 8 & 2 & 2 & -6 \\ 5 & 0 & 3 & -4 \\ -2 & -3 & 3 & 1 \\ 2 & 3 & -3 & -1 \\ 4 & 6 & -6 & -2 \end{bmatrix} = \begin{bmatrix} 3 & 3 \\ 1 & 2 \\ -1 & 1 \\ 1 & -1 \\ 2 & -2 \end{bmatrix} \begin{bmatrix} 2 & 2 & -1 & -2 \\ 0 & -1 & 2 & -1 \end{bmatrix}$$

72. (CS 189 Spring 2015 Introduction to Machine Learning Final, ex.Q9) (Low Dimensional Decompositions) Given a design matrix $X \in \mathbb{R}^{n \times d}$ with $n > d$, we can create a low dimensional decomposition approximation $\bar{X} = BC$, where $\bar{X} \in \mathbb{R}^{n \times d}$, $B \in \mathbb{R}^{n \times k}$, $C \in \mathbb{R}^{k \times d}$, and $k < d$. The following figure shows a diagram approximated by B times C:



We can formulate several low dimensional techniques as solving the following optimization, subject to various constraints:

$$\min_{B,C} \|X - BC\|_{\text{Fro}}^2,$$

where $\|\cdot\|_{\text{Fro}}^2$ denotes the squared Frobenius norm of a matrix, that is, the sum of its squared entries.

(a) Which machine learning technique corresponds to solving the above optimization problem with constraint $\mathcal{C}_1$: each row of B is a vector e_i (a vector of all zeros, except a one in position i)?

i. k-means
ii. k-medoids
iii. SVD of X

Solution: A

(b) Describe the B and C matrices that result from solving the optimization problem above with constraint $\mathcal{C}_1$.

Solution: The rows of C are the cluster centers (the means) and the rows of B indicate which cluster each point belongs to.

(c) Which machine learning technique corresponds to solving the above optimization problem with constraint $\mathcal{C}_2$: each column of B has norm equal to one?

i. k-means

ii. k-medoids

iii. SVD of X

Solution: C

(d) Describe B and C matrices that result from solving the optimization problem above with constraint $\mathcal{C}_2$.

Solution: B is the first k left singular vectors of X and C is the transpose of the first k right singular vectors of X scaled by the first k singular values of X.
$X = U\Sigma V^\top, B = U_k$, and $C = \Sigma_k V_k^\top$.

(e) Which machine learning technique corresponds to solving the above optimization problem with the constraints $\mathcal{C}_3$: each row of C is one of the rows from X and each row of B is a row e_i (a vector of all zeros, except a one in position i)?

i. k-means

ii. k-medoids

iii. SVD of X

Solution: B

(f) Describe the B and C matrices that result from solving the optimization problem above with constraints $\mathcal{C}_3$.

Solution: The rows of C are the medoids (points from X representing the cluster centers) and the rows of B indicate which cluster each point belongs to.

73. (CS 189 Spring 2017 Introduction to Machine Learning Final, Q1.10) Circle the correct answer. A low-rank approximation of a matrix can be useful for

(a) removing noise

(b) filling in unknown values

(c) discovering latent categories in the data

(d) matrix compression

Solution: A,B,C,D

2.3.2 Singular Value Decomposition - SVD

74. (CS246 Final Exam Solutions, Winter 2011, ex.6.b) Let us say we use the SVD to decompose a Users $\times$ Movies matrix M and then use it for prediction after reducing the dimensionality. Let the matrix have k singular values. Let the matrix M_i be the matrix obtained after reducing the dimensionality to i singular values. As a function of i, plot how you think the error on using M_i instead of M for prediction purposes will vary.

    Solution: Will reduce then increase.

75. (EPFL final exam 2017, ex.24) You are given your $D \times N$ data matrix X, where D represents the dimension of the input space and N is the number of samples. We discussed in the course the singular value decomposition (SVD). Recall that the SVD is not invariant to scaling and that empirically it is a good idea to remove the mean of each feature (row of X) and to normalize its variance to 1. Assume that X has this form except that the last row/feature is then multiplied by $\sqrt{2}$, i.e., it has variance (l_2^2 -norm) of 2 instead of 1.

    Recall that the SVD allows us to write X in the form $X = USV^\top$, where U and V are unitary and S is a $D \times N$ diagonal matrix with entries s_i that are non-negative and decreasing, called the singular values. Assume now that you add a feature, i.e., you add a row to X. Assume that this row is identical to the last row of X, i.e., you just replicate the last feature. Call the new matrix $\tilde{X}$. But assume also that for $\tilde{X}$ we normalize all rows to have variance 1.

    To summarize, X is the original data matrix, where all means have been taken out and all rows are properly normalized to have variance 1 except the last one that has variance 2. And $\tilde{X}$ is the original data matrix with the last row replicated, and all means have been taken out and all rows are properly normalized. Let X $= USV^\top$ be the SVD of X and let. $\tilde{X} = \tilde{U}\tilde{S}\tilde{V}^\top$ be the SVD of $\tilde{X}$.

    (a) Show that

        i. $\tilde{V} = V$

        ii. $\tilde{S}$ is equal to S with an extra all-zero row attached.

    (b) Based on the previous relationships and assuming that it is always best to run an SVD with "normalized" rows, what is better: If you KNOW that a feature is highly correlated to another feature a priori. Should you rather first run the SVD and then figure out what features

to keep or should you first take the highly correlated feature out and then run the SVD? Explain.

Solution:

(a) In the exercise you learned that the columns of V are the eigenvectors associated to $X^\top X$. And the non-zero singular values of X are the square roots of the non-zero eigenvalues of $X^\top X$. But for our case $\tilde{X}^\top \tilde{X} = X^\top X$, which proves the two claims.

(b) It is better to first take out the highly correlated feature. If not, then the previous calculations show that this is the same as having a row that is not properly normalized.

76. (EPFL final exam 2016, ex.1.19) Assume that we have a data matrix X of dimension $D \times N$ as usual. Suppose that its SVD is of the form $X = USV^\top$, where S is a diagonal matrix with $s_1 = N$ and $s_2 = s_3 = \cdots = s_D = 1$. Assume that we want to compress the data from D to 1 dimensions via a linear transform represented by a $1 \times D$ matrix C and reconstruct then via $D \times 1$ matrix R. Let $\hat{X} = RCX$ be the reconstruction. What is the smallest value we can achieve for $\|X - \hat{X}\|^2_{\text{Fro}}$?

(a) D

(b) D - 1

(c) N - D

(d) N - D + 1

(e) N - D - 1

(f) N - 1

(g) N

(h) ND

Solution: Answer b is correct. We have learned that the Frobenius norm of the difference is at least equal to the sum of squares of the singular values that we leave out. In our case each of them has value 1 and we leave out D - 1 of them.

77. (CS246: Mining Massive Data Sets Winter 2013 Final, ex.9)

$$M = \begin{array}{c} \\ Diane \\ Ethan \\ Frank \\ Grace \\ Hank \\ Ingrid \\ Joe \end{array} \begin{array}{c} \begin{array}{ccccc} MMDS & MachineLearning & DataStructures & Dynamics & Mechanics \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 1 & 1 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Matrix M represents the ratings of Engineering courses taken by Stanford students. Each row of M represents the given student's set of ratings and the columns of M represent theclasses. The labels for students and classes are provided along each row and column. An entry M_{ij} in M represents the student i's rating for class j. Now, the SVD decomposition of matrix M is found to be as follows:

$$USV^\top = \begin{bmatrix} .18 & 0 \\ .36 & 0 \\ .18 & 0 \\ .90 & 0 \\ 0 & .53 \\ 0 & .80 \\ 0 & .27 \end{bmatrix} \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \begin{bmatrix} .58 & .58 & .58 & 0 & 0 \\ 0 & 0 & 0 & .71 & .71 \end{bmatrix}$$

It is pretty clear that there are two concepts of classes here: the first three are Computer Science classes while the last two are Mechanical Engineering classes.

Suppose a new student named Tony has the following reviews: 4 for Machine Learning, 5 for Data Structures, and 2 for Dynamics. This gives a representation of Tony in the class space as [0 4 5 2 0].

(a) What is the representation of Tony in the concept space?

(b) Explain what this representation predicts about how much Tony would like the two remaining classes that he has not reviewed (MMDS and Mechanics).

(c) Another student named Bruce has the following reviews: 5 for MMDS, 2 for Machine Learning, 4 for Dynamics, and 5 for Mechanics. What is the representation of Bruce in the concept space?

(d) Using similarity defined as sim(a,b) $= \frac{a \cdot b}{\|a\| \cdot \|b\|}$ where $\|a\|$ is the L_2 norm, calculate the cosine similarity of the two users using their concept space vectors.

Solution:

(a) Given that the class space for Tony, we can represent it as a vector, $a = [0, 4, 5, 2, 0]$. Then the mapping to concept space is $aV = [5.220, 1.420]$.

(b) It shows that he prefers MMDS rather than Mechanics, since he has higher preferences toward CS classes.

(c) Similarly, we can represent Bruce's review as a vector, $b = [5, 2, 0, 4, 5]$, and obtain $bV = [4.06, 6.39]$.

(d) The similarity between Tony and Bruce is 0.739, which is simply obtained by setting the values in the right vectors.

78. (CS168, Spring 2018 CS168 Final Exam, ex.2) In this question, we will examine SVD in the context of exploratory data analysis. Assume that the Senate consists of 100 senators and that they considered 1,000 bills during a given time period, with each senator voting either "Yes" or "No" on each of the 1,000 bills. We can represent this data as a 100×1000 matrix M, where the entry $M_{i,j}$ is 1 if the ith senator voted "Yes" on the jth bill, and is 0 otherwise.

(a) To begin, lets assume that each senator is affiliated with either party A or party B, split roughly 50/50, and that all members of a party cast the same vote ("Yes/No") on a bill. Additionally, assume that each of the bills belongs to exactly one of the following three types: 1) supported by both parties, 2) supported only by party A, 3) supported only by party B. Under these assumptions, what is the rank of matrix M? Justify your answer with at most two sentences.

(b) For the remainder of this question, lets relax the above assumptions, and instead, just assume that, on most bills, members of the same party generally vote similarly. Consider the voting matrix M, and its singular value decomposition $M = U\Sigma V^\top$.

i. Would the top few left singular vectors (columns of U) have any natural interpretations in this case? Explain your answer in a sentence or two.

ii. Would the top few right singular vectors (columns of V) have any natural interpretations in this case? Explain your answer in a sentence or two.

iii. Given the SVD decomposition $M = U\Sigma V^\top$, how can you compute a good rank r approximation of M? In what formal sense is the

approximation you describe the "best" rank r approximation of M?

iv. Roughly what would the best rank-1 approximation for M correspond to? (For example. what is a plausible interpretation of the row and column vector that define this rank-1 matrix?)

v. Roughly what would you expect the best rank-2 approximation for M to correspond to, and how would this differ from the matrix M?

2.3.3 Latent Semantic Indexing/Analysis - LSI/LSA

79. (taken from Grossman and Frieder's Information Retrieval, Algorithms and Heuristics)

A "collection" consists of the following "documents":

d1: Shipment of gold damaged in a fire.

d2: Delivery of silver arrived in a silver truck.

d3: Shipment of gold arrived in a truck.

Suppose that we use the term frequency as term weights and query weights. The following document indexing rules are also used:

- stop words were not ignored
- text was tokenized and lowercased
- no stemming was used
- terms were sorted alphabetically

We wish to use this example to illustrate how LSI works.

Use Latent Semantic Indexing (LSI) to rank these documents for the query gold silver truck.

Solution:

Step 1: Set term weights and construct the term-document matrix A and query matrix:

Terms		d1	d2	d3		q
a		1	1	1		0
arrived		0	1	1		0
damaged		1	0	0		0
delivery		0	1	0		0
fire		1	0	0		0
gold	**A =**	1	0	1	**q =**	1
in		1	1	1		0
of		1	1	1		0
shipment		1	0	1		0
silver		0	2	0		1
truck		0	1	1		1

Step 2: Decompose matrix A matrix and find the U, S and V matrices, where $A = USV^\top$

$$\mathbf{U} = \begin{bmatrix} -0.4201 & 0.0748 & -0.0460 \\ -0.2995 & -0.2001 & 0.4078 \\ -0.1206 & 0.2749 & -0.4538 \\ -0.1576 & -0.3046 & -0.2006 \\ -0.1206 & 0.2749 & -0.4538 \\ -0.2626 & 0.3794 & 0.1547 \\ -0.4201 & 0.0748 & -0.0460 \\ -0.4201 & 0.0748 & -0.0460 \\ -0.2626 & 0.3794 & 0.1547 \\ -0.3151 & -0.6093 & -0.4013 \\ -0.2995 & -0.2001 & 0.4078 \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} 4.0989 & 0.0000 & 0.0000 \\ 0.0000 & 2.3616 & 0.0000 \\ 0.0000 & 0.0000 & 1.2737 \end{bmatrix}$$

$$\mathbf{V} = \begin{bmatrix} -0.4945 & 0.6492 & -0.5780 \\ -0.6458 & -0.7194 & -0.2556 \\ -0.5817 & 0.2469 & 0.7750 \end{bmatrix} \quad \mathbf{V^T} = \begin{bmatrix} -0.4945 & -0.6458 & -0.5817 \\ 0.6492 & -0.7194 & 0.2469 \\ -0.5780 & -0.2556 & 0.7750 \end{bmatrix}$$

Step 3: Implement a Rank 2 Approximation by keeping the first two columns of U and V and the first two columns and rows of S.

$$U \approx U_k = \begin{bmatrix} -0.4201 & 0.0748 \\ -0.2995 & -0.2001 \\ -0.1206 & 0.2749 \\ -0.1576 & -0.3046 \\ -0.1206 & 0.2749 \\ -0.2626 & 0.3794 \\ -0.4201 & 0.0748 \\ -0.4201 & 0.0748 \\ -0.2626 & 0.3794 \\ -0.3151 & -0.6093 \\ -0.2995 & -0.2001 \end{bmatrix} \qquad k = 2$$

$$S \approx S_k = \begin{bmatrix} 4.0989 & 0.0000 \\ 0.0000 & 2.3616 \end{bmatrix}$$

$$V \approx V_k = \begin{bmatrix} -0.4945 & 0.6492 \\ -0.6458 & -0.7194 \\ -0.5817 & 0.2469 \end{bmatrix} \qquad V^T \approx V_k^T = \begin{bmatrix} -0.4945 & -0.6458 & -0.5817 \\ 0.6492 & -0.7194 & 0.2469 \end{bmatrix}$$

Step 4: Find the new document vector coordinates in this reduced 2-dimensional space.

Rows of V holds eigenvector values. These are the coordinates of individual document vectors, hence

d1(-0.4945, 0.6492)

d2(-0.6458, -0.7194)

d3(-0.5817, 0.2469)

Step 5: Find the new query vector coordinates in the reduced 2-dimensional space.

$q = q^\top U_k S_k^{-1}$

Note: These are the new coordinate of the query vector in two dimensions. Note how this matrix is now different from the original query matrix q given in Step 1.

$$q = q^T U_k S_k^{-1} \qquad k = 2$$

$$q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} -0.4201 & 0.0748 \\ -0.2995 & -0.2001 \\ -0.1206 & 0.2749 \\ -0.1576 & -0.3046 \\ -0.1206 & 0.2749 \\ -0.2626 & 0.3794 \\ -0.4201 & 0.0748 \\ -0.4201 & 0.0748 \\ -0.2626 & 0.3794 \\ -0.3151 & -0.6093 \\ -0.2995 & -0.2001 \end{bmatrix} \begin{bmatrix} \frac{1}{4.0989} & 0.0000 \\ 0.0000 & \frac{1}{2.3616} \end{bmatrix}$$

$$q = \begin{bmatrix} -0.2140 & -0.1821 \end{bmatrix}$$

Step 6: Rank documents in decreasing order of query-document cosine similarities.

$$\text{sim}(q, d) = \frac{q \bullet d}{|q||d|}$$

$$\text{sim}(q, d_1) = \frac{(-0.2140)(-0.4945) + (-0.1821)(0.6492)}{\sqrt{(-0.2140)^2 + (-0.1821)^2}\sqrt{(-0.4945)^2 + (0.6492)^2}} = -0.0541$$

$$\text{sim}(q, d_2) = \frac{(-0.2140)(-0.6458) + (-0.1821)(-0.7194)}{\sqrt{(-0.2140)^2 + (-0.1821)^2}\sqrt{(-0.6458)^2 + (-0.7194)^2}} = 0.9910$$

$$\text{sim}(q, d_3) = \frac{(-0.2140)(-0.5817) + (-0.1821)(0.2469)}{\sqrt{(-0.2140)^2 + (-0.1821)^2}\sqrt{(-0.5817)^2 + (0.2469)^2}} = 0.4478$$

Ranking documents in descending order

$$d_2 > d_3 > d_1$$

We can see that document d2 scores higher than d3 and d1. Its vector is closer to the query vector than the other vectors.

80. (Un. of Zagreb - Faculty of Electrical Eng. and Computing - 30 april 2014 - midterm - Text analysis and retrieval, ex. Part II.18) Consider a word-document matrix consisting of documents $d_1, \ldots, d_4$ and words $w_1, \ldots, w_3$. In order to build an LSI model, an SVD of the matrix was performed as follows:

$$\begin{array}{c} \\ w_1 \\ w_2 \\ w_3 \end{array} \begin{array}{c} \begin{array}{cccc} d_1 & d_2 & d_3 & d_4 \end{array} \\ \begin{bmatrix} 5 & 3 & 0 & 1 \\ 3 & 2 & 2 & 6 \\ 0 & 0 & 8 & 7 \end{bmatrix} \end{array} = \begin{bmatrix} 0.2 & 0.8 & -0.6 \\ 0.5 & 0.4 & 0.7 \\ 0.8 & -0.4 & -0.4 \end{bmatrix} \begin{bmatrix} 12.3 & 0.0 & 0.0 & 0.0 \\ 0.0 & 6.7 & 0.0 & 0.0 \\ 0.0 & 0.0 & 2.1 & 0.0 \end{bmatrix} \begin{bmatrix} 0.2 & 0.1 & 0.6 & 0.7 \\ 0.8 & 0.5 & -0.4 & 0.0 \\ -0.3 & -0.1 & -0.7 & 0.7 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

$$= UDV^\top$$

(a) Compute the reconstructed version of document d_2 using only the two most important latent concepts ($k = 2$).

(b) Which concept (u_1,u_2 or u_3) has the most impact on reconstructing the documents d_3?

(c) Assuming we use only two latent concepts for reconstruction ($k = 2$), compute the similarity (any of the measures mentioned in class) of d_1 and d_2 without explicitly computing reconstructed documents.

81. (CS 189 Spring 2016 Introduction to Machine Learning Final) In latent semantic indexing, we compute a low-rank approximation to a term-document matrix. Which of the following motivate the low-rank reconstruction?

(a) Finding documents that are related to each other, e.g. of a similar genre

(b) The low-rank approximation provides a lossless method for compressing an input matrix

(c) In many applications, some principal components encode noise rather than meaningful structure

(d) Low-rank approximation enables discovery of nonlinear relations

Solution: A,C

82. (`https://stats.stackexchange.com/questions/152879/latent-semantic-indexing-and-`
Give **reasons for not centering the data for LSI**.

3 Introduction - Dimensionality reduction

83. (CS 189 Summer 2018 Introduction to Machine Learning - DIS5, ex.1) (Motivation: Dimensionality reduction)

In this problem sheet we explore the motivation for general dimensionality reduction in machine learning and derive from first principles why projection on the first eigenvectors of the covariance matrix of the data has some favorable properties. A deeper understanding on the advantages of PCA and other dimensionality reduction methods is conveyed in the homework.

In general, we assume the following scenario: Suppose we are given n points $x_1, \ldots, x_n$ in $\mathbb{R}^d$ and the dimension of the feature vectors is d (very big, like 10^3). By dimensionality reduction, we refer to a mapping $\psi : \mathbb{R}^d \to \mathbb{R}^k$ that maps vectors from $\mathbb{R}^d$ to $\mathbb{R}^k$ with $k << d$.

(a) (Motivation) Given n feature vectors of d dimensions, in which regimes of n; d and why would you want to reduce the dimensionality in practical machine learning applications? Think about the concept of regularization studied extensively in the past few weeks.

(b) (Computational aspect) Revisit this in the context of linear regression. What is the computational complexity of performing a linear regression of n data points in d dimensions with n ¿ d (say by solving the normal equations when $XX^\top$ is invertible)? If the projection was given to you for free, approximately how many operations would you save if you reduced the dimension from $d = 10^3$ to $d = 10$?

(c) (Brainstorming possible projections) What are some naive and less naive dimensionality reduction methods you could think of and what would their computational costs be (approximately)? Which methods require either previous data of the same distribution or the data itself, which are projection methods are independent of the data?

(d) (Desiderata for projection) With the above goals in mind, let's think about a concrete scenario where we want to do binary classification. What are intuitively good properties of the data which makes it easy or possible for a classification algorithm to work well? In the homework you will prove that PCA and random projections are guaranteed to preserve some of these properties.

Solution:

(a) In general:

There are two big motivations for dimensionality reduction. First, there is the simple one coming from the bias/variance tradeoff. You have seen that every feature essentially brings its own variance to the problem that scales like 1/n. So, when there are a lot of potential features that we could use, the optimal number of features that we do use might be fewer. Hence, dimensionality reduction. The second is computational. Processing the data is costly. If a system of linear equations has to be solved, the complexity is super-linear in the number of variables. Cutting down the number of features cuts down the number of variables that we need to solve for.

Reducing dimension here seems to be win-win. Though for computing the reduction, we need to be smart, else there are no computational savings.

In slightly more detail: There are three basic regimes $n << d$ (high dimensional data regime = underdetermined), $n > d$ and $n >> d$ (overdetermined):

- When $n << d$ (underdetermined system), we use regularization (or model selection) assuming low rank structure to avoid the over-fitting that would naturally occur.
- When $n >> d$ there might appear to be enough data to get good estimates of the variables. However, even here, we can potentially get some advantages from dimensionality reduction (and other forms or regularization) if there is lower-dimensional models have good approximation error. (You saw this with polynomials fitting the exponential function.). Another idea here (not explored that much in 189) is to do "sketching" to reduce the n samples to something that is fewer since there is plenty of data.
- Dimensionality reduction is in general most helpful in cases when $n << d$ or $n > d$ and the problem has lower effective dimension (otherwise $n << d$ obviously wouldn't give you a reasonable estimator in the first place).

(b) Solving linear regression requires O(nd2) for n ¿ d (and O(n2d) for n ¡ d). We discuss the case n ¿ d for which the computational complexity can be seen by considering the solution via normal equations, which is

$$w^* = (X^\top X)^{-1} X^\top y. (1)$$

Forming the matrix $X^\top X$ costs $O(nd^2)$ and inverting it costs $O(d^3)$. Forming $X^\top y$ costs $O(nd)$ and the final matrix multiplication of the

two $d \times d$ matrices costs $O(d^3)$. The total cost is therefore $O(d^2(n+d))$ which in the case $d < n$ is equal to $O(nd^2)$. An alternative approach to compute the computational complexity of L.R. is via SVD computation.

Reducing the dimension by a factor of 100 therefore gives a 10,000 fold speedup for solving the linear regression, which is quite substantial.

(c) Brainstorm methods: Throwing elements away, Random matrix (independent Gaussian or Bernoulli entries), Randomly subsampled Fourier or Hadamard matrices, PCA. If curious, more about these topics can be learned from `https://arxiv.org/pdf/1104.5557.pdf`.

Complexities:

- Randomized matrix: $O(dkn)$, data independent
- Randomized FT, Hadamard would take $O(ndlogk)$, data independent
- PCA (SVD + projection step) $O(d^2n)$ (for full) vs. $O(dkn)$ (approximate) + projection step $O(dkn)$, needs raw data to even compute the dimensions.

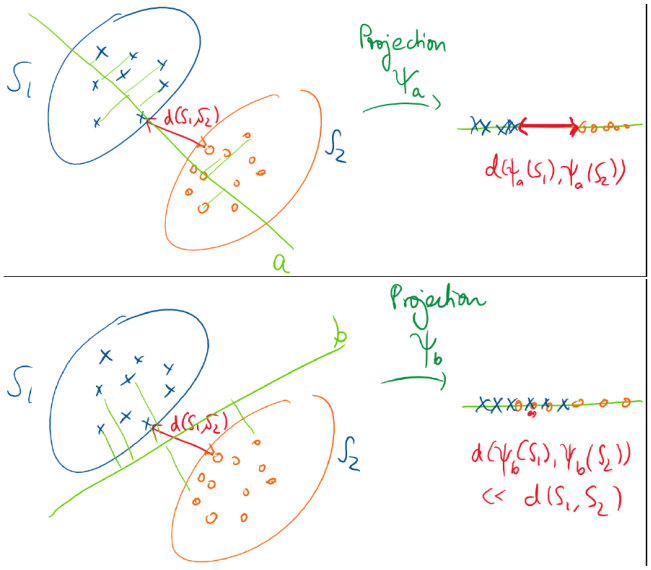
Exact SVD does not give an overall computational gain, since the computation for the projection matrix itself already takes $O(nd^2)$ computations. For randomized methods, $k > O(logn)$ are needed (see homework), random projections itself is already helpful ending up with $O(dnlogn)$. Approximate SVD methods get us improvements on the same scale. Using randomized FT or Hadamard instead gets us even smaller complexity $O(dnloglogn)$.

(d) Note that many notions vaguely mentioned here will be rigorously discussed in much more detail in later lectures, discussions, and homeworks. This is brought up here to motivate good properties of dimensionality reduction techniques. Intuitively speaking, if the minimum separation between samples belonging to two different classes is large, the classification task is "easy" in the sense of robust to noise. Mathematically, we may define the distance between sets as the minimum pairwise distance

$$d^2(S_1, S_2) := \min_{i \in S_1, j \in S_2} \|x_i - x_j\|_2^2$$

When the two sets are linearly separable, this notion is similar to the notion of gap and margin which we will cover later and is most commonly introduced in the context of support vector machines.

Projections which preserve Euclidean properties (like angles and norms) of the space in the reduced dimensional space are thus helpful to keep an originally "easy" classification problem, still "easy". Consider the following two dimensional sets for the two classes (crosses vs. circles) and two candidate projection directions.



Projection Ψ_a is "better" as it keeps both sets separated and the minimum distance also does not shrink very much in value.

In order to preserve minimum pairwise distance, it suffices if we preserve all pairwise distances (which you prove in homework). Preserving pairwise distances formally means, i.e., that

$$(1-\epsilon)\|x_i - x_j\|^2 \leq \|\Psi(x_i) - \Psi(x_j)\|^2 \leq (1+\epsilon)\|x_i - x_j\|^2, \forall i, j \in [N],$$

where we use $[N]$ as shorthand for all the indices of the data points.

4 Johnson-Lindenstrauss (JL) dimensionality reduction

There are also other two interesting exercises regarding JL/Random Projections in "PCA - Revision" section in solved exercises and in proposed exercises.

84. (CS 189 fall 2015 Introduction to Machine Learning - final, ex.Q8) The dimensions are high! And possibly hard too.

    In this problem, we will derive a famous result called the "Johnson-Lindenstrauss" lemma. Suppose you are given n arbitrary vectors $x^1, \ldots, x^n \in \mathbb{R}^{d\times 1}$. Let $k = 320 \log n$. Now consider a matrix $A \in \mathbb{R}^{k\times d}$ that is obtained randomly in the following manner: every entry of the matrix is chosen independently at random from $\mathcal{N}(0, 1)$. Define vectors $z^1, \ldots, z^n \in \mathbb{R}^{k\times 1}$ as $z^i = \frac{1}{\sqrt{k}} Ax^i$ for every $i \in \{1, \ldots, n\}$.

    (a) For any given $i \in \{1, \ldots, n\}$, what is the distribution of the random vector Ax^i? Your answer should be in terms of the vector x^i.

    (b) For any distinct $i, j \in \{1, \ldots, n\}$, derive a relation between $E[\|A(x^i - x^j)\|_2^2]$ and the value of $\|x^i - x^j\|_2^2$? More points for deriving the relation using your answer from 84a above.

    (c) It can be shown that for any fixed vector v, the random matrix A has the property that

    $$\frac{3}{4}\|v\|_2^2 \leq \|Av\|_2^2 \leq \frac{5}{4}\|v\|_2^2$$

    with probability at least $1 - \frac{1}{n^4}$. Using this fact, show that with probability at least $1 - \frac{1}{n^2}$, every pair (z^i, z^j) simultaneously satisfies $\frac{3}{4}\|x^i - x^j\|_2^2 \leq \|z^i - z^j\|_2^2 \leq \frac{5}{4}\|x^i - x^j\|_2^2$. (Think of how you would bound probabilities of multiple events. Only requires a very basic fact about probability and a basic thought.)

    (d) Describe, in at most two sentences, the usefulness of this result. (Think of n and d as having very large values, for instance, several billions).

    Solution:

    (a) To simplify notation, let $v = x^i$. Clearly, Av is a zero-mean jointly Gaussian vector. Let us compute the covariance: $E[(Av)(Av)^\top]$. Letting $a_j^\top$ denote the j-th row of A, we have that the j-th entry of vector

Av is $a_j^\top v$, and hence the (i,j)-th entry of $(Av)(Av)^\top$ is $(a_i^\top vv^\top a_j)$. It follows that the (i,j)-th entry of $E[(Av)(Av)^\top]$ is $E[a_i^\top vv^\top a_j] = E[v^\top a_i a_j^\top v] = v^\top E[a_i a_j^\top]v$. Now we have $E[a_i a_j^\top] = I$ if $i = j$ and 0 otherwise. Thus the covariance matrix is a diagonal matrix with each entry on the diagonal equal to $\|x^i\|_2^2$.

(b) Without using part 1: $E[\|A(x^i - x^j)\|_2^2] = E[(x^i - x^j)^\top A^\top A(x^i - x^j)] = (x^i - x^j)^\top E[A^\top A](x^i - x^j)$. $E[A^\top A] = kI$ and hence the answer is $k\|(x^i - x^j)\|_2^2$.

Using part 1: Now let $v = x^i - x^j$. Observe that $E[\|Av\|_2^2]$ is simply the sum of the variance of each entry of the vector Av. We computed these variances in part (1) as being equal to $\|v\|_2^2$. Since vector Av has length k, the sum of the variances equals $k\|v\|_2^2$.

(c) Specifically, let E_{ij} denote the event that the pair (z^i, z^j) does not satisfy the above bound. Then letting $v = x^i - x^j$, we have that $P(E_{ij}) \leq \frac{1}{n^4}$. The probability that any of the pairs fail is $P(\cup_{ij} E_{ij}) \leq \sum_{ij} P(E_{ij}) \leq n^2 \frac{1}{n^4} = \frac{1}{n^2}$

(d) Helps in reducing the dimensionality of the feature space and is especially useful for problems where only the pairwise distances need to be preserved.

85. (CS168, Spring 2018, Final Exam, ex.1.c) Consider applying the Johnson-Lindenstrauss transform (i.e., projecting all points onto a set of randomly chosen directions) to a set of n points. Will the projected points have the property that for each of the C_n^3 subsets of 3 points, the area of the triangle spanned by those three points in the original space will be close to the area of the triangle spanned by the projections of the three points? Justify your answer in at most two sentences.

5 Principal Component Analysis - PCA

5.1 PCA

86. (after ESLII.pdf pag 553) **FIRST EXERCISE FOR PCA**

Solve the following exercise:

- **for** $d = 1$
- **for any** d

The principal components of a set of data in $\mathbb{R}^D$ provide a sequence of best linear approximations to that data, of all ranks $d \leq D$.

Denote the observations by $x_1, x_2, \ldots, x_n$ and consider the rank-d linear model for representing them

$$f(\mu, \lambda) = \mu + V_d \lambda,$$

where μ is a location vector in $\mathbb{R}^D$, V_d is a $D \times d$ matrix with d orthogonal unit vectors as columns (i.e., an orthonormal basis for a subspace of $\mathbb{R}^D$; the choice of an orthonormal one can be viewed as due to its advantages), and λ is a d vector of parameters. This is the parametric representation of an affine hyperplane of rank d. The following two figures illustrate for $d = 1$ and $d = 2$.

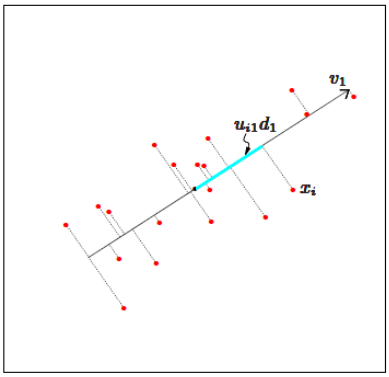


FIGURE 14.20. *The first linear principal component of a set of data. The line minimizes the total squared distance from each point to its orthogonal projection onto the line.*

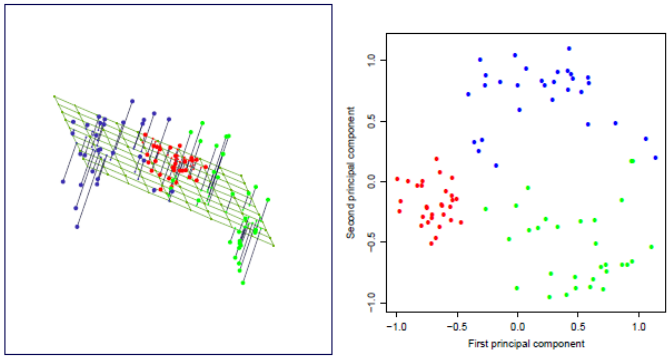


FIGURE 14.21. *The best rank-two linear approximation to the half-sphere data. The right panel shows the projected points with coordinates given by* $\mathbf{U}_2\mathbf{D}_2$*, the first two principal components of the data.*

Fitting such a model to the data by least squares amounts to minimizing the *reconstruction error*

$$\min_{\mu,\lambda\in\mathbb{R}^D;V_d\in\mathbb{R}^{D\times d}:V_d^TV_d=I}\sum_{i=1}^{n}\|x_i-\mu-V_d\lambda_i\|^2$$

(a) Partially optimize for μ and λ and obtain

$$\hat{\mu}=\bar{x}=\frac{x_1+\cdots+x_n}{n}$$

$$\hat{\lambda}_i=V_d^\top(x_i-\bar{x})$$

(b) The result from the previous subpoint leaves us to find the matrix V_d:

$$\min_{V_d\in\mathbb{R}^{D\times d}V_d^\top V_d=I}\sum_{i=1}^{n}\|(x_i-\bar{x})-V_dV_d^\top(x_i-\bar{x})\|^2$$

For convenience, we assume that $\bar{x}=0$ (otherwise we simply replace the observations by their centered versions $\tilde{x}_i=x_i-\bar{x}$).

Solve the new optimization problem.

Hint: Show that this problem is equivalent to another problem whose solution you already know.

(c) Complete the following sentence: Apart from minimizing the reconstruction error, PCA can be interpreted as maximizing ...

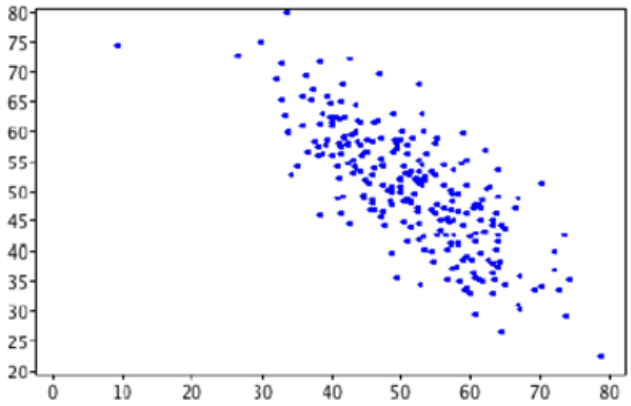
Hint: use the equivalent optimization problem to answer this.

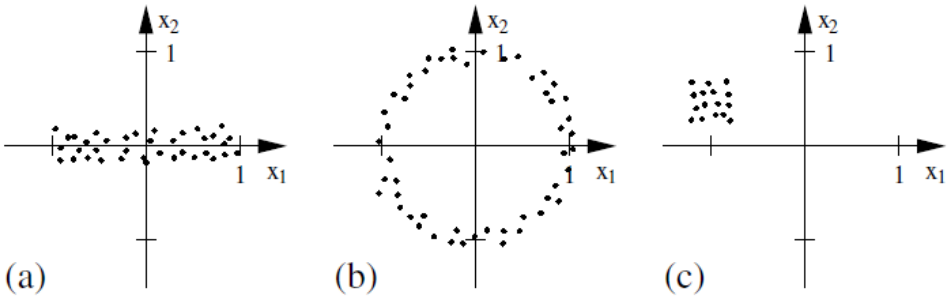
87. Suppose that our dataset consists of two points: $\begin{bmatrix}1\\1\end{bmatrix}$ and $\begin{bmatrix}2\\2\end{bmatrix}$. Draw the points on a grid and find the new coordinates of points **(principal components scores) using mainly the plot** you just drew.

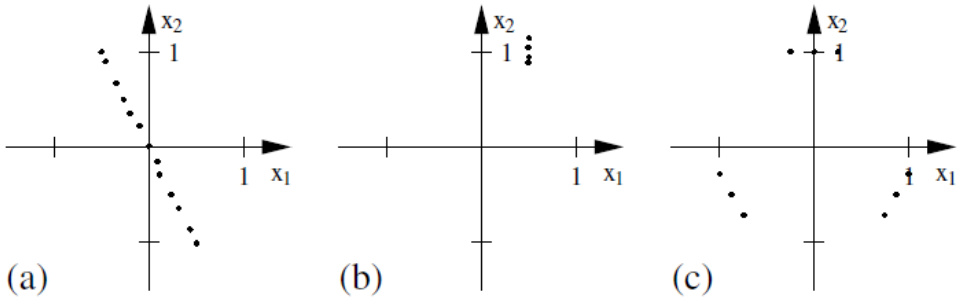
Partial solution: $\begin{bmatrix}-\sqrt{2}/2\\0\end{bmatrix}$ and $\begin{bmatrix}\sqrt{2}/2\\0\end{bmatrix}$

88. (LA4ML - Eigenvectors and Intro to PCA - Worksheet. Part Three. ex.1 - and "Exercises* on Principal Component Analysis Laurenz Wiskott Institut fur Neuroinformatik Ruhr-Universitat Bochum, Germany, EU 4 February 2017, ex.2.15.2 + 2.15.3") For the following data plot, take your best guess and draw the direction vectors of the first and second principal components (the eigenvectors of the covariance matrix).

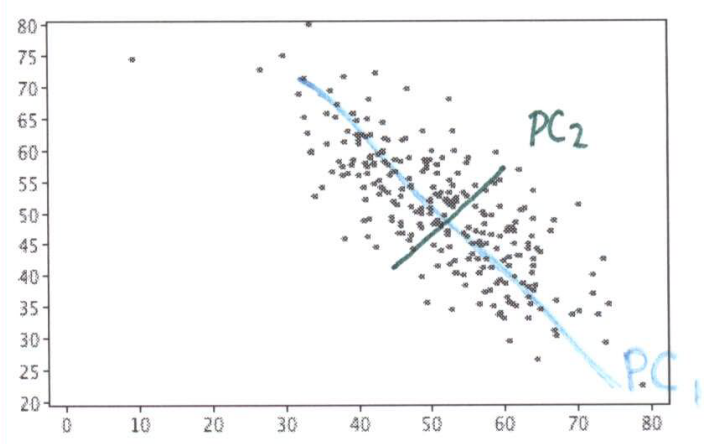
Observation: Principal direction vectors go through the centroid (mean) of points.







Solution:



89. **BY ALINA MUNTEANU** (CMU, 2011 spring, T. Mitchell, HW 5, pr. 2.3)

Consider 3 data points in the 2-d space: $(-1, -1)$, $(0, 0)$, $(1, 1)$.

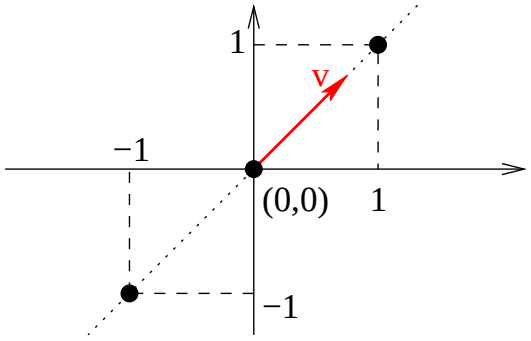
a. What is the first principal component (write down the actual vector)?

b. If we project the original data points into the 1-d subspace by the principal component you choose, what are their coordinates in the 1-d subspace? And what is the variance of the projected data?

c. For the projected data you just obtained above, now if we represent them in the original 2-d space and consider them as the reconstruction of the original data points, what is the reconstruction error?

Solution:

a. From the graphical representation of the points, it is clear that the first principal component is a vector of the form $v = (1,1)^T$. This vector should be normalized to have unit length, so $\mathbf{v} = \frac{1}{||(1,1)^T||}(1,1)^T = \frac{1}{\sqrt{2}}(1,1)^T = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)^T$ is the first principal component.



This value can be computed also by the eigen-decomposition of the covariance matrix (PCA algorithm). The covariance matrix of the data points is:

$$C = \frac{1}{3}\sum_{i=1}^{3}(x_i - \mu)(x_i - \mu)^T = \frac{1}{3}\left[\begin{pmatrix}-1\\-1\end{pmatrix}(-1,-1) + \begin{pmatrix}0\\0\end{pmatrix}(0,0) + \begin{pmatrix}1\\1\end{pmatrix}(1,1)\right]$$

$$C = \frac{1}{3}\left[\begin{pmatrix}1 & 1\\1 & 1\end{pmatrix} + \begin{pmatrix}0 & 0\\0 & 0\end{pmatrix} + \begin{pmatrix}1 & 1\\1 & 1\end{pmatrix}\right] = \begin{pmatrix}2/3 & 2/3\\2/3 & 2/3\end{pmatrix}$$

where $\mu = (0,0)^T$ is the mean of the data: $\mu = (\mu_1, \mu_2)^T$ and $\mu_1 = \mu_2 = \frac{-1+0+1}{3} = 0$.

Next, the eigenvalue of this matrix have to be computed, using the folowing equation:

$$\det(C - \lambda I) = 0 \Leftrightarrow \det\begin{pmatrix}2/3-\lambda & 2/3\\2/3 & 2/3-\lambda\end{pmatrix} = 0 \Leftrightarrow \left(\frac{2}{3} - \lambda\right)^2 - \frac{4}{9} = 0$$

$$\Leftrightarrow \lambda\left(\lambda - \frac{4}{3}\right) = 0 \quad \Rightarrow \lambda_1 = \frac{4}{3}, \lambda_2 = 0$$

The first principal component will be a (normalized) eigenvector corresponding to the greatest eigenvalue ($\lambda_1 = \frac{4}{3}$). This vector satisfies:

$$\left(C - \frac{4}{3}I\right)\mathbf{v} = 0 \Leftrightarrow \begin{pmatrix}-2/3 & 2/3\\2/3 & -2/3\end{pmatrix}\begin{pmatrix}v_1\\v_2\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix} \Leftrightarrow \begin{cases}-\frac{2}{3}v_1 + \frac{2}{3}v_2 = 0\\ \frac{2}{3}v_1 - \frac{2}{3}v_2 = 0\end{cases}$$

$\Rightarrow v_1 = v_2 \Rightarrow$ The first principal component is $\mathbf{v} = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)^T$

b. The projection of the point x_i into the subspace of $\mathbf{v}$ is computed using the formula:

$$z_i = \mathbf{v}^T(x_i - \mu)$$

The projection of $x_1 = (-1, -1)^T$ is: $z_1 = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)\binom{-1}{-1} = -\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2} = -\sqrt{2}$.

The projection of $x_2 = (0, 0)^T$ is: $z_2 = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)\binom{0}{0} = 0$.

The projection of $x_3 = (1, 1)^T$ is: $z_3 = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)\binom{1}{1} = \sqrt{2}$.

The variance of the projected data is $\frac{1}{3}\sum_{i=1}^{3}(z_i - \mu_z)^2 = \frac{1}{3}\sum_{i=1}^{3}(z_i)^2 = \frac{4}{3}$ (the sample mean $\mu_z = \frac{-\sqrt{2} + 0 + \sqrt{2}}{3} = 0$).

c. The reconstruction of the point x_i using the projected data z_i is computed using the formula:

$$\widehat{x_i} = \mathbf{v}z_i + \mu \Rightarrow \begin{cases} \widehat{x_1} = \mathbf{v}z_1 = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)^T - \sqrt{2} = (-1, -1)^T = x_1 \\ \widehat{x_2} = \mathbf{v}z_2 = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)^T 0 = (0, 0)^T = x_2 \\ \widehat{x_3} = \mathbf{v}z_3 = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)^T \sqrt{2} = (1, 1)^T = x_3 \end{cases}$$

As $\widehat{x_i} = x_i,\ \forall i \in \{1, 2, 3\} \Rightarrow$ the reconstruction error is 0.

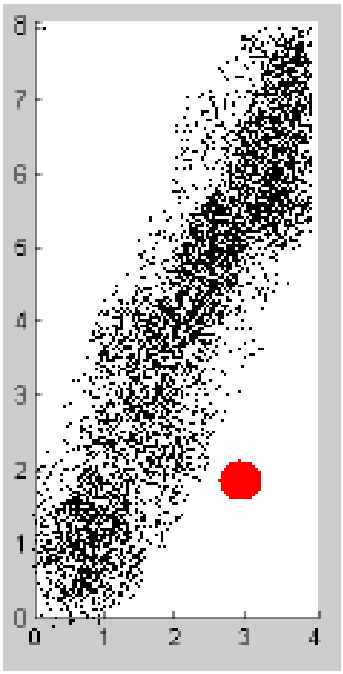
90. **BY ALINA MUNTEANU** (CMU, 2008 spring, T. Mitchell, HW 9, pr. 2)

a. Draw, and clearly label, the first principal component of the small black points shown below (the points were drawn by hand and so may not be exactly correct, but trust the general idea: you should not need to use a computer).

b. Reconstruct the large red point at (x=3, y=2) both graphically, using a vector diagram, and algebraically, showing the form, along with the result, of the reconstruction:

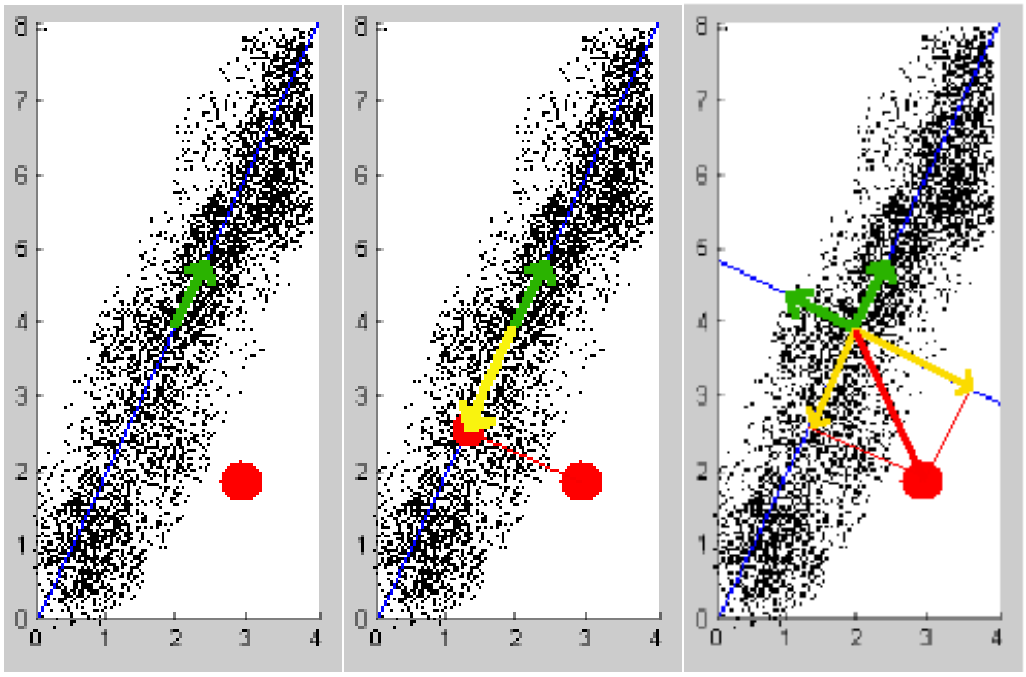
- using only the first principal component

- using the first and second principal components



Solution:

a. The mean of the small black data points is $\mu = (2, 4)^T$ and the direction of the first principal component is that of the line $((0, 0), (4, 8))$. The first principal component is a vector of length 1 that originate from the mean of the distribution. The graphical representation is presented in figure a.



a b1 b2

b. The first principal component is:

$$\mathbf{v}_1 = \frac{1}{||\binom{1}{2}||}\binom{1}{2} = \frac{1}{\sqrt{5}}\binom{1}{2} = \binom{\sqrt{5}/5}{2\sqrt{5}/5}$$

The projection of the point $x = (3,2)^T$, using only the first principal component, is

$$z = \mathbf{v}_1^T(x-\mu) = \left(\frac{\sqrt{5}}{5}, \frac{2\sqrt{5}}{5}\right)\left[\binom{3}{2} - \binom{2}{4}\right] = \left(\frac{\sqrt{5}}{5}, \frac{2\sqrt{5}}{5}\right)\binom{1}{-2} = \frac{\sqrt{5}}{5} - \frac{4\sqrt{5}}{5} = -\frac{3\sqrt{5}}{5}$$

The reconstruction of the point x using the projected data z is

$$\widehat{x} = \mathbf{v}_1 z + \mu = \binom{\sqrt{5}/5}{2\sqrt{5}/5}\left(-\frac{3\sqrt{5}}{5}\right) + \binom{2}{4} = \binom{-3/5}{-6/5} + \binom{2}{4} = \binom{7/5}{14/5}$$

The second principal component is a vector $\mathbf{v}_2$ with length 1, orthogonal to $\mathbf{v}_1$, so:

$$\mathbf{v}_2 = \frac{1}{||\binom{-2}{1}||}\binom{-2}{1} = \frac{1}{\sqrt{5}}\binom{-2}{1} = \binom{-2\sqrt{5}/5}{\sqrt{5}/5}$$

The projection of the point $x = (3, 2)^T$, using $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2)$ the first and second principal components, is

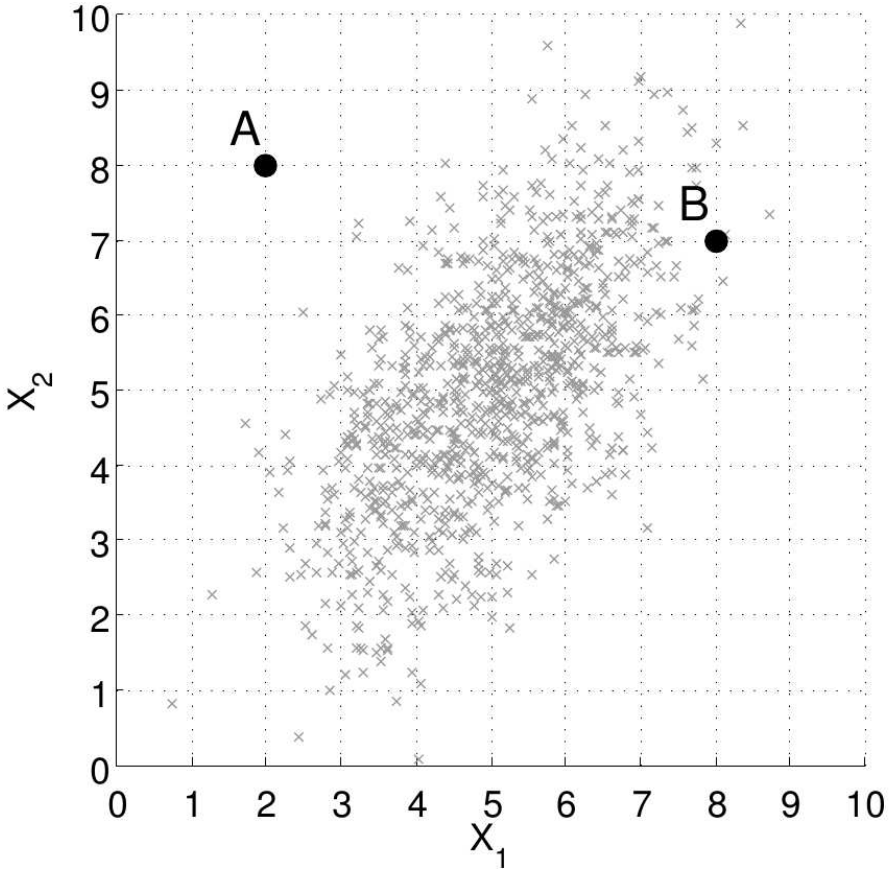
$$z' = \mathbf{v}^T(x - \mu) = \begin{pmatrix} \sqrt{5}/5 & 2\sqrt{5}/5 \\ -2\sqrt{5}/5 & \sqrt{5}/5 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \end{pmatrix} = \left(-\frac{3\sqrt{5}}{5}, -\frac{4\sqrt{5}}{5} \right)^T$$

The reconstruction of the point x using the projected data z' is

$$\widehat{x'} = \mathbf{v}z' + \mu = \begin{pmatrix} \sqrt{5}/5 & -2\sqrt{5}/5 \\ 2\sqrt{5}/5 & \sqrt{5}/5 \end{pmatrix} \left(-\frac{3\sqrt{5}}{5}, -\frac{4\sqrt{5}}{5} \right)^T + \begin{pmatrix} 2 \\ 4 \end{pmatrix} = \begin{pmatrix} 1 \\ -2 \end{pmatrix} + \begin{pmatrix} 2 \\ 4 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \end{pmatrix} = x$$

91. **BY ALINA MUNTEANU** (CMU, 2009 fall, Geoff Gordon, midterm, pr. 10)

Plotted in the figure are two dimensional data drawn from a multivariate Normal (Gaussian) distribution.



a. What is the mean of this distribution? Estimate the answer visually and round to the nearest integer.

b. Would the off-diagonal covariance $\Sigma_{1,2} = Cov(X_1, X_2)$ be: (a) negative (b) positive (c) approximately zero

Define v_1 and v_2 as the directions of the first and second principal component, with $||v_1|| = ||v_2|| = 1$. These directions define a change of basis

$$Z_1 = v_1^T \cdot (X - \mu) \qquad\qquad Z_2 = v_2^T \cdot (X - \mu)$$

c. Sketch and label v_1 and v_2 on the figure. The arrows should originate from the mean of the distribution. You do not need to solve the SVD, instead visually estimate the directions.

d. The covariance $Cov(Z_1, Z_2)$, is: (a) negative (b) positive (c) approximately zero

e. Which point (A or B) would have the higher reconstruction error after projecting onto the first principal component direction v_1 ?

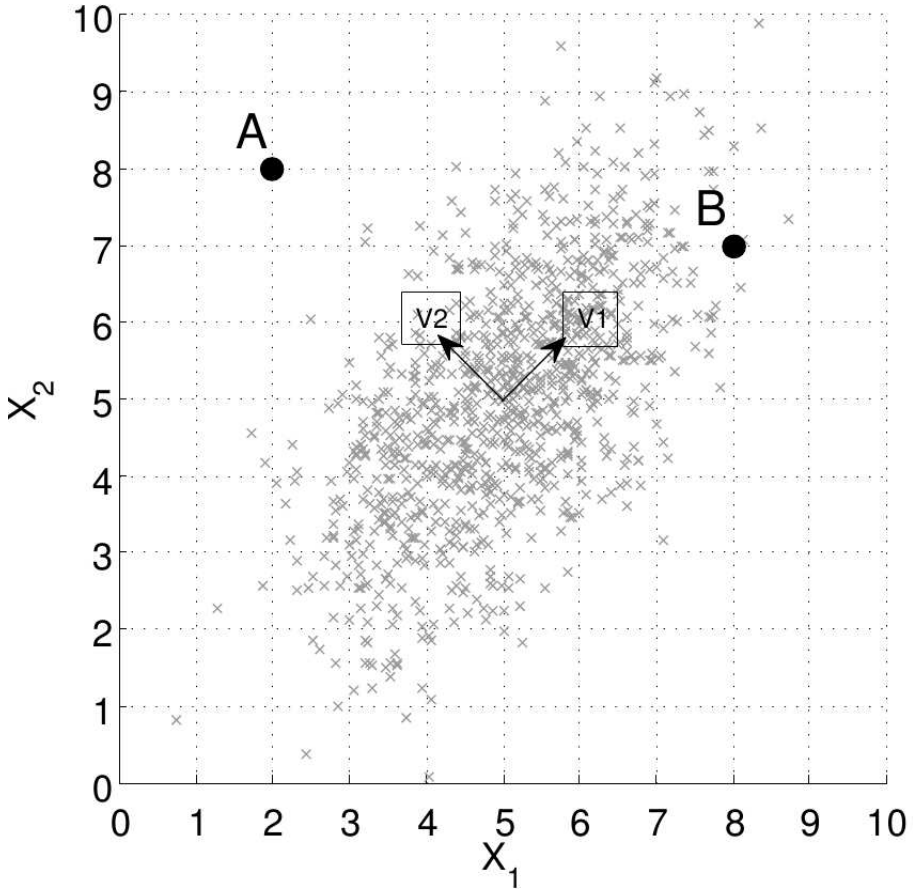
Solution:

a. The mean of the distribution is:

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} = \begin{pmatrix} E[X_1] \\ E[X_2] \end{pmatrix} = \begin{pmatrix} 5 \\ 5 \end{pmatrix}$$

b. The off-diagonal covariance $\Sigma_{1,2} = Cov(X_1, X_2)$ would be positive.

c. The first and second principal component are:



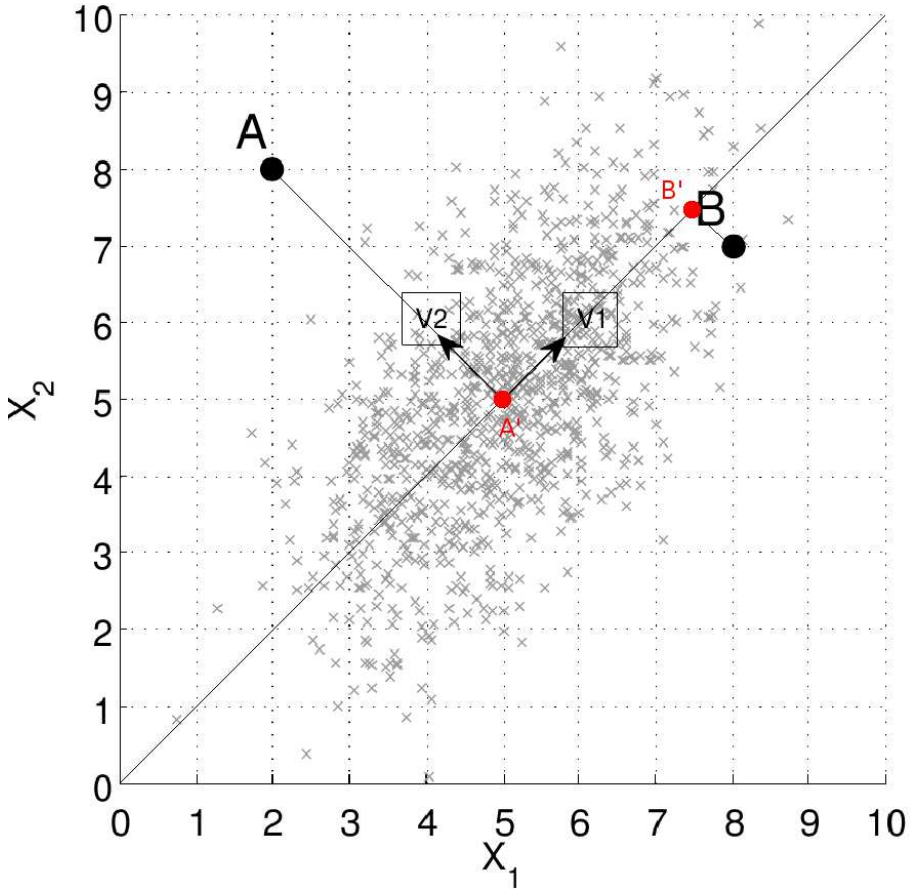
Their numeric values can be calculated as:

$$v_1 = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)^T \qquad v_2 = \left(-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)^T$$

d. The covariance $Cov(Z_1, Z_2)$, is approximately zero, because of the transformation.

e. From the graphically representation, it is clear that point A would have the higher reconstruction error after projecting onto the first principal com-

ponent direction v_1.



The projections of the points $A = (2, 8)^T$ and $B = (8, 7)^T$ onto the first principal component direction v_1 are:

$$Z_{1A} = v_1^T(A - \mu) = \left(-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)\left[\begin{pmatrix}2\\8\end{pmatrix} - \begin{pmatrix}5\\5\end{pmatrix}\right] = \left(-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)\begin{pmatrix}-3\\3\end{pmatrix} = 0$$

$$Z_{1B} = v_1^T(B - \mu) = \left(-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)\left[\begin{pmatrix}8\\7\end{pmatrix} - \begin{pmatrix}5\\5\end{pmatrix}\right] = \left(-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)\begin{pmatrix}3\\2\end{pmatrix} = \frac{5\sqrt{2}}{2}$$

After the reconstruction of the points, we obtain:

$$\widehat{A} = v_1 Z_{1A} + \mu = \begin{pmatrix}5\\5\end{pmatrix}$$

$$\widehat{B} = v_1 Z_{1B} + \mu = \left(-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)^T \frac{5\sqrt{2}}{2} + \begin{pmatrix}5\\5\end{pmatrix} = \left(\frac{5}{2}, \frac{5}{2}\right)^T + \begin{pmatrix}5\\5\end{pmatrix} = \begin{pmatrix}7.5\\7.5\end{pmatrix}$$

The reconstruction errors are: $err(A) = \sum_{i=1}^{2} \left(x_i(A) - x_i(\widehat{A})\right)^2 = (2-5)^2 + (8-5)^2 = 9$ and $err(B) = (8-7.5)^2 + (7-7.5)^2 = 1/4 + 1/4 = 1/2$.

92. **BY ALINA MUNTEANU** (CMU, 2009 spring, Ziv-Bar Joseph, final, pr. 10)

You have the following data:

data	x	y
X_1	5.51	5.35
X_2	20.82	24.03
X_3	−0.77	−0.57
X_4	19.30	19.38
X_5	14.24	12.77
X_6	9.74	9.68
X_7	11.59	12.06
X_8	−6.08	−5.22

You want to reduce the data into a single dimension representation. You are given the first principal component $v_1 = (0.694, 0.720)^T$.

a. What is the representation (projected coordinate) for data $X_1(x = 5.51, y = 5.35)$ in the first principal space?

b. What are the xy coordinates in the original space reconstructed using this first principal representation for data $X_1(x = 5.51, y = 5.35)$?

c. What is the representation (projected coordinate) for data $X_1(x = 5.51, y = 5.35)$ in the second principal space?

d. What is the reconstruction error if you use two principal components to represent original data?

Solution:

a. In order to calculate the projection of the data, we need to compute the mean of the distribution, $\mu = (\mu_x, \mu_y)^T$:

$$\mu_x = \frac{5.51 + 20.82 - 0.77 + 19.30 + 14.24 + 9.74 + 11.59 - 6.08}{8} = 9.293$$
$$\mu_y = \frac{5.35 + 24.03 - 0.57 + 19.38 + 12.77 + 9.68 + 12.06 - 5.22}{8} = 9.685$$

The projected coordinate for data $X_1 = (5.51, 5.35)^T$ in the first principal space is:

$$Z_1 = v_1^T(X_1 - \mu) = (0.694, 0.720)\left[\begin{pmatrix}5.51\\5.35\end{pmatrix} - \begin{pmatrix}9.293\\9.685\end{pmatrix}\right] = (0.694, 0.720)\begin{pmatrix}-3.783\\-4.335\end{pmatrix}$$
$$\Rightarrow Z_1 = -5.746$$

b. The xy coordinates in the original space reconstructed using this first principal representation for data X_1 are:

$$\widehat{X_1} = v_1 Z_1 + \mu = \begin{pmatrix} 0.694 \\ 0.720 \end{pmatrix} (-5.746) + \begin{pmatrix} 9.293 \\ 9.685 \end{pmatrix} = \begin{pmatrix} 5.305 \\ 5.547 \end{pmatrix}$$

c. The second principal component is $v_2 = (v_2^x, v_2^y)^T$, a vector orthonormal to $v_1 = (0.694, 0.720)^T \Leftrightarrow v_1 \cdot v_2 = 0 \Leftrightarrow 0.694 v_2^x + 0.720 v_2^y = 0$. We can choose $v_2 = (-0.720, 0.694)^T$.

The projected coordinate for data $X_1 = (5.51, 5.35)^T$ in the second principal space is:

$$Z_2 = v_2^T (X_1 - \mu) = (-0.720, 0.694) \begin{pmatrix} -3.783 \\ -4.335 \end{pmatrix} = -0.284$$

d. The xy coordinates in the original space reconstructed using two principal components for data X_1 are:

$$\widehat{X_1'} = (v_1, v_2) \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} + \mu = \begin{pmatrix} 0.694 & -0.720 \\ 0.720 & 0.694 \end{pmatrix} \begin{pmatrix} -5.746 \\ -0.284 \end{pmatrix} + \begin{pmatrix} 9.293 \\ 9.685 \end{pmatrix} = \begin{pmatrix} 5.509 \\ 5.350 \end{pmatrix}$$

The reconstruction error in this case is

$$\begin{aligned} err(X_1) &= \left(x(X_1) - x(\widehat{X_1'}) \right)^2 + \left(y(X_1) - y(\widehat{X_1'}) \right)^2 = \\ &= (5.51 - 5.509)^2 + (5.35 - 5.350)^2 = 0.001^2 = 10^{-6} \end{aligned}$$

and is due to the approximations.

93. **BY ALINA MUNTEANU**(CMU, 2009 spring, Ziv-Bar Joseph, Eric Xing, HW5, pr. 5)

Consider the covariance matrix for a Gaussian with mean $= (0, 0)$ and variance $= \sigma^2 \cdot I_2$ where σ^2 is a positive constant, and I_2 is a 2×2 identity matrix.

a. What are the two principle components for this matrix? What are their eigenvalues?

b. Given a data point $p = (x, y)$ from this distribution, what is the reconstructed data using the projection onto the first principal component of this matrix?

c. For this reconstructed value, what is the expected value of the reconstruction error (squared error between the true value and reconstructed value).

Solution:

a. In order to compute the principal components, we have to find the eigenvalues (λ) and eigenvectors ($\mathbf{v}$) of the covariance matrix ($\Sigma = \sigma^2 \cdot I_2$) after mean ($\mu$) centering the data for each attribute. Since $\mu = 0$, we can find the eigenvalues by solving:

$$\det(\Sigma - \lambda I_2) = 0 \Leftrightarrow \det \begin{pmatrix} \sigma^2 - \lambda & 0 \\ 0 & \sigma^2 - \lambda \end{pmatrix} = 0 \Leftrightarrow (\sigma^2 - \lambda)^2 = 0$$

$$\Rightarrow \lambda_1 = \lambda_2 = \sigma^2 \text{ are the eigenvalues of the covariance matrix } \Sigma$$

The eigenvector $\mathbf{v} = (v_x, v_y)^T$ will be any vector that satisfies:

$$(\Sigma - \lambda I_2) \cdot \mathbf{v} = 0 \Leftrightarrow \begin{pmatrix} \sigma^2 - \lambda & 0 \\ 0 & \sigma^2 - \lambda \end{pmatrix} \begin{pmatrix} v_x \\ v_y \end{pmatrix} = 0$$

Since $\lambda_1 = \lambda_2 = \sigma^2$, we have that

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} v_x \\ v_y \end{pmatrix} = 0 \Leftrightarrow 0 \cdot v_x + 0 \cdot v_y = 0$$

To ensure orthonormality of the components, we select the following vectors $\mathbf{v}_1 = (1, 0)^T$ and $\mathbf{v}_2 = (0, 1)^T$. (Because the Gaussian is symmetric around the origin, any two mutually perpendicular axes may be used. We can use the original axes for convenience.)

b. Given the properties of our covariance matrix, both components (i.e. $\mathbf{v}_1$ and $\mathbf{v}_2$) are equally important. If we project our data point p into $\mathbf{v}_1$ we obtain:

$$z_1 = \mathbf{v}_1^T \cdot (p - \mu) = (1, 0) \cdot \begin{pmatrix} x \\ y \end{pmatrix} = x$$

If we reconstruct our data point, we obtain:

$$\widehat{p} = \mathbf{v}_1 \cdot x + \mu = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot x = \begin{pmatrix} x \\ 0 \end{pmatrix}$$

c. The squared error between the true value (p) and the reconstructed value ($\widehat{p}$) is:

$$err = (x - x)^2 + (0 - y)^2 = y^2$$

Give the properties of the covariance, $Cov(x, y) = E(xy) - E(x)E(y)$, the expected value of this error is:

$$E(err) = E(y^2) = Cov(y, y) + E(y)^2 = \sigma^2 + 0 = \sigma^2$$

94. **BY ALINA MUNTEANU** (CMU, 2010 fall, Aarti Singh, HW 5, pr. 2.1)

An example of raw dataset, D_1, has 64 records from 64 different users, that is $n = 64$. Each record in D_1 is a vector of length 6830, in other words, 6830 features $p = 6830$. If we do Principal Component Analyses (PCA) of this dataset D_1, how many principal components with non-zero variance would we get? Explain why?

Solution:

Because, when $n < p$, it is necessarily true that the data lie on an n-dimensional subspace of the p-dimensional feature space, so there are only $n = 64$ orthogonal directions along which the data can vary at all. Alternately, we can imagine that there are $6830 - 64 = 6766$ other principal components, each with zero variance.

95. **BY ALINA MUNTEANU** (CMU, 2010 fall, Aarti Singh, HW 5, pr. 2.2)

D_2 is a data set about the information on each of US state. Therefore, we have $n = 50$. For each state, there are eight features, which is described in Table 1:

Table 1: Descriptions of the Eight Attributes

Attribute	Explanation
Population	in thousands
Income	dollars per capita
Illiteracy	Percent of the adult population unable to read and write
Life Exp	Average years of life expectancy at birth
Murder	Number of murders and non-negligent manslaughters per 100,000 people
HS Grad	Percent of adults who were high-school graduates
Frost	Mean number of days per year with low temperatures below freezing
Area	In square miles

In the following, we will do two different principal component analyses (PCAs) of this dataset D_2. The two PCAs are called as PCA_1 and PCA_2 respectively. The only difference between the two PCAs is that one has the variables standardized to variance 1 before calculating the covariance matrix and its eigenvalues, the other does not. The summary statistics for these variables (without standardizing to variance 1) are listed in Table 2:

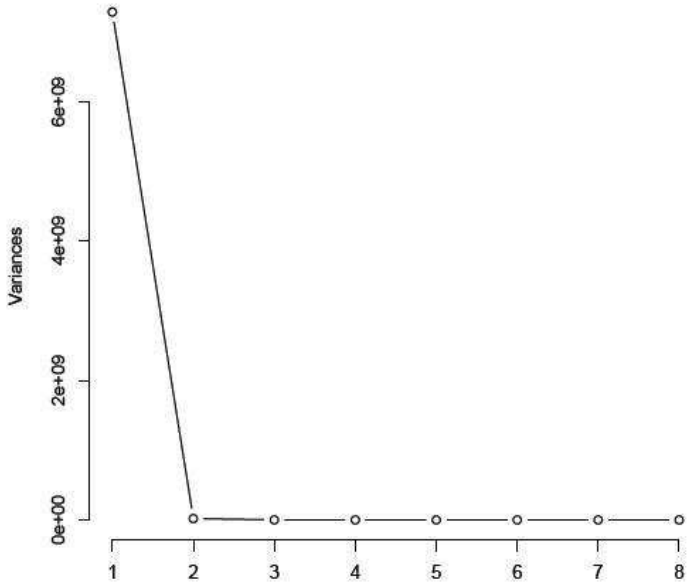
Table 2: Summary Statistics for the Eight Attributes

Attribute	Min	Median	Mean	Max
Population	365	2838	4246	21198
Income	3098	4519	4436	6315
Illiteracy	0.500	0.950	1.170	2.800
Life Exp	67.96	70.67	70.88	73.60
Murder	1.400	6.850	7.378	15.100
HS Grad	37.80	53.25	53.11	67.30
Frost	0.00	114.50	104.46	188.00
Area	1049	54277	70736	566432

Generally speaking, determining which principal components account for which parts of the variance can be done by looking at a Scree Plot. A Scree Plot is a simple line segment plot that shows the fraction of total variance in the data as explained or represented by each principal component(PC). The PCs are ordered, and by definition are therefore assigned a number label, by decreasing order of contribution to total variance. The PC with the largest fraction contribution is labeled with the label name. Such a plot when read left-to-right can often show a clear separation in fraction of total variance where the 'most important' components cease and the 'least important' components begin. The point of separation is often called the 'elbow'. (In the PCA literature, the plot is called a **'Scree' Plot** because it often looks like a 'scree' slope, where rocks have fallen down and accumulated on the side of a mountain.)

The Figures and Tables following show some displays for the PCA_1 and PCA_2 respectively, which you will need to use to answer the following questions.

The Scree Plot of PCA_1 is displayed in the folowing figure and projections of the features on to the first two PCs are listed in Table 3.

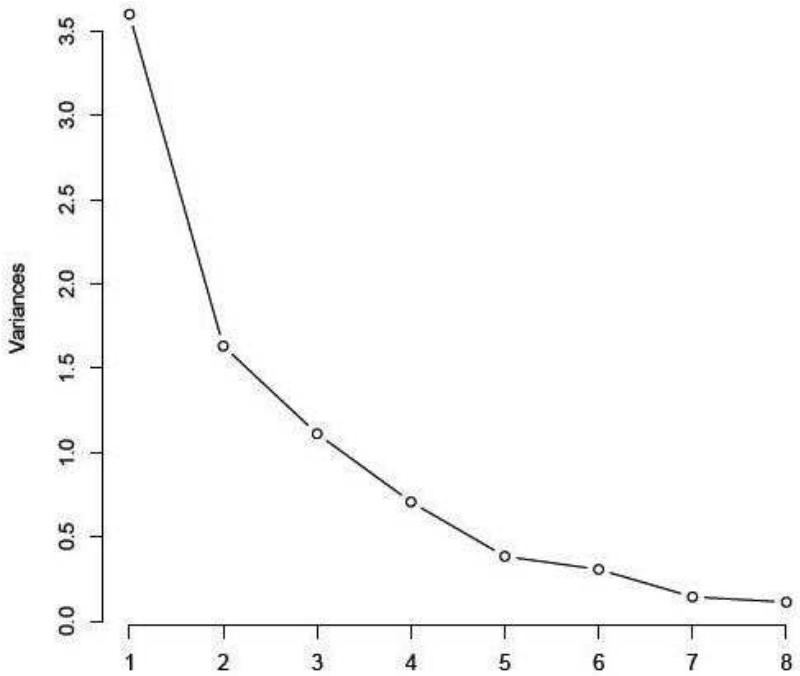


Scree plot forPCA_1

Table 3: Projections of the features on to the first two principal components of PCA_1.

Attribute	PC1	PC2
Population	1.18×10^{-3}	-1.00
Income	2.62×10^{-3}	-2.8×10^{-2}
Illiteracy	5.52×10^{-7}	-1.42×10^{-5}
Life Exp	-1.69×10^{-6}	1.93×10^{-5}
Murder	9.88×10^{-6}	-2.79×10^{-4}
HS Grad	3.16×10^{-5}	1.88×10^{-4}
Frost	3.61×10^{-5}	3.87×10^{-3}
Area	1.00	1.26×10^{-3}

The Scree Plot of PCA_2 is displayed in the folowing figure and projections of the features on to the first two PCs are listed in Table 4.



Scree plot forPCA_2

Table 4: Projections of the features on to the first two principal components of PCA_2.

Attribute	PC1	PC2
Population	0.1260	0.4110
Income	−0.2990	0.5190
Illiteracy	0.4680	0.0530
Life Exp	−0.4120	−0.0817
Murder	0.4440	0.3070
HS Grad	−0.4250	0.2990
Frost	−0.3570	−0.1540
Area	−0.0334	0.5880

a. Recall the only difference between the PCA_1 and PCA_2 is that one has the variables standardized to variance 1 before calculating the covariance matrix and its eigenvalues, the other does not. Based on the Scree Plots and Projections of the features on to the first two principal components tables

for PCA_1 and PCA_2 respectively, which one has the variables standardized to variance 1 before calculating the covariance matrix and its eigenvalues? Explain briefly.

b. From the Scree Plot of PCA_1, where is a "big gap" or elbow and what is the reasonable number of principal components to be retained?

c. Describe, in words, the first two principal components of PCA_1. (Describe which are the features most relevant to each PC and which are characteristics of the variance in the data was captured by each PC.)

d. From the Scree Plot of PCA_2, where is a "big gap" or elbow and what is the reasonable number of principal components to be retained?

e. Describe, in words, the first two principal components of PCA_2.(Describe which are the features most relevant to each PC and which are characteristics of the variance in the data was captured by each PC.)

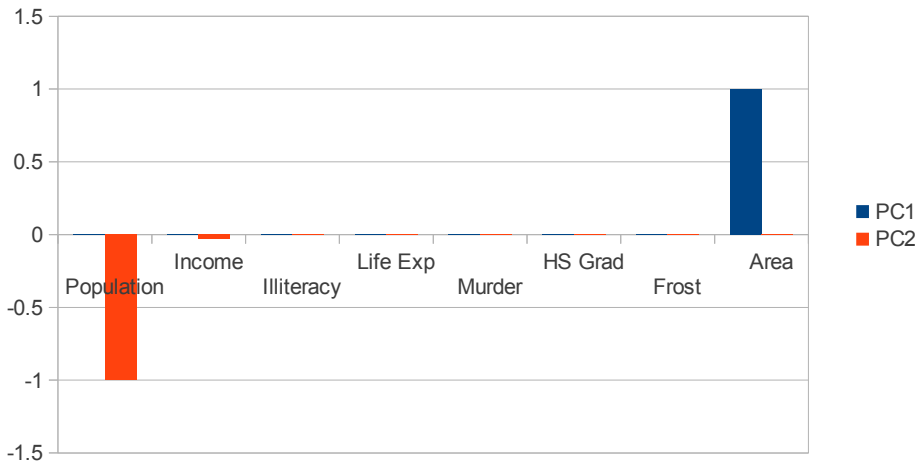
f. Would you rather do PCA_2 or PCA_1 for the PCA analysis? Pick one and explain your choice. (A choice with no or inadequate reasoning will get little or no credit.)

Solution:

a. PCA_2 is the one that has the variables standardized to variance 1 before calculating the covariance matrix and its eigenvalue.

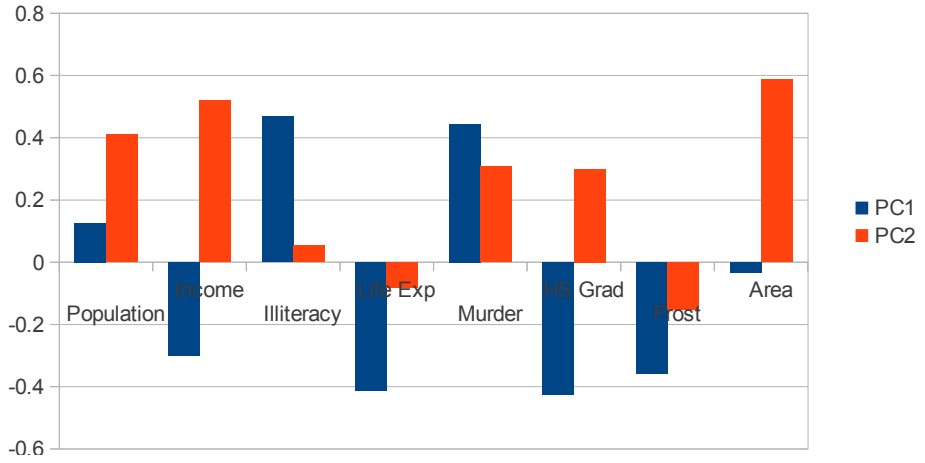
b. Between PC1 and PC2 there is a "big gap". So we only need on PC1. The rest 7 PCs are discarded.

c. From Table 1, PC1 is almost exactly the state's area, and PC2 almost exactly its population.

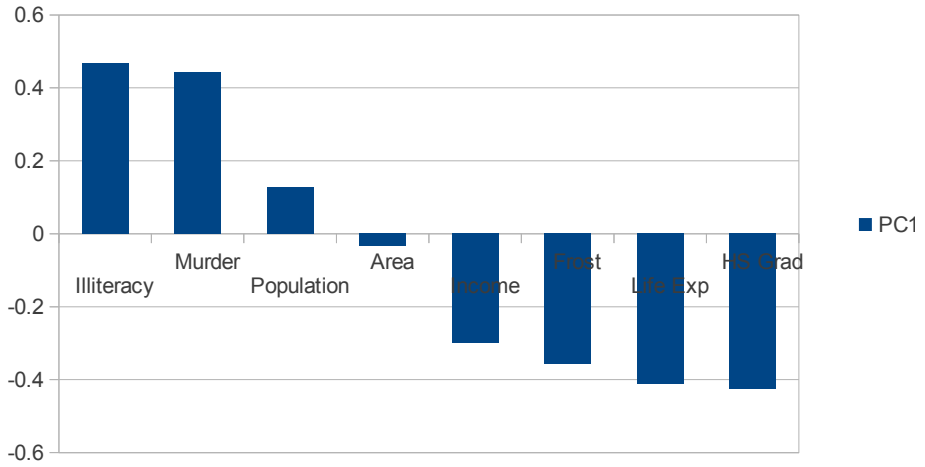


d. The elbow occurs between PC6 and PC7. So we only need the first six PC. The rest 2 PCs are discarded. (We also accept the following answer as long as the description is right - There is a big gap (elbow) between PC4 and PC5. So we only need the first four PC. The rest 4 PCs are discarded.)

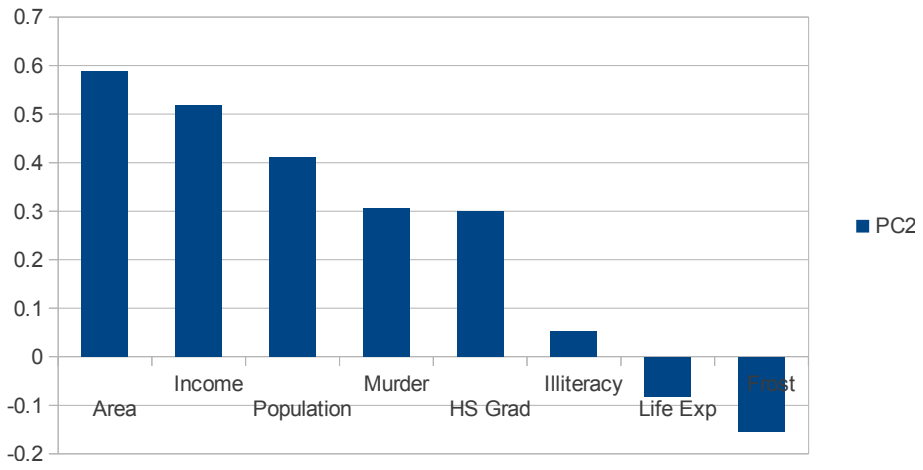
e. The representation of the data from table 2:



PC1 is high for states with high murder, high illiteracy, low education, and little frost. It separates warm, violent, ignorant states from cold, peaceful, educated ones.



PC2 is high for large, populous, rich states, especially if they tend to be more violent and more educated than the norm. It separates big, rich states from small, poor ones.

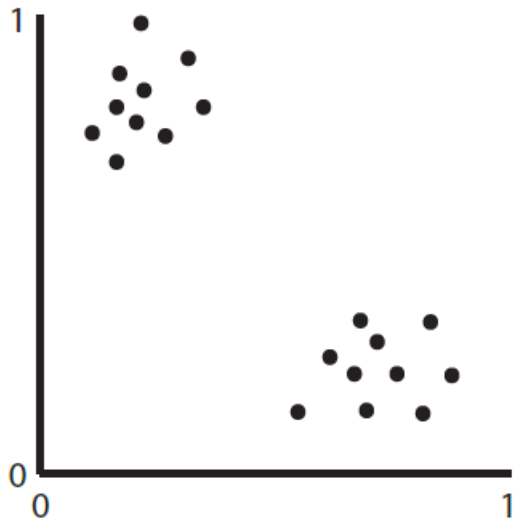


f. PCA_2 is a better choice . The features are not on the same scale, so P CA1 is very weird. PC1 for PCA_1 is almost exactly the state's area, just because the numbers there are immensely bigger than all the other features. (Look at the table of summary statistics.) PC2 is basically population, for the same reason. PCA_2 , on the other hand, is scaled to make the features comparable in size, and shows more interesting patterns.

96. (after CMU, 2017f, NBalcan, class test 3, pr. 2) **PCA vs uncentered PCA** Consider the following plot of data.

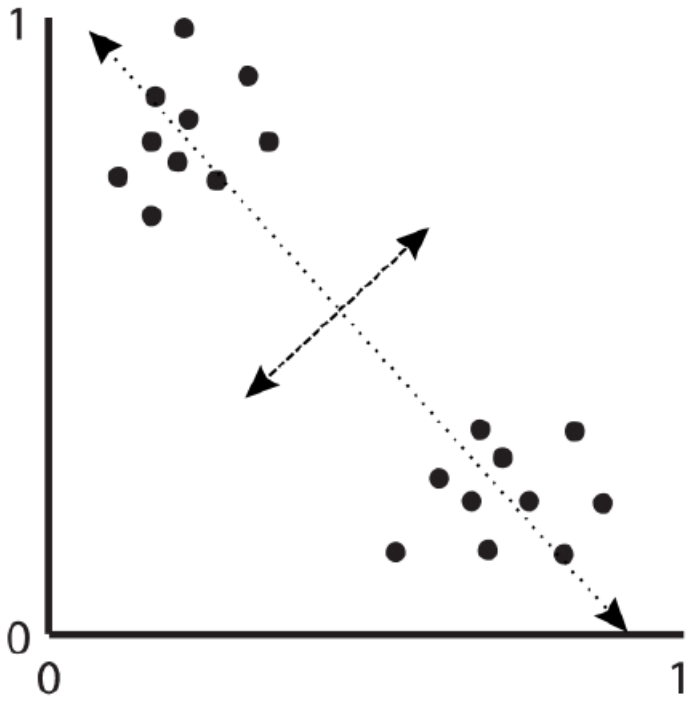
    (a) Draw arrows from the mean of the data to denote the direction and relative magnitudes of the principal components. (PCA = best **affine** subspace)

    (b) Draw arrows from the origin to denote the direction and relative magnitudes of the principal components when data was not centered. (uncentered PCA = best **linear** subspace)



Solution:

(a)



97. (LA4ML - Principal Components Analysis - Worksheet. Part One. ex.1) Suppose we have a dataset with 8 variables and we use standardized data (i.e., **correlation PCA**). What is the total amount of variance in our data?

Solution: 8

98. (CMU, 2016f, NBalcan, MGormley, HW5, pr.2.1) **Z-scoring before applying PCA?**

Consider the $3 \times N$ matrix X be a dataset with 3 variables and N samples. PCA follows these steps when applied to X:

(a) recenter X such that the sample mean $\frac{1}{N}\sum_{n=1}^{N} X_n = \mathbf{0}$

(b) identify ordered orthonormal principal components $u_1, u_2, u_3 \in \mathbb{R}^3$

(c) project recentered X onto the principal components: $u_i^T X$

(d) identify ordered variances $\sigma_1, \sigma_2, \sigma_3$ for the principal components, where $\sigma_i = \frac{1}{N}\sum_{n=1}^{N}(u_i^T X_n)^2$ and $\sigma_1 \geq \sigma_2 \geq \sigma_3$

We can also think about PCA as performing an eigendecomposition on the 3×3 covariance matrix Σ. Given the eigenvectors u_1, u_2, u_3 (i.e., the basis vectors), we can compute each eigenvalue (i.e., the variances) as $\sigma_i = u_i^T \Sigma u_i$. PCA orders the eigenvalues such that $\sigma_1 \geq \sigma_2 \geq \sigma_3$.

Now consider drawing the columns of $X = [X_1, \ldots, X_N]$ from the given multivariate Gaussian distribution:

$$X_n \sim \mathcal{N}\left(\mu = \begin{bmatrix} 5 \\ 5 \\ 10 \end{bmatrix}, \Sigma = \begin{bmatrix} 6 & -3 & 0 \\ -3 & 6 & 0 \\ 0 & 0 & 10 \end{bmatrix}\right)$$

We apply PCA to X and ask the following questions.

Note: For the following problems, feel free to use Matlab's built-in `pca`, `mvnrnd`, `cov2corr`, etc., to help guide your answers. Use $N \geq 1000$ samples when generating the data X. Do not include code in your answers. Provide analytical, exact solutions; do *not* give numerical solutions from Matlab.

Hint: The fastest way to solve these problems is with intuition and to check with Matlab.

(a) What are the ordered variances $\sigma_1, \sigma_2, \sigma_3$ for the ordered principal components?

(b) What are the ordered basis vectors $u_1, u_2, u_3 \in \mathbb{R}^3$ for the ordered principal components?

One limitation of PCA is that it is scale variant. For example, consider two variables, where one represents age (in years) and one represents height (in feet). Because the range of ages (0 to 100 years) is much larger than the range of height (4 to 7 feet), PCA will trivially identify the age variable for the first principal component. However, the first principal component should be able to reconstruct both age and height, which are likely correlated.

To overcome this, a typical procedure before applying PCA is to z-score each variable (i.e., for each x_i, subtract its mean and divide by its standard deviation). This transforms the covariance matrix Σ into a correlation matrix $\tilde{\Sigma}$. Note that $\hat{\Sigma}_{ij} = \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii}}\sqrt{\Sigma_{jj}}}$.

(c) Let's say we z-score X to obtain $\tilde{X}$. Then, $\tilde{X} = [\tilde{x}_1, \tilde{x}_2, \tilde{x}_3]^T \sim \mathcal{N}(\tilde{\mu}, \tilde{\Sigma})$. What are the elements of $\tilde{\mu}$ and $\tilde{\Sigma}$?

We now apply PCA to $\tilde{X}$.

(d) What are the ordered variances $\tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{\sigma}_3$ for the ordered principal components?

(e) What are the ordered basis vectors $\tilde{u}_1, \tilde{u}_2, \tilde{u}_3 \in \mathbb{R}^3$ for the ordered principal components?

(f) Do your answers for 4 and 5 differ from your answers to 1 and 2? Why or why not?

99. (LA4ML - Principal Components Analysis - Worksheet. Part One. ex.2) Suppose I have a dataset with 3 variables and the eigenvalues of the covariance matrix are

$\lambda_1 = 3, \lambda_2 = 2, \lambda_3 = 1$

(a) What **proportion of variance is explained** by the first principal component?

(b) What is the variance of the second principal component?

(c) What proportion of variance is captured by using both the first and second principal components?

Solution:

(a) 1/2

(b) 2

(c) 5/6

100. Suppose I have a dataset with 3 variables and the highest eigenvalue (of the covariance matrix) in absolute value is $\lambda = 4$ (computed via the *power iteration* algorithm). The variances of the initial three variables were: 1,5,10.

What **proportion of variance is explained** by the first principal component? **Do we have enough information to compute this?**

Solution: Yes. $\frac{4}{1+5+10}$

101. (after LA4ML - Principal Components Analysis - Worksheet. Part One. ex.3)

(a) The following output is produced after running PCA on the iris dataset:

Eigenvectors				
	Prin1	Prin2	Prin3	Prin4
Sepal_Length	0.521066	0.377418	-.719566	-.261286
Sepal_Width	-.269347	0.923296	0.244382	0.123510
Petal_Length	0.580413	0.024492	0.142126	0.801449
Petal_Width	0.564857	0.066942	0.634273	-.523597

Write the **two systems of linear equations** that can be induced from the table above.

(b) The following output is produced after running PCA on the iris dataset:

Eigenvectors		
	Prin1	Prin2
Sepal_Length	0.521066	0.377418
Sepal_Width	-.269347	0.923296
Petal_Length	0.580413	0.024492
Petal_Width	0.564857	0.066942

Write the **single system of linear equations** that can be induced from the table above.

102. (LA4ML - Principal Components Analysis - Worksheet. Part One. ex.3) (**PCA interpretation**) The following output is produced after running PCA on the iris dataset:

Eigenvalues of the Correlation Matrix				
	Eigenvalue	Difference	Proportion	Cumulative
1	2.91849782	2.00446735	0.7296	0.7296
2	0.91403047	0.76727360	0.2285	0.9581
3	0.14675688	0.12604204	0.0367	0.9948
4	0.02071484		0.0052	1.0000

Eigenvectors				
	Prin1	Prin2	Prin3	Prin4
Sepal_Length	0.521066	0.377418	-.719566	-.261286
Sepal_Width	-.269347	0.923296	0.244382	0.123510
Petal_Length	0.580413	0.024492	0.142126	0.801449
Petal_Width	0.564857	0.066942	0.634273	-.523597

(a) How much variance is the data is captured by a projection onto the span of the first three principal components?

(b) Which variable is most closely associated with PC2?

(c) Observations that have larger scores on PC3 are somewhat likely to have larger/smaller than average sepal lengths? (*circle one*)

(d) If you had to reduce the dimensions of this data down to 2 variables, which variables would you choose?

(e) What is the total amount of variance for this example? How do you know?

Solution:

(a) 99%

(b) sepal width

(c) smaller

(d) principal component 1 and principal component 2

(e) 4. The top box says eigenvalues of correlation matrix so this is correlation PCA and there are 4 variables in our data.

103. ("Exercises* on Principal Component Analysis Laurenz Wiskott Institut fur Neuroinformatik Ruhr-Universitat Bochum, Germany, EU 4 February 2017, ex.2.15.4") Given some data in $\mathbb{R}^3$ with the corresponding 3×3 second-moment matrix C with eigenvectors c_α and eigenvalues λ_α, with $\lambda_1 = 3$, $\lambda_2 = 1$, $\lambda_3 = 0.2$.

(a) Define a matrix $A \in \mathbb{R}^{2\times3}$ that maps the data into a two-dimensional space while preserving as much variance as possible.

(b) Define a matrix $B \in \mathbb{R}^{3\times2}$ that places the reduced data back into $\mathbb{R}^3$ with minimal reconstruction error. How large is the reconstruction error?

(c) Prove that AB is an identity matrix. Why would one expect that intuitively?

(d) Prove that BA is a projection matrix but not the identity matrix.

Solution:

(a) The dimension with least variance is spanned by the eigenvector of λ_3. The two-dimensional subspace with largest variance is spanned by the eigenvectors of λ_1 and λ_2. The corresponding matrix reads

$$A := \begin{bmatrix} c_1^\top \\ c_2^\top \end{bmatrix}$$

(b) Embedding the reduced data back into the $\mathbb{R}^3$ is done again with the eigenvectors.

$$B := (c_1, c_2)$$

The reconstruction error is the sum over eigenvalues of the neglected eigenvectors, which if $\lambda_3 = 0.2$ in this case.

(c) Intuitively the matrix AB corresponds to a mapping from $\mathbb{R}^2$ into $\mathbb{R}^3$ and back again. No information is lost in this process, which means that AB should be the identity matrix. We can also show formally that

$$AB = \begin{bmatrix} c_1^\top \\ c_2^\top \end{bmatrix} (c_1, c_2) = \begin{bmatrix} c_1^\top c_1 & c_1^\top c_2 \\ c_2^\top c_1 & c_2^\top c_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

(d) The defining property of a projection matrix is that applying it twice is the same as applying it once. Thus, we verify

$$(BA)(BA) = B(AB)A = B(1)A = BA$$

To show that BA is not the identity matrix we multiply it with the third eigenvector.

$$BAc_3 = (c_1, c_2) \begin{bmatrix} c_1^\top \\ c_2^\top \end{bmatrix} c_3$$

$$= (c_1, c_2) \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ since } c_3 \text{ is orthogonal to } c_1 \text{ and } c_2$$

$$= 0$$

$$\neq c_3$$

Thus, BA cannot be the identity matrix.

Alternatively one can argue that B is 3×2 and A is 2×3. Thus, BA can have at most rank 2 and therefore cannot be the identity matrix.

5.2 PCA and SVD

104. (CMU, 2012s, ZBarJoseph, HW5, ex. 2) In linear algebra, the singular value decomposition (SVD) is a factorization os a real matrix X as:

$$X = USV^\top$$

If the dimension of X is $m \times n$, where without loss of generality $m \geq n$, U is an $m \times n$ matrix, S is an $n \times n$ diagonal matrix and $V^\top$ is also an $n \times n$

matrix. Furthermore, U and V are orthonormal matrices: $UU^\top = I$ and $VV^\top - I$.

Consider a dataset of observations $\{x_n\}$ where $n = 1, \ldots, N$. We assume that the examples are zero-centered such that $\overline{x} = \sum_{n=1}^{N} x_n = 0$. PCA algorithm computes the covariance matrix:

$$S = \frac{1}{N} \sum_{n=1}^{N} x_n x_n^\top$$

The principal components $\{u_i\}$ are eigenvectors of S.

Let $X = [x_1, \ldots, x_N]$, a $D \times N$ matrix where each column is one example x_n. If $US'V^\top$ is a SVD of X.

(a) Show that the principal components $\{u_i\}$ are columns of U. This shows the relationship between PCA and SVD.

(b) When the number of dimensions is much larger than the number of datapoints ($D >> N$), is it better to do PCA by using the covariance matrix or using SVD?

(c) Consider the following data set:

$$D = \begin{bmatrix} 1 & 1 & 1 \\ \epsilon & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{bmatrix}$$

where ϵ is a tiny number. Each column is one example. First zero-center the data set and then do PCA using two techniques: 1) by using the covariance matrix and 2) by using SVD. What do you observe?

Solution:

(a)

$$S = XX^\top$$

The principal components $\{u_i\}$ are eigenvectors of S, which are vectors such that: $Su_i = \lambda_i u_i$.
$X = US'V^\top$ so,

$$\begin{aligned} S &= XX^\top \\ &= US'V^\top VS'U^\top \\ &= US'^2U^\top \end{aligned}$$

Therefore, the columns of U are eigenvectors of S.

(b) Assume that the complexity of SVD is $O(ND\min(N,D))$ and the complexity of solving eigenvector problem is $O(D^3)$, we should use SVD.

(c) First zero-center the data,

$$\overline{D} = \begin{bmatrix} 0 & 0 & 0 \\ \frac{2\epsilon}{3} & -\frac{1\epsilon}{3} & -\frac{1\epsilon}{3} \\ -\frac{1\epsilon}{3} & \frac{2\epsilon}{3} & -\frac{1\epsilon}{3} \\ -\frac{1\epsilon}{3} & -\frac{1\epsilon}{3} & \frac{2\epsilon}{3} \end{bmatrix}$$

The covariance matrix is,

$$S = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{2\epsilon^2}{3} & -\frac{1\epsilon^2}{3} & -\frac{1\epsilon^2}{3} \\ 0 & -\frac{1\epsilon^2}{3} & \frac{2\epsilon^2}{3} & -\frac{1\epsilon^2}{3} \\ 0 & -\frac{1\epsilon^2}{3} & -\frac{1\epsilon^2}{3} & \frac{2\epsilon^2}{3} \end{bmatrix}$$

This matrix is singular so it does not have an eigendecomposition. However, we still can do SVD of X.

105. (CMU, 2008f, EXing, final exam, ex.3.2) Given 6 data points in 5-d space, $(1,1,1,0,0)$, $(-3,-3,-3,0,0)$, $(2,2,2,0,0)$, $(0,0,0,-1,-1)$, $(0,0,0,2,2)$, $(0,0,0,-1,-1)$. We can represent these data points by a 6×5 matrix X, where each row corresponds to a data point:

$$X = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ -3 & -3 & -3 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & -1 & -1 \end{bmatrix}$$

(a) What is the sample mean of the data set?

(b) What is SVD of the data matrix X you choose?
hints: The SVD for this matrix must take the following form, where a, b, c, d, σ_1, σ_2 are the parameters you need to decide.

$$X = \begin{bmatrix} a & 0 \\ -3a & 0 \\ 2a & 0 \\ 0 & b \\ 0 & -2b \\ 0 & b \end{bmatrix} \times \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \times \begin{bmatrix} c & c & c & 0 & 0 \\ 0 & 0 & 0 & d & d \end{bmatrix}$$

(c) What is first principle component for the original data points?

(d) If we want to project the original data points into 1-d space by principle component you choose, what is the variance of the projected data?

(e) For the projected data in the previous subpoint, now if we represent them in the original 5-d space, what is the reconstruction error?

Solution:

(a) [0.0.0.0.0]

(b)

$$a = \pm 1/\sqrt{14} = \pm 0.267$$
$$b = \pm 1/\sqrt{6} = \pm 0.408$$
$$\sigma_1 = 1/(a \cdot c) = \sqrt{42} = 6.48$$
$$\sigma_2 = 1/(b \cdot d) = \sqrt{12} = 3.46$$
$$c = \pm 1/\sqrt{3} = \pm 0.577$$
$$d = \pm 1/\sqrt{2} = \pm 0.707$$

(c) $pc = \pm[c, c, c, 0, 0] = \pm[0.577, 0.577, 0.577, 0, 0]$ (Intuition: First, we want to notice that the first three data points are co-linear, and so do the last three data points. And also the first three data points are orthogonal to the rest three data points. Then, we want notice that the norm of the first three are much bigger than the last three, therefore, the first pc has the same direction as the first three data points)

(d) $var = \sigma_1^2/6 = 7$ (Intuition: we just the keep the first three data points, and set the rest three data points as $[0, 0, 0, 0, 0]$ (since they are orthogonal to pc), and then compute the variance among them)

(e) $var = \sigma_2^2/6 = 2^1$ (Intuition, since the first three data points are orthogonal with the rest three, here the rerr is the just the sum of the norm of the last three data points (2+8+2=12), and then divided by the total number (6) of data points, if we use average definition

5.3 Affine Subspace Identification - ASI

106. (CMU, 2014f, EXing, BPoczos, HW3, ex.1.2)

Affine Subspace Identification (ASI) We will now motivate the dimensionality reduction problem in a slightly different perspective. The resulting algorithm has many similarities to PCA. We will refer to method as Affine Subspace Identification (ASI).

You are given data $(x_i)_{i=1}^n$, $x_i \in \mathbb{R}^D$. Let $X = [x_1^\top; \ldots; x_n^\top] \in \mathbb{R}^{n \times D}$. We suspect that the data actually lies in a d dimensional affine subspace plus some gaussian noise. Our objective is to

nd a d dimensional representation z for x this can be used as a preprocessing step before an algorithm. In particular, we are not after any interpretation for this lower dimensional representation. We will assume $d < n$ and that the span of the data has dimension larger than d. Further, our method should work whether $n > D$ or $n < D$. We wish to

find parameters $A \in \mathbb{R}^{D \times d}$, $b \in \mathbb{R}^D$ and a lower dimensional representation $Z \in \mathbb{R}^{n \times d}$ so as to minimize

$$J(A, b, Z) = \frac{1}{n} \sum_{i=1}^{n} \|x_i - Az_i - b\|^2$$

Here $Z = [z_1^\top; \ldots; z_n^\top]$ and z_i is the representation for x_i.

(a) Let $C \in \mathbb{R}^{d \times d}$ be invertible and $d \in \mathbb{R}^d$. Show that both (A_1, b_1, Z_1) and (A_2, b_2, Z_2) achieve the same value on the objective J. Here, $A_2 = A_1 C^{-1}$, $b_2 = b_1 - A_1 C^{-1} d$ and $Z_2 = Z_1 C^\top + 1 d^\top$.

Therefore in order to make the problem determined we need to impose some constraint on Z. We will assume that the z_i's have zero mean covariance Ψ where Ψ is given.

$$\overline{Z} = \frac{1}{n} \sum_{i=1}^{n} z_i = 0, S = \frac{1}{n} \sum_{i=1}^{n} z_i z_i^\top = \Psi$$

(b) Outline a procedure to solve the above problem. Specify how you would obtain A,Z,b which minimize the objective and satisfy the constraints. Hint: The rank k approximation of a matrix in Frobenius norm is obtained by taking its SVD and then zeroing out all but the first k singular values.

(c) You are given a new point x_*. Give the rule to obtain the d dimensional representation x_* for the new point.

Solution:

First observe that we can write the objective as

$$J(A, b, Z) = \|X - ZA^\top - 1b^\top\|_{\text{Fro}}^2$$

Further the two constraints on Z can be written as $Z^\top 1 = 0$ and $Z^\top Z = n\Psi$.

(a) We will show that $Z_1 A_1^\top + 1b_1^\top = Z_2 A_2^\top + 1b_2^\top$, so both values will achieve the same objective.

$$\begin{aligned} Z_2 A_2^\top + 1b_2^\top &= (Z_1 C^\top + 1d^\top) C^{-\top} A_1^\top + 1(b_1 - A_1 C^{-1} d)^\top \\ &= Z_1 C^\top C^{-\top} A_1^\top + 1d^\top C^{-\top} A_1^\top + 1b_1^\top - 1^\top d C^{-\top} A_1^\top \\ &\quad Z_1 A_1^\top + 1b_1^\top \end{aligned}$$

(b) First note that the problem does not constraint b in any way so we can take the derivative and set it to zero.

$$\begin{aligned} a = \nabla_b J &= 2(X - ZA^\top - 1b^\top)^\top 1 \\ &= Z^\top 1^\top - AZ^\top 1 - b^\top 1^\top 1 \\ b &= \frac{1}{n} X^\top 1 = \overline{X} \end{aligned}$$

Here, first note that $1^\top 1 = n$. If we can find Z to satisfy the constraint $Z^\top 1 = 0$ then the last step holds. We will assume this and then show that we can find such a Z.

To minimize wrt A,Z, note that $ZA^\top$ needs to be the best rank k approximation to $X - 1b^\top$ in Frobenius norm. We can do this by taking the SVD of $\overline{X} = X - 1b^\top$ and then zeroing out the last $\min\{n, d\} - k$ singular values. Let $\tilde{X} = U\Sigma V^\top$ be the SVD. Here, $U \in \mathbb{R}^{n \times n}$, $\Sigma \in \mathbb{R}^{n \times D}$ is diagonal and $V \in \mathbb{R}^{D \times D}$. Denote the first d columns of U and V by U_d, and the top $d \times d$ block of Σ by Σ_d. The rank d approximation is given by, $ZA^\top = U_d \Sigma_d V_d^\top$.

Now, if we choose $Z = \sqrt{n} U_d \Psi^{1/2}$ and $A^\top = \frac{1}{\sqrt{n}} \Psi^{-1/2} \Sigma_d V_d^\top$

$$Z^\top Z = n\Psi^{1/2} U_d^\top U_d \Psi^{1/2} = n\Psi$$

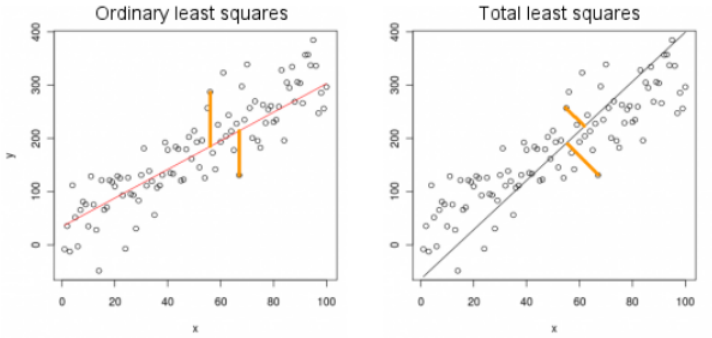
Finally, we need to chow that this Z satisfies $Z^\top 1 = 0$. For this first note that $\tilde{X}^\top 1 = V\Sigma U^\top 1 = 0$ which implies $U^\top 1 = 0$ since $V\Sigma$ is full rank (as $\dim(\text{span}(X))$ ¿, d). Then $U_d^\top 1 = 0 \Rightarrow Z^\top 1 = 0$.

(c) $z_* = A^+(x_* - b)$, where A^+ is the MP inverse. This gives the projection of $x_* - b$ onto the column space of A.

5.4 PCA and Least Squares

5.4.1 Total Least Squares - TLS

107. (CS 189 Introduction to Machine Learning Spring 2018 HW5, ex.2)



(from `https://stats.stackexchange.com/questions/13152/how-to-perform-orthogonal-`

In most of the models we have looked at so far, we've accounted for noise in the observed y measurement and adjusted accordingly. However, in the real world it could easily be that our feature matrix X of data is also corrupted or noisy. Total least squares is a way to account for this. Whereas previously we were minimizing the y distance from the data point to our predicted line because we had assumed the features were definitively accurate, now we are minimizing the entire distance from the data point to our predicted line. In this problem we will explore the mathematical intuition for the TLS formula. We will then apply the formula to adjusting the lighting of an image which contains noise in its feature matrix due to inaccurate assumptions we make about the image, such as the image being a perfect sphere.

Let X and y be the true measurements. Recall that in the least squares problem, we want to solve for w in $\min_w \|Xw - y\|$. We measure the error as the difference between Xw and y, which can be viewed as adding an error term ϵ_y such that the equation $Xw = y + \epsilon_y$ has a solution:

$$\min_{\epsilon_y, w} \|\epsilon_y\|_2, \text{ subject to } XW = y + \epsilon_y$$

Although this optimization formulation allows for errors in the measurements of y, it does not allow for errors in the feature matrix X that is measured from the data. In this problem, we will explore a method called total least squares that allows for both error in the matrix X and the vector y, represented by ϵ_X and ϵ_y, respectively. For convenience, we absorb the negative sign into ϵ_y and ϵ_X and define true measurements y and X like so:

$$y^{\text{true}} = y + \epsilon_y$$

$$X^{\text{true}} = X + \epsilon_X$$

Specifically, the total least squares problem is to find the solution for w in the following minimization problem:

$$\min_{\epsilon_y, \epsilon_X, w} \|[\epsilon_X, \epsilon_X]\|_{\text{Fro}}^2, \text{ subject to } (X + \epsilon_X)w = y + \epsilon_y (4)$$

where the matrix $[\epsilon_X, \epsilon_y]$ is the concatenation of the columns of ϵ_X with the column vector y. Notice that the minimization is over w because it's a free parameter, and it does not necessarily have to be in the objective function. Intuitively, this equation is finding the smallest perturbation to the matrix of data points X and the outputs y such that the linear model can be solved exactly. The constraint in the minimization problem can be rewritten as

$$[X + \epsilon_X, y + \epsilon_y] \begin{bmatrix} w \\ -1 \end{bmatrix} = 0 (5)$$

(a) Let the matrix $X \in \mathbb{R}^{n \times d}$ and $y \in \mathbb{R}^n$ and note that $\epsilon_X \in \mathbb{R}^{n \times d}$ and $\epsilon_y \in \mathbb{R}^n$. Assuming that $n > d$ and $\text{rank}(X + \epsilon_X) = d$, explain why $\text{rank}([X + \epsilon_x, y + \epsilon_y]) = d$.

(b) For the solution w to be unique, the matrix $[X + \epsilon_X; y + \epsilon_y]$ must have exactly d linearly independent columns. Since this matrix has $d + 1$ columns in total, it must be rank-deficient by 1. Recall that the Eckart-Young-Mirsky Theorem tells us that the closest lower-rank matrix in the Frobenius norm is obtained by discarding the smallest singular values. Therefore, the matrix $[X + \epsilon_X; y + \epsilon_y]$ that minimizes

$$\|[\epsilon_X, \epsilon_y]\|_{\text{Fro}}^2 = \|[X^{\text{true}}, y^{\text{true}}] - [X, y]\|_{\text{Fro}}^2$$

is given by the following

$$[X + \epsilon_X, y + \epsilon_y] = U \begin{bmatrix} \Sigma_d & \\ & 0 \end{bmatrix} V^\top$$

where $[X, y] = U\Sigma V^\top$.

Suppose we express the SVD of $[X, y]$ in terms of submatrices and vectors:

$$[X, y] = \begin{bmatrix} U_{xx} & u_{xy} \\ u_{yx}^\top & u_{yy} \end{bmatrix} \begin{bmatrix} \Sigma_d & \\ & \sigma_{d+1} \end{bmatrix} \begin{bmatrix} V_{xx} & v_{xy} \\ v_{yx}^\top & v_{yy} \end{bmatrix}^\top$$

$u_{xy} \in \mathbb{R}^{n-1}$ is the first $(n-1)$ elements of the $(d+1)$0th column of U, $u_{yx}^\top \in \mathbb{R}^d$ is the first d elements of the n-th row of U, u_{yy} is the n-th element of the $(d+1)$-th column of U, $U_{xx} \in \mathbb{R}^{(n-1)\times d}$ is the $(n-1) \times d$ top left submatrix of U.

Similarly, $v_{xy} \in \mathbb{R}^d$ is the first d elements of the $(d+1)$-th column of V, $v_{yx}^\top \in \mathbb{R}^d$ is the first d elements of the $(d+1)$-th row of V, v_{yy} is the $(d+1)$-th elements of the $(d+1)$-th column of V, $V_{xx} \in \mathbb{R}^{d\times d}$ is the $d \times d$ top left submatrix of V, σ_{d+1} is the $(d+1)$-th eigenvalue of $[X, y]$, Σ_d is the diagonal matrix of the d largest singular values of $[X, y]$.

Using this information show that the

$$[\epsilon_X, \epsilon_y] = -\begin{bmatrix} u_{xy} \\ u_{yy} \end{bmatrix} \sigma_{d+1} \begin{bmatrix} v_{xy} \\ y_{yy} \end{bmatrix}^\top$$

(c) Using the result from the previous part and the fact that v_{yy} is not 0 (see notes on Total Least Squares), find a nonzero solution to $[X + \epsilon_X, y + \epsilon_y] \begin{bmatrix} w \\ -1 \end{bmatrix} = 0$ and thus solve for w in Equation

$$[X + \epsilon_X, y + \epsilon_y] \begin{bmatrix} w \\ -1 \end{bmatrix} = 0.$$

HINT: Looking at the last column of the product $[X, y]V$ may or may not be useful for this problem, depending on how you solve it.

(d) From the previous part, you can see that $\begin{bmatrix} w \\ -1 \end{bmatrix}$ is a right-singular vector of $[X, y]$. Show that

$$(X^\top X - \sigma_{d+1}^2 I)w = X^\top y$$

Solution:

(a) Given that the constraint in the minimization problem from Equation (4) is satisfied, the last column of the matrix $[X + \epsilon_X; y + \epsilon_y]$. which is $y + \epsilon_y$ can be expressed as a linear combination of the remaining columns in the matrix. Therefore, the rank of the matrix is the rank of the remaining columns, which we previously assumed was d.

(b) We have

$$[X, y] + [\epsilon_X, \epsilon_y] = \begin{bmatrix} U_{xx} & u_{xy} \\ u_{yx}^\top & u_{yy} \end{bmatrix} \begin{bmatrix} \Sigma_d & \\ & 0 \end{bmatrix} \begin{bmatrix} V_{xx} & v_{xy} \\ v_{yx}^\top & v_{yy} \end{bmatrix}^\top$$

We can substract the second equation from the first to solve for $[\epsilon_X, \epsilon_y]$:

$$[X, y] - ([X, y] + [\epsilon_X, \epsilon_y]) = \begin{bmatrix} U_{xx} & u_{xy} \\ u_{yx}^\top & u_{yy} \end{bmatrix} \begin{bmatrix} \Sigma_d & \\ & \sigma_{d+1} \end{bmatrix} \begin{bmatrix} V_{xx} & v_{xy} \\ v_{yx}^\top & v_{yy} \end{bmatrix}^\top$$

$$- \begin{bmatrix} U_{xx} & u_{xy} \\ u_{yx}^\top & u_{yy} \end{bmatrix} \begin{bmatrix} \Sigma_d & \\ & 0 \end{bmatrix} \begin{bmatrix} V_{xx} & v_{xy} \\ v_{yx}^\top & v_{yy} \end{bmatrix}$$

$$-[\epsilon_x, \epsilon_y] = \begin{bmatrix} U_{xx} & u_{xy} \\ u_{yx}^\top & u_{yy} \end{bmatrix} \left(\begin{bmatrix} \Sigma_d & \\ & \sigma_{d+1} \end{bmatrix} - \begin{bmatrix} \Sigma_d & \\ & 0 \end{bmatrix} \right) \begin{bmatrix} V_{xx} & v_{xy} \\ v_{yx}^\top & v_{yy} \end{bmatrix}^\top$$

$$[\epsilon_X, \epsilon_y] = - \begin{bmatrix} U_{xx} & u_{xy} \\ u_{yx}^\top & u_{yy} \end{bmatrix} \begin{bmatrix} 0 & \\ & \sigma_{d+1} \end{bmatrix} \begin{bmatrix} V_{xx} & v_{xy} \\ v_{yx}^\top & v_{yy} \end{bmatrix}$$

$$= - \begin{bmatrix} u_{xy} \\ u_{yy} \end{bmatrix} \sigma_{d+1} \begin{bmatrix} v_{xy} \\ v_{yy} \end{bmatrix}^\top$$

(c) Following the hint, we look at the last column of the product $[X, y]V$, which is given by

$$[X, y] \begin{bmatrix} v_{xy} \\ v_{yy} \end{bmatrix} = \begin{bmatrix} u_{xy} \\ u_{yy} \end{bmatrix} \sigma_{d+1}$$

and we can substitute this into the equation above to get

$$[\epsilon_X, \epsilon_y] = -[X, y] \begin{bmatrix} v_{xy} \\ v_{yy} \end{bmatrix} \begin{bmatrix} v_{xy} \\ v_{yy} \end{bmatrix}^\top$$

and thus

$$[X, y] + [\epsilon_X, \epsilon_y] = [X, y] - [X, y] \begin{bmatrix} v_{xy} \\ v_{yy} \end{bmatrix} \begin{bmatrix} v_{xy} \\ v_{yy} \end{bmatrix}^\top$$

$$[X + \epsilon_X, y + \epsilon_y] \begin{bmatrix} v_{xy} \\ v_{yy} \end{bmatrix} = [X, y] \begin{bmatrix} v_{xy} \\ v_{yy} \end{bmatrix} - [X, y] \begin{bmatrix} v_{xy} \\ v_{yy} \end{bmatrix} = 0$$

This makes the nonzero $\begin{bmatrix} v_{xy} \\ v_{yy} \end{bmatrix}$ be in the nullspace, and hence almost gives us what we want $\begin{bmatrix} w \\ -1 \end{bmatrix}$, except for scaling and having v_{yy} instead of $a - 1$. We know from the notes on Total Least Squares that v_{yy} is not 0 almost surely for continuous noise, so we can just rescale to find $w = -v_{xy} v_{yy}^{-1}$ from Equation (5).

(d) Since $\begin{bmatrix} w \\ -1 \end{bmatrix}$ is a right-singular vector of $[Xy]$, it is a right-eigenvector of the matrix

$$[Xy]^\top [Xy] = \begin{bmatrix} X^\top X & X^\top y \\ y^\top X & Y^\top y \end{bmatrix}$$

So to find it we solve

$$\begin{bmatrix} X^\top X & X^\top y \\ y^\top X & y^\top y \end{bmatrix} \begin{bmatrix} w \\ -1 \end{bmatrix} = \sigma_{d+1}^2 \begin{bmatrix} w \\ -1 \end{bmatrix}$$

Ignore the bottom equation and consider the solution of the top equation:

$$X^\top X w - X^\top y = \sigma_{d+1}^2 w$$

which can be rewritten as

$$(X^\top X - \sigma_{d+1}^2 I) w = X^\top y$$

This result is like ridge regression, but with a negative regularization constant!

108. (CS189 Spring 2018 midterm, ex.8.(d)) During training of your model, both independent variables in the matrix $X \in \mathbb{R}^{n \times d}$ and dependent target variables $y \in \mathbb{R}^n$ are corrupted by noise. At test time, the data points you are computing predictions for, x_{test}, are noiseless. Which method(s) should you use to estimate the value of $\hat{w}$ from the training data in order to make the most accurate predictions y_{test} from the noiseless test input data, X_{test}? Assume that you make predictions using $y_{\text{test}} = X_{\text{test}} \hat{w}$.

(a) OLS

(b) Weighted Least Squares

(c) Ridge regression

(d) TLS

Solution: TLS since the test data has no noise while the training data does. This means that we are looking for the true relationship (as opposed to the best predictor for the training data) and total least squares is the one that does this.

Because TLS was shown to be equivalent to ridge-regression with a negative regularizer, points were not taken off for marking that as well.

5.4.2 PCA and Ridge Regression

109. (CS 189 Introduction to Machine Learning Fall 2018 Midterm, ex.2) OR some in (CS 189 Introduction to Machine Learning Spring 2018 Midterm, ex.6 - some longer answers + last subpoint) A Spectral View of Linear Regression OR **Ridge Regression vs. PCA**

Assume we are given training data in the form of the matrix $X \in \mathbb{R}^{n \times d}$ where the rows are the d-dimensional feature vectors x_i and $y \in \mathbb{R}^n$ which is the vector of the corresponding target values. We do not assume that X is full rank, and take its rank to be r. Note that $d \leq n$.

Recall that the compact singular value decomposition is $X = U\Sigma V^\top$ where $U \in \mathbb{R}^{n \times d}$, $V \in \mathbb{R}^{d \times d}$ and $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_d)$. We denote the n-dimensional column vectors of U by u_i and the d-dimensional column vectors of V by v_i. Furthermore, let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d$.

In this problem, we consider the result of two different linear regression techniques: ridge regression and applying ordinary least squares after using PCA to reduce the feature dimension from d to k (PCA-OLS). In particular, we compare the predicted value by of a new datapoint x by writing an expression of the form:

$$\hat{y}(x) = x^\top w = x^\top \sum_{i=1}^{d} \rho(\sigma_i) v_i u_i^\top y$$

In the following questions you will find the form of the spectral $\rho(\sigma)$ for ridge regression and PCA-OLS.

(a) Recall that the ridge regression optimizer is defined (for $\lambda > 0$) as

$$w_{\text{ridge}} = \text{argmin}_{w \in \mathbb{R}^d} \|Xw - y\|_2^2 + \lambda \|w\|_2^2.$$

Show that the closed-form solution for w_{ridge} has the form

$$w_{\text{ridge}} = V\text{diag}(\rho_\lambda(\sigma_1), \ldots, \rho_\lambda(\sigma_d))U^\top y,$$

and find the ridge-regression spectral function ρ_λ.

(b) Using the expression for w_{ridge} from the previous part, write down the ridge regression predictor function in form of :

$$\hat{y}(x) = x^\top w = x^\top \sum_{i=1}^{d} \rho(\sigma_i) v_i u_i^\top y$$

(c) The ordinary least squares problem on the reduced k-dimensional PCA feature space (PCA-OLS) can be written as

$$\bar{w}_{\text{PCA}} = \text{argmin}_{w \in \mathbb{R}^d} \|XV_k w - y\|^2$$

where the columns of V_k consist of the first k right singular vectors of X. The expression embeds the raw feature vectors onto the top k principal components by the transformation $V_k^\top x_i$. Assume the PCA dimension is less than the rank of the data matrix, $k \leq r$.

Write down the expression for the optimizer $\bar{w}_{\text{PCA}} \in \mathbb{R}^k$ in terms of U, y and the singular values of X.

Hint: $k \leq r$ implies that the matrix of PCA embedded data matrix XV_k is full rank.

(d) Now, use the expression for $\bar{w}_{\text{PCA}}$ from the previous part write down the predictor function in the form of

$$\hat{y}(x) = x^\top w = x^\top \sum_{i=1}^{d} \rho(\sigma_i) v_i u_i^\top y.$$

In doing so, you should define the form of the PCA-OLS spectral function ρ_k.

(e) Let

$$\hat{y}_{\text{test}} = x_{\text{test}}^\top \sum_{i=1}^{d} v_i \beta_i u_i^\top y.$$

For the following part, $d = 5$. The following $\beta = (\beta_1, \ldots, \beta_5)$ (written out to two significant figures)are teh results of OLS (i.e., that we would get from ridge regression in the limit $\lambda \to 0$), λ-ridge-regression, and

k-PCA-OLS for some X, y (identical for each method) and $\lambda = 1$, $k = 3$. Write down which procedure was used for each of the three sub-parts below.

We hope this helps you intuitively see the connection between these three methods.

Hint: It is not necessary to find the singular values of X explicitly, or to do any numerical computations at all.

i. $\beta = (0.01, 0.1, 0.5, 0.1, 0.01)$
ii. $\beta = (0.001, 0.1, 1, 0, 0)$
iii. $\beta = (0.001, 0.1, 1, 10, 100)$

Solution:

(a) First, recall that

$$w_{\text{ridge}} = (X^\top X + \lambda I)^{-1} X^\top y.$$

Then plugging in the SVD of X,

$$\begin{aligned} w_{\text{ridge}} &= (V\Sigma^2 V^\top + \lambda I)^{-1} V\Sigma U^\top y \\ &= V(\Sigma^2 + \lambda I)^{-1}\Sigma U^\top y \end{aligned}$$

Thus we see that

$$\rho_\lambda(\sigma_i) = \frac{\sigma_i}{\lambda + \sigma_i^2}.$$

(b) The resulting prediction for ridge reads

$$\begin{aligned} \hat{y}_{\text{ridge}} &= x^\top V \text{diag}(\frac{\sigma_i}{\lambda + \sigma_i^2}) U^\top y \\ &= x^\top \sum_{i=1}^{d} \frac{\sigma_i}{\lambda + \sigma_i^2} v_i u_i^\top y \end{aligned}$$

(c) Apply OLS on the new matrix XV_k to obtain

$$\begin{aligned} \tilde{w}_{\text{PCA}} &- [(XV_k)^\top (XV_k)]^{-1} (XV_k)^\top y \\ &= [V_k^\top V\Sigma^2 V^\top V_k]^{-1} V_k X^\top y \\ &= \Sigma_k^{-1} U_k^\top y \end{aligned}$$

(d) The resulting prediction for PCA reads (note that you need to project it first!)

$$\begin{aligned}\hat{y}_{\text{PCA}} &= x^\top V_k \tilde{w}_{\text{PCA}} \\ &= x^\top V_k \Sigma_k^{-1} U_k^\top y \\ &= x^\top \sum_{i=1}^{k} \frac{1}{\sigma_i} v_i u_i^\top y\end{aligned}$$

$$\rho_k(\sigma_i) = \begin{cases} \frac{1}{\sigma_i} & 1 \leq k \\ 0 & i > k \end{cases}$$

(e) Ridge, 3-PCA-OLS, OLS

Reasoning: The prediction for OLS is the same as for PCA with $k = d$.

$$\hat{y}_{\text{OLS}} = x^\top \sum_{i=1}^{d} \frac{1}{\sigma_i} v_i u_i^\top y$$

Putting all pieces together, we can thus see that PCA does "hard shrinkage" or "hard cutoff" (i.e., sets to zero) of the last $k+1, \ldots, d$ coefficients β_i, whereas ridge regression does "soft shrinkage" (i.e., shrinks towards zero) of the coefficients.

5.5 PCA and Whitening

110. (from `https://cbrnr.github.io/2018/12/17/whitening-pca-zca/`) Whitening (also known as sphering) is a linear transformation used for decorrelating signals. Applied to EEG, this means that the original channel time series (which tend to be highly correlated) are transformed into uncorrelated signals with identical variances. The term whitening is derived from white noise (which in turn draws its name from white light), which consists of serially uncorrelated samples. Whitening thus transforms a random vector into a white noise vector with uncorrelated components.

Theoretically, there are infinitely many possibilities to perform a whitening transformation. We will explore two popular whitening methods in more detail, namely principal component analysis (PCA) and zero-phase component analysis (ZCA), which was introduced by Bell and Sejnowski (1997). These methods are commonly used in EEG/MEG analysis as a preprocessing step prior to independent component analysis (ICA) (see this previous

post on how to remove ocular artifacts with ICA). If you want to delve into the matter more deeply, Kessy et al. (2018) discuss even more possible whitening methods.

Mathematically, whitening transforms a random vector x (the original EEG channel time series) into a random vector z using a whitening matrix W:

$$z = Wx$$

Importantly, the original covariance cov(x)=Σ becomes cov(z)=I after the transformation – the identity matrix. This means that all components of z have unit variance and all correlations have been removed.

Both PCA and ZCA are based on the eigenvectors and eigenvalues of the (empirical) covariance matrix. In particular, the covariance matrix can be decomposed into its eigenvectors U and eigenvalues Λ as:

$$\Sigma = U\Lambda U^\top$$

(a) **PCA Whitening** (Let us say we are in the when we do not want to reduce dimensionality.)

i. Compute the covariance of the principal components.
Observation: This will not be a whitening transform, but a **decorrelation** one.

ii. Show that the whitening matrix W^{PCA} for PCA can be written as:

$$W^{\text{PCA}} = \Lambda^{-1/2}U^\top$$

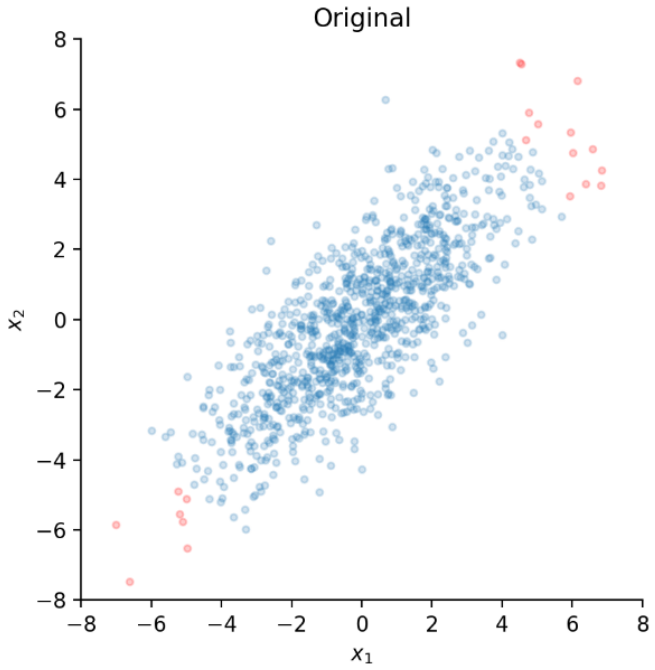
(b) **ZCA Whitening** (from `https://stats.stackexchange.com/questions/117427/what-is-the-difference-between-zca-whitening-and-pca-whitening`; see proof in a photo ZCA.jpg) Now we want to find a transformation (which will be called the ZCA transformation) that satisfies the following property: it results in whitened data that is as close as possible to the original data (in the least squares sense):

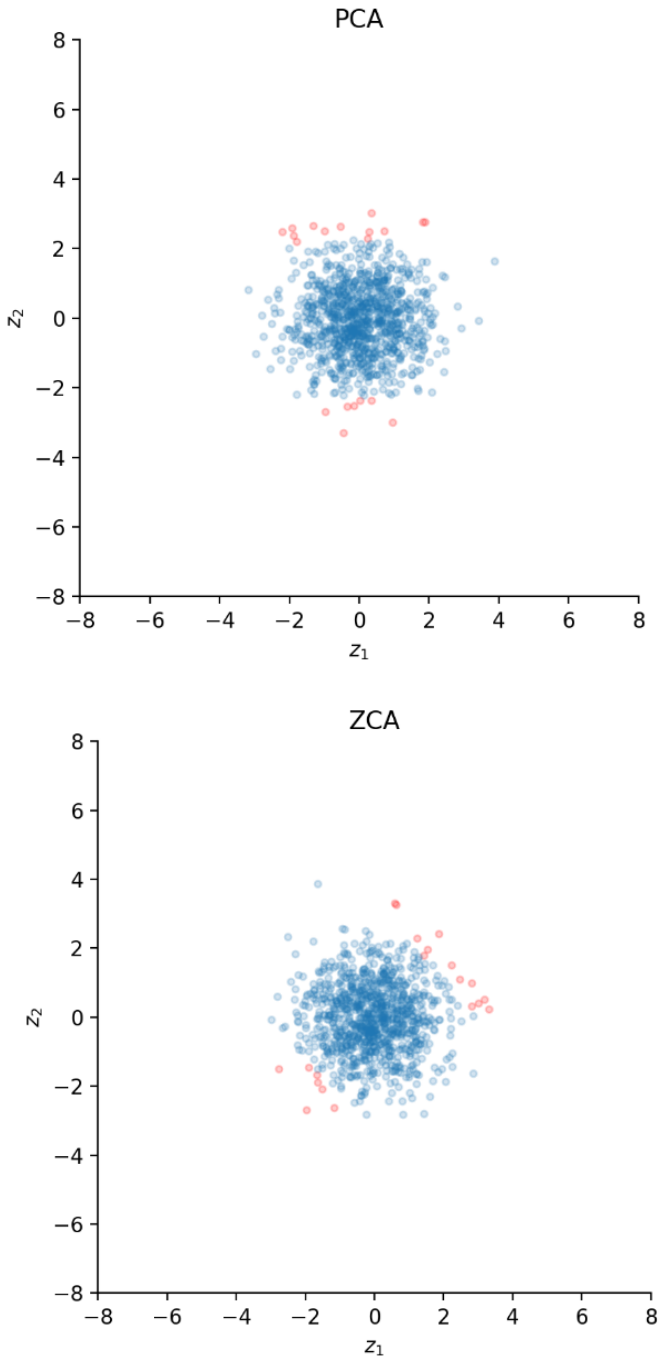
$$W^{\text{ZCA}} = \arg \min_{AX\text{-whitened}} \|X - AX\|^2$$

where $X \in \mathbb{R}^{d\times n}$, d - number of attributes, n - number of observations.
Prove that the result of this optimization problem can be written as:

$$W^{\text{ZCA}} = U\Lambda^{-1/2}U^\top$$

Observation: See the results of PCA whitening and ZCA whitening on the following dataset. In contrast to PCA, ZCA has preserved the orientation of the original data points. This can be observed from the positions of the red dots, which are aligned along the same direction as the original data. This property has given this whitening transformation its name "zero-phase", because it minimally distorts the original phase (i.e. orientation) of the data.



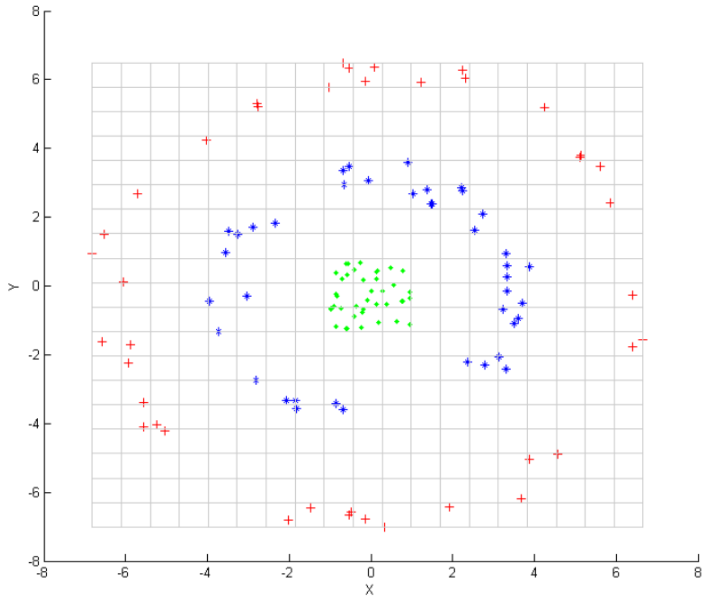


5.6 Dual PCA and Kernel PCA

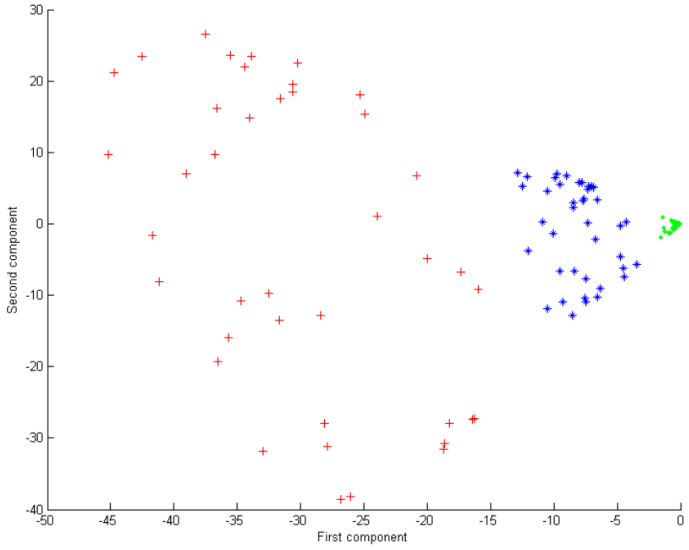
111. (after Bishop page 569) **Dual PCA/PCA for high-dimensional data** $(d >> n)$ Let $X \in \mathbb{R}^{d \times n}$ be a matrix with data, where n - number of

instances and d - number of attributes. In PCA, we compute the eigen-decomposition of $XX^\top$. Typical algorithms for finding the eigenvectors of a $b \times b$ matrix have a **computation cost** that scales like $O(b^3)$. In our case, the computational cost is $O(d^3)$. This is good in a usual context, where $n \geq d$. But if we are in the context of $n << d$, we will find that at least $d - n + 1$ of the eigenvectors are zero, corresponding to eigenvectors along whose directions the data set had zero variance and, furthermore, the cost will be $O(d^3)$. Prove that we can do better and obtain a computational cost of $O(n^3)$ which is better than $O(d^3)$. This new algorithm is called **dual PCA**, in contrast with the classic PCA. Suppose that the computational cost of multiplying $A \in \mathbb{R}^{c \times d}$ by $B \in \mathbb{R}^{d \times e}$ is $O(cde)$.

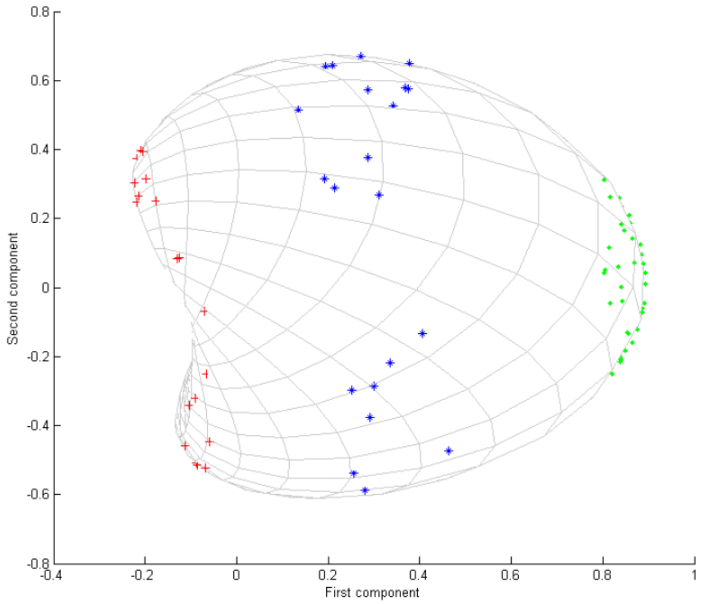
112. (CS189 spring 2018 midterm, ex.7) **Kernel PCA**



(Input points before kernel PCA; taken from `https://en.wikipedia.org/wiki/Kernel_principal_component_analysis`)



(Output after kernel PCA with $k(x, y) = (x^\top y + 1)^2$. The three groups are distinguishable using th first component only. taken from `https://en.wikipedia.org/wiki/Kernel_principal_component_analysis`)



(Output after kernel PCA, with a Gaussian kernel; taken from `https://en.wikipedia.org/wiki/Kernel_principal_component_analysis`)

In lectures, discussion, and homework, we learned how to use PCA to do dimensionality reduction by projecting the data to a subspace that captures most of the variability. This works well for data that is roughly Gaussian shaped, but many real-world high dimensional datasets have underlying low-dimensional structure that is not well captured by linear subspaces. However, when we lift the raw data into a higher-dimensional feature space by means of a nonlinear transformation, the underlying low-dimensional structure once again can manifest as an approximate subspace. Linear dimensionality reduction can then proceed. As we have seen in class so far, kernels are an alternate way to deal with these kinds of nonlinear patterns without having to explicitly deal with the augmented feature space. This problem asks you to discover how to apply the "kernel trick" to PCA.

Let $X \in \mathbb{R}^{n \times \mathcal{L}}$ be the data matrix, where n is the number of samples and $\mathcal{L}$ is the dimension of the raw data. Namely, the data matrix contains the data points $x_j \in \mathbb{R}^{\mathcal{L}}$ as rows

$$X = \begin{bmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_n^\top \end{bmatrix} \in \mathbb{R}^{n \times \mathcal{L}}$$

(a) Compute $XX^\top$ in terms of the singular value decomposition $X = U\Sigma V^\top$ where $U \in \mathbb{R}^{n \times n}$, $\Sigma \in \mathbb{R}^{n \times \mathcal{L}}$ and $V \in \mathbb{R}^{\mathcal{L} \times \mathcal{L}}$. Notice that $XX^\top$ is the matrix of pairwise Euclidean inner products for the data points. How would you get U if you only had access to $XX^\top$?

(b) Given a new test point $x_{\text{point}} \in \mathbb{R}^{\mathcal{L}}$, one central use of PCA is to compute the projection of x_{test} onto the subspace spanned by the k top singular vectors $v_1, \ldots, v_k$. Express the scalar projection $z_j = v_j^\top x_{\text{test}}$ onto the j-th principal component as a function of the inner products

$$Xx_{\text{test}} = \begin{bmatrix} < x_1, x_{\text{test}} > \\ \vdots \\ < x_n, x_{\text{test}} > \end{bmatrix}.$$

Assume that all diagonal entries of Σ are nonzero and non-increasing, that is $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$.

Hint: Express $V^\top$ in terms of the singular values Σ, the left singular vectors U and the data matrix X. If you want to use the compact form of the SVD, feel free to do so.

(c) How would you define kernelized PCA for a general kernel function $k(x_i; x_j)$ (to replace the Euclidean inner product $< x_i, x_j >$)? For example, the RBF kernel $k(x_i; x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{\delta^2}\right)$. Describe this in terms of a procedure which takes as inputs the training data points $x_1, x_2, \ldots, x_n \in \mathbb{R}^{\mathcal{L}}$ and the new test point $x_{\text{test}} \in \mathbb{R}^{\mathcal{L}}$, and outputs the analog of the previous part's z_j coordinate in the kernelized PCA setting. You should include how to compute U from the data, as well as how to compute the analog of Xx_{test} from the previous part.

Invoking the SVD or computing eigenvalues/eigenvectors is fine in your procedure, as long as it is clear what matrix is having its SVD or eigenvalues/eigenvectors computed. The kernel $k(\cdot;\cdot)$ can be used as a black-box function in your procedure as long as it is clear what arguments it is being given.

Solution:

(a) By plugging in the compact SVD decomposition $X = U\Sigma V^\top$ and using $U^\top U = I$ we get

$$X^\top X = V\Sigma U^\top U\Sigma V^\top = V\Sigma^2 V^\top.$$

Similarly, with $V^\top V = I$ we get

$$XX^\top = U\Sigma V^\top V\Sigma U^\top = U\Sigma^2 U^\top.$$

Notice from the last line that U are the eigenvectors of $XX^\top$ with eigenvalues $\sigma_1^2, \sigma_2^2, \ldots, \sigma_l^2$ where $\sigma_1, \sigma_2, \ldots, \sigma_d$ are the singular values of X and can therefore be computed by performing an eigendecomposition of $XX^\top$.

(b) By multiplying the compact SVD $X = U\Sigma V^\top$ on both sides with $U^\top$, we get $U^\top X = \Sigma V^\top$ and multiplying both sides of the new equation with Σ^{-1}, we obtain

$$V^\top = \Sigma^{-1} U^\top X.$$

Therefore we get

$$z_j = v_j^\top x_{\text{test}} = \frac{1}{\sigma_j} u_j^\top X x_{\text{test}}$$

(c) For kernelizing PCA, we replace inner products $< x_i, x_j >$ with $k(x_i, x_j)$ and $< x_i, x_{\text{test}} >$ with $k(x_i, x_{\text{test}})$, the procedure is then:

i. Pbtain the vectors u_j as eigenvectors from the eigendecomposition of the kernelized counterpart of the Gram matrix: $K \in \mathbb{R}^{n \times n}$ with $K_{ij} = k(x_i, x_j)$. The eigenvalues should be sorted in decreasing order. They are all non-negative real numbers because of the properties of kernels - the K matrix must be positive semi-definite.

ii. Kernelize the inner products $z_j = \frac{1}{\sigma_j} u_j^\top X x_{\text{test}}$ from the previous part by using:

$$z_j = \frac{1}{\sigma_j} u_j^\top \begin{bmatrix} k(x_1, x_{\text{test}}) \\ k(x_2, x_{\text{test}}) \\ \vdots \\ k(x_n, x_{\text{test}}) \end{bmatrix},$$

where the σ_j are the square roots of the eigenvalues for the martix K above generated by using the kernel on all pairs of training points. Because these are non-negative real numbers, the square root is well defined.

113. (Radford, 2014s, pr. 3) (Not necessarily Dual PCA, but $D > n$)

We have two i.i.d. observations of seven variables, as follows:

$$5, 7, 8, 2, 3, 5, 2$$

$$3, 3, 6, 6, 1, 1, 0$$

(a) Find a 7-dimensional vector of length one that points in the direction of the first principal component of this data. Explain how you obtained it.

(b) Find he projection on this principal component of the new observation shown below:

$$4, 1, 9, 3, 2, 2, 1$$

Solution:

(a) First, we subtract the sample means from the two observed vectors, giving the following centred data:

$$1, 2, 1, -2, 1, 1, 1$$

$$-1, -2, -1, 2, -1, -2, -1$$

With only two training cases, each of these vectors must point in the direction of the first principal component. Taking the first, its length is 4, so one vector of length 1 is in the direction of the first principal component is

$$\left[\frac{1}{4}, \frac{1}{2}, \frac{1}{4}, -\frac{1}{2}, \frac{1}{4}, \frac{1}{2}, \frac{1}{4}, \right]^\top$$

The other possible answer is the negation of the above.

It's also possible to answer this question by computing

$$XX^\top = \begin{bmatrix} 16 & -16 \\ -16 & 16 \end{bmatrix}$$

and then finding its eigenvectors, $[1, -1]^\top$ and $[1, 1]^\top$, which have eigenvalues 32 and 0. PC1 is in the direction $X^\top[1, -1]^\top$. After scaling to unit length, this gives the same answer as above.

(b) Subtracting the sample means from the training data gives $[0, -4, 2, -1, 0, -1, 0]^\top$. The dot product of this with the PC1 vector from the first subpoint is -3/2.

114. (MIT, 2015s, final, 6.2) In using medical data for prediction, one may often have to face "small data" rather than "big data" problems. One of the guest lectures emphasized ways to deal with issues arising from applying machine learning methods in the "small data" regime. These include (select all that apply)

(a) Dimensionality reduction such as PCA

(b) Shrinkage (using estimates tied to broader categories to inform more specific ones)

(c) Expanding feature vectors to include more potentially useful features

(d) Making high dimensional feature vectors sparse

5.7 Revision

115. (CMU, 2006f, final, pr.1.h) Give one similarity and one difference between **feature selection and PCA**.

Solution:

similarity: reduce the dimension of data

difference: feature selection finds a subset of features, while PCA produces a smaller, new set

116. (CS189 Spring 2018 - Introduction to Machine Learning HW5, ex.3) **(PCA and Random Projections)** In this question, we revisit the task of dimensionality reduction. Dimensionality reduction is useful for several purposes, including but not restricted to, visualization, storage, faster computation etc. While reducing dimension is useful, it is desirable to demand that such reductions preserve some properties of the original data. Often, certain geometric properties like distance and inner products are important to perform certain machine learning tasks. And as a result, we may want to perform dimensionality reduction but ensuring that we approximately maintain the pairwise distances and inner products.

While you have already seen many properties of PCA so far, in this question we investigate if random projections are a good idea for dimensionality reduction. A few advantages of random projections over PCA can be: (1) PCA is expensive when the underlying dimension is high and the number of principal components is also large (however note that there are several very fast algorithms dedicated to doing PCA), (2) PCA requires you to have access to the feature matrix for performing computations. The second requirement of PCA is a bottle neck when you want to take only a low dimensional measurement of a very high dimensional data, e.g., in FMRI and in compressed sensing. In such cases, one needs to design a projection scheme before seeing the data. We now turn to a concrete setting to study a few properties of PCA and random projections.

Suppose you are given n points $x_1, \ldots, x_n$ in $\mathbb{R}^d$. Define the $n \times d$ matrix $X = \begin{bmatrix} x_1^\top \\ \vdots \\ x_n^\top \end{bmatrix}$ where each row of the matrix represents one of the given points. In this problem, we will consider a few low-dimensional linear embedding $\psi : \mathbb{R}^d \to \mathbb{R}^k$ that maps vectors from $\mathbb{R}^d$ to $\mathbb{R}^k$.

Let $X = U\Sigma V^\top$ denote the singulat value decomposition of the matrix X. Assume that $n \geq d$ and let $\sigma_1 \geq \cdots \geq \sigma_d$ denote the singular values of X.

Let $v_1, \ldots, v_d$ denote the columns of the matrix V. We now consider the following k-dimensional PCA embedding $\psi_{\text{PCA}}(x) = (v_1^\top x, \ldots, v_k^\top x)^\top$. Note that this embedding projects a d-dimensional vector on the linear span of the set $\{v_1, \ldots, v_k\}$ and that $v_i^\top x$ denotes the i$^{\text{th}}$ coordinate of the projected vector in the new space.

We begin with a few matrix algebra relationships, and use them to investigate certain mathematical properties of PCA and random projections in the first few parts, and then see them in action on a synthetic dataset in

the later parts.

Notation: The symbol $[n]$ stands for the set $\{1, \ldots, n\}$.

(a) What is the ij^{th} entry of the matrices $XX^\top$ and $X^\top X$? Express the matrix $XX^\top$ in terms of U and Σ, and, express the matrix $X^\top X$ in terms of Σ and V.

(b) Show that

$$\psi_{\text{PCA}}(x_i)^\top \psi_{\text{PCA}}(x_j) = x_i^\top V_k V_k^\top x_j, \text{ where } V_k = [v_1 \ldots v_k].$$

Also show that $V_k V_k^\top = V I^k V^\top$, where the matrix I^k denotes a $d \times d$ diagonal matrix with first k diagonal entries as 1 and all other entries as zero.

(c) Suppose that we know the first k-singular values are the dominant singular values. In particular, we are given that

$$\frac{\sum_{i=1}^k \sigma_i^4}{\sum_{i=1}^d \sigma_i^4} \geq 1 - \epsilon,$$

for some $\epsilon \in (0, 1)$. Then show that the PCA projection to the first k-right singular vectors preserves the inner products on average:

$$\frac{1}{\sum_{i=1}^n \sum_{j=1}^n (x_i^\top x_j)^2} \sum_{i=1}^n \sum_{j=1}^n |(x_i^\top x_j) - (\psi_{\text{PCA}}(x_i)^\top \psi_{\text{PCA}}(x_j))|^2 \leq \epsilon.$$

Thus, we find that if there are dominant singular values, PCA projection can preserve the inner products on average. Hint: Using previous two parts and the definition of Frobenius norm might be useful.

(d) Now consider a different embedding $\psi : \mathbb{R}^d \to \mathbb{R}^k$ which preserved all pairwise distances and norms up-to a multiplicative factor, that is,

$$(1 - \epsilon)\|x_i\|^2 \leq \|\psi(x_i)\|^2 \leq (1 + \epsilon)\|x_i\|^2, \forall i \in [n] \quad (11)$$

and

$$(1-\epsilon)\|x_i - x_j\|^2 \leq \|\psi(x_i) - \psi(x_j)\|^2 \leq (1+\epsilon)\|x_i - x_j\|^2, \forall i, j \in [n] \quad (12)$$

where $0 < \epsilon << 1$ is a small scalar. Further assume that $\|x_i\| \leq 1, \forall i \in [n]$. Show that the embedding ψ satisfying equations (11) and (12) preserves each pairwise inner product:

$$|\psi(x_i)^\top \psi(x_j) - (x_i^\top x_j)| \leq C\epsilon, \forall i, j \in [n],$$

for some constant C. Thus, we find that if an embedding approximately preserves distances and norms upto a small multiplicative factor, and the points have bounded norms, then inner products are also approximately preserved upto an additive factor.

Hint: You may use the Cauchy-Schwarz inequality.

(e) Now we consider the random projection using a Gaussian matrix as introduced in the section. In next few parts, we work towards proving that if the dimension of projection is moderately big, then with high probability, the random projection preserves norms and pairwise distances approximately as described in equations above (first 2, not the third).

Consider the random matrix $J \in \mathbb{R}^{k \times d}$ with each of its entry being i.i.d. $\mathcal{N}(0,1)$ and consider the map $\psi_J : \mathbb{R}^d \to \mathbb{R}^k$ such that $\psi_J(x) = \frac{1}{\sqrt{k}} Jx$. Show that for any fixed non-zero vector u, the random variable $\frac{\|\psi_J(u)\|^2}{\|u\|^2}$ can be written as

$$\frac{1}{k} \sum_{i=1}^{k} Z_i^2$$

where Z_i's are i.i.d. $\mathcal{N}(0,1)$ random variables.

(f) (BONUS) For i.i.d. $Z_i \sim \mathcal{N}(0,1)$, we have the following probability bound

$$P[|\frac{1}{k} \sum_{i=1}^{k} Z_i^2| \notin (1-t, 1+t)] \leq 2e^{-kt^2/8}, \forall t \in (0,1).$$

Note that this bound suggests that $\sum_{i=1}^{k} Z_i^2 \approx \sum_{i=1} E[Z_i^2] = k$ with high probability. In other words, sum of square of Gaussian random variables concentrates around its mean with high probability. Using this bound and the previous part, now show that if $k \geq \frac{16}{\epsilon^2} \log(\frac{N}{\delta})$, then

$$P[\forall i, j \in [n], i \neq j, \frac{\|\psi_J(x_i) - \psi_J(x_j)\|^2}{\|x_i - x_j\|^2} \in (1-\epsilon, 1+\epsilon)] \geq 1 - \delta.$$

That is show that for k large enough, with high probability the random projection J approximately preserves the pairwise distances. Using this result, we can conclude that random projection serves as a good tool for dimensionality reduction if we project to enough number of dimensions. This result is popularly known as the Johnson-Lindenstrauss Lemma. Hint 1: The following (powerful technique

cum) bound might be useful: For a set of events A_{ij}, we have

$$P[\cap_{i,j} A_{ij}] = 1 - P[(\cap_{i,j} A_{ij})^c]1 - P[\cup_{i,j} A_{ij}^c] \geq 1 - \sum_{ij} P[A_{ij}^c].$$

You may define the event $A_{ij} = \left\{ \frac{\|\psi_J(x_i) - \psi_J(x_j)\|^2}{\|x_i - x_j\|^2} \in (1-\epsilon, 1+\epsilon) \right\}$ and use the union bound above.

(g) Suppose there are two clusters of points $S_1 = \{u_1, \ldots, u_n\}$ and $S_2 = \{v_1, \ldots, v_m\}$ which are far apart, i.e., we have

$$d^2(S_1, S_2) = \min_{u \in S_1, v \in S_2} \|u - v\|^2 \geq (1-\epsilon)\gamma$$

if $k \geq \frac{C}{\epsilon^2} \log(m+n)$ for some constant C. Note that such a property can help in several machine learning tasks. For example, if the clusters of features for different labels were far in the original dimension, then this problem shows that even after randomly projecting the clusters they will remain far enough and a machine learning model may perform well even with the projected data. We now turn to visualizing some of these conclusions on a synthetic dataset.

Solution:

(a) Let x_{ik} denote the k-th entry of the vector x_i. Then, we have

$$(XX^\top)_{ij} = \left[\begin{bmatrix} x_1^\top \\ \vdots \\ x_n^\top \end{bmatrix} \begin{bmatrix} x_1 & \ldots & x_n \end{bmatrix} \right]_{ij} = x_i^\top x_j = \sum_{k=1}^{d} x_{ik} x_{jk}$$

and

$$(X^\top X)_{ij} = \sum_{k=1}^{n} (X^\top)_{ik} (X)_{kj} = \sum_{k=1}^{n} x_{ki} x_{kj}$$

Furthermore, we have

$$XX^\top = U\Sigma^2 U^\top \text{ and } X^\top X = V\Sigma^2 V^\top$$

(b) Note that

$$\psi_{\text{PCA}}(x) = \begin{bmatrix} v_1^\top x \\ \vdots \\ v_k^\top x \end{bmatrix} = \begin{bmatrix} v_1^\top \\ \vdots \\ v_k^\top \end{bmatrix} x = V_k^\top x$$

and thus we have

$$\psi_{\text{PCA}}(x_i)^\top \psi_{\text{PCA}}(x_j) = x_i^\top V_k V_k^\top x_j$$

as required.

For the second part, we note that for two matrices

$$A = \begin{bmatrix} a_1 & \dots & a_d \end{bmatrix}, B = \begin{bmatrix} b_1^\top \\ \vdots \\ b_d^\top \end{bmatrix}$$

we have

$$AB = \sum_{i=1}^{d} a_i b_i^\top (6).$$

Furthermore, given a diagonal matrix Γ of size d-by-d, with $(\Gamma)_{ii} = \gamma_i$, we have

$$A\Gamma B = \sum_{i=1}^{d} \gamma_i a_i b_i^\top (7).$$

Using equation (6), we have

$$V_k V_k^\top = \sum_{i=1}^{k} v_i v_i^\top$$

Furthermore, using equation (7), we obtain that

$$V I^k V^\top = \sum_{i=1}^{d} v_i (I^k)_{ii} v_i^\top = \sum_{i=1}^{k} v_i v_i^\top.$$

since $(I^k)_{ii} = 1$ for $i = 1, \dots, k$ and 0 for $i > k$. The students are not required to prove so rigorously. The key take away from this part is to bo comfortable with different ways wo write matrices, and keep in mind the representations (6) and (7).

(c) Using the solution from the first subpoint, we have

$$(XX^\top)_{ij} = x_i^\top x_j (9)$$

Using the solution from the second subpoint, we have

$$\Psi = \begin{bmatrix} \psi_{\text{PCA}}(x_1)^\top \\ \vdots \psi_{\text{PCA}}(x_n)^\top \end{bmatrix} = XV_k (10)$$

We have

$$\sum_{i=1}^{n}\sum_{j=1}^{n}|(x_i^\top x_j)-(\psi_{\text{PCA}}(x_i)^\top\psi_{\text{PCA}}(x_j))|^2 \overset{(i)}{=} \|XX^\top-\Psi\Psi^\top\|_{\text{Fro}}^2$$

$$\overset{(ii)}{=} \|U\Sigma^2U^\top - XV_kV_k^\top X^\top\|_{\text{Fro}}^2$$

$$\overset{(iii)}{=} \|U\Sigma^2U^\top - U\Sigma V^\top VI^kV^\top V\Sigma U^\top\|_{\text{Fro}}^2$$

$$\overset{(iv)}{=} \|U\Sigma^2U^\top - U\Sigma I_k\Sigma U^\top\|_{\text{Fro}}^2,$$

where for step (i) we applied the definition of the Frobenius norm and the equations (9) and (10), for step (ii) we have used the solution from the first subpoint to replace $XX^\top = U\Sigma^2U^\top$ and for step (iii), we have used the singular value decomposition $X = U\Sigma V^\top$, and the second result from the second subpoint. For step (iv) we make use of the factor that $V^\top V = I$.

Now we use the fact that Frobenius norm is unitary invariant (Discussion 1, problem 2(b)). Thus, we have

$$\|U\Sigma^2U^\top - U\Sigma I_k\Sigma U^\top\|_{\text{Fro}}^2 = \|U\Sigma^2 - \Sigma I_k\Sigma U^\top\|_{\text{Fro}}^2 = \|\Sigma^2-\Sigma I_k\Sigma\|_{\text{Fro}}^2 = \sum_{i=k+1}^{d}\sigma_i^4.$$

We now use the fact that

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i^\top x_j)^2 = \|XX^\top\|_{\text{Fro}}^2 = \|\Sigma^2\|_{\text{Fro}}^2 = \sum_{i=1}^{d}\sigma_i^4,$$

we obtain that

$$\frac{1}{\sum_{i=1}\sum_{j=1}(x_i^\top x_j)^2}\sum_{i=1}^{n}\sum_{j=1}^{n}|(x_i^\top x_j)-(\psi_{\text{PCA}}(x_i)^\top\psi_{\text{PCA}}(x_j))|^2 = \frac{\sum_{i=k+1}^{d}\sigma_i^4}{\sum_{i=1}^{d}\sigma_i^4}$$

$$= 1 - \frac{\sum_{i=1}^{k}\sigma_i^4}{\sum_{i=1}^{d}\sigma_i^4}$$

$$\leq \epsilon.$$

(d) We will show that

$$-3\epsilon \leq (x_i^\top x_j - \psi(x_i)^\top\psi(x_j)) \leq 3\epsilon$$

Taking absolute value, we obtain the claimed result for $C = 3$.
Upper bound: Expanding the second inequality from equation (!2), we obtain

$$\|\psi(x_i)\|^2+\|\psi(x_j)\|^2-2\psi(x_i)^\top\psi(x_j) \leq (1+\epsilon)\|x_i\|^2+(1+\epsilon)\|x_j\|^2-2(1+\epsilon)x_i^\top x_j.$$

We use the lower bounds from equation (11):

$$\|\psi(x_i)\|^2 \geq (1-\epsilon)\|x_i\|^2$$
$$\|\psi(x_j)\|^2 \geq (1-\epsilon)\|x_j\|^2.$$

And hence we have

$$(1-\epsilon)\|x_i\|^2+(1-\epsilon)\|x_j\|^2-2\psi(x_i)^\top\psi(x_i) \leq (1+\epsilon)\|x_i\|^2+(1+\epsilon)\|x_j\|^2-2(1+\epsilon)x_i^\top x_j.$$

Moving terms around we obtain

$$\begin{aligned} 2(x_i^\top x_j - \psi(x_i)^\top\psi(x_j)) &\leq 2\epsilon(\|x_i\|^2 + \|x_j\|^2 - x_i^\top x_j) \\ &\leq \epsilon(\|x_i\|^{@} + \|x_j\|^2 + \|x_i\|\|x_j\|) \\ &\leq 6\epsilon, \end{aligned}$$

since $\|x_i\| \leq 1$. Thus we have

$$(x_i^\top x_j - \psi(x_i)^\top\psi(x_j)) \leq 3\epsilon.$$

Lower bound: Expanding the first inequality from equation (12), we obtain

$$\|\psi(x_i)\|^2+\|\psi(x_j)\|^2-2\psi(x_i)^\top\psi(x_j) \geq (1-\epsilon)\|x_i\|^2+(1-\epsilon)\|x_j\|^2-2(1-\epsilon)x_i^\top x_j.$$

We use the upper bounds from equation (11):

$$\|\psi(x_i)\|^2 \leq (1+\epsilon)\|x_i\|^2$$
$$\|\psi(x_j)\|^2 \leq (1+\epsilon)\|x_j\|^2.$$

And hence we have

$$(1+\epsilon)\|x_i\|^2+(1+\epsilon)\|x_j\|^2-2\psi(x_i)^\top\psi(x_j) \geq (1-\epsilon)\|x_i\|^2+(1-\epsilon)\|x_j\|^2-2(1-\epsilon)x_i^\top x_j.$$

Moving terms around we obtain

$$\begin{aligned} 2(x_i^\top x_j - \psi(x_)^\top\psi(x_j)) &\geq -2\epsilon(\|x_i\|^2 + \|x_j\|^2 - x_i^\top x_j) \\ &\geq -2\epsilon(\|x_i\|^2 + \|x_j\|^2 + \|x_i\|\|x_j\|) \\ &\geq -6\epsilon, \end{aligned}$$

since $\|x_i\| \leq 1$. And thus we are done.

(e) Let $J = \begin{bmatrix} J_1^\top \\ \vdots \\ J_k^\top \end{bmatrix}$. Then, we have

$$\frac{\|\psi_J(u)\|^2}{\|u\|^2} = \frac{1}{k}\sum_{i=1}^{k} \frac{(J_i^\top u)^2}{\|u\|^2}.$$

Note that

$$J_i \overset{i.i.d.}{\sim} \mathcal{N}(0, I_d),$$

and hence

$$(J_i^\top u) \sim \mathcal{N}(0, \|u\|^2),$$

and consequently, we have

$$Z_i = \frac{J_i^\top u}{\|u\|} \overset{i.i.d.}{\sim} \mathcal{N}(0,1).$$

As a result, we conclude that

$$\frac{\|\psi_J(u)\|^2}{\|u\|^2} = \frac{1}{k}\sum_{i=1}^{k} \frac{(J_i^\top u)^2}{\|u\|^2} = \frac{1}{k}\sum_{i=1}^{k} Z_i^2$$

where Z_i's are i.i.d. $\mathcal{N}(0,1)$ random variables.

(f) We define the event as given in the hint. For any $i, j \in [n], i \neq j$, define

$$A_{ij} = \left\{ \frac{\|\psi_J(x_i) - \psi_J(x_j)\|^2}{\|x_i - x_j\|^2} \in (1-\epsilon, 1+\epsilon) \right\}$$

Note that the event

$$\mathcal{A} = \left\{ \forall i, j \in [n], i \neq j, \frac{\|\psi_J(x_i) - \psi_J(x_j)\|^2}{\|x_i - x_j\|^2} \in (1-\epsilon, 1+\epsilon) \right\}$$

is equivalent to the intersection of all events A_{ij}, since all events A_{ij} must occur for $\mathcal{A}$ to occur.

That is

$$\mathcal{A} = \cap_{i\in[n], j\in[n], i\neq j} A_{ij}.$$

Note that there are $\binom{n}{2}$ events. Now as given in the hint, we have

$$P[\mathcal{A}] = P\left[\cap_{i\in[n], j\in[n], i\neq j} A_{ij}\right]$$

$$= 1 - P\left[\left(\cap_{i\in[n],j\in[n],i\neq j} A_{ij}\right)^c\right]$$
$$= 1 - P\left[\cup_{i\in[n],j\in[n],i\neq j} A_{ij}^c\right]$$
$$\geq 1 - \sum_{i\in[n],j\in[n],i\neq j} P[A_{ij}^c].$$

Now we have

$$P[A_{ij}^c] = P\left[\frac{\|\psi_J(x_i) - \psi_J(x_j)\|^2}{\|x_i - x_j\|^2} \notin (1-\epsilon, 1+\epsilon)\right]$$
$$\overset{\text{5-th subpoint}}{=} P\left[\left|\frac{1}{k}\sum_{i=1}^{k} Z_i^2\right| \notin (1-\epsilon, 1+\epsilon)\right]$$
$$\overset{\text{prob.bnd}}{\leq} 2e^{-k\epsilon^2/8}$$

Thus, we have that

$$P[\mathcal{A}] \geq 1 - \sum_{i\in[n],j\in[n],i\neq j} P[A_{ij}^2]$$
$$\geq 1 - \sum_{i\in[n],j\in[n],i\neq j} 2e^{-k\epsilon^2/8}$$
$$\geq 1 - \binom{n}{2} 2e^{-k\epsilon^2/8}$$
$$\geq 1 - n(n-1)e^{-k\epsilon^2/8}$$
$$\geq 1 - n^2\epsilon^{-k\epsilon^2/8}.$$

Now to make this probability greater than δ, we set k such that

$$n^2 e^{-k\epsilon^2/8} \leq \delta \Rightarrow 2\log n - k\frac{\epsilon^2}{8} \leq \log\delta \Rightarrow k \geq \frac{8}{\epsilon^2}(2\log n - \log\delta).$$

Note that since $\delta < 1$, we have that $k \geq \frac{16}{\epsilon^2}\log(\frac{N}{\delta})$ implies that $k \geq \frac{8}{\epsilon^2}(2\log n - \log\delta)$, and hence we are done.

(g) Using previous part, we obtain that pairwise squared distances between $m+n$ points are approximately preserved up to a factor of $(1-\epsilon, 1+\epsilon)$ if $k \geq C\log(m+n)/\epsilon^2$. Now, we consider any pair of points which minimize the objective $\min_{u\in S_1, v\in S_2} \|\psi_J(u) - \psi_J(v)\|^2$. Let's call these points u_{i^*} and v_{j^*}. Then we have

$$\min_{u\in S_1, v\in S_2} \|\psi_J(u) - \psi_j(v)\|^2 = \|\psi_J(u_{i^*}) - \psi_J(v_{j^*})\|^2$$

$$\geq (1-\epsilon)\|u_{i^*} - v_{j^*}\|^2$$
$$\geq (1-\epsilon) \min_{u \in S_1, w \in S_2} \|u - v\|^2$$
$$= (1-\epsilon)\gamma,$$

and we are done.

117. (CMU, 2012s, ZBarJoseph, final, pr. 7.3)

The key assumption of a **naive Bayes (NB)** classifier is that features are independent, which is not always desirable. Suppose that linear principal components analysis (PCA) is first used to transform the features, and NB is then used to classify data in this low-dimensional space. Is the following statement true? Justify your answers.

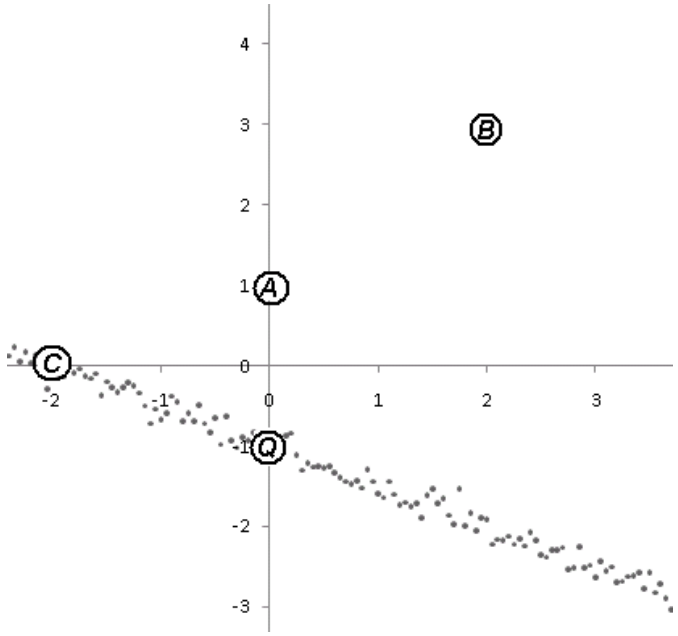
The independent assumption of NB would now be valid with PCA transformed features because all principal components are orthogonal and hence uncorrelated.

Solution:

This statement is false. First, uncorrelation is not equivalent to independence. Second, transformed features are not necessarily uncorrelated if the original features are correlated in a nonlinear way.

118. (CMU, 2013f, WCohen, EXing, sample exam, pr. 7) **kNN and PCA** Recall the homework question regarding Principal Component Analysis (PCA). The image below shows a similar distribution of small black points along with the large query point, labeled 'Q' (at x=0, y=-1). In addition, there are now three additional candidate neighbor points, labeled 'A' (at x=0, y=1), 'B' (at x=2, y=3), and 'C' (at x=-2, y=0). Assume each point has zero area and is centered directly at the coordinates given.

(a) In the original basis (that is, before doing any kind of dimensionality reduction), list, in order from closest to furthest, the three neighbors (A, B, C) of Q. Be sure to clearly note any ties:



(b) Find the first principal component of the small black dots in the above drawing. Redraw Q, A, B and C below, as reconstructed using only this first principal component:

(c) In this new basis you found above, (that is, reconstructing the points using only the first principal component), list, in order from closest to furthest, the three neighbors (A, B, C) of Q. Be sure to clearly note any ties:

(d) Find the second principal component of the small black dots in the original drawing. Redraw Q, A, B, and C below, as reconstructed using only this second principal component:

(e) In this new basis you found above, (that is, reconstructing the points using only the second principal component), list, in order from closest to furthest, the three neighbors (A, B, C) of Q. Be sure to clearly note any ties:

119. (CMU, 2012s, ZBarJoseph, final, pr. 2.4)

Consider **neural networks** where each neuron has a linear activation function, i.e., each neurons output is given by $g(x) = c + b\frac{1}{n}\sum_{i=1}^{n} W_i x_i$, where b and c are two fixed real numbers and n is the number of incoming links to that neuron.

(a) Suppose you have a single neuron with a linear activation function g() as above and input $x = x_0, \ldots, x_n$ and weights $W = W_0, \ldots, W_n$.

Write down the squared error function for this input if the true output is a scalar y, then write down the weight update rule for the neuron based on gradient descent on this error function.

(b) Now consider a network of linear neurons with one hidden layer of m units, n input units, and one output unit. For a given set of weights $w_{k;j}$ in the input-hidden layer and W_j in the hidden-output layer, write down the equation for the output unit as a function of $w_{k;j}$, W_j , and input x. Show that there is a single-layer linear network with no hidden units that computes the same function.

(c) Now assume that the true output is a vector y of length o. Consider a network of linear neurons with one hidden layer of m units, n input units, and o output units. If o ¿ m, can a single-layer linear network with no hidden units be trained to compute the same function?Briefly explain why or why not.

(d) The model in the previous subpoint combines dimensionality reduction with regression. One could also reduce the dimensionality of the inputs (e.g. with PCA) and then use a linear network to predict the outputs. Brie y explain why this might not be as effective as 3) on some data sets.

Solution:

(a) Error function: $(y - c - \frac{b}{n} W^\top x)^2$
Update rule: $W_i \leftarrow W_i + \lambda 2 x_i (y - c - \frac{b}{n} W^\top x)$

(b) Without loss of generality, we assume that the weights encapsulate the linear activation function for this sub-question.
$y \approx \sum_j W_j \sum_k w_{k,j} x_j = \sum_k (\sum_j W_j w_{k,j}) x_k = \sum_k \beta_k x_k$
Or, $W^\top w^\top x = \beta^\top x$ where $\beta^\top = W^\top w^\top$

(c) No. The hidden layer effectively imposes a rank constraint on the learned coefficients that a single layer network will not be able to enforce during training.

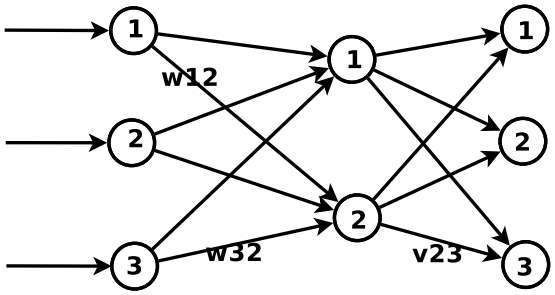
(d) If some of the linearly independent input dimensions are not correlated with the output, then PCA on the inputs alone will not regularize effectively. The model in 3) reduces dimensionality based on the predictive capacity of the input dimensions.

120. (CMU, 2017f, NBalcan, HW5, pr. 5.2)

Dimensionality Reduction and Representation Learning

In this question, we explore the relation between PCA, kernel PCA and auto encoder neural networks (trained to output the same vector they receive as input). We will use n and d to denote the number and dimensionality of the given data points respectively.

(a) Consider an auto encoder with a single hidden layer of k nodes. Let w_{ij} denote the weight of the edge from the i^{th} input node to the j^{th} hidden node. Similarly, let v_{ij} denote the weight of the edge from the i^{th} hidden node to the j^{th} output node. Show how you can set the activation functions of hidden and output nodes as well as the weights w_{ij} and v_{ij} such that the resulting auto encoder resembles PCA.



(b) Kernel PCA is a non-linear dimensionality reduction technique where a principal vector v_j is computed as a linear combination of training examples in the feature space

$$v_j = \sum_{i=1}^{n} \alpha_{ij} \phi(x_i).$$

Computing the principal component of a new point x can then be done using kernel evaluations

$$z_j(x) = \langle v_j, \phi(x) \rangle = \sum_{i=1}^{n} \alpha_{ij} \langle \phi(x_i), \phi(x) \rangle = \sum_{i=1}^{n} \alpha_{ij} k(x_i, x).$$

You will show that kernel PCA can be represented by a neural network. First we define a *kernel node*. A kernel node with a vector w_i of incoming weights and an input vector x computes the output $y = k(x, w_i)$.

Show that, given a data set $x_1, \ldots, x_n$, there exists a network with a single hidden layer and the output of the network is the kernel principal

components $z_1(x), \ldots, z_k(x)$ for a given input x. Specify the number of nodes in the input, output and hidden layers, the type and activation function of hidden and output nodes , and the weights of the edges in terms of $\alpha, x_1, \ldots, x_n$.

(c) What is the number of parameters (weights) required to store the network in the previous question?

(d) Another way to do non-linear dimensionality reduction is to train an auto encoder with non-linear activation functions (e.g. sigmoid) in the hidden layers. State one advantage and one disadvantage of that approach compared to kernel PCA.

Solution:

(a) The PCA can be represented by the linear transformations

$$z = U^\top x$$
$$\hat{x} = Uz$$

where U is a matrix of the top k eigenvectors. Therefore is we use linear units with a hidden layer of size k, set $w_{ij} = U_{ij}$ and set $v_{ij} = U_j$, then we get the same effect of PCA. Specifically, the hidden layer is equivalent to z.

(b) We will have a hidden layer of n kernel units such that $w_i = x_i$ and k linear output units such that $v_{ij} = \alpha_{ij}$.

(c) It requires $nd + nk$ parameters to store w, v.

(d) An advantage is that it does not require a number of parameters that grows with n so we can achieve non-linearity with smaller models. A disadvantage is that there is no tractable algorithm that guarantees finding the global optimum.

121. (CMU, 2012f, TMitchell, ZBar-Joseph, midterm, part of pr. 9.c) Nearly all the algorithms we have learned about in this course have a tuning parameter for regularization that adjusts the bias/variance tradeoff, and can be used to protect against overfitting. More regularization tends to cause less overfitting. For the PCA algorithm, we point out the tuning parameter. If you increase the parameter, does it lead to MORE or LESS regularization? (In other words, MORE bias (and less variance), or LESS bias (and more variance)?) For the blank, please write MORE or LESS.

Dimension reduction as preprocessing: Instead of using all features, reduce the training data down to k dimensions with PCA, and use the PCA projections as the only features.

Higher k means ... **regularization**.

Solution: less

122. (LA4ML - slides - Double Jeopardy Round OR specified if other source) Circle the correct answer and justify your choice:

(a) If your data matrix has 1000 observations on 40 variables, then how many principal components exist?

i. Impossible to tell from this information
ii. 40000
iii. 1000
iv. 40

Solution: D

(b) The first principal component is...

i. A statistic that tells you how much **multicollinearity** is in your data
ii. A scalar that tells you how much total variance is in the data
iii. The first column in your data matrix
iv. A vector that points in the direction of maximum variance in the data

Solution: D

(c) The principal component scores are...

i. Statistics which tell you the importance of each principal component
ii. The coordinates of your data in the new basis of principal components
iii. Statistics which tell you how much each variable relates to each principal component
iv. Relatively random

Solution: B

(d) The eigenvalues of the covariance matrix...

i. Are always orthogonal

ii. Add up to 1
iii. Tell you how much variance exists along each principal component
iv. Tell you the proportion of variance explained by each principal component

Solution: C

(e) The total amount of variance in a data set is...

i. The sum of all entries in the covariance matrix
ii. The sum of the eigenvalues of the covariance matrix
iii. The sum of the variances of each variable
iv. Both 122(e)ii and 122(e)iii.

Solution: D

(f) PCA is a special case of the Singular Value Decomposition (SVD), when your data is either centered or standardized.

i. True
ii. False

Solution: A

(g) **Principal Component Regression**...

i. Can give you meaningful beta parameters for your original variables
ii. Attempts to solve the problem of severe multicollinearity in predictor variables
iii. Is a biased regression technique and should be used only as a last resort when you cannot omit correlated variables
iv. All of the above

Solution: D

(h) To perform principal components regression, we simply compute the principal components of our data, which are uncorrelated, and use all of them in our model in place of the original variables.

i. True
ii. False

Solution: B

(i) (LA4ML - Applications of PCA - Worksheet. Part One, ex.7) **Covariance and Correlation PCA** are exactly the same thing.

i. True

ii. False

Solution: B

(j) (Final CS 189 Fall 2015 Introduction to Machine Learning, ex.Q1.(8)) Dimensionality reduction can be used as pre-processing for machine learning algorithms like decision trees, kd-trees, neural networks etc.

i. True

ii. False

Solution: A

(k) (Final CS 189 Fall 2015 Introduction to Machine Learning, ex. Q1.(13)) Whitening the data doesn't change the first principal direction.

i. True

ii. False

Solution: B

(l) (Final CS 189 Fall 2015 Introduction to Machine Learning, ex. Q1.(14)) PCA can be kernelized.

i. True

ii. False

Solution: A

(m) (Final CS 189 Spring 2014 Introduction to Machine Learning, ex.Q2.(b)) Given d-dimensional data $\{x_i\}_{i=1}^N$, you run PCA and pick P principal components. Can you always reconstruct any data point x_i for $i \in \{1, \ldots, N\}$ from the P principal components with zero reconstruction error?

i. Yes, if $P < d$

ii. Yes, if $P = d$

iii. Yes, if $P < n$

iv. No, always

Solution: B

(n) (CMU, 2013f, WCohen, EXing, sample exam, pr. 2.6)

When using Principle Components Analysis, the top n Principle Components have the following property: they provide a lower-dimensional representation of the original data, and one that allows reconstructing the original data with the minimum sum of squared errors.

i. True

ii. False

(o) (CMU, 2009s, ZBJoseph, final exam, ex.1.10) Can SVD and PCA produce the same projection result?

i. Yes

ii. No

Solution:

Yes. When the data has a zero mean vector, otherwise you have to center the data first before taking SVD.

5.8 Spectral++

5.8.1 Spectral clustering - BDA version

123. (a) (CS 189 Spring 2016 Introduction to Machine Learning Final, ex.Q1.13) Which of the following are true of spectral graph partitioning methods?

i. They find the cut with minimum weight

ii. They minimize a quadratic function subject to one constraint: the partition must be balanced

iii. They use one or more eigenvectors of the Laplacian matrix

iv. The Normalized Cut was invented at Stanford

Solution: C

(b) (CS 189 Spring 2016 Introduction to Machine Learning Final, Q1.18) Suppose you want to split a graph G into two subgraphs. Let L be G's Laplacian matrix. Which of the following could help you find a good split?

i. The eigenvector corresponding to the second-largest eigenvalue of L.

ii. The left singular vector corresponding to the second-largest singular value of L

iii. The eigenvector corresponding to the second-smallest eigenvalue of L

iv. The left singular vector corresponding to the second-smallest singular value of L

Solution: C,D

(c) (CS 189 Spring 2016 Introduction to Machine Learning Final, ex.Q1.28) In the derivation of the spectral graph partitioning algorithm, we relax a combinatorial optimization problem to a continuous optimization problem. This relaxation has the following effects.

i. The combinatorial problem requires an exact bisection of the graph, but the continuous algorithm can produce (after rounding) partitions that aren't perfectly balanced

ii. The combinatorial problem cannot be modified to accommodate vertices that have different masses, whereas the continuous problem can

iii. The combinatorial problem requires finding eigenvectors, whereas the continuous problem requires only matrix multiplication

iv. The combinatorial problem is NP-hard, but the continuous problem can be solved in polynomial time

Solution: A,D

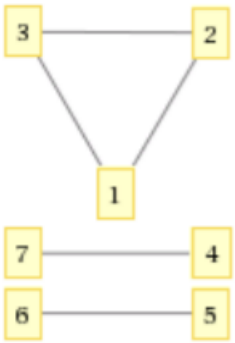
124. (CS 189 Spring 2017 Introduction to Machine Learning Final, ex.Q7)

(a) Write down the Laplacian matrix L_G of the following graph G. Every edge has weight 1.

Solution:

$$\begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$

(b) Find three orthogonal eigenvectors of L_G, all having eigenvalue 0.



Solution:

$$x = \begin{bmatrix}1\\1\\1\\0\\0\\0\\0\end{bmatrix}, y = \begin{bmatrix}0\\0\\0\\1\\0\\0\\1\end{bmatrix}, z = \begin{bmatrix}0\\0\\0\\0\\1\\1\\0\end{bmatrix}$$

(c) Use two of those three eigenvectors (it doesn't matter which two) to assign each vertex of G a spectral vector in $\mathbb{R}^2$. Draw these vectors in the plane, and explain how they partition G into three clusters. (Optional alternative: if you can draw 3D figures well, you are welcome to use all three eigenvectors and assign each vertex a spectral vector in $\mathbb{R}^3$.)

Solution:

The eigenvectors x and y give the embedding

$$1, 2, 3 \to \begin{bmatrix}1\\0\end{bmatrix}, 4, 7 \to \begin{bmatrix}0\\1\end{bmatrix}, 5, 6 \to \begin{bmatrix}0\\0\end{bmatrix}$$

Each of the three clusters is mapped to a single point in $\mathbb{R}^2$.

(d) Let K_n be the complete graph on n vertices (every pair of vertices is connected by an edge of weight 1) and let L_{K_n} be its Laplacian matrix. The eigenvectors L_{K_n} are $v_1 = 1 = [1 \dots 1]^\top$ and every vector that is orthogonal to 1. What are the eigenvalues of L_{K_n}?

Solution: $\lambda_1 = 0$ and $\lambda_2 = \dots = \lambda_n = n$.

(e) What property of these eigenvalues gives us a hint that the complete graph does not have any good partitions?

Solution: λ_2 is large, so there is no low-sparsity cut. (The optimal cut has sparsity $\geq \lambda_2/2$.)

5.8.2 Spectral clustering - ML version

125. (CMU, 2004f, HW3, ex.2)

Intro from CMU, 2010f, ASingh, HW5, pr. 3:

There is a class of clustering algorithms, called spectral clustering algorithms, which has recently become quite popular. Many of these algorithms

are quite easy to implement and perform well on certain clustering problems compared to more traditional methods like k-means. In this problem, we will try to develop some intuition about why these approaches make sense and implement one of these algorithms.

Before beginning, we will review a few basic linear algebra concepts you may find useful for some of the problems.

- If A is a matrix, it has an eigenvector v with eigenvalue λ if $Av = \lambda v$.
- For any $m \times m$ symmetric matrix A, the *Singular Value Decomposition* of A yields a factorization of A into

  $$A = USU^T$$

  where U is an $m \times m$ orthogonal matrix (meaning that the columns are pairwise orthogonal) and $S = diag(|\lambda_1|, |\lambda_2|, \ldots, |\lambda_m|)$ where the λ_i are the eigenvalues of A.

Given a set of m data points $x_1, \ldots, x_m$, the input to a spectral clustering algorithm typically consists of a matrix, A, of pairwise similarities between datapoints. A is often called the *affinity matrix*. The choice of how to measure similarity between points is one which is often left to the practitioner. A very simple affinity matrix can be constructed as follows:

$$A(i,j) = A(j,i) = \begin{cases} 1 & \text{if } d(x_i, x_j) < \Theta \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $d(x_i, x_j)$ denotes Euclidean distance between points x_i and x_j.

The general idea of spectral clustering is to construct a mapping of the datapoints to an eigenspace of A with the hope that points are well separated in this eigenspace so that something simple like k-means applied to these new points will perform well.

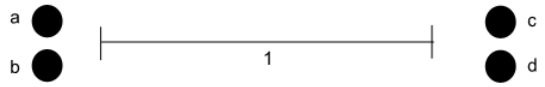


Figure 1: Simple dataset.

As an example, consider forming the affinity matrix for the dataset in Figure 1 using Equation 1 with $\Theta = 1$. Then we get the affinity matrix in Figure 2(a).

$$A = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 1 & 1 & 0 & 0 \\ b & 1 & 1 & 0 & 0 \\ c & 0 & 0 & 1 & 1 \\ d & 0 & 0 & 1 & 1 \end{array} \qquad \tilde{A} = \begin{array}{c|cccc} & a & c & b & d \\ \hline a & 1 & 0 & 1 & 0 \\ c & 0 & 1 & 0 & 1 \\ b & 1 & 0 & 1 & 0 \\ d & 0 & 1 & 0 & 1 \end{array}$$

(a) (b)

Figure 2: Affinity matrices of Figure 1 with $\Theta = 1$.

Now for this particular example, the clusters $\{a, b\}$ and $\{c, d\}$ show up as nonzero blocks in the affinity matrix. This is, of course, artificial, since we could have constructed the matrix A using any ordering of $\{a, b, c, d\}$. For example, another possibile affinity matrix for A could have been as in Figure 2(b).

The key insight here is that the eigenvectors of matrices A and $\tilde{A}$ have the same entries (just permuted). The eigenvectors with nonzero eigenvalue of A are: $e_1 = (.7, .7, 0, 0)^T$, $e_2 = (0, 0, .7, .7)^T$. And the nonzero eigenvectors of $\tilde{A}$ are: $e_1 = (.7, 0, .7, 0)^T$, $e_2 = (0, .7, 0, .7)^T$. Spectral clustering embeds the original data points in a new space by using the coordinates of these eigenvectors. Specifically, it maps the point x_i to the point $(e_1(i), e_2(i), \ldots, e_k(i))$ where $e_1, \ldots, e_k$ are the top k eigenvectors of A. We refer to this mapping as the spectral embedding. See Figure 3 for an example.

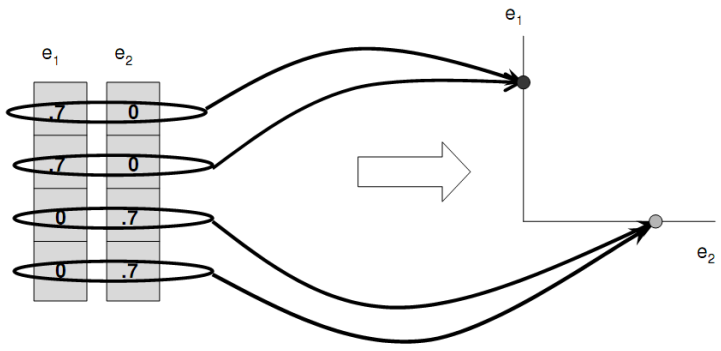
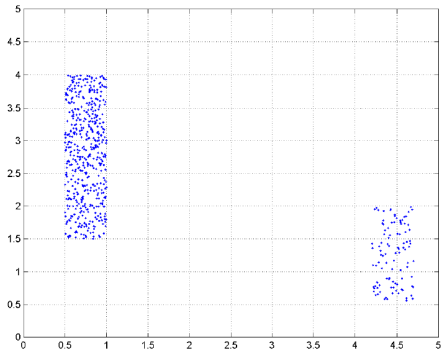


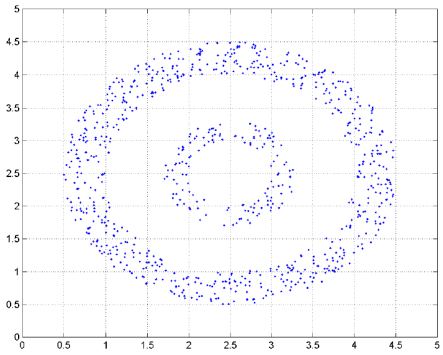
Figure 3: Using the eigenvectors of A to embed the data points. Notice that the points $\{a, b, c, d\}$ are tightly clustered in this space.

In this problem we will analyze the operation of one of the variants of spectral clustering methods on two datasets shown in the following figure. For

each of the datasets (unless directed otherwise) please answer the following questions.



(a)



(b)

(a) The first step is to build an affinity matrix. The matrix defines the degree of similarity between points.

i. Suppose we use the L2 norm to construct the following affinity matrix (let x_i denote an i-th datapoint):

$$A(i,j) = A(j,i) = \begin{cases} 1 & \text{if } |x_i - x_j|_2 < \theta \\ 0 & \text{otherwise} \end{cases}$$

What θ value would you choose and why?

ii. Suppose instead we use Gaussian kernel for our affinity matrix:

$$A(i,j) = \exp\left(-\frac{|x_i - x_j|_2}{2\sigma^2}\right)$$

What σ value would you choose and why?

(b) The second step is to compute first k dominant eigenvectors of the affinity matrix, where k is the number of clusters we want to have.

For the dataset in figure (a) and the affinity matrix defined by equation 1 is there a value of for which you can compute analytically eigenvalues corresponding to the first two dominant eigenvectors? If not, explain why not. If yes, compute and write these eigenvalues down.

(c) The third step is to cluster the rows of the matrix Y into k clusters using K-means (or a similar algorithm), where Y is constructed by placing k dominant eigenvectors into columns and re-normalizing the rows (to make each row a unit vector).

For the dataset in figure (a) and the affinity matrix defined by equation 1 write down your best guess for the coordinates of $k = 2$ cluster centers.

(d) Finally, given the clusters on matrix Y, a point x_i is declared to be in cluster j iff the i-th row of Y is in cluster j.

i. What are the final clusters you would expect to obtain for each of the datasets? Provide a rough sketch of the clusters to give an idea.

ii. What are the clusters you would expect to obtain if using EM algorithm for Gaussian Mixture Models with 2 clusters? Also provide a rough sketch of the clusters.

Solution:

(a) In general, you want to choose such a parameter that the similarity between points in different clusters is 0 (or close to it), while the similarity between neighboring points in the same cluster is close to 1. The answers to this question are based purely on eye-balling. So, for the affinity matrix in the first subpoint and the dataset in figure (a) θ in between about 2.5 and 3 will result in an ideal case. For the dataset in figure (b) it is less clear, but a value of 0.5, for example, will give us what we want. For the Gaussian kernel, we want to set σ to obtain the same effect (even though we can not really achieve 0 similarity). So, for example, for the dataset in figure (a) $\sigma = 0.5$ and for the dataset in figure (a) $\sigma = 0.3$ should separate points reasonably.

(b) Yes, consider $\theta = 2.6$. It will result in the affinity matrix that is a block matrix: $A = \begin{bmatrix} 1_{n\times n} & 0 \\ 0 & 1_{m\times m} \end{bmatrix}$ where $1_{n\times n}$ is a block of ones of size n by n, and $1_{m\times m}$ a block of ones of size m by m. n is the number of points in left cluster, and m is the number of points in the right cluster.

Such matrix has one eigenvalue $\lambda_1 = n$ (with a corresponding eigenvector $v_1 = [1, \ldots, 1_n, 0, \ldots, 0]^\top$), and a second eigenvalue $\lambda_2 = m$ (with a corresponding eigenvector $v_2 = [0, \ldots, 0_n, 1, \ldots, 1]^\top$). These are in fact the only eigenvalues of A, since $\text{rank}(A)$ is clearly 2.

(c) Given the eigenvectors $v_1 = [1, \ldots, 1_n, 0, \ldots, 0]^\top$ and $v_2 = [0, \ldots, 0_n, 1, \ldots, 1]^\top$,

we have: $Y = \begin{bmatrix} 1 & 0 \\ \vdots & \vdots \\ 1_n & 0_n \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \end{bmatrix}$

If we run K-means on the rows of Y, we get two clusters with centers at $< 1, 0 >$ and $< 0, 1 >$.

(d) i. This should result in almost perfect clustering (that is, will cluster each connected component together).

ii. The dataset in figure (a) should be clustered perfectly, while the dataset in figure (b) will probably be split in two clusters - half of the inner and outer circles in one cluster, and the other halves of the circles in the second cluster. Thus, the dataset in figure (b) is clustered incorrectly (assuming we want each circle to form its own cluster)

126. (CMU, 2004f, HW3, ex.3) The version of spectral clustering we have studied in class made use of matrix $A = D^{-1/2}WD^{-1/2}$. W is an affinity matrix with $w_{ij} = w_{ji}$ being a non-negative distance between points x_i and x_j. D is a diagonal matrix whose i-th diagonal element, d_{ii}, is the sum of W's i-th row. In the following you will need to prove several properties about A that are important for a good understanding of spectral clustering. For the proofs you might find useful the following property: for any symmetric matrix B with all nonnegative entries if u is an eigenvector with all positive entries, then no other independent eigenvector of B has the same eigenvalue.

(a) Show that a vector $v_1 = [d_{11}^{1/2}, d_{22}^{1/2}, \ldots, d_{nn}^{1/2}]^\top$ is an eigenvector of A with an eigenvalue $\lambda_1 = 1$.

(b) Prove that $\lambda_1 = 1$ is the largest eigenvalue of A.

(c) Prove that all eigenvectors orthogonal to v_1 will have an eigenvalue strictly smaller than 1.

(d) Show that $P^\infty = D^{-1/2}(v_1 v_1^\top)D^{1/2}$, where $P = D^{-1}W$ is the probability transition matrix.

Solution:

(a) We need to show that $Av_1 = v_1$.

$$D^{-1/2}WD^{-1/2}v_1 =$$
$$D^{-1/2}WD^{-1/2}[d_{11}^{1/2}, d_{22}^{1/2}, \ldots, d_{nn}^{1/2}]^\top =$$
$$D^{-1/2}W[1, 1, \ldots, 1]^\top =$$
$$\left[\frac{\sum_j(w_{1j})}{d_{11}^{1/2}}, \ldots, \frac{\sum_j(w_{nj})}{d_{nn}^{1/2}}\right]^\top =$$
$$[d_{11}^{1/2}, \ldots, d_{nn}^{1/2}]^\top =$$
$$v_1$$

(b) We have just shown that A has an eigenvalue of $\lambda_v = 1$ with a corresponding fully positive eigenvector v. We need to show no eigenvector can have an eigenvalue larger than 1. We prove by contradiction. Suppose not and there exists an eigenvector u, s.t. $\lambda_u > 1$.

Let us derive a scaled version of this vector, $u' = c \ldots u$, where $c = \min_{i;s.t.u_i>0} \frac{v_i}{u_i}$ (Such c exists because v is fully positive, u is orthogonal to v since A is symmetric, and therefore in order to have $u \cdot v = 0$, u must contain both negative and positive elements). Clearly, u' is still an eigenvector of A with the same eigenvalue. We also now have the following: $\forall i, u'_i \leq v_i$ and $\exists j$,$u'_j = v_j$. As a result, $v - u' \geq 0$, where 0 is a column vector of zeros. Consequently, we must have $A(v - u') \geq 0$ since A is non-negative.

On the other hand, $A(v - u') = Av - Au' = \lambda_v v - \lambda_u u' = v - \lambda_u u'$. However, since $\lambda_u > 1$, we will have $v_j - \lambda_u u'_j < 0$ for j, $u'_j = v_j$. Thus, $A(v - u') \ngeq 0$.

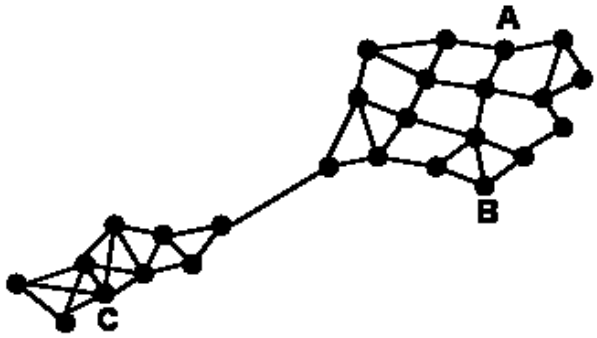
(c) This follows directly from the hint and the property we proved in the second subpoint.

(d) $P^\infty = D^{-1/2}(A^\infty)D^{1/2} = D^{-1/2}(\lambda_1 v_1 v_1^\top + \lambda 2 v_2 v_2^\top + \ldots)D^{1/2}$

Since the eigenvectors of a symmetric matrix A can always be made orthonormal, the dot-product of any two distinct eigenvectors will be 0 and the dot-product of same eigenvectors will be 1. Hence, $(\lambda_1 v_1 v_1^\top + \lambda_2 v_2 v_2^\top + \ldots)^\infty = (\lambda_1^\infty v_1 v_1^\top + \lambda_2 v_2 v_2^\top + \ldots)$. Finally, since $\forall i > 1$, $|\lambda_i < 1|$, $P^\infty = D^{-1/2}(v_1 v_1^\top)D^{1/2}/$ (Here, we took a bit of a shortcut: in this subpoint we have proved that $\lambda_i < 1$, but the fact that $\lambda_i > -1$ follows exactly from the same arguments as in the second subpoint except using $\max_{i;s.t.:u_i<0}$ instead os $\min_{i;s.t.:u_i>0}$ in the computations of c).

127. (CMU, 2004f, midterm, ex.9)

Consider the graph below. Let W be the distance matrix for this graph where $w_{i,j} = 1$ iff there is an edge between nodes i and j and otherwise $w_{i,j} = 0$. We will define the matrices D and P as we did in class by setting $D_{i,i=\sum_j w_{i,j}}$ and $P = D^{-1}W$. As we mentioned in class, P is the probability transition matrix for this graph. We denote by $P^t_{i,j}$ the i, j entry in the matrix P raised to the power of t.



For each of the expressions below, replace ? with either $<$, $>$ or $=$ and briefly explain your reasoning.

(a) $P^{20}_{A,C} ? P^{100}_{A,C}$

(b) $P^{20}_{A,B} ? P^{100}_{A,B}$

(c) $\sum_j P^{20}_{A,j} ? \sum_j P^{100}_{A,j}$

(d) $P^{\infty}_{B,A} ? P^{\infty}_{B,C}$

Solution:

(a) $P^{20}_{A,C} < P^{100}_{A,C}$
As the power of P increases it is more likely to transition to another cluster. Since A and C are in different clusters, it is more likely to end up in C when we take 100 steps than when we take 20 steps.

(b) $P^{20}_{A,B} > P^{100}_{A,B}$
A and B are in the same cluster. It is more likely to stay in the same cluster when the power of P is low (few steps) than for higher powers of P (many steps).

(c) $\sum_j P^{20}_{A,j} = \sum_j P^{100}_{A,j}$
P^t for any t is a probability transition matrix and so its rows always sum to 1.

(d) $P^{\infty}_{B,A} < P^{\infty}_{B,C}$
At the limit, the point we end at is independent of the point we started at. Thus, we need to evaluate the (fixed) probability of ending in A vs. the probability of ending in C. In class, we have shown that this probability is proportional to the components of the first eigenvector of the symmetric matrix we defined ($D^{-1/2}WD^{-1/2}$). In class (and in the problem set) we have derived the actual values for the entries of this vector. As we showed, these entries are the square root of the sum of the rows of W. Since in our case rows sum up to the out degree (or in degree) of the nodes, the probability that we will end up at a certain point is proportional to the connectivity of that point. Since C is connected to 5 other nodes whereas A is only connected to 3, $P^{\infty}_{B,A} < P^{\infty}_{B,C}$.

128. (CMU, 2009s, ZBJoseph, final exam, ex.1.9) Assume we would like to use spectral clustering to cluster n elements. We are using the k nearest neighbor method we discussed for generating the graph that would be used in the clustering procedure. Following this process:

(a) What is the maximum number of nodes that a single node is connected to?

(b) What is the minimum number of nodes that a single node is connected to?

Solution:

(a) $n - 1$

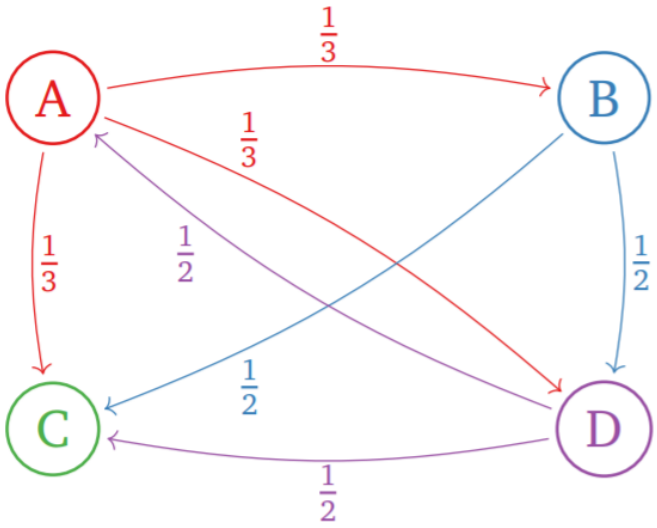
(b) k

129. (CMU, 2008f, EXing, final, pr. 1.1) True or False: PCA and Spectral Clustering (such as Andrew Ng's) perform eigendecomposition on two different matrices. However, the size of these two matrices are the same.

Solution: False

5.8.3 Ranking Webpages

130. (`https://textbooks.math.gatech.edu/ila/ila.pdf` ILA pag 354) What is the **PageRank** vector for the following internet? (Use the damping factor $p = 0.15$.).



Which page is the most important? Which is the least important?

Solution:

First we compute the modified importance matrix:

$$A = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix} \overset{\text{modify}}{=} A' = \begin{bmatrix} 0 & 0 & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{3} & 0 & \frac{1}{4} & 0 \\ \frac{1}{3} & \frac{1}{2} & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & \frac{1}{4} & 0 \end{bmatrix}$$

Choosing the damping factor $p = 0.15$, the Google Matrix is

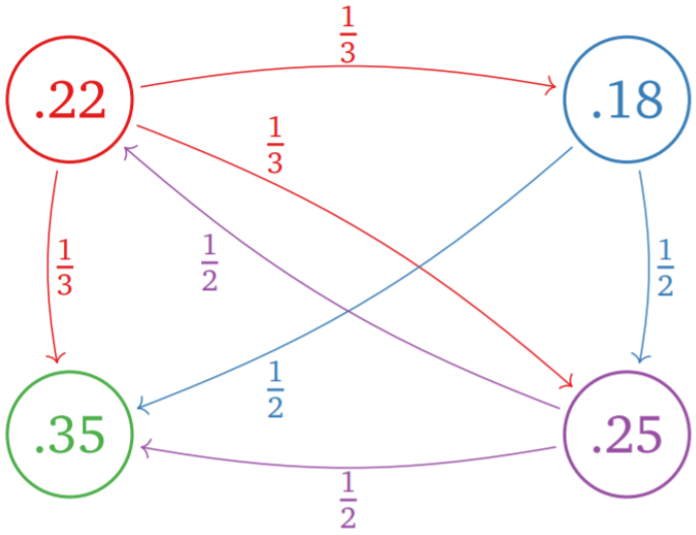
$$M = 0.85 \cdot \begin{bmatrix} 0 & 0 & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{3} & 0 & \frac{1}{4} & 0 \\ \frac{1}{3} & \frac{1}{2} & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & \frac{1}{4} & 0 \end{bmatrix} + 0.15 \cdot \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}$$

$$\approx \begin{bmatrix} 0.0375 & 0.0375 & 0.2500 & 0.4625 \\ 0.3208 & 0.0375 & 0.2500 & 0.0375 \\ 0.3208 & 0.4625 & 0.2500 & 0.4625 \\ 0.3208 & 0.4625 & 0.2500 & 0.0375 \end{bmatrix}$$

The PageRank vector in the steady state:

$$w \approx \begin{bmatrix} 0.2192 \\ 0.1752 \\ 0.3558 \\ 0.2498 \end{bmatrix}$$

This is the PageRank:



Page C is the most important, with a rank of 0.558, and page B is the least important, with a rank of 0.1752.

131. (CMU, 2017f, NBalcan, HW5, pr.2.1) **Ranking of Webpages**

The goal of webpage ranking algorithms is to find a small set of most authoritative pages relevant to the query. In this problem we will see how PCA can be used to rank webpages on the Internet. Here, we will consider the **Hyperlink Induced Topic Search (HITS) algorithm**.

Let's view the Internet as a directed graph $G = (V; E)$, where every webpage, denoted as n_i, is a node in V , and every hyperlink from n_i to n_j , denoted as e_{ij} , is a directed edge in E. Before describing the HITS algorithm we define two terms:

- Authorities: pages that are relevant and are linked to by many other pages.

- Hubs: pages that link to many related authorities.

As a first step, a focused subgraph of the Internet is constructed based on the query. In the second step HITS finds authoritative results using link analysis. It assigns each webpage ni an authority score x_i and a hub score y_i. The authority score x_i is a scaled sum of the hub scores of other webpages pointing to webpage ni. The hub score is the scaled sum of the authority scores of other webpages that webpage ni is pointing out to. Let x and y be the vector of authority scores and hub scores, respectively. Also, let A be the adjacent matrix of the graph G, i.e., $A_{ij} = 1$ if $e_{ij} \in E$ and $A_{ij} = 0$ otherwise. HITS performs the following two updates iteratively

$$y \leftarrow \frac{Ax}{\|Ax\|}$$
$$x \leftarrow \frac{A^\top y}{\|A^\top y\|}$$

(a) Show that the unit norm eigenvectors of $AA^\top$ (for y) and $A^\top A$ (for x) give fixed points of the algorithm.

(b) Show that, in general, y and x converge to the unit-norm eigenvectors associated with the maximum eigenvalue of $AA^\top$ and $A\top A$, respectively. Explain why not any other eigenvector.

Solution:

(a) Suppose xt is the value of x after t iterations and similarly suppose y_t is the value of y after t iterations. Then we have

$$x_{t+1} = \frac{(A^\top A)x_t}{\|(A^\top A)x_t\|}$$

(Note that this is the power iteration). If x is a fixed point of the above equation then it should satisfy

$$x = \frac{(A^\top A)x}{\|(A^\top A)x\|}$$

This clearly shows that the unit norm eigenvectors of $A^\top A$ give the fixed points of the algorithm for x. We can use similar argument to show that the unit norm eigenvectors of $AA^\top$ give the fixed points of the algorithm for y.

(b) We have

$$x_t = \frac{(A^\top A)^t x_0}{\|(A^\top A)^t x_0\|}$$

Let $x_0 = \sum_{i=1}^{|V|} \alpha_i u_i$, where u_i is the eigenvector of $A^\top A$ corresponding to i-th largest eigenvalue, λ_i (for simplicity assume that $\lambda_1 > \lambda_2$). Suppose $\alpha_1 \neq 0$. Then we have

$$(A^\top A)^t x_0 = \sum_{i=1}^{|V|} \alpha_i (\lambda_i)^t u_t$$

and $\|(A^\top A)^t x_0\| = \sum_{i=1}^{|V|} \alpha_i (\lambda_i)^t$. This shows that for large t, $\frac{(A^\top A)^t x_0}{\|(A^\top A)^t x_0\|}$ converges to u_i. This shows that, as long as $\lambda_1 \neq 0$, x converges to the unit norm eigenvector of $A^\top A$ with maximum eigenvalue.

6 Non-negative Matrix Factorization - NMF

132. (after `https://www.jjburred.com/research/pdf/jjburred_nmf_updates.pdf`)The goal of Non-negative Matrix Factorization (**NMF**) is to decompose a matrix of non-negative (i.e., zero or positive) elements into a product of two factor matrices, both of them containing also non-negative elements. It is common to use the notation X for the input matrix (of size $M \times N$), W for the first factor matrix (sometimes called basis matrix or feature matrix, of size $M \times K$) and H for the second factor matrix (sometimes called coefficient matrix or activation matrix, of size $K \times N$). The resulting factorization is often approximate:

$$X \approx WH$$

The optimization problem is as follows:

$$\{\hat{W}, \hat{H}\} = \arg \min_{W \in \mathbb{R}^{M \times K}, H \in \mathbb{R}^{K \times N}; w_{mk}, h_{kn} \geq 0} \|X - WH\|_{\text{Fro}}^2$$

where $\hat{W}$ and $\hat{H}$ are the obtained output factors, and w_{mk} and h_{kn} denote the entries of the factor matrices.

Derive an algorithm to solve this optimization problem. Hint: see proofs at Detailed derivation of multiplicative update rules for NMF `https://www.jjburred.com/research/pdf/jjburred_nmf_updates.pdf`

7 LDA, GDA, QDA, FDA

7.1 Linear/Gaussian Discriminant Analysis - LDA=GDA

133. (Stanford, Andrew Ng, 2009f, HW1, ex.4) **Gaussian discriminant analysis (GDA) = Linear discriminant analysis (LDA)**

See also the following interesting paper: `https://arxiv.org/pdf/1906.02590.pdf`

Suppose we are given a dataset $\{(x^{(i)}, y^{(i)}); i = 1, \ldots, m\}$ consisting of m independent examples, where $x^{(i)} \in \mathbb{R}^n$ are n-dimensional vectors, and $y^{(i)} \in \{0, 1\}$. We will model the joint distribution of (x, y) according to:

$$p(y) = \phi^y (1-\phi)^{1-y}$$

$$p(x|y=0) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_0)^\top \Sigma^{-1}(x-\mu_0)\right)$$
$$p(x|y=1) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_1)^\top \Sigma^{-1}(x-\mu_1)\right)$$

Here, the parameters of our model are ϕ, Σ, μ_0 and μ_1. (Note that while there're two different mean vectors μ_0 and mu_1, there's only one covariance matrix Σ.)

(a) Suppose we have already fit ϕ, Σ, μ and μ_1, and now want to make a prediction at some new query point x. Show that the posterior distribution of the label at x takes the form of a logistic function, and can be written

$$p(y=1|x; \phi, \Sigma, \mu_0, \mu_1) = \frac{1}{1+\exp(-\theta^\top x)}$$

where θ is some appropriate function of ϕ, Σ, μ_0, μ_1. (Note: To get your answer into the form above, for this part of the problem only, you may have to redefine the $x^{(i)}$'s to be $n+1$ - dimensional vectors by adding the extra coordinate $x_0^{(i)} = 1$, like we did in class.)

(b) For this part of the problem only you may assume (the dimension of x) is 1, so that $\Sigma = [\sigma^2]$ is just a real number, and likewise the determinant of Σ is given by $|\Sigma| = \sigma^2$. Given the dataset, we claim that the maximum likelihood estimates of the parameters are given by

$$\phi = \frac{1}{m}\sum_{i=1}^{m} 1\{y^{(i)} = 1\}$$

$$\mu_0 = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}}$$

$$\mu_1 = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}$$

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^\top$$

The log-likelihood of the data is

$$l(\phi, \mu_0, \mu_1, \Sigma) = \log \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}; \phi, \mu_0, \mu_1, \Sigma)$$

$$= \log \prod_{i=1}^{m} p(x^{(i)}|y^{(i)}; \mu_0, \mu_1, \Sigma)p(y^{(i)}; \phi)$$

By maximizing l with respect to the four parameters, prove that the maximum likelihood estimates of ϕ, μ_0, μ_1 and Σ are indeed as given in the formulas above. (You may assume that there is at least one positive and one negative example, so that the denominators in the definitions of μ_0 and μ_1 above are non-zero.)

(c) Without assuming that $n = 1$, show that the maximum likelihood estimates of ϕ,μ_0,μ_1 and Σ are as given in the formulas in the second part. [Note: If you're fairly sure that you have the answer to this part right, you don't have to do the second part, since that's just a special case.]

Solution:

(a) Since the given formulae are conditioned on y, use Bayes rule to get:

$$p(y = 1|x; \phi, \Sigma, \mu_0, \mu_1) = \frac{p(x|y = 1; \phi, \Sigma, \mu_0, \mu_1)p(y = 1; \phi, \Sigma, \mu_0, \mu_1)}{p(x; \phi, \Sigma, \mu_0, \mu_1)}$$

$$= \frac{p(x|y = 1; \dots)p(y = 1; \dots)}{p(x|y = 1; \dots)p(y = 1; \dots) + p(x|y = 0; \dots)p(y = 0; \dots)}$$

$$= \frac{\exp(-\frac{1}{2}(x-\mu_1)^\top\Sigma^{-1}(x-\mu_1))\phi}{\exp(-\frac{1}{2}(x-\mu_1)^\top\Sigma^{-1}(x-\mu_1))\phi + \exp(-\frac{1}{2}(x-\mu_0)^\top\Sigma^{-1}(x-\mu_0))(1-\phi)}$$

$$\frac{1}{1+\frac{1-\phi}{\phi}\exp(-\frac{1}{2}(x-\mu_0)^\top\Sigma^{-1}(x-\mu_0)+\frac{1}{2}(x-\mu_1)^\top\Sigma^{-1}(x-\mu_1))}$$

$$= \frac{1}{1+\exp(\log(\frac{1-\phi}{\phi})-\frac{1}{2}(x-\mu_)^\top\Sigma^{-1}(x-\mu_0)+\frac{1}{2}(x-\mu_1)^\top\Sigma^{-1}(x-\mu_1))}$$

$$= \frac{1}{1+\exp(\frac{1}{2}(-2\mu_0^\top\Sigma^{-1}x+\mu_0^\top\Sigma^{-1}\mu_0+2\mu_1^\top\Sigma^{-1}x-\mu_1^\top\Sigma^{-1}\mu_1)+\log(\frac{1-\phi}{\phi}))}$$

where we have simplified the denominator in the penultimate step by expansion, i.e.,

$$-\frac{1}{2}(x-\mu_0)^\top\Sigma^{-1}(x-\mu_0)+\frac{1}{2}(x-\mu_1)^\top\Sigma^{-1}(x-\mu_1)$$

$$-\frac{1}{2}(x^\top\Sigma^{-1}x-\mu_0^\top\Sigma^{-1}x-x^\top\Sigma^{-1}\mu_0+\mu_0^\top\Sigma^{-1}\mu_0-x^\top\Sigma^{-1}x+\mu_1^\top\Sigma^{-1}x+x^\top\Sigma^{-1}\mu_1-\mu_1^\top\Sigma^{-1}\mu_1)$$

$$= -\frac{1}{2}(\mu_0^\top\Sigma^{-1}x-(x^\top\Sigma^{-1}\mu_0)^\top+\mu_0^\top\Sigma^{-1}\mu_0+\mu_1^\top\Sigma^{-1}x+(x^\top\Sigma^{-1}\mu_1)^\top-\mu_1^\top\Sigma^{-1}\mu_1)$$

Recall that th qeustion was to find θ for $\exp(-\theta^\top x)$, after adding a constant intercept term $x_0 = 1$, we have θ equal to:

$$\begin{bmatrix}\frac{1}{2}(\mu_0^\top\Sigma^{-1}\mu_0-\mu_1^\top\Sigma^{-1}\mu_1)\log(\frac{1-\phi}{\phi})\\ \Sigma^{-1}\mu_1-\Sigma^{-1}\mu_0\end{bmatrix}$$

(b) The derivation follows from the more general one for the next part.

(c) First, derive the expression for the log-likelihood of the training data:

$$l(\phi,\mu_0,\mu_1,\Sigma) = \log\prod_{i=1}^m p(x^{(i)}|y^{(i)};\mu_0,\mu_1,\Sigma)p(y^{(i)};\phi)$$

$$= \sum_{i=1}^m \log p(x^{(i)}|y^{(i)};\mu_0,\mu_1,\Sigma) + \sum_{i=1}^m \log p(y^{(i)};\phi)$$

$$\approx \sum[\frac{1}{2}\log\frac{1}{|\Sigma|}-\frac{1}{2}(x^{(i)}-\mu_{y^{(i)}})^\top\Sigma^{-1}(x^{(i)}-\mu_{y^{(i)}})+y^{(i)}\log\phi+(1-y^{(i)})\log(1-\phi)]$$

where constant terms independent of the parameters have been ignored in the last expression.

Now, the likelihood is maximized by setting the derivative (or gradient) with respect to each of the parameters to zero.

$$\frac{\partial l}{\partial \phi} = \sum_{i=1}^{m} [\frac{y^{(i)}}{\phi} - \frac{1 - y^{(i)}}{1 - \phi}]$$

$$= \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}{\phi} - \frac{m - \sum_{i=1}^{m} 1\{y^{(i)} = 1\}}{1 - \phi}$$

Setting this equal to zero and solving for ϕ gives the maximum likelihood estimate.

For μ_0, take the gradient of the log-likelihood, and then use the same kinds of tricks as were used to analytically solve the linear regression problem.

$$\nabla_{\mu_0} l = -\frac{1}{2} \sum_{i:y^{(i)}=0} \nabla_{\mu_0} (x^{(i)} - \mu_0)^\top \Sigma^{-1} (x^{(i)} - \mu_0)$$

$$= -\frac{1}{2} \sum_{i:y^{(i)}=0} \nabla_{\mu_0} [\mu_0^\top \Sigma^{-1} \mu_0 - {x^{(i)}}^\top \Sigma^{-1} \mu_0 - \mu_0^\top \Sigma^{-1} x^{(i)}]$$

$$= -\frac{1}{2} \sum_{i:y^{(i)}=0} \nabla_{\mu_0} \mathrm{tr} [\mu_0^\top \Sigma^{-1} \mu_0 - x^{(i)} \Sigma^{-1} \mu_0 - \mu_0^\top \Sigma^{-1} x^{(i)}]$$

$$= -\frac{1}{2} \sum_{i:y^{(i)}=0} [2\Sigma^{-1} \mu_0 - 2\Sigma^{-1} x^{(i)}]$$

The last step uses matrix calculus identities (specifically, those given in page 8 of the lecture notes), and also the fact that Σ (and thus Σ^{-1}) is symmetric.

Setting this gradient to zero given the maximum likelihood estimate for μ_0. The derivation for μ_1 is similar to the one above.

For Σ, we find the gradient with respect to $S = \Sigma^{-1}$ rather than Σ just to simplify the derivation (note that $|S| = \frac{1}{|\Sigma|}$). You should convince yourself that the maximum likelihood estimate Σ_m as $S_m^{-1} = \Sigma_m$.

$$\nabla_S l = \sum_{i=1}^{m} \nabla_S [\frac{1}{2} \log |S| - \frac{1}{2} \underbrace{(x^{(i)} - \mu_{y^{(i)}})^\top}_{b_i^\top} S \underbrace{(x^{(i)} - \mu_{y^{(i)}})}_{b_i}]$$

$$= \sum_{i=1}^{m} [\frac{1}{2|S|} \nabla_S |S| - \frac{1}{2} \nabla_S b_i^\top S b_i]$$

But, we have the following identities:

$$\nabla_S |S| = |S|(S^{-1})^\top$$

$$\nabla_S b_i^\top S b_i = \nabla_S \text{tr}(b_i^\top S b_i) = \nabla_S \text{tr}(S b_i b_i^\top) = b_i b_i^\top$$

In the above, we again used matrix calculus identities, and also the commutativity os the trace operator for square matrices. Putting these into the original equation, we get:

$$\nabla_S l = \sum_{i=1}^{m} [\frac{1}{2} S^{-1} - \frac{1}{2} b_i b_i^\top]$$

$$= \frac{1}{2} \sum_{i=1}^{m} [\Sigma - b_i b_i^\top]$$

Setting this to zero gives the required maximum likelihood estimate for Σ.

134. (CS 189 Spring 2016 Introduction to Machine Learning Midterm, ex.Q4) Let's derive the decision boundary when one class is Gaussian and the other class is exponential. Our feature space is one-dimensional ($d = 1$), so the decision boundary is a small set of points. We have two classes, named N for normal and E for exponential. For the former class ($Y = N$), the prior probability is $\pi_N = P(Y = N) = \frac{\sqrt{2\pi}}{1+\sqrt{2\pi}}$ and the class conditional $P(X|Y = N)$ has the normal distribution $\mathcal{N}(0, \sigma^2)$. For the latter, the prior probability is $\pi_E - P(Y - E) = \frac{1}{1+\sqrt{2\pi}}$ and the class conditional has the exponential distribution

$$(X = x|Y = E) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x \geq 0 \\ \lambda 0 & \text{if } x < 0 \end{cases}$$

Write an equation in x for the decision boundary. (Only the positive solutions of your equation will be relevant; ignore all $x < 0$.) Use the 0-1 loss function. Simplify the equation until it is quadratic in x. (You don't need to solve the quadratic equation. It should contain the constants σ and λ. Ignore the fact that 0 might or might not also be a point in the decision boundary.) Show your work, starting from the posterior probabilities.

Solution:

Ignoring the possibility of $x = 0$, the decision boundary is the set of positive solutions to

$$P(Y = N|X = x) = P(Y = E|X = x)$$

$$\frac{P(X = x|Y = N)P(Y = N)}{P(X = x)} = \frac{P(X = x|Y = E)P(Y = E)}{P(X = x)}$$

$$\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \frac{\sqrt{2\pi}}{1 + \sqrt{2\pi}} = \lambda e^{-\lambda x} \frac{1}{1 + \sqrt{2\pi}}$$

$$-\ln -\frac{x^2}{2\sigma^2} = \ln \lambda - \lambda x$$

$$0 = \frac{x^2}{2\sigma^2} - \lambda x + \ln \lambda + \ln \sigma.$$

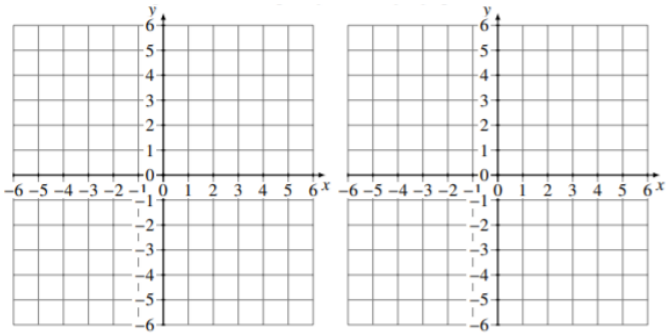
Note that the last term can be abbreviated to $\ln(\lambda\sigma)$. The last line above is not necessary for full credit; the second-last line counts as a "quadratic equation". The first line of math also is not necessary for full credit, but Bayes' Theorem must implicitly be present.

135. (CS 189 Spring 2017 Introduction to Machine Learning Midterm, ex.Q4) Consider a two-class classification problem in $d = 2$ dimensions. Points from these classes come from multivariate Gaussian distributions with a common mean but different covariance matrices.

$$X_C \sim \mathcal{N}\left(\mu = \begin{bmatrix}1\\1\end{bmatrix}, \Sigma_C = \begin{bmatrix}4 & 0\\0 & 4\end{bmatrix}\right)$$

$$X_D \sim \mathcal{N}\left(\mu = \begin{bmatrix}1\\1\end{bmatrix}, \Sigma_D = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right)$$

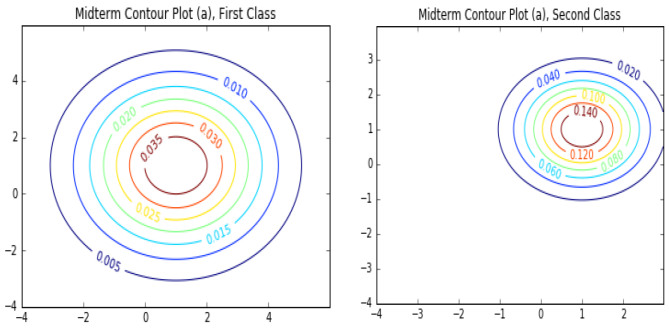
(a) Plot some isocontours of the probability distribution function $P(\mu, \Sigma_C)$ of X_C on the left graph. (The particular isovalues don't matter much, so long as we get a sense of the isocontour shapes.) Plot the isocontours of $P(\mu, \Sigma_D)$ for the same isovalues (so we can compare the relative spacing) on the right graph.



(b) Suppose that the priors for the two classes are $\pi_C = \pi_D = \frac{1}{2}$ and we use the 0-1 loss function. Derive an equation for the points x in the Bayes optimal decision boundary and simplify it as much as possible. What is the geometric shape of this boundary? (Hint: try to get your equations to include the term $|x - \mu|^2$ early, then keep it that way.) (Hint 2: you can get half of these points by guessing the geometric shape.)

Solution:

(a) As there are no covariance terms, the isocontours are axis-aligned. As the variances are equal, the isocontours are circles. The student should make an attempt to demonstrate that they understand that higher standard deviations results in larger "gaps" between significant isocontours, as below.



(b)

$$P(Y = 1|X = P(Y = 2|X))$$

$$P(X|Y = 1)\pi_C = P(X|Y = 2)\pi_D$$

$$\frac{1}{2\pi\sqrt{|\Sigma_C|}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma_C^{-1}(x-\mu)\right) = \frac{1}{2\pi\sqrt{|\Sigma_D|}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma_D^{-1}(x-\mu)\right)$$

$$\frac{1}{4}\exp\left(-\frac{1}{2}\frac{1}{4}|x-\mu|^2\right) = \exp\left(-\frac{1}{2}|x-\mu|^2\right)$$
$$-\frac{1}{8}|x-\mu|^2 - \ln 4 = -\frac{1}{2}|x-\mu|^2$$
$$|x-\mu|^2 = \frac{8}{3\ln 4}$$

This is a circle with center $(1,1)$ and radius $\sqrt{\frac{8\ln 4}{3}} \approx 1.92$.

136. (a) (CS 189 Spring 2016 Introduction to Machine Learning Midterm, ex. Q1.(h)) Gaussian discriminant analysis

i. models $P(Y=y|X)$ as a Gaussian
ii. models $P(Y=y|X)$ as a logistic function
iii. is an example of a generative model
iv. can be used to classify points without ever computing an exponential

Solution: B,C,D

(b) (CS 189 Spring 2016 Introduction to Machine Learning Midterm, ex. Q1.(s)) In Gaussian discriminant analysis, if two classes come from Gaussian distributions that have different means, may or may not have different covariance matrices, and may or may not have different priors, which decision boundary shapes are possible?

i. a hyperplane
ii. a nonlinear quadratic surface (quadratic = the isosurface of a quadratic function)
iii. a surface that is not a quadratic
iv. the empty set (the classifier always returns the same class)

Solution: A,B,D

(c) (CS 189 Spring 2017 Introduction to Machine Learning Midterm, ex. Q1.(m)) In LDA/GDA, what are the effects of modifying the sample covariance matrix as $\tilde{\Sigma} = (1-\lambda)\Sigma + \lambda I$, where $0 \leq \lambda \leq 1$?

i. $\tilde{\Sigma}$ is positive definite
ii. $\tilde{\Sigma}$ is invertible
iii. Increases the eigenvalues of Σ by λ
iv. The isocontours of the quadratic form of $\tilde{\Sigma}$ are closer to spherical

Solution: A,B,D

7.2 Quadratic Discriminant Analysis - QDA

137. (CS 189 Spring 2019 Introduction to Machine Learning Midterm, ex.Q3) QDA

(a) Consider 12 labeled data points sampled from three distinct classes:

Class 0: $\begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}-2\\0\end{bmatrix}, \begin{bmatrix}5\\3\end{bmatrix}, \begin{bmatrix}-3\\-5\end{bmatrix}$

Class 1: $\begin{bmatrix}\sqrt{2}\\\sqrt{2}\end{bmatrix}, \begin{bmatrix}-\sqrt{2}\\\sqrt{2}\end{bmatrix}, \begin{bmatrix}4\sqrt{2}\\-\sqrt{2}\end{bmatrix}, \begin{bmatrix}4\sqrt{2}\\-\sqrt{2}\end{bmatrix}, \begin{bmatrix}-4\sqrt{2}\\-\sqrt{2}\end{bmatrix}$

Class 2: $\begin{bmatrix}3\\5\end{bmatrix}, \begin{bmatrix}1\\3\end{bmatrix}, \begin{bmatrix}8\\6\end{bmatrix}, \begin{bmatrix}0\\-2\end{bmatrix}$

For each class $C \in \{0, 1, 2\}$, compute the class sample mean μ_C, the class sample covariance matrix Σ_C, and the estimate of the prior probability π_C that a point belongs to class C. (Hint: $\mu_1 = \mu_0$ and $\Sigma_2 = \Sigma_0$)

(b) Sketch one or more isocontours of the QDA-produced normal distribution or quadratic discriminant function (they each have the same contours) for each class. The isovalues are not important; the important aspects are the centers, axis directions, and relative axis lengths of the isocontours. Clearly label the centers of the isocontours and to which class they correspond.

(c) Suppose that we apply LDA to classify the data given in the first part. Why will this give a poor decision boundary?

Solution:

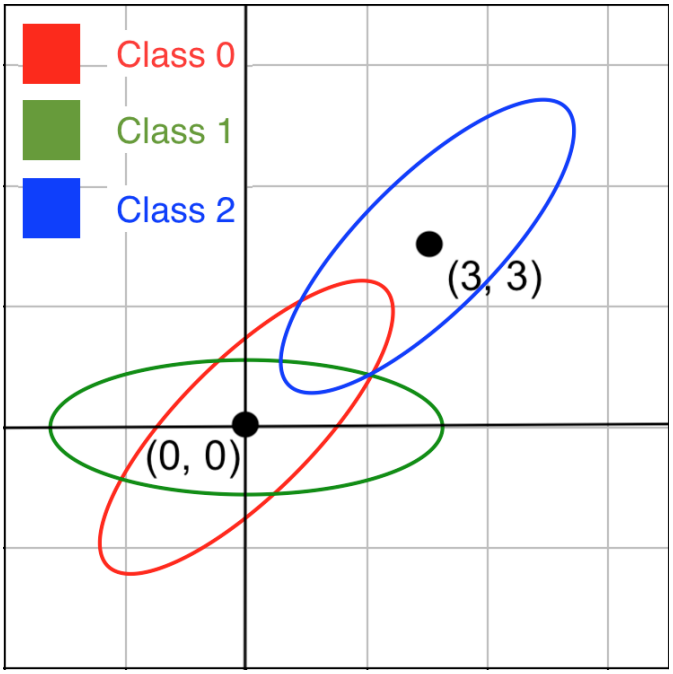
(a) Class 0: Mean is $\begin{bmatrix}0\\0\end{bmatrix}$, covariance is $\begin{bmatrix}9.5 & 7.5\\7.5 & 9.5\end{bmatrix}$, prior is $\frac{1}{3}$

Class 1: Mean is $\begin{bmatrix}0\\0\end{bmatrix}$, covariance is $\begin{bmatrix}17 & 0\\0 & 2\end{bmatrix}$, prior is $\frac{1}{3}$

Class 2: Mean is $\begin{bmatrix}3\\3\end{bmatrix}$, covariance is $\begin{bmatrix}9.5 & 7.5\\7.5 & 9.5\end{bmatrix}$, prior is $\frac{1}{3}$

(b) The ellipses for classes 0 and 1 both need to be centered around the origin. The ellipses for class 0 should be aligned on a 45 degree rotation of the coordinate axes with more variance along the [1; 1] direction than the [1; -1] direction. The ellipses for class 1 should be axis aligned with more variance along the x-axis. The ellipses for class 2 must be a translation of the ellipses for class 0.

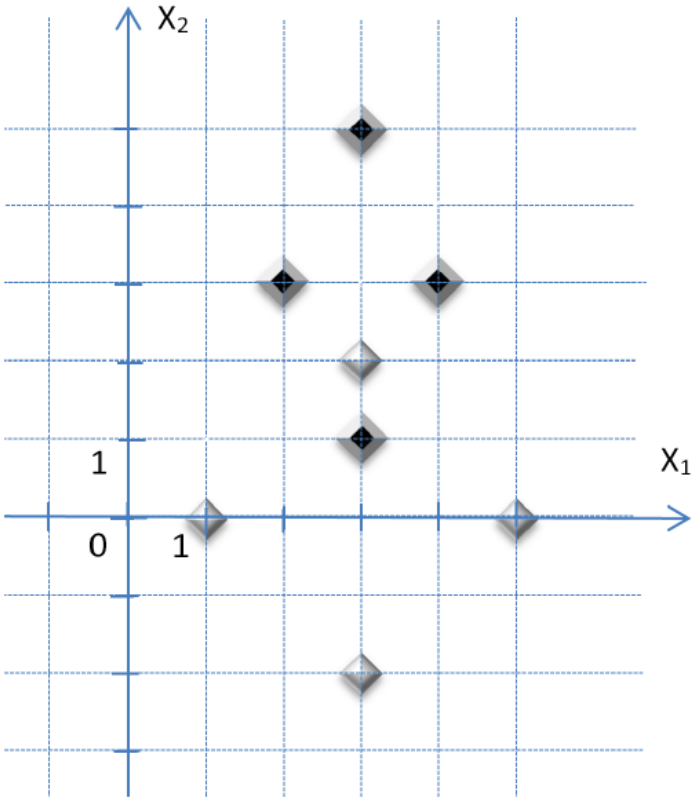
Note: If incorrect covariance matrices were calculated in the first part, full credit on this part should still be possible so long as each ellipse is centered correctly around the appropriate mean and the variance is in the appropriate directions.



(c) The discriminant functions for classes 0 and 1 would have the exact same mean and covariance, so there would be no decision boundary between them.

138. (CS 189 Spring 2018 - DIS9, ex.1) QDA

We are training data for a two class classification problem as laid out in the figure.The black dots are examples of the positive class ($y = +1$) and the white dots examples of the negative class ($y = -1$).



(a) Draw on the figure the position of the class centroids $\mu_{(+)}$ and $\mu_{(-)}$ for the positive and negative class respectively, and indicate them as circled $(+)$ and $(-)$. Give their coordinates:

$$\mu_{(+)} = \begin{bmatrix} \dots \\ \dots \end{bmatrix}, \mu_{(-)} = \begin{bmatrix} \dots \\ \dots \end{bmatrix}$$

(b) Compute the covariance matrices for each class:

$$\Sigma_{(+)} = \begin{bmatrix} \dots \\ \dots \end{bmatrix}, \Sigma_{(-)} = \begin{bmatrix} \dots \\ \dots \end{bmatrix}$$

(c) Assume each class has data distributed according to a bi-variate Gaussian, centered on the class centroids computed in the first question above. Draw on the figure the contour of equal likelihood $p(X = x|Y = y)$ going thourgh the data samples, fo each class. Indicate with light lines the principal axes of the data distribution for each class.

(d) Compute the determinant and the inverse of $\Sigma_{(+)}$ and $\Sigma_{(-)}$:

$$|\Sigma_{(+)}| = \ldots, |\Sigma_{(-)}| = \ldots$$

$$\Sigma_{(+)}^{-1} = \ldots, \Sigma_{(-)}^{-1} = \begin{bmatrix} \ldots \\ \ldots \end{bmatrix}$$

(e) The likelihood of examples of the positive class is given by:

$$p(X = x|Y = +1) = \frac{1}{2\pi|\Sigma_{(+)}|^{1/2}} \exp(-\frac{1}{2}(x - \mu_{(+)})^\top \Sigma_{(+)}^{-1}(x - \mu_{(+)}))$$

and there is a similar formula for $p(X = x|Y = -1)$. Compute $f_{(+)}(x) = \log(p(X = x|Y = +1))$ and $f_{(-)}(x) = \log(p(X = x|Y = -1))$. Compute $f_{(+)}(x) = \log(p(X = x|Y = -1))$. Then compute the discriminant function $f(x) = f_{(+)} - f_{(-)}$:

$$f_{(+)}(x) = \ldots$$

$$f_{(-)}(x) = \ldots$$

$$f(x) = \ldots$$

(f) Draw on the figure for each class contours increasing equal likelihood. Geometrically construct the Bayes optimal decision boundary. Compare to the formula obtained with $f(x) = 0$ after expression x_2 as a function of x_1:

$$x_2 = \ldots$$

What type of function is it?

(g) Now assume $p(Y = -1) \neq p(Y = +1)$, how does it change the decision boundary?

Solution:

(a)

$$\mu_{(+)} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

$$\mu_{(-)} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

(b)

$$\Sigma_{(+)} = \begin{bmatrix} 1/2 & 0 \\ 0 & 2 \end{bmatrix}$$

$$\Sigma_{(-)} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

(c)

(d)

$$|\Sigma_{(+)}| = 1, |\Sigma_{(-)}| = 4$$

$$\Sigma_{(+)}^{-1} = \begin{bmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$$

$$\Sigma_{(-)}^{-1} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$$

(e)

$$f_{(+)}(x) = -\frac{1}{2}(x - \mu_{(+)})^\top \Sigma_{(+)}^{-1}(x - \mu_{(+)}) - \log(2\pi|\Sigma_{(+)})$$

$$= -(x_1 - 3)^2 - \frac{1}{4}(x_2 - 3)^2 - \log(2\pi)$$

$$f_{(-)}(x) = -\frac{1}{2}(x - \mu_{(-)})^\top \Sigma_{(-)}^{-1}(x - \mu_{(-)}) - \log(2\pi|\Sigma_{(-)}|^{1/2})$$

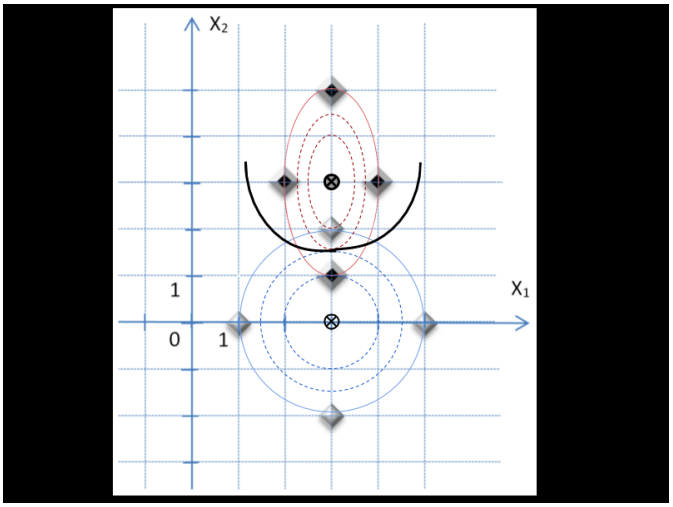
$$= -\frac{1}{4}(x_1 - 3)^2 - \frac{1}{4}x_2^2 - \log(4\pi)$$

$$f(x) = -\frac{1}{2}((x - \mu_{(+)})^\top \Sigma_{(+)}^{-1}(x - \mu_{(-)})^\top \Sigma_{(-)}^{-1}(x - \mu_{(-)})) - \log\left(\frac{|\Sigma_{(+)}|}{|\Sigma_{(-)}|}\right)$$

$$= -\frac{3}{4}(x_1 - 3)^2 + \frac{3}{2}x_2 - \frac{9}{4} + \log(2)$$

(f) We put $f(x) = 0$ so:

$$x_2 = \frac{1}{2}(x_1 - 3)^2 + \frac{3}{2} - \frac{2}{3}\log(2)$$



(g) We get that:

$$\log(p(X = x, Y = +1)) = f_{(+)}(x) + \log(p(Y = +1))$$
$$\log(p(X = x, Y = -1)) = f_{(-)}(x) + \log(p(X = x, Y = -1)) :$$

$$f_{(+)}(x) + \log(p(Y = +1)) = f_{(-)}(x) + \log(p(Y = -1))$$
$$f(x) + \log\left(\frac{p(Y = +1)}{p(Y = -1)}\right) = 0$$
$$x_2 = \frac{1}{2}(x_1 - 3)^2 + \frac{3}{2} - \frac{2}{3}\log(2) + \frac{2}{3}\log\frac{p(Y = -1)}{p(Y = +1)}$$

The boundary is shifted by $\frac{2}{3}\log\frac{p(Y=-1)}{p(Y=+1)}$

139. (CS 189 Spring 2019 Introduction to Machine Learning Midterm, Q1.(i)) Which of the following apply to linear discriminant analysis?

(a) You calculate the sample mean for each class

(b) It approximates the Bayes decision rule

(c) You calculate the sample covariance matrix using the mean of all the data points

(d) The model produced by LDA is never the same as the model produced by QDA

Solution: A,B

A: Calculating the sample mean within each class is part of LDA by definition

C: You calculate the sample covariance using the mean for each class, not the mean of all the data points

B: LDA finds what the Bayes decision rule would be under the assumption the class conditionals have normal distributions, parameterized by the sample means and covariance

D: QDA can produce the same covariance for each class as LDA

7.3 Fisher Discriminant Analysis - FDA

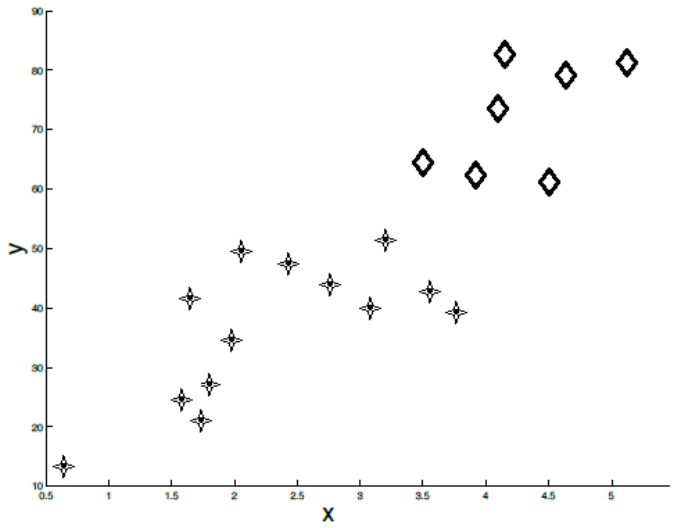
7.3.1 PCA issue

140. (CMU, 2016s, Matt Gormley, Final Exam Review, ex.4 - exams/lecture29-final) In the following plots, a train set of data points X belonging to two classes on $\mathbb{R}^2$ are given, where the original features are the coordinates (x, y). For each, answer the following questions:

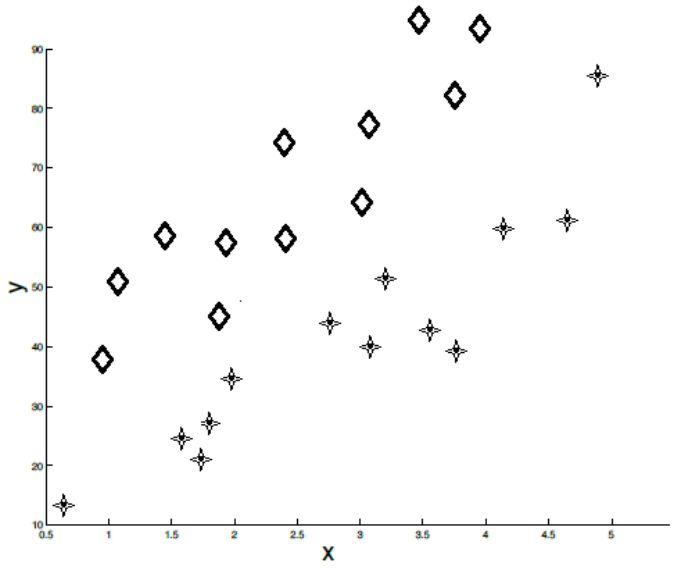
    (a) Draw all the principal components.

    (b) Can we correctly classify this dataset by using a threshold function after projecting onto one of the principal components? If so, which principal component should we project onto? If not, explain in 1–2 sentences why it is not possible.

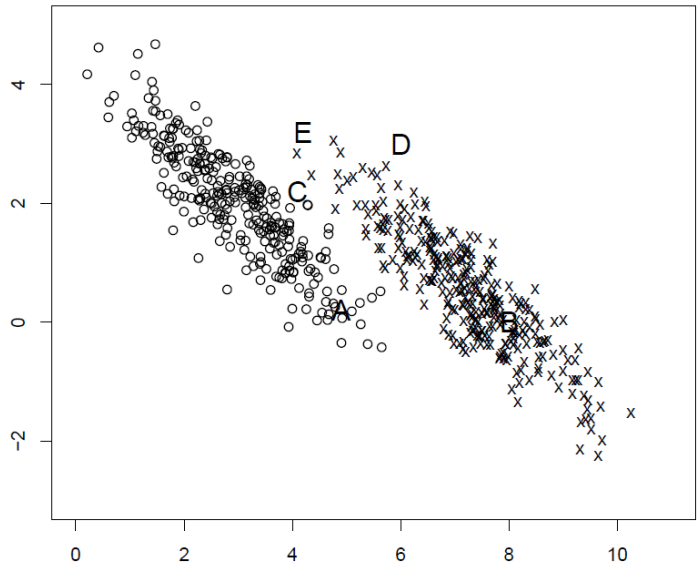
    Dataset 1:



Dataset 2:

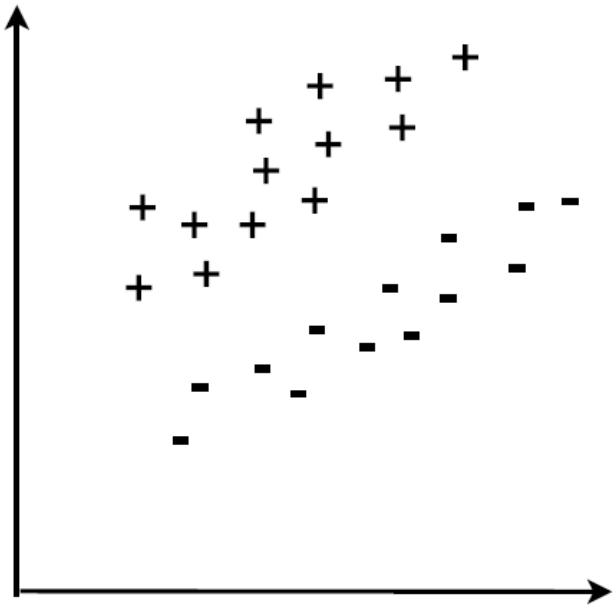


141. (Radford, 2008f, final exam, ex.4.b) The scatterplot below shows the values of two variables for 300 observations from each of the two classes (600 total). The observations are marked by class - either, O or X:



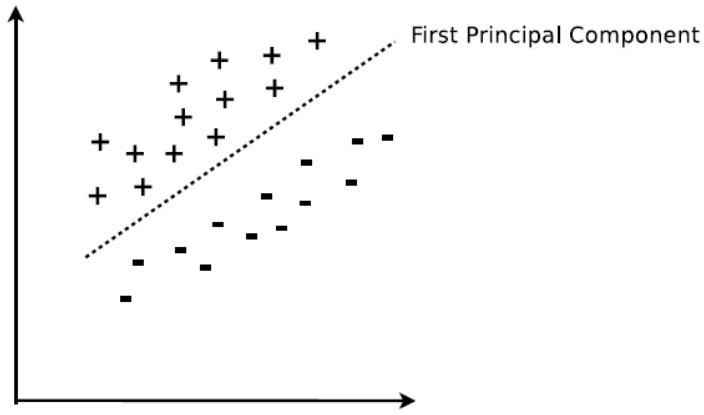
Suppose that we reduced these observations from teo variables to a single variable by projecting them on the first principal direction, as found using all 600 observations, from both classes. How well would you expect to be able to classify a new observation on the basis of its projection on this principal component? Would it work just as well as with the original variables? Or almost as well? Or much worse? Or perhaps classification would not be possible at all? Explain.

142. (CMU, 2017f, NBalcan, midterm,ex.3.4) What happens if we perform PCA as a preprocessing step and project the data onto the first principal component before performing classification on the data in the following figure? Does the data remain linearly separable?



Solution:

The following figure shows the first principal component of the data. It can be seen that the data becomes inseparable after projection onto the first principal component.



7.3.2 FDA

143. (MIT, 2004f, HW2, ex.2)

See the following paper: `https://arxiv.org/pdf/1906.09436.pdf`.

Also, a next research topic: `https://arxiv.org/pdf/1910.05437.pdf`.

We would like to classify vectors x of dimension d into one of two classes, $y = 0$ or $y = 1$. Assume that we know in advance that data from each class is sampled according to Gaussian distributions of equal covariance:

$$p(x|y=0;\mu_0,\Sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu_0)^\top \Sigma^{-1}(x-\mu_0)\right)$$

$$p(x|y=1;\mu_1,\Sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu_1)^\top \Sigma^{-1}(x-\mu_1)\right)$$

To classify each point x optimally (in the sense of minimizing the expected classification error) we must assign it to the class y that maximizes the posterior probability

$$P(y|x) = \frac{p(x|y)P(y)}{p(x)}$$

The resulting decision boundary, separating the two classes, is defined by the equation

$$\log \frac{P(y=1|x)}{P(y=0|x)} = \log \frac{p(x|y=1)P(y=1)}{p(x|y=0)P(y=0)} = 0$$

(a) Assuming μ_0, μ_1, Σ, $P(y)$ are known, show that the decision boundary is given by the following line

$$(\mu_1-\mu_0)^\top \Sigma^{-1}\left(x - \frac{\mu_1+\mu_2}{2}\right) + \log\frac{P(y=1)}{P(y=0)} = 0$$

Can a linear logistic regression model give rise to the same decision boundary? (Hint: if A is symmetric then $v^\top Au = u^\top Av$).

(b) In practice the parameters of the two Gaussians are unknown but we are given n_0 samples from class $y = 0$ and n_1 samples from class $y = 1$. Let $\hat{\mu}_0$ and $\hat{\Sigma}_0$ be the mean and covariance of the samples from class $y = 0$; similarly, we define $\hat{\mu}_1$ and $\hat{\Sigma}_1$ based on the samples from class $y = 1$.

In Fisher linear discriminant analysis we find w such that when each point is projected onto the line $t \cdot w$, $t \in \mathbb{R}$, the classes are "maximally" separable by a simple threshold. The criterion for finding w is to

maximize the separation of the projected means over the projected variances:

$$\frac{(\hat{\mu}_1^\top w - \hat{\mu}_0^\top w)^2}{n_0 w^\top \hat{\Sigma}_0 w + n_1 w^\top \hat{\Sigma}_1 w}$$

To simplify our calculation we first write the above criterion as

$$\frac{(m^\top w)^2}{w^\top S w}$$

where m is a vector and S is a positive semi-definite and symmetric matrix. What are m ans S?

(c) When S is positive definite, we can write it as $S = R^\top R$, where R is invertible (the square root of the matrix). Since R is invertible we can always search for $v = Rw$ instead of w directly. Write the criterion in terms of v and show that the maximizing solution is given by

$$\hat{v} = R^{-\top} m$$

where $R^{-\top} = (R^\top)^{-1}$. Provide the resulting expression for $\hat{w}$. (Hint: maximum of $a^\top (v/\|v\|)$ over v is obtained by any v proportional to a.)

Solution:

(a) As mentioned in the text of the problem, the points on the decision boundary satisfy:

$$\log \frac{p(x|y=1)P(y=1)}{p(x|y=0)P(y=0)} = 0$$

The $\sqrt{2\pi|\Sigma|}$ in the normal distributions cancel because of the ratios, while the exponentials are canceled by the logarithm:

$$\frac{1}{2}[(x-\mu_1)^\top \Sigma^{-1}(x-\mu_1) - (x-\mu_0)^\top \Sigma^{-1}(x-\mu_0)] + \log \frac{P(y=1)}{P(y=0)} = 0$$

$$\frac{1}{2}[2\mu_1^\top \Sigma^{-1} x - 2\mu_0^\top \Sigma^{-1} x + \mu_0^\top \Sigma^{-1}\mu_0 - \mu_1^\top \Sigma^{-1}\mu_1] + \log \frac{P(y=1)}{P(y=0)} = 0$$

$$(\mu_1 - \mu_0)^\top \Sigma^{-1} x + \frac{1}{2}[\mu_0^\top \Sigma^{-1}\mu_0 - \mu_1^\top \Sigma^{-1}\mu_1] + \log \frac{P(y=1)}{P(y=0)} = 0$$

$$(\mu_1 - \mu_0)^\top \Sigma^{-1} x - \frac{1}{2}(\mu_1 - \mu_0)^\top \Sigma^{-1}(\mu_1 + \mu_0) + \log \frac{P(y=1)}{P(y=0)} = 0$$

and the assertion follows.

The decision boundary of logistic regression is also linear, and with the appropriate sample of training points, any linear separation can be the result of training logistic regression on some data. Thus the optimal decision boundary can be the decision boundary of a linear logistic regression model.

In fact, if the two Gaussians have the same covariance and we sample a large number of points as training data, the linear logistic regression decision boundary converges to the optimal decision.

(b) The following optimization problem:

$$\hat{w} = \arg\max_{w} \frac{(\hat{\mu}_1^\top w - \hat{\mu}_0^\top w)^2}{n_0 w^\top \hat{\Sigma}_0 w + n_1 w^\top \hat{\Sigma}_1 w}$$

can be easily written as:

$$\hat{w} = \arg\max_{w} \frac{[(\hat{\mu}_1 - \hat{\mu}_0)^\top w]^2}{w^\top [n_0 \hat{\Sigma}_0 + n_1 \hat{\Sigma}_1] w}$$

Therefore

$$m = \hat{\mu}_1 - \hat{\mu}_0$$
$$S = n_0 \hat{\Sigma}_0 + n_1 \hat{\Sigma}_1$$

S is symmetric and positive semi-definite because $\hat{\Sigma}_0$ and $\hat{\Sigma}_1$ are $n_0, n_1 \geq 0$.

(c) To write the criterion in terms of v, we substitute w by $R^{-1}v$:

$$\frac{(m^\top w)^2}{w^\top S w} = \frac{(n^\top R^{-1} v)^2}{(R^{-1}v)^\top S R^{-1} v} = \frac{((R^{-\top} m)^\top v)^2}{v^\top v} = \left[(R^{-\top} m)^\top \frac{v}{\|v\|} \right]^2$$

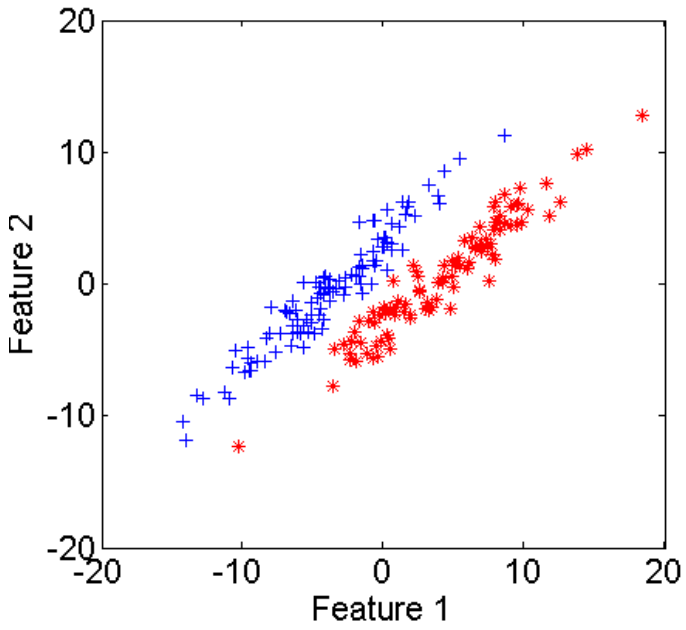
where we have used $S = R^\top R$.

The criterion takes the form of the square of a dot product between the fixed vector $\hat{v} = R^{-\top} m$ and the vector of norm 1 given by $v/\|v\|$. The only degree of freedom over which to optimize is the angle between the two vectors. But if two vectors have fixed norms, their dot product is maximized when they have the same direction (the inequality $v^\top u \leq \|v\| \cdot \|u\|$ holds). Thus $v \equiv \hat{v}$ maximizes the criterion (as well as any scalar multiple of $\hat{v}$). Moreover:

$$\hat{w} = R^{-1}\hat{v} = R^{-1}R^{-\top}m = S^{-1}m = (n_0\hat{\Sigma}_0 + n_1\hat{\Sigma}_1)^{-1}(\hat{\mu}_1 - \hat{\mu}_0)$$

7.3.3 PCA and FDA

144. (UNIVERSITY OF OSLO Faculty of Mathematics and Natural Sciences Exam: INF 4300 / INF 9305 – Digital image analysis Date: Thursday December 1, 2016, ex.4 - exams/New Folder (6): 4. interesting questions PCA (exmpl + ex)) You are given a 2D dataset with 2 classes as plotted in the figure below.



The data has the following properties:

Covariance matrix C: $\begin{bmatrix} 36 & 21 \\ 21 & 19 \end{bmatrix}$

Eigenvectors of C: $v_1 = \begin{bmatrix} -0.55 \\ 0.83 \end{bmatrix}$, $v_2 = \begin{bmatrix} 0.83 \\ 0.55 \end{bmatrix}$

Eigenvalues of C: $\lambda_1 = 5$, $\lambda_2 = 51$

(a) Explain the criterion function that principal component analysis (PCA) optimizes.

(b) Explain if PCA requires any normalization of the input data

(c) Which direction vector gives the first principal component?

(d) PCA is a linear transform $y = A^\top x$ of the input data x. What is A for the data example?

(e) How much of the variance in the data is explained by the first principal component?

(f) Which geometrical relation is there between the first and the second principal component?

(g) From the data listed above, we can construct the covariance matrix of the transform data y. What is the covariance matrix of y? No computation is needed.

(h) An alternative to PCA is Fisher's linear discriminant. Which criterion function does Fisher use?

(i) Do either PCA or Fisher have any limitations regarding how many features the transform can produce? Justify you answer.

Solution:

(a) PCA find the direction with maximum variance, which is equivalent to minimizing the signal representation error. (Either answer is correct)

(b) PCA normally use the correlation matrix, but we can use the covariance matrix if we subtract the mean.

(c) The direction given by the vector $[0.83, 0.55]^\top$

(d) $A = \begin{bmatrix} 0.83 & -0.55 \\ 0.55 & 0.83 \end{bmatrix}$

(e) Given by the eigenvalues $51/(51+5) = 91\%$

(f) They are perpendicular (and the correlation between them is zero)

(g) The variance is equal to the eigenvalues, and the covariance is zero: $\text{cov}(y) = \begin{bmatrix} 51 & 0 \\ 0 & 5 \end{bmatrix}$

(h) Fisher maximize the between-class scatter and minimize the within-class scatter, with function $J = w^\top S_w w / w^\top S_E w$

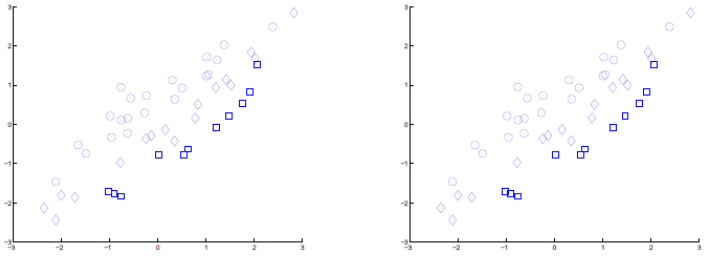
(i) No limitations with PCA, but with Fisher max dimension K-1, if K is the number of classes. This is because the rank of SB is K-1

145. (CMU, 2011s, TMitchell, HW5, ex.2.1) Principal components analysis (PCA) reduces the dimensionality of the data by finding projection direction(s) that minimizes the squared errors in reconstructing the original data or

equivalently maximizes the variance of the projected data. On the other hand, Fisher's linear discriminant is a supervised dimension reduction method, which, given labels of the data, finds the projection direction that maximizes the between-class variance relative to the within-class variance of the projected data.

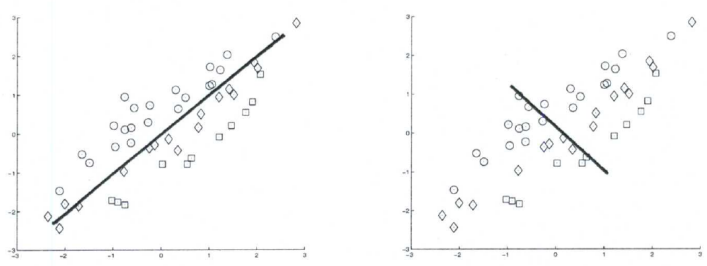
In the following figure, draw the first principal component direction in the left figure, and the

first Fisher's linear discriminant direction in the right figure. Note: for PCA, ignore the fact that points are labeled (as round, diamond or square) since PCA does not use label information. For linear discriminant, consider round points as the positive class, and both diamond and square points as the negative class (since in the course lecture we only discuss the two-class case).



Solution:

The PCA and LDA directions are shown in the following figure:



1(a) First PCA component

1(b) First LDA component

8 Factor Analysis - FA

8.1 FA

146. (after CMU, 2014f, EXing, BPoczos, HW3, pr. 1.3)

(see also their solution and `http://cs229.stanford.edu/notes/cs229-notes9.pdf` and in my dissertation to create a good solution)

Factor analysis is a generative model for linear dimensionality reduction. As before we will assume a $d(< n)$ dimensional latent space. However, this time we assume the following generative process for the data.

$$\mathbb{R}^d \ni z \sim \mathcal{N}(0, I)$$

$$\mathbb{R}^D \ni x|z \sim \mathcal{N}(Az + b, \Psi)$$

where Ψ is already known. The model says that we first sample a d dimensional Gaussian with zero mean and variance I. Then we map if to D dimensions by computing $Az + b$. Finally, we add some spherical Gaussian noise.

We will use an EM procedure to learn the parameters A, b, Ψ. So far we were only looking at EM with discrete latent variables. In this case we aill look at EM with a parametric continuous latent space.

The following results will be useful for us:

Fact 1 (Conditional of a Gaussian). Say $(Y_1, Y_2), Y_i \in \mathbb{R}^{d_i}$ is Gaussian distributed.

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12}^\top & \Sigma_{22} \end{bmatrix} \right)$$

Then, conditional on $Y_1 = y_1$ the distribution for Y_2 is

$$Y_2|Y_1 = y_1 \sim \mathcal{N}(\mu_2 + \Sigma_{12}^\top \Sigma_{11}^{-1}(y_1 - \mu_1), \Sigma_{22} - \Sigma_{12}^\top \Sigma_{11}^{-1} \Sigma_{12})$$

Fact 2 (Some Matrix Derivatives). Let $X \in \mathbb{R}^{r \times c}$ and $u \in \mathbb{R}^r$, $v, w \in \mathbb{R}^c$.

$$\nabla_X v^\top X^\top u = uv^\top$$

$$\nabla_X v^\top X^\top X w = X(vw^\top + wv^\top)$$

(a) Write down the joint distribution of (z, x). Use this to derive the marginal distribution of x and the conditional distribution $z|x$.

The conditional distribution $z|x$ will be useful for us when performing the E-step. In addition, one option for the lower dimensional representation of x_i, will be the conditional mean $E[z|x_i]$.

(b) First obtain the Maximum Likelihood Estimate for b. This does not require EM and can be done easily.

(c) Write down the E-step update at the $(t+1)^{\text{th}}$ iteration.

(d) Now write down the M-step.

Solution for their question (we modified it...):

$$z \sim \mathcal{N}(0, \Psi)$$

$$x|z \sim \mathcal{N}(Az + b, \nu^2 I)$$

(a) First note that we can write x conditioned on z as $x|z = Az + b + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \nu^2 I)$.

The joint distribution will be Gaussian since x is just a linear transformation of z. To specify the joint distribution we should know the mean of x, the variance of x and the covariance between x and z. They can be computed as follows.

$$E[x] = E[Az + b + \epsilon] = 0 + b = b$$

$$E[(z - E[z])(x - E[x])^\top] = E[z(x-b)^\top] = E[z(Az + b + \epsilon)^\top] - E[zb^\top]$$
$$= E[zz^\top A^\top + zb^\top + z\epsilon^\top] = \Psi A^\top$$
$$E[(x - E[x])(x - E[x])^\top] = E[(x-b)(x-b)^\top] = E[(Az + \epsilon)(Az + \epsilon)^\top]$$
$$= E[Azz^\top A^\top + 2\epsilon z^\top A^\top + \epsilon\epsilon^\top] = A\Psi A^\top + \nu^2 I$$

Therefore, we can write

$$\begin{bmatrix} z \\ x \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ b \end{bmatrix}, \begin{bmatrix} \Psi & \Psi A^\top \\ A\Psi^\top & A\Psi A^\top \nu^2 I \end{bmatrix} \right)$$

Using the hints given in the question, the marginal for x and the conditional $z|x$ can be written as

$$x \sim \mathcal{N}(b, A\Psi A^\top + \nu^2 I)$$

$$z|x \sim \mathcal{N}(\Psi A^\top (A\Psi A^\top + \nu^2 I)^{-1}(x - b), \Psi - \Psi A^\top (A\Psi A^\top + \nu^2 I)^{-1} A\Psi^\top)$$

We will denote the conditional mean and variance of $z|x$ by $\mu_{z|x}$ and $\Sigma_{z|x}$ respectively.

(b) The log likelihood for A, b, ν given data $\{x_i\}_{i=1}^n$ is

$$l(A, b, \nu) = \log \prod_{i=1}^{n} p(x_i) = \sum_{i=1}^{n} \log p(x_i)$$

$$= \sum_{i=1}^{n} -\frac{D}{2}\log(2\pi) - \frac{1}{2}\log\det(A\Psi A^\top + \nu^2 I) - \frac{1}{2}(x_i - b)^\top (A\Psi A^\top + \nu^2 I)^{-1}(x_i - b)$$

The MLE for b can be obtained easily. By taking the derivative of l w.r.t. b and setting it to 0 we have,

$$\nabla_b l = \sum_{i=1}^{n} (A\Psi A^\top + \nu^2 I)^{-1}(x_i - b) = 0 \Rightarrow b = \frac{1}{n}\sum_{i=1}^{n} x_i$$

since $(A\Psi A^\top + \nu^2 I)^{-1}$ is full rank.

(c) (Obtain a lower bound on the log likelihood using $R(z_i|x_i)$ - the conditional distribution for z_i given x_i.) To perform EM, we use Jensen's inequality to construct the following lower bound.

$$l(A, b, \nu) \leq \sum_{i=1}^{n} \int R(z_i|x_i) \log \frac{p(x_i, z_i; A, b, \Psi)}{R(z_i|x_i)}$$

$$= \sum_{i=1}^{n} E_{R(z_i|x_i)}[\log p(x_i|z_i; A, b, \nu)] + C$$

$$= \sum_{i=1}^{n} E_{R(z_i|x_i)}\left[\log\left(\frac{1}{(2\pi)^{D/2}\nu^D}\exp\left(-\frac{(x_i - b - Az_i)^\top (x_i - b - Az_i)}{2\nu^2}\right)\right)\right] + C$$

$$= \sum_{i=1}^{n} E_{R(z_i|x_i)}\left[\log\left(-\frac{D}{2}\log(2\pi) - D\log(\nu) - \frac{1}{2\nu^2}(x_i - b - Az_i)^\top (x_i - b - Az_i)\right)\right] + C$$

$$= -nD\log(\nu) + \frac{-1}{2\nu^2}\sum_{i=1}^{n} E_{R(z_i|x_i)}[\|x_i - b\|^2 - 2z_i^\top A^\top (x_i - b) + z_i A^\top A z_i] + C'$$

Here C, C' are constants that do not depend on A, b, ν. Let us call this lower bound l_b.

(d) In the M-step we will maximze the lower bound above w.r.t. the parameters A, ν simply by taking the derivative and setting it to zero. First take the derivative w.r.t. using the hints given in the question,

$$\nabla_A l_b(A, b, \nu) = \frac{-1}{2\nu^2}\sum_{i=1}^{n} E_{R(z_i|x_i)}[2A(z_i z_i^\top) - 2(x_i - b)z_i^\top] \Rightarrow$$

$$A(\sum_i E_{R(z_i|x_)} z_i z_i^\top) = \sum_i (x_i - b) E_{R(z_i|x_i)} z_i^\top \Rightarrow A = (\sum_i (x_i - b)\mu_{z_i|x_i}^\top)$$
$$(\sum_i \mu_{z_i|x_i}\mu_{z_i|x_i}^\top + \Sigma_{z_i|x_i})^{-1}$$

Similarly for ν,

$$\frac{\partial l_b}{\partial \nu} = -\frac{nD}{\nu} + \frac{1}{\nu^3}\sum_{i=1}^n E_{R(z_i|x_i)}[\|x_i - b\|^2 - 2z_i^\top A^\top (x_i - b) + z_i A^\top A z_i] \Rightarrow$$

$$\nu^2 = \frac{1}{nD}\sum_{i=1}^n (\|x_i - b\|^2 - 2\mu_{z_i|x_i}^\top A^\top (x_i - b) + \|A\mu_{z_i|x_i}\|^2 + \text{diag}(A\Sigma_{z_i|x_i}A^\top)^\top 1)$$

For the last step we used the fact that $E_{R(z_i|x_i)} z_i = \mu_{z_i|x_i}$, and from the properties of the Gaussian, $E_{R(z_i|x_i)}\|Az_i\|^2 = \|A\mu_{z_i|x_i}\|^2 + \text{diag}(A\Sigma_{z_i|x_i}A^\top)^\top 1$. We can perform the M-step via the above update equations for A and ν.

Observations:

(a) There are many ways to fit an FA model. One possibility is using the EM algorithm.

(b) For
$$z \sim \mathcal{N}(0, I)$$
$$\epsilon \in \mathcal{N}(0, \Psi)$$
$$x = \mu + \Lambda z + \epsilon$$
there are two important notions:
- Λ - the matrix of **factor loadings**
- z - the score(s) of standardized factors, or **standardized score(s)**

(c) The **loadings can be defined** in at least 2 equivalent ways:
- from $x = \mu + \Lambda z + \epsilon$ - coefficients in linear combination predicting a variable by the standardized components
- from $\begin{bmatrix} z \\ x \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ \mu \end{bmatrix}, \begin{bmatrix} I & \Lambda^\top \\ \Lambda & \Lambda\Lambda^\top + \Psi \end{bmatrix}\right)$ cross-covariance matrix between original variables and standardized factors

(d) (from `https://youtu.be/QGd06MTRMHs?t=2102`)

	Model $p(X)$	Not probabilistic
Data lies in "subspace"	FA	PCA
Data lies in "groups"	GMM	k-means

147. (Bishop book, ex.12.19) Show that the factor analysis model is **invariant under rotations of the latent space rotations**.

Solution:

To see this we define a rotated latent space vector $\tilde{z} = Rz$ where R is an $M \times M$ orthogonal matrix, and similarly defining a modified factor loading matrix $\tilde{W} = WR$. Then we note that the latent space distribution $p(z)$ depends only on $z^\top z = \tilde{z}^\top \tilde{z}$, where we have used $R^\top R = I$. Similarly, the conditional distribution of the observed variable $p(x|z)$ depends only on $Wz = \tilde{W}\tilde{z}$. Thus the joint distribution takes the same form for any choice of R. This is reflected in the predictive distribution $p(x)$ which depends on W only through the quantity $WW^\top = \tilde{W}\tilde{W}^\top$ and hence is also invariant to different choices of R.

148. (Bishop book, ex.12.25) Consider a linear-Gaussian latent-variable model having a latent space distribution $p(z) = \mathcal{N}(x|0, I)$ and a conditional distribution for the observed variable $p(x|z) = \mathcal{N}(x|Wz + \mu, \Phi)$ where Φ is an arbitrary symmetric, positive-definite noise covariance matrix. Now suppose that we make a nonsingular linear transformation of the data variables $x \to Ax$, where A is a $D \times D$ matrix. If μ_{ML}, W_{ML} and Ψ_{ML} represent the maximum likelihood solution corresponding to the original untransformed data, show that $A\mu_{\text{ML}}$, AW_{ML}, and $A\Phi_{\text{ML}}A^\top$ will represent the corresponding maximum likelihood solution for the transformed data set.

Finally, show that the form of the model is preserved in two cases: (i) A is a diagonal matrix and Φ is a diagonal matrix. This corresponds to the case of factor analysis. The transformed Φ remains diagonal, and hence **factor analysis is covariant under component-wise re-scaling of the data variables**; (ii) A is orthogonal and Φ is proportional to the unit matrix so that $\Phi = \sigma^2 I$. This corresponds to probabilistic PCA. The transformed Φ matrix remains proportional to the unit matrix, and hence **probabilistic PCA is covariant under a rotation of the axes of data space**, as is the case for conventional PCA.

Solution: Following the discussion of section 12.2, the log likelihood function for this model can be written as

$$L(\mu, W, \Phi) = -\frac{ND}{2}\ln(2\pi) - \frac{N}{2}\ln[WW^\top + \Phi]$$

$$-\frac{1}{2}\sum_{n=1}^{N}\{(x_n - \mu)^\top (WW^\top + \Phi)^{-1}(x_n - \mu)\}$$

where we have used (12.43).

If we consider the log likelihood function for the transformed data set we obtain

$$L_A(\mu, W, \Phi) = -\frac{ND}{2}\ln(2\pi) - \frac{N}{2}\ln[WW^\top + \Phi]$$

$$-\frac{1}{2}\sum_{n=1}^{N}\{(Ax_n - \mu)^\top (WW^\top + \Phi)^{-1}(Ax_n - \mu)\}$$

Solving for the maximum likelihood estimator for μ in the usual way we obtain

$$\mu_A = \frac{1}{N}\sum_{n=1}^{N} Ax_n = A\overline{x} = A\mu_{\mathrm{ML}}$$

Back-substituting into the log likelihood function, and using the definition of the sample covariance matrix (12.3), we obtain

$$L_A(\mu, W, \Phi) = -\frac{ND}{2}\ln(2\pi) - \frac{N}{2}\ln[WW^\top + \Phi]$$

$$-\frac{1}{2}\sum_{n=1}^{N}\mathrm{Tr}((WW^\top + \Phi)^{-1}ASA^\top)$$

We can cast the final term into the same form as the corresponding term in the original log likelihood function if we first define

$$\Phi_A = A\Phi^{-1}A^\top, W_A = AW$$

With these definitions the log likelihood function for the transformed data set takes the form

$$L_A(\mu_A, W_A, \Psi_A) = -\frac{ND}{2}\ln(2\pi) - \frac{N}{2}\ln[W_A W_A^\top + \Phi_A]$$

$$-\frac{1}{2}\sum_{n=1}^{N}((x_n - \mu_A)^\top (W_A W_A^\top + \Phi_A)^{-1}(x_n - \mu_A)) - N\ln|A|$$

This takes the same form as the original log likelihood function apart from an additive constant $-\ln|A|$. Thus the maximum likelihood solution in the new variables for the transformed data set will be identical to that in the old variables. We now ask whether specific constraints on Φ will be preserved by this re-scaling. In the case of probabilistic PCA the noise covariance

Φ is proportional to the unit matrix and takes the form $\sigma^2 I$. For this constraint to be preserved we require $AA^\top = I$ so that A is an orthogonal matrix. This corresponds to a rotation of the coordinate system. For factor analysis Φ is a diagonal matrix, and this property will be preserved if A is also diagonal since the product of diagonal matrices is again diagonal. This corresponds to an independent re-scaling of the coordinate system. Note that in general probabilistic PCA is not invariant under component-wise re-scaling and factor analysis is not invariant under rotation. These results are illustrated in the following figure.

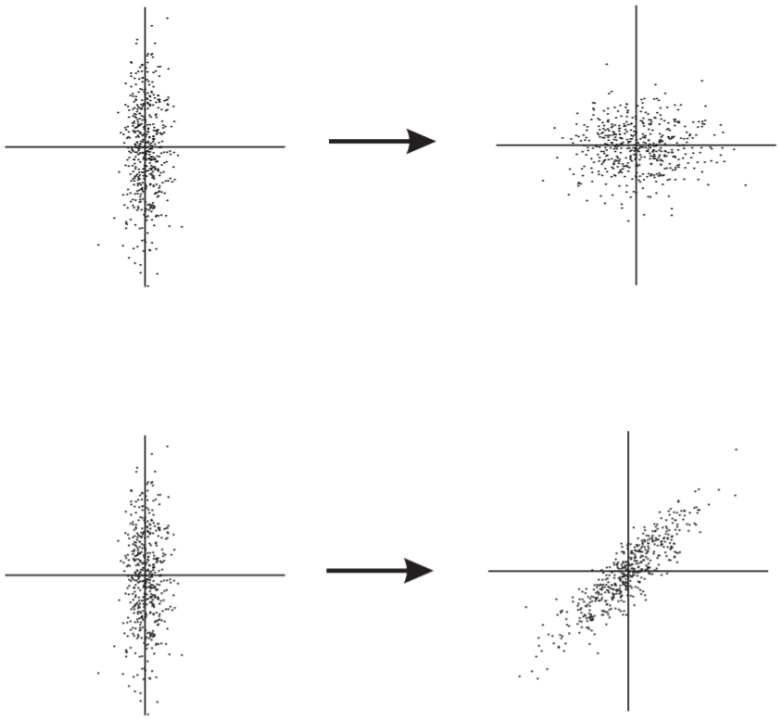


Figure 7 Factor analysis is covariant under a componentwise re-scaling of the data variables (top plots), while PCA and probabilistic PCA are covariant under rotations of the data space coordinates (lower plots).

149. (from Unsupervised ML, Helsinki, Aapo Hyvarinen, Ex. set 4, ex.5) The optimization problem for **quartimax** is to maximize

$$J(U) = \sum_{ij} G((AU)_{ij})$$

under the constraint of orthogonality of U.

(a) Derive an iterative update rule for the matrix U which solves the optimization problem for $G(y) = y^4$.

(b) Show that $J(U)$ is constant for all orthogonal U if $G(y) = y^2$.

150. (Radford, 2012s, third test, ex.2) Recall that in a factor analysis model an observed data point, x, is modeled using M latent factors as

$$x = \mu + Wz + \epsilon$$

where μ is a vector of means for the p components of x, W is a $p \times M$ matrix, z is a vector of M latent factors, assumed to have independent $\mathcal{N}(0,1)$ distributions, and ϵ is a vector of p residuals, assumed to be independent, and to come from normal distributions with mean zero. The variance of ϵ_j is σ_j^2.

Suppose that $p = 5$ and $M = 2$, and that the parameters of the mode are mean $\mu = [0, 0, 0, 0, 0]^\top$, residual standard deviations $\sigma_1 = 1$, $\sigma_2 = 1$, $\sigma_3 = 2$, $\sigma_4 = 2$, $\sigma_5 = 2$ and

$$W = \begin{bmatrix} 1 & 2 \\ -1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

(a) Find the covariance matrix for x. Show your work.

(b) Suppose that we don't observe vectors x of dimension five, but rather we observe vectors y of dimension four, where $y_1 = x_1$, $y_2 = 3x_2$, $y_3 = -x_3$, and $y_4 = 2x_4 + x_5$. Assuming that the distribution of x is given by the factor analysis model with parameters above, write down a factor analysis model (including values of its parameters) for the distribution of y.

Solution:

(a)

$$\text{Cov}(x) = E[(Wz + \epsilon)(Wz + \epsilon)^\top] = WW^\top + \text{diag}(\sigma_1^2, \ldots, \sigma_5^2)$$

$$= \begin{bmatrix} 6 & 1 & 1 & 1 & 2 \\ 1 & 3 & -1 & -1 & 1 \\ 1 & -1 & 5 & 1 & 0 \\ 1 & -1 & 1 & 5 & 0 \\ 2 & 1 & 0 & 0 & 5 \end{bmatrix}$$

(b) Using the relation of y to x and the model for x above, we can write $y = W' + z + \epsilon'$, where

$$W' = \begin{bmatrix} 1 & 2 \\ -3 & 3 \\ -1 & 0 \\ 2 & 1 \end{bmatrix}$$

The standard deviations of the ϵ'_i will be $\sigma'_1 = \sigma_1 = 1$, $\sigma'_2 = 3\sigma_2 = 3$, $\sigma'_3 = \sigma_3 = 2$, and $\sigma'_4 = \sqrt{4\sigma_4^2 + \sigma_5^2} = \sqrt{20}$.

151. (Radford, 2008f, final exam, ex.2) The distribution of a vector X of length $p = 4$ is given by a factor analysis model with $k = 1$ common factors, in which the true values of the parameters are mean $\mu = 0$, covariance of specific factors $\Psi = 4I$ and the loadings matrix $L = [3, 2, 1, 0]^\top$.

(a) Give another value for the loadings matrix, L, that will produce the same distribution for X. No explanation is required.

(b) Find the covariance matrix of X.

(c) Find the conditional distribution of the common factor given that we observe that the first component of X is -2. Assume that the remaining three components of X are not observed, and that we know the true values of the model parameters.

152. IS IT OK TO INCLUDE THIS? (Exercises of Multivariate Statistical Methods for Engineering and Management Master in Industrial Engineering and Management 2nd Semester 2010/2011, Factor Analysis section ex.1 - exams/New Folder (8)) Consider the one-factor model associated with the standardized random vector $X = (X_1, \ldots, X_5)^\top$, represented by the following equations:

$$X_1 = 0.65f + \epsilon_1$$
$$X_2 = 0.84f + \epsilon_2$$
$$X_3 = 0.70f + \epsilon_3$$
$$X_4 = 0.32f + \epsilon_4$$
$$X_5 = 0.28f + \epsilon_5$$

Admit the usual assumptions: $E(f) = 0$, $E(e) = 0$, $Var(f) = 1$, $Var(e) = \text{diag}(\Psi_1, \ldots, \Phi_5)$, $Cov(f, e) = 0$.

(a) Compute the communalities of the manifest variables with the common factor.

(b) What are the unique variances associated with each manifest variable?

(c) Compute the correlation between the following sets of manifest variables:

i. X_1 and X_2
ii. X_2 and X_3
iii. X_2 and X_5
iv. X_4 and X_5

(d) What is the shared variance between each m

8.2 Probabilistic Principal Component Analysis - PPCA

153. (Stanford, 2007f, ANg, HW4, pr.2) (the solution must be modified using our method of deriving an EM algo)

In this problem we look at the relationship between two unsupervised learning algorithms we discussed in class: Factor Analysis and Principal Component Analysis.

Consider the following joint distribution over (x, z) where $z \in \mathbb{R}^k$ is a latent random variable

$$z \sim \mathcal{N}(0, I)$$
$$x|z \sim \mathcal{N}(Uz, \sigma^2 I)$$

where $U \in \mathbb{R}^{n \times k}$ is a model parameters and σ^2 is assumed to be a known constant. This model is often called Probabilistic PCA. Note that this is nearly identical to the factor analysis model except we assume that the variance of $x|z$ is a knwon scaled identity matrix rather than the diagonal parameter matrix, Ψ, and we do not add an additional μ term to the mean (though this last difference is just for simplicity of presentation). However, as we will see, it turns out that $\sigma^2 \to 0$, this model is equivalent to PCA.

For simplicity, you can assume for the remainder of the problem that $k = 1$, i.e., that U is a column vector in $\mathbb{R}^n$.

(a) Use the rules for manipulating Gaussian distributions to determine the joint distribution over (x, z) and the conditional distribution of $z|x$.

[Hint: for later parts of this problem, it will help significantly if you simplify your solution for the conditional distribution using the identity we first mentioned in problem set n.1: $(\lambda I+BA)^{-1}B = B(\lambda I+AB)^{-1}$.]

(b) Using these distribution, derive an EM algorithm for the model. Clearly state the E-step and the M-step of the algorithm.

(c) As $\sigma^2 \to 0$ show that if the EM algorithm converges to a parameter vector U^* (and such convergence is guaranteed by the argument presented in class), then U^* must be an eigenvector of the sample covariance matrix $\Sigma = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}{x^{(i)}}^\top$ - i.e., U^* must satisfy

$$\lambda U^* = \Sigma U^*.$$

[Hint: When $\sigma^2 \to 0$, $\Sigma_{z|x} \to 0$, so the E step only needs to compute the means $\mu_{z|x}$ and not the variances. Let $w \in \mathbb{R}^m$ be a vector containing all these means, $w_i = \mu_{z^{(i)}|x^{(i)}}$, and show that the E-step and M-step can be expressed as

$$w = \frac{XU}{U^\top U}, U = \frac{X^\top w}{w^\top w}$$

respectively. Finally, show that if U does not change after this update, it must satisfy the eigenvector equation shown above.]

Observation: (after `https://stats.stackexchange.com/questions/208731/what-is-principal-subspace-in-probabilistic-pca` and Bishop - Pattern Recog p.574-576) Probabilistic PCA (PPCA) is the following latent variable model

$$z \sim \mathcal{N}(0, I)$$
$$x \sim \mathcal{N}(Wz + \mu, \sigma^2 I),$$

where $x \in \mathbb{R}^p$ is one observation and $z \in \mathbb{R}^q$ is a latent variable vector; usually $q << p$. Note that this differs from factor analysis in only one little detail: error covariance structure in PPCA is $\sigma^2 I$ and in FA it is an arbitrary diagonal matrix Ψ.

Tipping and Bishop, 1999, Probabilistic Principal Component Analysis (`http://research.microsoft.com/pubs/67218/bishop-ppca-jrss.pdf`) prove the following theorem: the maximum likelihood solution for PPCA can be obtained analytically (i.e., there is a closed-form solution for PPCA; for FA, there is no closed-form solution!) and is given by:

$$\sigma^2_{\text{ML}} = \frac{1}{p-q}\sum_{i=q+1}^{p} \lambda_i$$

$$W_{\text{ML}} = U_q(\Lambda_q - \sigma^2_{\text{ML}} I)^{1/2} R$$

where U_q is a matrix of q leading principal directions (eigenvectors of the covariance matrix), Λ_q is the diagonal matrix of corresponding eigenvalues λ_i, σ^2_{ML} is also given by an explicit formula, and R is an arbitrary $q \times q$ rotation matrix (corresponding to rotations in the latent space).

Also, $E[z|x] = (W^\top W + \sigma^2 I)^{-1} W^\top (x - \bar{x})$. By plugging in W_{MLE}: $E[z|x] = (W^\top_{\text{MLE}} W_{\text{MLE}} + \sigma^2 I)^{-1} W^\top_{\text{MLE}} (x - \bar{x})$

There are two similarities to PCA:

- the columns of W_{MLE} span the principal subspace of the data space
- when $\sigma^2 \to 0$, then the posterior mean ($E[z|x]$) reduces to

  $$(W^\top_{\text{MLE}} W_{\text{MLE}})^{-1} W^\top_{\text{MLE}} (x - \bar{x})$$

  which represents an orthogonal projection of the data point onto the latent space, and so we recover the standard PCA model.

Solution:

(a) To compute the joint distribution, we compute the means and covariances of x and z. First, $E[z] = 0$ and

$$E[x] = E[Uz + \epsilon] = UE[z] + E[\epsilon] = 0, \text{ where } \epsilon \sim \mathcal{N}(0, \sigma^2 I)$$

Since both x and z have zero mean

$$\Sigma_{zz} = E[zz^\top] = I (= 1, \text{ since z is a scalar when k = 1})$$

$$\Sigma_{zx} = E[(Uz + \epsilon)z^\top] = UE[zz^\top] + E[\epsilon z^\top] = U$$

$$\Sigma_{xx} = E[(Uz + \epsilon)(Uz + \epsilon)^\top] = E[Uzz^\top U^\top + \epsilon z^\top U^\top + Uz\epsilon^\top + \epsilon\epsilon^\top]$$
$$= UE[zz^\top]U^\top + E[\epsilon\epsilon^\top] = UU^\top + \sigma^2 I$$

Therefore,

$$\begin{bmatrix} z \\ x \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & U^\top \\ U & UU^\top + \sigma^2 I \end{bmatrix} \right)$$

Using the rules for conditional Gaussian distributions, $z|x$ is also Gaussian with mean and covariance

$$\mu_{z|x} = U^\top (UU^\top + \sigma^2 I)^{-1} x = \frac{U^\top x}{U^\top U + \sigma^2}$$

$$\Sigma_{z|x} = 1 - U^\top (UU^\top + \sigma^2 I)^{-1} U = 1 - \frac{U^\top U}{U^\top U + \sigma^2}$$

where in both cases the last equality comes from the identity in the hint.

(b) Even though $z^{(i)}$ is a scalar value, in this problem we continue to use the notation ${z^{(i)}}^\top$, etc. to make the similarities to the Factor Analysis case obvious.

For the E-step, we compute the distribution $Q_i(z^{(i)}) = p(z^{(i)}|x^{(i)}; U)$ by computing $\mu_{z^{(i)}|x^{(i)}}$ and $\Sigma_{z^{(i)}|x^{(i)}}$ using the above formulas.

For the M-step, we need to maximize

$$\sum_{i=1}^{m} \int_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}|z^{(i)}; U) p(z^{(i)})}{Q_i(z^{(i)})}$$

$$= \sum_{i=1}^{m} E_{z^{(i)} \sim Q_i} [\log p(x^{(i)}|z^{(i)}; U) + \log p(z^{(i)}) - \log Q_i(z^{(i)})]$$

Taking the gradient with respect to U equal to zero, dropping terms that don't depend on U, and omitting the subscript on the expectation, this becomes

$$\nabla_U \sum_{i=1}^{m} E[\log p(x^{(i)}|z^{(i)}; U)] = \nabla_U \sum_{i=1}^{m} E[-\frac{1}{2\sigma^2}(x^{(i)} - Uz^{(i)})^\top (x^{(i)} - Uz^{(i)})]$$

$$= -\frac{1}{2\sigma^2} \sum_{i=1}^{m} \nabla_U E[\mathrm{tr} {z^{(i)}}^\top U^\top U z^{(i)} - 2\mathrm{tr} {z^{(i)}}^\top U^\top x^{(i)}]$$

$$= -\frac{1}{2\sigma^2} \sum_{i=1}^{m} E[U z^{(i)} {z^{(i)}}^\top - x^{(i)} {z^{(i)}}^\top]$$

$$= \frac{1}{2\sigma^2} \sum_{i=1}^{m} [-U E[z^{(i)} {z^{(i)}}^\top] + x^{(i)} E[{z^{(i)}}^\top]]$$

using the same reasoning as in the Factor Analysis class notes. Setting this derivative to zero gives

$$U = (\sum_{i=1}^{m} x^{(i)} E[{z^{(i)}}^\top])(\sum_{i=1}^{m} E[z^{(i)} {z^{(i)}}^\top])^{-1}$$

$$= (\sum_{i=1}^{m} x^{(i)} \mu_{z^{(i)}|x^{(i)}}^\top)(\sum_{i=1}^{m} \Sigma_{z^{(i)}|x^{(i)}} + \mu_{z^{(i)}|x^{(i)}} \mu_{z^{(i)}|x^{(i)}}^\top)^{-1}$$

All these terms were calculated in the E-step, so this is our final M step update.

(c) For the E step, when $\sigma^2 \to 0$, $\mu_{z^{(i)}|x^{(i)}} = \frac{U^\top x^{(i)}}{U^\top U}$, so using w as defined in the hint we have

$$w = \frac{XU}{U^\top U}$$

as desired.

As mentioned in the hint, when $\sigma^2 \to 0$, $\Sigma_{z^{(i)}|x^{(i)}} = 0$, so

$$U = \left(\sum_{i=1}^{m} x^{(i)} \mu_{z^{(i)}|x^{(i)}}^\top\right)\left(\sum_{i=1}^{m} \Sigma_{z^{(i)}|x^{(i)}} + \mu_{z^{(i)}|x^{(i)}} \mu_{z^{(i)}|x^{(i)}}^\top\right)^{-1}$$

$$= \left(\sum_{i=1}^{m} x^{(i)} w_i\right)\left(\sum_{i=1}^{m} w_i w_i\right)^{-1} = \frac{X^\top w}{w^\top w}$$

For U to remain unchanged after an update requires that

$$U = \frac{X^\top \frac{XU}{U^\top U}}{\frac{U^\top X^\top}{U^\top U}\frac{XU}{U^\top U}} = X^\top XU \frac{U^\top U}{U^\top X^\top XU} = X^\top XU \frac{1}{\lambda}$$

proving the desired equation.

8.3 PCA: an FA point of view

154. (after LA4ML - (Eigenvectors and Intro to PCA - Worksheet. Part Three, ex.2) and after `https://stats.stackexchange.com/questions/612/is-pca-followed-by-a`
PCA loadings

Suppose your data contained the variables *VO2_max*, *mile pace*, and *weight* in that order. What variable is the most important/influential to the first principal component, in each of the following cases:

(a) The first principal component for this data is the eigenvector (with its norm equal to 1) of the covariance matrix (i.e., the coefficients/weights in the linear combination):

$$\begin{bmatrix} 0.26 \\ 0.53 \\ 0.80 \end{bmatrix}$$

Observation: If X is the mean-centered data matrix with rows as variables and columns as observations. If its SVD is $X = USV^\top$, then the *weights* mentioned above are in U.

(b) The **covariance** vector between the initial attributes and the first principal component is:

$$\begin{bmatrix} 4 \cdot 0.26 \\ 4 \cdot 0.53 \\ 4 \cdot 0.80 \end{bmatrix}$$

Observation: If X is the mean-centered data matrix with rows as variables and columns as observations. If its SVD is $X = USV^\top$, then the *covariances* mentioned above are in US^2. Prove it!

(c) The **loadings** vector corresponding to the first component (i.e., the **covariance vector between the initial attributes and the standardized first principal component**) is:

$$\begin{bmatrix} \frac{2 \cdot 0.26}{\sqrt{10-1}} \\ \frac{2 \cdot 0.53}{\sqrt{10-1}} \\ \frac{2 \cdot 0.80}{\sqrt{10-1}} \end{bmatrix}$$

Observation: If X is the mean-centered data matrix with rows as variables and columns as observations. If its SVD is $X = USV^\top$, then the *loadings* mentioned above are in $U\frac{S}{\sqrt{n-1}}$(n is the number of observations). Prove it!

(d) The **rescaled/standardized loadings** vector corresponding to the first principal component (i.e., the **covariance vector between the standardized initial attributes and the standardized first component**; or: the **correlation between the initial attributes and the first component**) is:

$$\begin{bmatrix} \frac{2 \cdot 0.26}{10\sqrt{10-1}} \\ \frac{2 \cdot 0.53}{25\sqrt{10-1}} \\ \frac{2 \cdot 0.80}{40\sqrt{10-1}} \end{bmatrix}$$

Observation: If X is the mean-centered data matrix with rows as variables and columns as observations. If its SVD is $X = USV^\top$, then the *standardized loadings* mentioned above are in $DU\frac{S}{\sqrt{n-1}}$(n is the number of observations, D - diagonal matrix to make X standardized on each attribute - mean 0, variance 1). Prove it!

Observation: Intuitively, any of the 4 criteria can be used to obtain the most important/influential variable to the first principal component. The answers to the first 3 will be the same. The answer to the fourth one can differ. In practice, the first and the

last ones are used (maybe because they both have the values in [-1,1]), but the last criterion (via rescaled/standardized loadings) is preferred due to its statistical interpretation (i.e., the correlation between the initial variables and the principal components).

Observation: the terminology of *loadings* (and *standardized scores*) is **borrowed from FA**.

Solution: Choose the the attribute with the highest absolute value.

155. Given the following factorization via SVD:

(a) Draw the **biplot corresponding to classic PCA (loading + standardized scores)**. Calibrate the biplot. Which are the **advantages** of this biplot vs other general biplots?

Hint: for the advantages see: `https://www.researchgate.net/profile/Ehab_Mahdy/post/Can_you_please_help_me_interpret_these_biplots/attachment/59d621b779197b8077980161/AS%3A297914397151239%401448039735565/download/Biplot.pdf` and `https://stats.stackexchange.com/questions/141085/positioning-the-arrows-on-a-pca-biplot`)

(b) Draw the **biplot corresponding to classic PCA (loading + standardized scores)**, but using **correlation PCA**. Calibrate the biplot. What other **advantages** come from using this biplot?

Observation: There is an infinity of biplots corresponding to PCA (because you can introduce a rotation/scaling/other type of matrix between the matrices in the SVD), but the one highlighted here (loadings + standardized scores) is the classic one (see `https://www.jstor.org/stable/2334381?seq=1/subjects`)! Some of them have advantages over the others or want to emphasize a specific detail.

DO NOT FORGET TO INSERT DATA, i.e., THE MATRIX FACTORIZATION.

156. (LA4ML - (Application of PCA - Worksheet. Part One. ex. 1,2,3)) **(Rotating PCs (with the idea borrowed from Factor Analysis))**

(a) What is the point of rotating principal components? What do you gain by doing so?

(b) After rotation, does the first principal component/factor still explain as much variance as it did initially?

(c) Since the principal components are originally decided by determining the subspace with maximum variance, wouldn't a rotation of these components explain less variance? Can you explain this?

Solution:

(a) Rotating principal components makes our factors (new variables) more interpretable by enforcing sparsity in the factor matrix (i.e., making each factor have only a few variables with relatively high loadings).

(b) No, after rotation the first factor will not explain the same amount of variance, some of that variance will be shifted to other components.

(c) No, once the dimension of the subspace is decided on, the variance of the data in that lower dimensional space stays constant regardless of how I rotate the axes. While each component may not explain the same amount of variance as it did before rotation, the total variance of all the components together will remain the same.

157. (after `https://stats.stackexchange.com/questions/123063/is-there-any-good-reason`
PCA as $\Sigma \approx WW^\top$

PCA can be interpreted as a (best) low-rank matrix approximation not only for the mean-centered data matrix $X \in \mathbb{R}^{d\times n}$ (n - number of observations and d - number of dimensions), but also for the sample covariance matrix $\Sigma = \frac{1}{n-1}XX^\top$. Prove this by showing that PCA loadings are a solution for the following optimization problem:

$$\min_{W\in\mathbb{R}^{d\times k}} \|\Sigma - WW^\top\|^2_{\text{Fro}}$$

Observations:

(a)

$$\begin{aligned} \text{PCA}: &\ \boldsymbol{\Sigma} \approx \mathbf{W}\mathbf{W}^\top \\ \text{PPCA}: &\ \boldsymbol{\Sigma} \approx \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I} \\ \text{FA}: &\ \boldsymbol{\Sigma} \approx \mathbf{W}\mathbf{W}^\top + \boldsymbol{\Psi} \end{aligned}$$

(b)

$$\text{minimizing} \left\{ \begin{array}{l} \|\boldsymbol{\Sigma} - \mathbf{W}\mathbf{W}^\top\|^2_{\text{Fro}} \\ \|\boldsymbol{\Sigma} - \mathbf{W}\mathbf{W}^\top - \sigma^2\mathbf{I}\|^2_{\text{Fro}} \\ \|\boldsymbol{\Sigma} - \mathbf{W}\mathbf{W}^\top - \boldsymbol{\Psi}\|^2_{\text{Fro}} \end{array} \right\} \text{leads to} \left\{ \begin{array}{c} \text{PCA} \\ \text{PPCA} \\ \text{FA} \end{array} \right\} \text{loadings},$$

(c) Because $\Sigma_{\text{PCA}} \approx WW^\top$, we can say that PCA tries to approximate?/explain/preserve the covariances and the variances between/of the components. Because $\Sigma_{\text{FA}} \approx WW^\top + \Psi$, we can say that FA tries to approximate?/explain/preserve the covariances between the components (the variances can have any value because of Ψ which gives sufficient degrees of freedom; ?most of the time, they will be set to the actual existing variances).

8.4 Revision

158. (LA4ML - from course - Jeopardy Round) Circle the correct answer and justify your choice:

(a) The **standardized loadings** on a principal component tell you

i. The variance of each variable on that principal compoennt
ii. How correlated each variable is with that principal component
iii. Absolutely nothing
iv. How much each observation weighs along that principal component

Solution: B

(b) What is the purpose or motivation behind the **rotations of factors in Factor Analysis**?

i. The original factors were not orthogonal, so we need to adjust them
ii. The first factor does not explain enough variance. By rotating, we can explain more variance
iii. The loadings of the variables are difficult to interpret, by rotating we get new factors which more clearly represent combinations of original variables
iv. The rotation helps spread the observations out so that we can more clearly see different groups or classes in the data

Solution: C

(c) When we perform a rotation of principal components (with the idea borrowed from Factor Analysis), we may not explain as much total variance as we did before rotation, but we will always have more variance between clusters of data points.

i. True

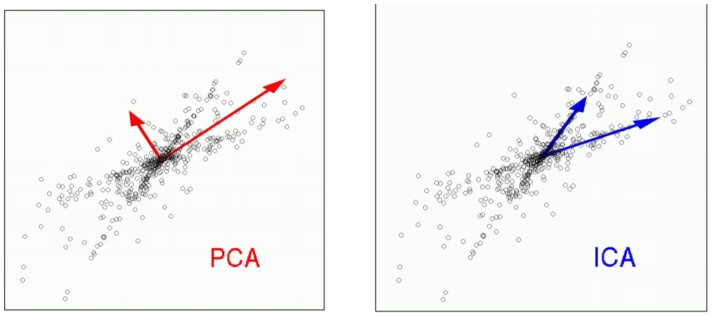
ii. False

Solution: B

9 Independent Component Analysis - ICA

159. (after `http://cs229.stanford.edu/notes/cs229-notes11.pdf` - see the solution also here)

Our next topic is Independent Components Analysis (ICA). Similar to PCA, this will find a new basis in which to represent our data. However, the goal is very different.



(image taken from `https://www.cs.cmu.edu/~bapoczos/Classes/ML10715_2015Fall/slides/ICA.pdf` slide 10)

As a motivating example, consider the "cocktail party problem." Here, n speakers are speaking simultaneously at a party, and any microphone placed in the room records only an overlapping combination of the n speakers' voices. But let's say we have n different microphones placed in the room, and because each microphone is a different distance from each of the speakers, it records a different combination of the speakers' voices. Using these microphone recordings, can we separate out the original n speakers' speech signals?

To formalize this problem, we imagine that there is some data $s \in \mathbb{R}^n$ that is generated via n independent sources. What we observe is

$$x = As$$

where A is an unknown square matrix called the mixing matrix. Repeated observations gives us a dataset $\{x^{(i)}; i = 1, \ldots, m\}$ and our goal is to recover the sources $s^{(i)}$ that had generated our data ($x^{(i)} = As^{(i)}$).

In our cocktail party problem, $s^{(i)}$ is an n-dimensional vector, and $s_j^{(i)}$ is the sound that speaker j was uttering at time i. Also, $x^{(i)}$ is an n-dimensional vector, and $x_j^{(i)}$ is the acoustic reading recorded by the microphone j at time i.

Let $W = A^{(-1)}$ be the **unmixing matrix**. Our goal is to find W, so that given our microphone recordings $x^{(i)}$, we can recover the sources by computing $s^{(i)} = Wx^{(i)}$. For notational convenience, we also let $w_i^\top$ denote the i-th row of W, so that

$$W = \begin{bmatrix} -w_1^\top - \\ \vdots \\ -w_n^\top - \end{bmatrix}$$

Thus, $w_i \in \mathbb{R}^n$, and the j-th source can be recovered by computing $s_j^{(i)} = w_j^\top x^{(i)}$.

(a) Discuss the ICA ambiguities.

(b) After the discussion on ICA ambiguities, you will be able to briefly describe the ICA model as follows:

$$s_i \sim \text{Non-Gaussian}(\dots) \text{ - any distribution, but Gaussian}$$

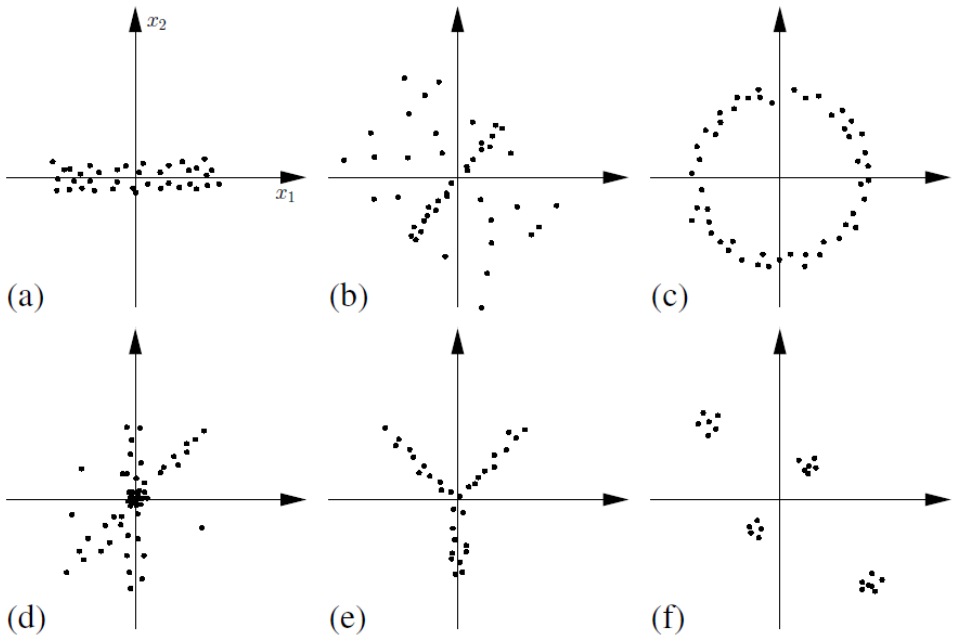
$$\forall i \neq j : s_i, s_j \text{ - independent}$$

$$x = As$$

Derive an ICA algorithm based on MLE and the gradient ascent method.

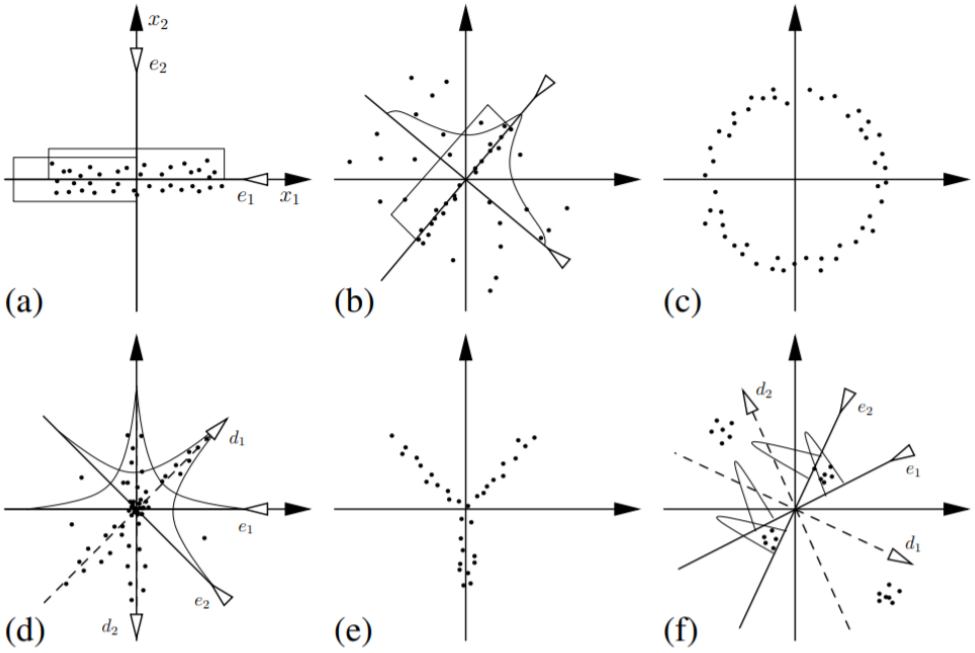
Observation: There are many ways to fit an ICA model. One possibility is using MLE and the gradient ascent method.

160. (Exercises* on Independent Component Analysis Laurenz Wiskott Institut fur Neuroinformatik Ruhr-Universitat Bochum, Germany, EU 4 February 2017, ex. 1.1.1) Decide whether the following distributions can be linearly separated into independent components. If yes, sketch the (not necessarily orthogonal) axes onto which the data must be projected to extract the independent components. Draw on these axes also the marginal distributions of the corresponding components.



Solution:

Solution:



Vectors e_1 and e_2 used for extracting the independent components are drawn with inverted arrow heads. Vectors d_1 and d_2 used for mixing the sources are drawn with dashed lines and proper arrow heads, but only if different from e_1 and e_2. All vectors are drawn with same length, which is not correct in some cases, because the variances of the extracted components might be wrong. If unmixing is not possible, no additional vectors are drawn. The distributions of the extracted sources are drawn on the e_1- and e_2-axes. Notice that they should be normalized to 1, i.e. they should have the same area.

Extra question:Elaborate on (f) (do not simplify this to a parallelogram). How exactly do the distribution and extraction vectors come about and relate to each other? How do you write the mixing and unmixing process in matrix notation?

161. Is it OK to include it? (Fordham University, ML CISC 5800, HW3, ex.A.3)

Review:

We can reconstruct an estimate of data point x using components u and their corresponding weights z

$$\tilde{x} = \sum_q z_q u^q$$

where $\tilde{x}$ is the estimate of x. However, the reconstruction will be inaccurate. A standard measure of the inaccuracy between an original data vector x and the estimated version of this vector $\tilde{x}$ is called "mean squared error":

$$E(x, \tilde{x}) = \sum_j (x_j - \tilde{x}_j)^2$$

If our original data point is $\begin{bmatrix} 2 \\ 0 \\ -1 \\ 2 \\ 0 \end{bmatrix}$ and our estimate is $\begin{bmatrix} 1.8 \\ 0.5 \\ -0.8 \\ 2.3 \\ -0.3 \end{bmatrix}$, the error will be

$(2-1.8)^2 + (0-0.5)^2 + (-1+0.8)^2 + (2-2.3)^2 + (0+0.3)^2 = 0.2^2 + 0.5^2 + 0.2^2 + 0.3^2 + 0.3^2 = 0.04 + 0.25 + 0.04 + 0.09 + 0.09;\ E = 0.51$

Question:

We have the following components

$$u^1 = \begin{bmatrix} 0.4 \\ 0.7 \\ 0.4 \\ 0.4 \end{bmatrix}, u^2 = \begin{bmatrix} 0.6 \\ 0.8 \\ 0 \\ 0 \end{bmatrix}, u^3 = \begin{bmatrix} 0.4 \\ 0 \\ 0 \\ 0.9 \end{bmatrix}$$

For each data point, we have three corresponding reconstruction weights:

$$x^1 = \begin{bmatrix} 1.3 \\ 0.5 \\ 0.4 \\ 2.5 \end{bmatrix}, z_1 = 1, z_2 = 0, z_3 = 2$$

$$x^2 = \begin{bmatrix} 2.5 \\ 2.4 \\ 0.1 \\ 1.2 \end{bmatrix}, z_1 = 0, z_2 = 3, z_3 = 1.5$$

(a) What are the estimated vectors for x^1 and x^2, based on the corresponding z values above?

(b) What is the mean squared error between the estimated and actual data vectors for x^1 and x^2?

(c) Do we expect the components u and weights z in this question are derived from ICA, PCA, or NMF? Explain your answer (in 1-2 sentences).

Solution:

(a)
$$\tilde{x}_1 = \begin{bmatrix} 1.2 \\ 0.7 \\ 0.4 \\ 2.2 \end{bmatrix}, \tilde{x}^2 = \begin{bmatrix} 2.4 \\ 2.4 \\ 0 \\ 1.35 \end{bmatrix}$$

(b) Error x^1: $0.1^2 + 0.2^2 + 0^2 + 0.3^2 = 0.14$
Error x^2: $0.1^2 + 0^2 + 0.1^2 + 0.15^2 = 0.0425$

(c) NMF and ICA - non-orthogonal. Particularly consistent with NMF, since all weights and components are non-negative.

162. Is it OK to include it? (Fordham University, ML CISC 5800, HW3, ex.A.4) Presume the following are independent components:

$$u^1 = \begin{bmatrix} 0 \\ 0.67 \\ 0.67 \\ 0.33 \end{bmatrix}, u^2 = \begin{bmatrix} 0.9 \\ -0.4 \\ 0 \\ 0.2 \end{bmatrix}, u^3 = \begin{bmatrix} 0.3 \\ 0.9 \\ -0.3 \\ 0 \end{bmatrix}$$

(a) Which independent component u best describes each of the data points x below? (In other words, which single component can mostly closely reconstruct each data point below?)

(b) What is the corresponding weight $z^i_{q_j}$ for this strongest component?

$$x^1 = \begin{bmatrix} -1.5 \\ -4.7 \\ 1.2 \\ 0.2 \end{bmatrix}, x^2 = \begin{bmatrix} -0.3 \\ 3.2 \\ 2.9 \\ 0.9 \end{bmatrix}, x^3 = \begin{bmatrix} 4.5 \\ 3.1 \\ -1.6 \\ -3.1 \end{bmatrix}$$

Solution:

(a) u^q which has greatest dot product with x^i

$$x^1 : u^3$$
$$x^2 : u^1$$
$$x^3 : u^3$$

(b) Take dot product with the associated u^q

$$z^1_3 = -0.45 - 4.23 - 0.36 = -5.04$$
$$z^2_3 = 2.14 + 1.94 + 0.30 = 4.38$$
$$z^3_3 = 1.3 + 2.7 + 0.5 = 4.9$$

163. (CMU, 2014s, BPoczos, ASingh, midterm, ex.1.20)

Answer True or False. Justify your answer very briefly in 1-2 sentences.

The goal os independent component analysis is to transform the datapoints with a linear transformation in such a way that they become linearly independent.

Solution: ICA transforms the datapoints with a linear transformation in such a way that they become *statistically* and not linearly independent.

10 Canonical Correlation Analysis - CCA

10.1 CCA

164. (CS 189 spring 2018 hw6, ex.2)

The goal of canonical correlation analysis (CCA) is to find linear combinations that maximize the correlation of two random vectors. We are given two zero-mean random vectors X, Y in $\mathbb{R}^d$. Any linear combination of the coordinates of the random vector X can be written as $a^\top X$, and similarly, a linear combination of the coordinates of the random vector Y can be written as $b^\top Y$. Note that $a^\top X$, $b^\top Y$ in $\mathbb{R}$ are scalar random variables.

The goal of CCA can be summarized as solving the following optimization problem

$$\rho = \max_{a,b \in \mathbb{R}^d} \rho(a^\top X, b^\top Y)$$

where the correlation coefficient $\rho(P, Q)$ between two zero-mean scalar random variables P and Q is defined by $\rho(P, Q) = \frac{E[PQ]}{\sqrt{E[P^2]E[Q^2]}}$.

The zero-mean jointly-Gaussian case expresses why we care about correlation. Two jointly Gausian zero-mean random variables that are uncorrelated are also independent of each other. Furthermore, if we want to best estimate (in a mean-squared sense) a particular scalar zero Gaussian random variable Y based on a vector X of jointly Gaussian zero-men random variables, then we are going to pick a weight vector w that is aligned with $E[XY]$. It is straightforward algebra to see that this direction maximizes the correlation coefficient $\rho(\frac{w^\top}{\|w\|}X, Y)$.

In this problem, we will work our way towards finding a solution to the problem of canonical correlation analysis using the singular value decomposition. In parts (a) to (c), we derive useful results regarding the singular value decomposition. In the later parts, you will use this result and see that canonical correlation analysis is equivalent to the singular value decomposition in a rotated and stretched coordinate system.

(a) Let $n \geq d$. For a matrix $A \in \mathbb{R}^{n \times d}$ with full column-rank and singular value decomposition $A = U\Sigma V^\top$, we know that the singular values are given by the diagonal entries of Σ, and the left singular vectors are the columns of the matrix U and the right singular vectors are the

columns of the matrix V. Note that both the matrices U and V have orthonormal columns.

Show that $A = \sum_{i=1}^{d} \sigma_i u_i v_i^\top$, where the ith singular value is denoted by $\sigma_i = \Sigma_{ii}$ and u_i and v_i are the ith left and right singular vectors, respectively.

(b) With the setup above, show that

i. $A^\top A$ has i-th eigenvalue σ_i^2, with associated eigenvector v_i

ii. $AA^\top$ has i-th eigenvalue Σ_i^2, with associated eigenvector u_i.

Notice that both of the above matrices are symmetric.

(c) Use the first part to show that

$$\sigma_1(A) = \max_{u:\|u\|_2=1; v:\|v\|_2=1} u^\top A v$$

where $\sigma_1(A)$ is the maximum singular value of A.

Additionally, show that if A has unique a unique maximum singular value, then the maximizers (u^*, v^*) above are given by the first left and right singular vectors, respectively.

Hint 1: You may or may not find the following fact useful: We can express any $u \in \mathbb{R}^n : \|u\|_2 = 1$ as a linear combination of left singular vectors $\{u_i\}$ of the matrix A, and any vector $v \in \mathbb{R}^d$ as a linear combination of the right singular vectors $\{v_i\}$ of the matrix A.

Hint 2: You may find the following results useful: For any two vectors $a, b \in \mathbb{R}^d$, we have

- Cauchy-Schwarz inequality: $|a^\top b| \leq \|a\|_2 \|b\|_2$, with equality only when b is a scaled version of a.
- Holder's inequality: $|a^\top b| \leq \|a\|_1 \|b\|_\infty$, Here, the l_1 and l_∞ norms of a vector v are defined by $\|v\|_1 = \Sigma_i |v_i|$ and $\|v\|_\infty = \max_i |v_i|$. Let us say that the vector b is fixed; then one way to achieve equality in the Holder inequality is to have: Let i be such that $|b_i| = \|b\|_\infty$. Set $a_i = \|a\|_1$, and $a_j = 0$ for all $j \neq i$.

Using the hint, we may write a unit norm vector $u = \sum_{i=1}^{n} a_i u_i$, where u_i are the n left singular vectors of the matrix A. Notice that since u is a unit norm vector, we must have $\|u\|_2^2 = \sum_{i=1}^{n} a_i^2 = 1$. Similarly, $v = \sum_{i=1}^{d} b_i v_i$, where v_i are the d right singular vectors of the matrix A and $\|v\|_2^2 = \sum_{i=1}^{d} b_i^2 = 1$.

Alternatively, notice that for any $u : \|\|_2 = 1$, we have $\|U^\top u\|_2 = \|u\|_2 = 1$ by unitary invariance of the l_2 norm (since $U^\top \in \mathbb{R}^{n \times n}$

has orthonormal columns). Similarly, we have $\|V^\top v\|_2 = \|v\|_2 = 1$. Since $U^\top$ and $V^\top$ are full rank matrices, maximizing over u and v is equivalent to maximizing over a linear combination of them, and so we have

$$\max_{u:\|u\|_2=1;v:\|v\|_2=1} v^\top U\Sigma V^\top v = \max_{a:\|a\|_2=1;b:\|b\|_2=1} a^\top \Sigma b.$$

Now notice that $\Sigma \in \mathbb{R}^{n\times d}$ has its last $n-d$ rows equal to zero. Therefore, expanding out the last expression similarly to the first part, we have reduced the problem to solving

$$\max_{a:\|a\|_2=1;b:\|b\|_2=1} \sum_{i=1}^{d} \sigma_i a_i b_i.$$

Now, we see that there is no value in setting any entry of a or b to be negative, since σ_i is always positive. We still therefore assume that $a_i \geq 0$ and $b_i \geq 0$ for all i. Also, denote $a_i b_i = c_i$, and collect the c_i values into a d-dimensional vector c. Similarly, collect the σ_i values into a vector σ. We therefore have that for any unit norm vectors u, v, the following inequality holds:

$$u^\top Av = c^\top \sigma \leq \|c\|_2\|\sigma\|_\infty = \|c\|_1\sigma_1(A),$$

where we have used Holder's inequality. Now, note that $\|c\|_1 = a^\top b$ since all the entries were assumed to be possitive, and using Cauchy Schwarz inequality, we know that $a^\top b \leq \|a\|_2\|b\|_2 = 1$. We have thus shown that $u^\top Av \leq \sigma_1(A)$ for all unit norm vectors u and v.

In order to show that the maximum is attained, choose $u = u_1$, and $v = v_1$, and note that this corresponds to choosing a and b to be the first basis vector in $\mathbb{R}^d$, which we denote by e_1. Thus, we have

$$u_1 Av_1 = s_1\Sigma e_1 = \sigma_1(A).$$

We have thus shown that the maximum $\sigma_1(A)$ is indeed attained by the claimed maximizers $u^* = u_1$ and $v^* = v_1$.

(d) Let us now look at the canonical correlation analysis problem, where we are given the covariance of two zero-mean random vectors $X, Y \in \mathbb{R}^d$ as follows:

$$E[XX^\top] = \Sigma_{XX}, E[YY^\top] = \Sigma_{YY}, \text{ and } E[XY^\top] = \Sigma_{XY}.$$

The goal is to find two directions/unit-vectors a and b so that the resulting scalar random variables $a^\top X$ and $b^\top Y$ have maximal correlation coefficient $\rho(a^\top X, b^\top Y)$. Show that the canonical correlation analysis problem can be rewritten as

$$\rho = \max_{a,b \in \mathbb{R}^d} \frac{a^\top \Sigma_{XY} b}{(a^\top \Sigma_{XX} a)^{1/2} (b^\top \Sigma_{YY} b)^{1/2}}$$

Conclude that if (a^*, b^*) is a maximizer above, then $(\alpha a^*, \beta b^*)$ is a maximizer for any $\alpha, \beta > 0$. We see that scaling the vectors a and b does not affect their correlation coefficient.

(e) Let us simplify our optimization problem. Assume that the covariance matrices Σ_{XX} and Σ_{YY} are full-rank. We choose matrices $\Sigma_{XX}^{-1/2}$, $\Sigma_{YY}^{-1/2}$ to whiten the vectors X and Y respectively. Note that for the vector $\tilde{X} = \Sigma_{XX}^{-1/2} X$, we have

$$E[\tilde{X}\tilde{X}^\top] = E[(X\Sigma_{XX}^{-1/2})^\top (X\Sigma_{XX}^{-1/2})] = 1$$

One may do a similar computation for the whitened vector $\tilde{Y} = \Sigma_{YY}^{-1/2} Y$, and conclude that its covariance matrix is identity too! Such a whitening step simplifies our computations, both at algebraic and conceptual level.

Rewrite the optimization problem from the previous part, using the aforementioned whitening in the following form:

$$\rho = \max_{c:\|c\|_2=1; d:\|d\|_2=1} c^\top \Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1/2} d$$

(f) Recall that the vectors (a^*, b^*) denote maximizers in the previous part of the problem. Using the simplification and the above parts show that

ρ^2 i the maximum eigenvalue of the matrix

$$\Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{XY}^\top \Sigma_{XX}^{-1/2}$$

Hint: An appropriate change of variables may make your life easier.

(g) Following the previous part's setup, show that $c^* = \Sigma_{XX}^{-1/2} a^*$ is an eigenvector corresponding to the maximum eigenvalue of the matrix

$$\Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{XY}^\top \Sigma_{XX}^{-1/2}$$

(h) Following the previous part's setup, show that $d^* = \Sigma_{YY}^{1/2} b^*$ is an eigenvector corresponding to the maximum eigenvalue of the matrix

$$\Sigma_{YY}^{-1/2} \Sigma_{XY}^\top \Sigma_{XX}^{-1} \Sigma_{XY} \Sigma_{YY}^{-1/2}$$

(i) Argue why such a CCA is not meaningful when the random vectors X and Y are uncorrelated, where by this we mean that $\text{cov}(X_i, Y_j) = 0$ for all i, j.

(j) Suppose you happen to know that X and Y^2 (where a squared-vector Y^2 is defined by squaring each entry of the vector Y) share a linear relationship, how could you still use CCA effectively with the given data?

(k) Why do you think that understanding CCA is relevant for machine learning?

Solution:

(a) There are mny ways to compute the answer to this problem; we illustrate one of them. We assume that we have the full SVD $A = U\Sigma V^\top$, where $U \in \mathbb{R}^{n \times n}$, $\Sigma \in \mathbb{R}^{n \times d}$, and $V \in \mathbb{R}^{d \times d}$.

Begin by computing the matrix $U\Sigma$, and note that $U \in \mathbb{R}^{n \times n}$ and $\Sigma \in \mathbb{R}^{n \times d}$. Think of this matrix multiplication as d vector multiplications; we therefore have

$$(U\Sigma)_i = U \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \sigma_i \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \sigma_i u_i,$$

where as before, we have used A_i to denote the ith column of a matrix A. Stacking these up, we have

$$U\Sigma = \begin{bmatrix} \sigma_1 u_1 \dots \sigma_d u_d \end{bmatrix}$$

Now notice that a matrix product can be computed using outer products. In other words, we have $AB^\top = \sum_i A_i B_i^\top$ (write out a simple 2×2 example to see exactly why). Therefore, we have

$$U\Sigma V^\top = \sum_{i=1}^{d} \sigma_i u_i v_i^\top.$$

(b) i. From the above part, we have

$$A^\top A = (\sum_{i=1}^{d} \sigma_i u_i v_i^\top)^\top (\sum_{j=1}^{d} \sigma_j u_j v_j^\top) = (\sum_{i=1}^{d} \sigma_i v_i u_i^\top)(\sum_{j=1}^{d} \sigma_j u_j v_j^\top).$$

Now notice that $u_i^\top u_j = 0$ unless $i = j$, in which case $u_i^\top u_i = 1$. Therefore, expanding the above multiplication, we see that only the terms where $i = j$ remain, and we have

$$A^\top A = \sum_{i=1}^{d} \sigma_i^2 v_i v_i^\top.$$

Consequently, we have

$$A^\top A v_j = \sum_{i=1}^{d} \sigma_i^2 v_i v_i v_j^\top = \sigma_i^2 v_j,$$

where we have used the fact that $v_i^\top v_j = 0$ unless $i = j$,in which case $v_j^\top v_j = 1$.
In words, we have shown that v_j is an eigenvector of $A^\top A$ with associated eigenvalue σ_j^2. This holds for all $j = \{1, 2, \ldots, d\}$.

ii. The second part proceeds exactly as above, where we show that

$$AA^\top = (\sum_{i=1}^{d} \sigma_i u_i v_i^\top)(\sum_{j=1}^{d} \sigma_j u_j v_j^\top)^\top = (\sum_{i=1}^{d} \sigma_i u_i v_i^\top)(\sum_{j=1}^{d} \sigma_j v_j u_j^\top).$$

Now notice that $v_i^\top v_j = 0$ unless $i = j$, in which case $v_i^\top v_i = 1$. Therefore, expanding the above multiplication, we see that only the terms where $i = j$ remain, and we have

$$AA^\top = \sum_{i=1}^{d} \sigma_i^2 u_i u_i^\top.$$

Consequently, we have

$$AA^\top u_j = \sum_{i=1}^{d} \sigma_i^2 u_i^\top u_j = \sigma_i^2 u_j,$$

where we have used the fact that $u_i^\top u_j = 0$ unless $i = j$, in which case $u_j^\top u_j = 1$.

In words, we have shown that u_j is an eigenvector of $AA^\top$ with associated eigenvalue σ_j^2. This holds for all $j = \{1, 2, \ldots, d\}$. Alternatively, we could have used the matrix definition of the SVD and the spectral theorem, and this answer will also get full credit.

(c) Using the fact that $b^\top Y$ is a scalar, we have $b^\top Y = Y^\top b$. We can now apply linearity of expectation to see that

$$E[(a^\top X)(b^\top Y)] = E[a^\top X Y^\top b] = a^\top E[XY^\top]b = a^\top \Sigma_{XY} b.$$

By a similar argument, we have

$$E[(a^\top X)(a^\top X)] = a^\top \Sigma_{XX} a, \text{ and}$$

$$E[(b^\top Y)(b^\top Y)] = b^\top \Sigma_{YY} b.$$

Substituting these quantities into the definition of the correlation coefficient, we have

$$\rho = \max_{a,b \in \mathbb{R}^d} \frac{a^\top \Sigma_{XY} b}{(a^\top \Sigma_{XX} a)^{1/2} (b^\top \Sigma_{YY} b)^{1/2}}.$$

In order to see the last conclusion, simply scale a and b by α and β respectively, and notice that

$$\frac{(\alpha a)^\top \Sigma_{XY} (\beta b)}{((\alpha a)^\top \Sigma_{XX} (\alpha a))^{1/2} ((\beta b)^\top \Sigma_{YY} (\beta b))^{1/2}} = \frac{a^\top \Sigma_{XY} b}{(a^\top \Sigma_{XX} a)^{1/2} (b^\top \Sigma_{YY} b)^{1/2}}$$

via cancellation in the numerator and denominator. Thus the maximization problem does not change either.

(d) Using the results from the previous subpoint we can write the optimization problem as follows:

$$\rho = \max_{a,b, \in \mathbb{R}^d} \frac{a^\top \Sigma_{XY} b}{(a^\top \Sigma_{XX} a)^{1/2} (b^\top \Sigma_{YY} b)^{1/2}}$$

$$= \max_{a,b \in \mathbb{R}^d} \frac{(\Sigma_{XX}^{1/2} a)^\top \Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1/2} (\Sigma_{YY}^{1/2} b)}{((\Sigma_{XX}^{1/2} a)^\top \Sigma_{XX}^{-1/2} \Sigma_{XX} \Sigma_{XX}^{-1/2} (\Sigma_{XX}^{1/2} a))^{1/2} ((\Sigma_{YY}^{1/2} b)^\top \Sigma_{YY}^{-1/2} \Sigma_{YY} \Sigma_{YY}^{-1/2} (\Sigma_{YY}^{-1/2} b))^{1/2}}$$

$$= \max_{a,b \in \mathbb{R}^d} \frac{(\Sigma_{XX}^{1/2} a)^\top \Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1/2} (\Sigma_{YY}^{1/2} b)}{((\Sigma_{XX}^{1/2} a)^\top (\Sigma_{XX}^{1/2} a))^{1/2} ((\Sigma_{YY}^{1/2} b)^\top (\Sigma_{YY}^{1/2} b))^{1/2}}$$

$$= \max_{c,d \in \mathbb{R}^d} \frac{c^\top \Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1/2} d}{(c^\top c)^{1/2} (d^\top d)^{1/2}}$$

From the previous part, we know that scaling c and d does not change the objective. Consequently, we may assume that c and d have unit norm, and so

$$\rho = \max_{c:\|c\|_2=1;d:\|d\|_2=1} c^\top \Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1/2} d.$$

(e) Let use denote $A = \Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1/2}$. From the third subpoint, we knoe that the value of ρ is the maximum singular value of the matrix A. We can see that $AA^\top = \Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{XY}^\top \Sigma_{XX}^{-1/2}$ and using the second subpoint it follows that ρ^2 is the maximum eigenvalue of the matrix $AA^\top$.

(f) From the third part $c^* = \Sigma_{XX}^{1/2} a^*$ is the maximal left vector of A. We now apply the second part, which state that c^* is the maximal eigenvector of the matrix

$$AA^\top = \Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1/2} (\Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1/2})^\top$$

(g) Similarly, from the third subpoint $d^* = \Sigma_{YY}^{1/2} b^*$ is the maximal right vector of A and from the second subpoint d^* is the maximal eigenvector of the matrix

$$A^\top A = (\Sigma_X X^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1/2})^\top \Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1/2}.$$

(h) Since $\Sigma_{XY} = 0$, CCA returns $\rho = 0$ and $a = b = 0$, and is therefore useless.

(i) If we know that there is a non-linear relationship between the variables that is quadratic, we may perform CCA on the variables X and $Z = Y^2$, between which we expect a linear relationship to exist. This is similar to the trick we played in moving from linear regression to polynomial regression.

(j) Canonical Correlation Analysis (CCA) is a fundamental statistical technique for characterizing the linear relationships between two multidimensional variables. As such it allows learning features that may exist in multiple sources. For example one can be given two sources recording, one audio and one video, both sources might be noisy in different ways, but both encode the same speech pattern.

This is a linear modeling approach that then inspires us for nonlinear neural net architectures that have bottlenecks.

165. (CS 189 Introduction to Machine Learning Fall 2019 Jennifer Listgarten, Stella Yu HW 04, ex.4) Assume that you have a database of images of the words typed in two different fonts. $X, Y \in \mathbb{R}^{n \times d}$ corresponds to the dataset of font 1 and font 2 respectively. Think of the database X as being composed on n independent draws (samples) from a random variable $X \in \mathbb{R}^d$, and similarly Y as n draws from a random variable $Y \in \mathbb{R}^d$. your goal is to use machine learning to build a text recognition of word images.

(a) Explain why you should want to consider using CCA in this problem.

(b) Assume that the data matrices X and Y include zero-mean features of the word images. Given two unit-length vectors $u, v \in \mathbb{R}^d$, compute the correlation coefficient of the random variables x, y projected onto u, v, i.e., compute the correlation coefficient between $u^\top x$ and $v^\top y$. Correlation coefficient between two scalar random variables P and Q s computed by:

$$\rho(P, Q) = \frac{cov(P, Q)}{\rho_P \rho_Q}$$

(c) Assume that the features of matrix X are rescaled to have values between -1 and 1. How does this change the correlation coefficient?

166. (FINM 331: DATA ANALYSIS FOR FINANCE AND STATISTICS FALL 2015 PROBLEM SET 3, ex.2) OR (Multivariate Analysis Homework 3 A49109720 Yi-Chen Zhang April 13, 2018, ex. 10.2 - with answers) IS IT OK?

The 2×1 random vectors X and Y have joint mean vector and joint covariance matrix

$$\mu = \begin{bmatrix} \mu_X \\ \mu_Y \end{bmatrix} \in \mathbb{R}^{4\times 4}, \Sigma = \begin{bmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_Y \end{bmatrix} \in \mathbb{R}^{4\times 4},$$

where

$$\mu_X = \begin{bmatrix} -3 \\ 2 \end{bmatrix}, \mu_Y = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

and

$$\Sigma_X = \begin{bmatrix} 8 & 2 \\ 2 & 5 \end{bmatrix}, \Sigma_Y = \begin{bmatrix} 6 & -2 \\ -2 & 7 \end{bmatrix}, \Sigma_{YX}^\top = \Sigma_{XY} = \begin{bmatrix} 3 & 1 \\ -1 & 3 \end{bmatrix}.$$

(a) Calculate the canonical correlation ρ_1 (the largest), ρ_2 (the second largest).

(b) Find the canonical correlation variables (U_1, V_1) and (U_2, V_2) corresponding to ρ_1 and ρ_2.

(c) Let $U = [U_1, U_2]^\top$ and $V = [V_1, V_2]^\top$. Evaluate
$E\left(\begin{bmatrix} U \\ V \end{bmatrix}\right)$ and $\text{Cov}\left(\begin{bmatrix} U \\ V \end{bmatrix}\right) = \begin{bmatrix} \Sigma_U & \Sigma_{UV} \\ \Sigma_{VU} & \Sigma_V \end{bmatrix}$

(d) Comment on the correlation structure between and within U and V.

Solution:

(a) The inverse ans square root of the inverse of Σ_{11} and Σ_{22} are calculated by R compiled in the Appendix. We have

$$\Sigma_{11}^{-1} = \begin{bmatrix} 0.1389 & -0.0556 \\ -0.0556 & 0.2222 \end{bmatrix}, \Sigma_{11}^{-1/2} = \begin{bmatrix} 0.3667 & -0.0667 \\ -0.0667 & 0.4667 \end{bmatrix}$$

$$\Sigma_{22}^{-1} = \begin{bmatrix} 0.1842 & 0.0526 \\ 0.0526 & 0.1579 \end{bmatrix}, \Sigma_{22}^{-1/2} = \begin{bmatrix} 0.4243 & 0.0645 \\ 0.0645 & 0.3921 \end{bmatrix}$$

Since $\rho^2 = (\rho_2^2, \rho_2^2)$ are the eigenvalues of the matrix $\Sigma_{11}^{-1/2}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\Sigma_{11}^{-1/2}$ with corresponding (2×1) eigenvectors h_1, h_2. (The quantities ρ^2 are alto the eigenvalues of the matrix $\Sigma_{22}^{-1/2}\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1/2}$ with corresponding (2×1) eigenvectors f_1, f_2.)

$$\Sigma_{11}^{-1/2}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\Sigma_{11}^{-1/2} = \begin{bmatrix} 0.2756 & -0.0322 \\ -0.0322 & 0.2690 \end{bmatrix}$$

and

$$\Sigma_{22}^{-1/2}\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1/2} = \begin{bmatrix} 0.2946 & -0.0234 \\ -0.0234 & 0.2500 \end{bmatrix}$$

The eigenvalues are $(\rho_1^2, \rho_2^2) = (0.3046, 0.2400)$ with the corresponding eigenvectors $H = (h_1, h_2)$ and $Q = (f_1, f_2)$, respectively. Here

$$h_1 = \begin{bmatrix} -0.7422 \\ 0.6702 \end{bmatrix}, h_2 = \begin{bmatrix} -0.6702 \\ -0.7422 \end{bmatrix}, f_1 = \begin{bmatrix} -0.9184 \\ 0.3936 \end{bmatrix}, f_2 = \begin{bmatrix} -0.3936 \\ -0.9193 \end{bmatrix}$$

So the canonical correlations $(\rho_1, \rho_2) = (0.5519, 0.4899)$.

(b) The canonical variate pairs:

$$U_1 = h_1^\top \Sigma_{11}^{-1/2} X^{(1)} = -0.3168 X_1^{(1)} + 0.3622 X_2^{(1)}$$

$$V_1 = f_1^\top \Sigma_{22}^{-1/2} X^{(2)} = -0.3647 X_1^{(2)} + 0.0951 X_2^{(2)}$$

$$U_2 = h_2^\top \Sigma_{11}^{-1/2} X^{(1)} = -0.1962 X_1^{(1)} - 0.3017 X_2^{(1)}$$

$$V_2 = f_1^\top \Sigma_{22}^{-1/2} X^{(2)} = -0.2263 X_1^{(2)} - 0.3858 X_2^{(2)}$$

(c) Since $U = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = H^\top \Sigma_{11}^{-1/2} X^{(1)}$ and $V = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = Q^\top \Sigma_{22}^{-1/2} X^{(2)}$

$$E \begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} H^\top \Sigma_{11}^{-1/2} \mu^{(1)} \\ Q^\top \Sigma_{22}^{-1/2} \mu^{(2)} \end{bmatrix} = \begin{bmatrix} 1.6749 \\ -0.0146 \\ 0.0951 \\ -0.3858 \end{bmatrix}$$

$$\text{Cov} \begin{bmatrix} U \\ V \end{bmatrix} = \text{Cov} \begin{bmatrix} H^\top \Sigma_{11}^{-1/2} X^{(1)} \\ Q^\top \Sigma_{22}^{-1/2} X^{(2)} \end{bmatrix}$$

$$= \begin{bmatrix} H^\top \Sigma_{11}^{-1/2} \Sigma_{11} \Sigma_{11}^{-1/2} H & H^\top \Sigma_{11}^{-1/2} \Sigma_{12} \Sigma_{22}^{-1/2} Q \\ Q^\top \Sigma_{22}^{-1/2} \Sigma_{21} \Sigma_{11}^{-1/2} H & Q^\top \Sigma_{22}^{-1/2} \Sigma_{22} \Sigma_{22}^{-1/2} Q \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0.5519 & 0 \\ 0 & 1 & 0 & 0.4899 \\ 0.5519 & 0 & 1 & 0 \\ 0 & 0.4899 & 0 & 1 \end{bmatrix}$$

The above result shows that $\text{Corr}(U_k, V_k) = \rho_k$ and

$$\text{Var}(U_k) = \text{Var}(V_k) = 1$$

$$\text{Cov}(U_k, U_l) = \text{Corr}(U_k, U_l) = 0, k \neq l$$
$$\text{Cov}(V_k, V_l) = \text{Corr}(V_k, V_l) = 0, k \neq l$$
$$\text{Cov}(U_k, V_l) = \text{Corr}(U_k, V_l) = 0, k \neq l$$

for $k, l = 1, 2$.

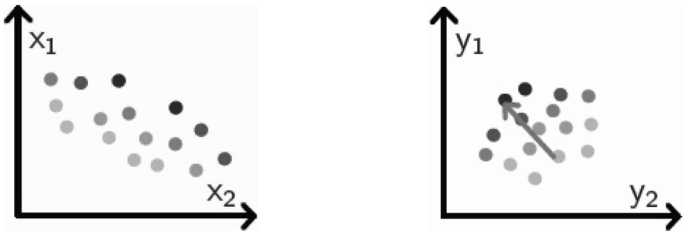
167. (Stanford, Sta306b midterm test, 2015s, ex.5 - exams/New folder) An investigator has mammograms (breast X-rays) for 50 women with breast cancer. He extracts 10 clinically important features from each of these images. He also has gene expression data for 20, 000 genes from a biopsy of each of their tumors. He wants to understand the important changes in gene expression that occur in breast cancer, and especially changes that result in structural changes in the cells and those that may affect response to treatment and survival. Suggest a way of doing this, and and suggest any additional data that he should collect for this purpose.

Solution:

Sparse CCA would be useful. Perhaps sparse on the gene side, but no L1 penalty for the 10 clinical features (there's so few of them). Additional data? Survival time would be really helpful.

168. (CMU, 2011s, TMitchell, HW5, ex.2.2) Canonical correlation analysis (CCA) handles the situation that each data point (i.e., each object) has two representations (i.e., two sets of features), e.g., a web page can be represented by the text on that page, and can also be represented by other pages linked to that page. Now suppose each data point has two representations x and y, each of which is a 2-dimensional feature vector (i.e., $x = [x_1, x_2]^\top$ and $y = [y_1, y_2]^\top$).

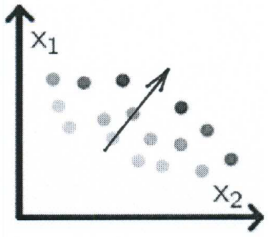
Given a set of data points, CCA finds a pair of projection directions (u, v) to maximize the sample correlation $\hat{\text{corr}}(u^\top x)(v^\top y)$ along the directions u and v. In other words, after we project one representation of data points onto u and the other representation of data points onto v, the two projected representations $u^\top x$ and $v^\top y$ should be maximally correlated (intuitively, data points with large values in one projected direction should also have large values in the other projected direction).



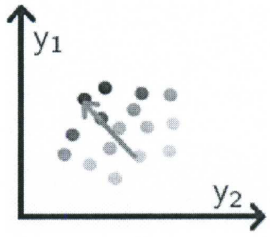
Now we can see data points shown in the figure above, where each data point has two representations $x = [x_1, x_2]^\top$ and $y = [y_1, y_2]^\top$. Note that data are paired: each point in the left figure corresponds to a specific point in the right figure and vice versa, because these two points are two representations of the same object. Different objects are shown in different gray scales in the two figures (so you should be able to approximately figure out how points are paired). In the right figure we've given one CCA projection direction v, draw the other CCA projection direction u in the left figure.

Solution:

The CCA projection direction is shown in the following figure (on the left).



2(a)



2(b)

10.2 Revision

169. (CS189 Spring 2018 HW9, ex. 3) **LDA and CCA**

Consider the following random variable $X \in \mathbb{R}^d$, generated using a mixture of two Gaussians. Here, the vectors $\mu_1, \mu_2 \in \mathbb{R}^d$ are arbitrary (mean) vectors, and $\Sigma \in \mathbb{R}^{d \times d}$ represents a positive definite (covariance) matrix/ For now, we will assume that we know all of these parameters.

Draw a label $L \in \{1, 2\}$ such that the label 1 ischosen with probability π_1 (and consequently, label 2 with probability $pi_2 = 1 - \pi_1$), and geenrate $X \sim \mathcal{N}(\mu_L, \Sigma)$.

(a) Now given a particular $X \in \mathbb{R}^d$ generated from the above model, we wich to find its label. Write out the decision rule corresponding to the following estimates of L:

- MLE
- MAP

Your decision rule should take the form of a threshold: if some function $f(X) > T$, then choose the label 1, otherwise choose the label 2. When are these two decision rules the same? Hint: investigate the ratio between the two likelihood functions and the ratio between the two posterior probabilities respectively.

(b) You should have noticed that the function f is linear in its argument X, and takes the form $w^\top(X - y)$. We will now show that CCA defined on a suitable set of random variables leads to precisely the same decision rule.

Let $Y \in \mathbb{R}^2$ be a one hot vector denoting the realization of the label l, i.e., $Y_l = 1$ if $L = l$, and zero otherwise. Let $\pi_1 = \pi_2 = 1/2$. Compute the covariance matrices Σ_{XX}, Σ_{XY} and Σ_{YY} as a function of

μ_1, μ_2, Σ. Recall that the random variables are not zero-mean. Hint: when computing the covariance matrices, the tower property of the expectation is useful.

(c) Let us now perform CCA o the two random variables. Recall that in order to find the first canonical directions, we look for vector $u \in \mathbb{R}^d$ and $v \in \mathbb{R}^2$ such that $\rho(u^\top C, v \top Y)$ is maximized.

Show that the maximizing u^* is proportional to $\Sigma^{-1(\mu_1-\mu_2)}$. Recall that u^* is that "direction" of X that contributes most to predicting Y. What is the relationship between u^* and the function $f(X)$ computed in the first subpoint of this exercise?

Hint: The Sherman-Morrison formula for matrix inversion may be useful:

Suppose $A \in \mathbb{R}^{d\times d}$ is an invertible square matrix and $a, b \in \mathbb{R}^d$ are column vectors. Then,

$$(A + ab^\top)^{-1} = A^{-1} - \frac{A^{-1}ab^\top A^{-1}}{1 + b^\top A^{-1}a}$$

170. (CS 189 Introduction to Machine Learning Summer 2018 DIS6, ex. 1) **OLS, Ridge Regression, TLS, PCA and CCA**

In this discussion, we will review several topics we have learnt so far. We emphasize on their basic attributes, including the objective functions, the generative models as well as the explicit form of solutions. You will also learn the connection and distinction between those methods.

(a) What problem does each of the methods trying to solve? What are their objective functions? Can you write out their solutions in a closed form? What are the probabilistic perspectives for OLS, ridge regression and total least squares?

(b) Suppose you have a matrix $X \in \mathbb{R}^{n\times d}$ and vector $y \in \mathbb{R}^{n\times 1}$. Use PCA to compute the first k principal components of $[Xy]$. Show how this solution would relate to a TLS solution to the problem.

(c) Among OLS, Ridge and TLS, what method would you use when: (1) observation X is noisy (2) X is not noisy and $d >> n$ (3) X is not noisy and $d << n$?

(d) How do OLS, ridge and TLS interact with the matrix $X^\top X$ in the closed form solutions? What are the eigenvalues of the matrix being inverted in the closed form solutions? Do you have any intuitions of why the eivenvalues changes in those manners?

(e) Suppose you have a multi-variate regression problem, i.e., the feature matrix is $X \in \mathbb{R}^{n \times p}$ and the regression target is $Y \in \mathbb{R}^{n \times q}$ and $q > 1$. We know a prior that the number of regression targets is large and there are strong correlations between the multiple regression targets. For example, consider you have $n = 100$ samples. Each example has $p = 500$ features, and there are $q = 1000000$ regression targets.

There are two approaches you can solve the problem. The first approach is treat the multivariate regression problem as q independent ridge regression problems. The second one is that first compute the CCA between X and Y, which gives two projection matrices U_k and V_k, then use q independent ridge regressions to fit $Y_c \equiv YV_k$ from $X_c \equiv XU_k$, i.e., solve for W that satisfy $X_C W \approx Y_c$. The final predictor is given by: $Y_{\text{predict}} = X(U_k W V_k^{-1})$. What's the pros and cons of each approach?

Solution:

(a) OLS assumes a linear model with noisy y, its objective function is: $\arg\min_w \|Xw - y\|^2$. Its underlying generative model is $y = Xw + \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$. The closed form solution if $(X^\top X)^{-1} X^\top y$.

Ridge regression adds extra regularization terms on top of OLS. In the generative model aspect, it puts a Gaussian prior on w, i.e., $y = Xw + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \lambda I)$ and $w \sim \mathcal{N}(0, I)$. The corresponding optimization problem if $\arg\min_w \|Xw - y\|^2 + \lambda \|w\|^2$. The closed form solution is $(X^\top X + \lambda I)^{-1} X^\top y$.

TLS is another modification of OLS by no longer assuming we have noiseless X. To be specific, the generative model is $y + \epsilon_y = (X + \epsilon_x)w$, where $\epsilon_w, \epsilon_y \sim \mathcal{N}(0, I)$. The objective in this case is: $\arg\min_{\epsilon_x, \epsilon_y} \|[\epsilon_x \epsilon_y]\|_{\text{Fro}}^2$, subject to $(X + \epsilon_x)w = y + \epsilon_y$. And the closed form solution is $(X^\top X - \sigma_{d+1}^2 I)^{-1} X^\top y$, where σ_{d+1} is the least singular value of $[Xy]$.

PCA deals with the dimension reduction of X. For the one dimensional case, the objective is to $\arg\max_{\|u\|=1} \|Xu\|^2$. The closed form solution is given by first do SVD on $X = U\Sigma V^\top$, then $u = V_1$, where V_1 is the first column of V.

CCA models relationship between two point sets X and Y. The objective is $\arg\max_{u,v} \rho(X_{rv}^\top u, Y_{rv}^\top v)$. The solution is: $u = W_x D_x u_d$ and $v = W_y D_y v_d$, where the whitening step gives: $W_x = V_x \Sigma_x^{-1} V_x^\top$ and V_x, Σ_x given by the SVD $X = U_x \Sigma_x V_x^\top$. The whitened matrix has the property of $(XW_x)^\top (XW_x) = I$. Similar results hold

for W_y. The decorrelation step gives $D_x = U_r$ and $D_y = V_r$, where $X + w^\top Y_w = U_r \Sigma_r V_r^\top$. After decorrelation, $(X_w D_x)^\top (Y_w D_y) = I$.

(b) Using what we've learned in PCA, we know that the SVD decomposition of the matrix becomes $[Xy] = U\Sigma V$. From there, we pick the resulting k vectors such that we start at $u = V_1$, where V_1 is the first column of V and go up to $u = V_k$ is the k-th column of V.

How is the TLS solution found? TLS minimizes $\|[\epsilon_X \epsilon_y]\|_{\text{Fro}}^2$, subject to the constraint of:

$$[X + \epsilon_X, y + \epsilon_y][w, -1]^\top = 0$$

Since we want to find the minimum noise, that after perturbation, has at least one zero eigenvalue. By Eckhart-Young theorem, we know that after perturbation, has at least one zero eigenvalue. By Eckhart-Young theorem, we know that the noise perturbed $[X + \epsilon_X; y + \epsilon_y]$ should be the best rank-d approximation to the original matrix $[X; y]$. Taking a step back, we see that TLS seeking a rank-d best approximation has the same objective function as PCA when reducing the dimensionality to d.

(c) (1) When X is noisy, TLS is preferred, since it models the noise explicitly in the model. It can recover the latent structure by explicitly removing the noise. However, note that when doing out of sample prediction, TLS is better only when the testing X is noiseless or has less noise than the training X.

(2) In this case, the problem is under-constraint and ridge regression fits best. TLS assumes that the X has noise. The matrix inversion step in OLS $((X\top X)^{-1})$ will be unstable due to $d > n$. Thus TLS and OLS do not fit this scenario.

(3) In this case, OLS fits the best, if there are no linear relationship among the features. The problem is over-constraint and generally the matrix inversion step $((X\top X)^{-1})$ is stable. However, if X does not have full column rank, the matrix inversion will fail. In that case, ridge regression with small regularization is preferred over OLS.

(d) OLS invert the matrix $X^\top X$, ridge invert the regularized matrix $X^\top X + \lambda I$ and TLS invert the $X^\top X - \sigma_{d+1}^2 I$ matrix. Thus those three methods either add, subtract an identity matrix. For the OLS case, it could be interpreted as adding $0I$. The eigenvalues in the ridge regression adds λ to the eigenvalues of the OLS case, while the TLS subtract σ_{d+1}^2. Ridge regression regularizes the model by having an extra $\lambda\|w\|^2$ loss.

It turns out that it is also helping the eigenvalues of $X^\top X$ to stay away from zero, and thus improving the numerical stability. TLS on the other hand, seems to decrease the numerical stability. However, as shown below, it could be think of as removing the components generated by noise.

$$E[X^\top X]$$
$$E[(X^* + \epsilon_x)^\top (X^* - \epsilon_x)]$$
$$E[X^{\top *} X + E[\epsilon_x^\top \epsilon_x]]$$
$$E[X^{\top *} X + \sigma^2_{\text{noise}} I]$$

In this case, we know that even the smallest eigenvalues is not zero, which comes from the noise. The TLS explicitly remove that noise by subtracting the σ^2_{d+1}.

(e) The ridge regression is assuming that each of the fitting target is independent. On the other hand, the CCA approach has taken into account of the potential correlation among the fitting targets. Having taken advantage of the target correlation, CCA might have higher statistical efficiency than the set of independent ridge regressions. For example, the ridge regression use $n = 100$ examples to fit each target, with $p = 500$ dimensional features. It will results in severe overfitting in general. However, if we assume that the regression targets have large correlation with each other, conceptually we are using $nq = 10E8$ examples to fit each target, with the same number of features. Obviously CCA in this case will be much better than the first approach.

The independent ridge regression assumes identity Gaussian noise on each of the fitting target. On the other hand, the noise model for the CCA approach depends on the data, influenced via the computed matrices U_k and V_k. It is conceptually much harder to make sense of this noise model and usually people use a noise model that does not depend on the data.