EE 046202 - Technion - Unsupervised Learning & Data Analysis

Homework 1 - Dimensionality Reduction

Agenda

- Questions
  - Matrix derivatives refresher
  - CCA
  - t-SNE Gradient
  - PCA
  - PCA & Friends (Exam-ish Question) - BONUS
- Python Exercise - KPCA Implementation
- Python Exercise - PCA and TSNE

Use as many cells as you need

אפשר גם לכתוב בעברית, אבל עדיף באנגלית

- Code Tasks are denoted with: </>
- Questions (which you need to answer in a Markdown cell) are denoted with: ?
- $\LaTeX$ Cheat-Sheet (to write equations)
  - Another Cheat-Sheet

Students Information

- Fill in

Name	Campus Email	ID
Student 1	student_1@campus.technion.ac.il	123456789
Student 2	student_2@campus.technion.ac.il	987654321

Submission Guidelines

- Maximal garde: **100** (even with the bonus, the grade will not be above 100).
  - Example: if you got 5 points bonus, but you were right in all sections, your grade will still be 100 (and not 105).
  - Example: if you got 5 points bonus, and 6 points were deducted for wrong answers, your grade will be 99.

- Submission only in **pairs**.
  - Please make sure you have registered your group in Moodle (there is a group creation component on the Moodle where you need to create your group and assign members).
- **ANSWERS TO THEORETICAL/MATHEMATICAL QUESTIONS**:
  - **Typed - 5 points bonus**: you can type directly in a Markdown cell using Latex (see cheatsheets above), or use Word, Overleaf, LyX...
    - This is a really good practice, we encourage you to practice your math typing skills.
- SAVE THE NOTEBOOKS WITH THE OUTPUT, CODE CELLS THAT WERE NOT RUN WILL NOT GET ANY POINTS!
- What you have to submit:
  - If you have answered the questions in the notebook, you should submit this file only, with the name: `ee046202_hw1_id1_id2.ipynb`.
  - If you answered the questions in a different file you should submit a `.zip` file with the name `ee046202_hw1_id1_id2.zip` with content:
    - `ee046202_hw1_id1_id2.ipynb` - the code tasks
    - `ee046202_hw1_id1_id2.pdf` - answers to questions.
  - No other file-types (`.py` , `.docx` ...) will be accepted.
- Submission on the course website (Moodle).
- **Latex in Colab** - in some cases, Latex equations may no be rendered. To avoid this, make sure to not use *bullets* in your answers ("* some text here with Latex equations" -> "some text here with Latex equations").

Working Online and Locally

- You can choose your working environment:

  1. `Jupyter Notebook` , **locally** with Anaconda or **online** on Google Colab
     - Colab also supports running code on GPU, so if you don't have one, Colab is the way to go. To enable GPU on Colab, in the menu: `Runtime` → `Change Runtime Type` → `GPU` .
  2. Python IDE such as PyCharm or Visual Studio Code.
     - Both allow editing and running Jupyter Notebooks.
- Please refer to `Setting Up the Working Environment.pdf` on the Moodle or our GitHub (https://github.com/taldatech/ee046202-unsupervised-learning-data-analysis) to help you get everything installed.

- If you need any technical assistance, please go to our Piazza forum (`hw1` folder) and describe your problem (preferably with images).

Keyboard Shortcuts

- Run current cell: **Ctrl + Enter**
- Run current cell and move to the next: **Shift + Enter**
- Show lines in a code cell: **Esc + L**
- View function documentation: **Shift + Tab** inside the parenthesis or `help(name_of_module)`
- New cell below: **Esc + B**
- Delete cell: **Esc + D, D** (two D's)

Tip

If you find it more convenient, you can copy the section to a new cell, and answer the question just right below it. For example:

Question 0

1. What is the best course in the Technion?
2. Why does no one pick Bulbasaur as first pokemon?
3. Why is there no superhero named Catman?

Answers - Q0

Q0 - Section 1

- Q: What is the best course in the Technion?

ANAM!

Q0 - Section 2

- Q: Why does no one pick Bulbasaur as first pokemon?

It is really a riddle....

Q0 - Section 3

- Q: Why is there no superhero named Catman?

I got nothing.

Question 1 - Matrix derivatives

Vector & Matrix Deriviatives

- $\nabla_x Ax = A^T$
- $\nabla_x x^T Ax = (A + A^T)x$
- $\frac{\partial}{\partial A} \ln |A| = A^{-T}$
- $\frac{\partial}{\partial A} Tr[AB] = B^T$

Using the above, we will use the following:

1. $\nabla_\mu \mu^T \Sigma^{-1} x_i = \Sigma^{-1} x_i$
2. $\nabla_\mu \mu^T \Sigma^{-1} \mu = (\Sigma^{-1} + \Sigma^{-T})\mu$
3. $\frac{\partial}{\partial \Sigma^{-1}} \ln |\Sigma^{-1}| = \Sigma^T = \Sigma$
4. $\frac{\partial}{\partial \Sigma^{-1}} Tr[\Sigma^{-1} \sum_{i=1}^n (\overline{x_i} - \overline{\mu})(\overline{x_i} - \overline{\mu})^T] = \sum_{i=1}^n (\overline{x_i} - \overline{\mu})(\overline{x_i} - \overline{\mu})^T$

Question time

Let $X_1, \dots, X_N \sim \mathcal{N}(\mu, \Sigma)$ i.i.d., $\mu \in \mathcal{R}^d,\ \Sigma \in \mathcal{R}^{d \times d}$.

1. Find the MLE estimator $\hat{\mu}_{MLE}$.
   - Reminder: finding the MLE estimator is done by comparing the gradient to zero.
   - Use the vector and matrix derivatives above.
2. Find the MLE estimator $\hat{\Sigma}_{MLE}$
   - You should use **The Trace Trick** - $\sum_{i=1}^n (\overline{x_i} - \overline{\mu})^T \Sigma^{-1} (\overline{x_i} - \overline{\mu}) = \sum_{i=1}^n Trace\big((\overline{x_i} - \overline{\mu})^T \Sigma^{-1} (\overline{x_i} - \overline{\mu})\big) = Trace\big(\Sigma^{-1} \sum_{i=1}^n (\overline{x_i} - \overline{\mu})(\overline{x_i} - \overline{\mu})^T\big)$.
   - It may be easier to use $\frac{\partial}{\partial \Sigma^{-1}}$.

Question 2 - CCA

In the lecture, we defined the CCA problem and the matrix

$$M = \Sigma_{xx}^{-1/2}\Sigma_{xy}\Sigma_{yy}^{-1/2} \in \mathbb{R}^{D_x \times D_y}$$

.

Show that the solution of the CCA problem is obtained via principal eigenvectors of

$$MM^T \in \mathbb{R}^{D_x \times D_x},\ M^TM \in \mathbb{R}^{D_y \times D_y}$$

. Hint: Use Lagrange multipliers, find singular vectors for M, M^T.

Question 3 - t-SNE Gradient

Recall the objective of t-SNE algorithm:

$$C = KL(P||Q) = \sum_{k,l \neq k} p_{lk} \log \frac{p_{lk}}{q_{lk}}$$

Calculate the gradient of the objective function with respect to the low-dimensionality mappings y_i, i.e. calculate $\frac{\partial C}{\partial y_i}$.

Hint:

- Denote $(1 + ||y_i - y_j||^2)^{-1} = E_{ij}^{-1}$.
    - Notice that $E_{ij} = E_{ji}$.

Question 4 - PCA

1. Assume we have a collection of data with m samples and n features per sample: $S \in \mathcal{R}^{m \times n}$. Assume the data is already centered. Let $C = S^TS$, C is an $n \times n$ symmetric matrix and thus diagonalizable. Assum that C has eigenvalues

   $$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n.$$

   Moreover, C has an **orthonormal** basis of eigenvectors

   $$v_1, v_2, \ldots, v_n \text{ such that } Cv_i = \lambda_i v_i.$$

   In particular,

   $$v_i^T C v_i = \lambda_i v_i^T v_i = \lambda_i ||v_i||_2^2 = \lambda_i.$$

   Claim: v_1 represents the direction of the largest variance of S. *Prove* the claim by showing the following: if $u = a_1v_1 + a_2v_2 + \ldots + a_nv_n$ is a unit vector in $\mathcal{R}^n$ then $u^TCu \leq \lambda_1$, i.e., v_1 is the direction of the largest variance of S.

2. **Theorem**: Let $X \in \mathcal{R}^{d \times n}$ be the data matrix with n samples and d features, $W = XX^T$ the covariance matrix (note the difference in definition from section 1), $u_1, \ldots, u_d$ the eigenvectors of W and $\lambda_1, \ldots, \lambda_d$ the corresponding eigenvalues. Let $P_{u_i} : \mathcal{R}^d \rightarrow \mathcal{R}^d$ be the projection operator onto the subspace $\text{span}\{u_i\}$. Then

   $$\sum_{j=1}^{n} ||P_{u_i} x_j||_2^2 = \lambda_i.$$

Prove the theorem.

- Frobenius Norm: $||A|||_F^2 = \sum_{i=1}^m \sum_{j=1}^n ||a_{ij}||_2^2$
- $||A||_F^2 = Tr(AA^T)$
- $W = UDU^{-1} = UDU^T$, D is a diagonal matrix with the eigenvalues on the diagonal.
- The projection operator matrix: $P_{u_i} = u_i u_i^T$

Question 5 - PCA & Friends (Exam-ish Question) - BONUS

1. Canonical Correlation Analysis (CCA): Consider two zero-mean random vectors, $x \in \mathbb{R}^{D_x}$ and $y \in \mathbb{R}^{D_y}$. We would like to find a one-dimensional representation of x and y that **maximizes the correlation** between them. Let these 1D representations be given by $a^T x$ and $b^T y$, respectively, $a \in \mathbb{R}^{D_x}, b \in \mathbb{R}^{D_y}$.
   - This problem can be formulated as the optimization problem

     $$\max_{a,b} \mathbb{E}\left[(a^T x)(b^T y)\right] \text{ s.t. } \mathbb{E}\left[(a^T x)^2\right] = 1, \mathbb{E}\left[(b^T y)^2\right] = 1$$

     Motivate each term and explain its origin.
   - Relate the previous expression to PCA.
2. Explain in your own words the difference of the statistical and geometric view point. Specifically,
   - Discuss the difference in optimization criteria according to these different view points.
   - Explain the redundancy in the MSE solution and explain why it is absent in variance max formulation.
3. Show that the conventional linear PCA algorithm is recovered as a special case of Kernal PCA. That is, if we choose the linear kernel function given by $k(x, y) = x^T y$ we get the PCA algorithm.

Question 6 - Python - KPCA Implementation

1. Implement the KPCA algorithm (do not use scikit-learn's implementation).

   - The KPCA function should have the form `KPCA(X, d, kernel)` where:
     - $X \in \mathbb{R}^{D \times N}$ - the data matrix with N data samples and D features.
     - d is the final dimension of each data point.
     - `kernel` is a **function** which receives x, y and returns $k(x, y)$ for these points.
       - When applying the KPCA, you can use Python's `lambda` function to write a one-line function.
       - If you don't want to use `lambda`, you can use the regular functions with `def func(): ...`, but you will need to change the signature of KPCA to `KPCA(X, d, kernel, *args)`, where `args` is a list of parameters that will be fed into `kernel`.
         - For example, for the Gaussian kernel, you will define a function in the form `GaussianKernel(x,y, sigma)`, and then you will call KPCA like that: `KPCA(X,d, GaussianKernel, sigma): ... k = GaussianKernel(x_1, x_2, sigma) ...`
       - Read more: *args, Lambda functions
     - For the eigenvectors and eigenvalues, use `np.linalg.eigh`.
     - How to make sure your implementation is correct? You can compare with scikit-learn's KPCA (see Tutorial 1).
2. Create a dataset of 1000 points using scikit-learn's `sklearn.datasets.make_s_curve` (use the default parameters). Apply PCA (use the one from the tutorial or scikit-learn's) and your KPCA. Use the Gaussian Kernel and a second (nonlinear) kernel of your choice. Explain how you picked σ for the Gaussian Kernel. Plot the original dataset (in **3D**), and its representations in 2D you obtained (4 plots in total: Original, PCA, KPCA (Gaussian), KPCA (Other)).

In [1]:

```
# imports
# you can add more if you need
import numpy as np
```

```python
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn.datasets import make_s_curve
from sklearn.decomposition import PCA
```

In [ ]:

```python
"""
Your Code Here (add as many cells as you need)
"""
```

Question 7 - Python - Dimensionality reduction methods

In this exercise we are going to compare different dimensionality reduction techniques on the Wine dataset. The wine dataset is a classic and very easy multi-class classification dataset. There are 3 types of wine, 178 examples with 13 features each. Even though it is a very dataset for classification, we will do unsupervised analysis to compare different aspects of dimensionality reduction methods. Let's look at the data.

In [1]:

```python
# imports for the question
# you can add more if you wish (but it is not really needed)
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA, KernelPCA
from sklearn.preprocessing import StandardScaler
from sklearn.datasets import load_wine, load_digits

X, y = load_wine(return_X_y=True)
```

Task 7.1 - Importance of Feature Scaling

- Perform PCA on the data (`X`) to `n_components=2` and plot it.
- Scale the data using `StandardSaler()` to create `X_scaled`, perform PCA on `X_scaled` to `n_components=2` and plot it.
- Color the datapoints according to their class: `ax.scatter(X[:,0], X[:,1], c=y)`

Note: for all algorithms, use constant seed (`random_state=...`), for example: `PCA(n_component=2, random_state=random_state)`

In [ ]:

```python
# your code here - you can use as many cells as you need
```

Question 7.1 - Importance of Feature Scaling

- Why is it so important to perform feature scaling before performing dimensionality reduction, espicially for PCA?
- Describe the results of Task 7.1

Task 7.2 - T-SNE

- Perform t-SNE on the scaled dataset to `n_components=2` and plot it.
- Find the `perplexity`, that yields the best-looking results.

In [ ]:

```python
# your code here - you can use as many cells as you need
```

Task 7.3 - Robustness to Noise & Outliers

In this task we are going to test how robust are t-SNE and PCA to noisy features and outliers. We will have 2 new datasets:

1. `X_noisy` - random normal noise ($\mathcal{N}(0,1)$) is added to the current features after scaling.
2. `X_skewed` - 20 random samples (random feature values) are added to the sample (total samples: 178 + 20 = 198)

The tasks:

- Perform standardization to create `X_noisy_scaled`, `X_skewed_scaled`
- Peform PCA and t-SNE to both datasets (4 in total) to `n_components=2` (as before) and plot the results (4 plots, can be in pairs and can be a 2X2 plot, don't forget to put titles).
  - You should tune the `perplexity` for t-SNE

In [ ]:

```
# your code here - you can use as many cells as you need
```

Question 7.3 - Robustness to Noise & Outliers

- Explain the results. In your answer, describe the effect of adding noise and outliers, which algorithm is more affected and why?

Credits

- Icons from Icon8.com - https://icons8.com
- Datasets from Kaggle - https://www.kaggle.com/