Exercise 2

AMTH/CPSC 445a/545a - Fall Semester 2016

September 21, 2017

> Compress your solutions into a single zip file titled `<lastname and initials>_assignment2.zip`, e.g. for a student named Tom Marvolo Riddle, `riddletm_assignment2.zip`. Include a single PDF titled `assignment2.pdf` and any MATLAB or Python scripts specified. Your homework should be submitted to Canvas before Thursday, October 5, 2017 at 1:00 PM.
>
> Programming assignments should use built-in functions in MATLAB or Python; In general, Python implementations may use the `scipy` stack [1]; however, exercises are designed to emphasize the nuances of data mining algorithms - if a function exists that solves an entire problem (either in the MATLAB standard library or in the `scipy` stack), please consult with the TA before using it.

Problem 1

Sketch a "unit circle" around $(0, 0)$ in each of the following distance metrics:

1. Euclidean
2. Manhattan
3. Supremum
4. $L^{\frac{2}{3}}$
5. Mahalanobis with covariance $\Sigma = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 1 \end{bmatrix}$

Include the five sketches in `assignment2.pdf`.

Remark. The wiki page for L^p spaces may prove useful for #4 [2]. Notice that technically, $L^{\frac{2}{3}}$ is not a formal distance *metric*, but rather a dissimilarity. However, a "unit circle" can still be plotted for it.

Problem 2

Prove that the following are distance metrics:

1. $1 - J(x, y)$, where J is the Jaccard coefficient
2. Minimal angle between unit vectors (i.e., $\arccos(\text{cosine\_similarity}(x, y))$, where $\|x\| = \|y\| = 1$).

3. Length of shortest path (i.e., Geodesic distance) on a weighted undirected graph $G(V, E, w)$, where V are the set vertices, E is the set of edges and $w : E \times E \to (0, \infty)$ are the weights.

4. Mutual set difference $d(A, B) = |A \backslash B| + |B \backslash A|$, where A and B are arbitrary sets. Show that this distance is equivalent to an L^1 distance.

Include the four proofs in `assignment2.pdf`.

Problem 3

Prove that given N data points in $\mathbb{R}^n$ centered around their mean (i.e., the corresponding $N \times n$ data matrix X satisfies $\sum_{i=1}^{N} X(i, j) = 0$ for each $j = 1, \ldots, n$), the k-dimensional coordinates (for $k < n$) given by PCA and by MDS are identical.
Include this proof in `assignment2.pdf`.

Problem 4

Using the template `script4.py` or `script4.m` complete the following problem.

- The template will load a list of cities `c` and distance matrix D. In MATLAB notation the distance `D(i,j)` corresponds to the distance between city `c{i}` and city `c{j}` (In Python notation `D[i,j]` is the distance between city `c[i]` and city `c[j]`).
- Implement and use the (classical) Multidimensional Scaling (MDS) algorithm to assign each city two-dimensional coordinates in $\mathbb{R}^2$. Use Singular Value Decomposition (`svd` in MATLAB, or `np.linalg.svd` in Python).
- Plot a point at the MDS coordinates of each city and label the point with the name of the corresponding city.
- **Optional** Repeat using eigendecomposition via `eig` or `np.linalg.eig`. Note the similarities and differences in the two plots. Are the two plots different beyond isometry?

Include the produced plot in `assignment2.pdf`.

Notice

Do not use any built-in MDS function (e.g. `cmdscale` in MATLAB) or functions from outside the scipy stack in Python (e.g. `sklearn.manifold.MDS`). Use the template `script4.m` or `script4.py` depending on if you are using MATLAB or Python. [Python] In general you should be able to complete the assignment with only the imported packages in the template.

Problem 5

Using the template `script5.py` or `script5.m` complete the following problem. Generate a toy example to test Mahalanobis distances & PCA and visualize them using scatter and quiver plots, based on the following steps:

Part A: data generation

- Sample 1000 points uniformly from the interval $[0, 10]$ on the horizontal axis in $\mathbb{R}^2$(i.e., x-coordinate in $[0, 10]$ and y-coordinate being 0);
- Choose an angle θ and rotate your points about the origin using a rotation matrix R_θ formed using this angle;
- Add 2-dimensional Gaussian noise to generate the toy example data.

Part B: Mahalanobis and principal components

- Find mean and covariance of the data;
- Compute the Mahalanobis distance of each data point from the mean of the data;
- Compute the two principal components ϕ_1 & ϕ_2 (i.e., covariance eigenvectors) and associated eigenvalues λ_1 & λ_2 .

Part C: Visualization

- Produce a 2D scatter plot of the generated data, where each data point is colored by its Mahalanobis distance from the data mean. For this plot use filled markers (using the flag 'filled') of size of 9 (using the parameter 'S');
- Use the quiver command to add to this plot the two vectors $\sqrt{\lambda_1}\phi_1$ & $\sqrt{\lambda_2}\phi_2$ whose directions is determined by the principal components and their length is determined by the corresponding eigenvalues. For this plot set the color of the vectors to be black (using the flag 'k') and the line width to 4 (using the 'LineWidth' option);
- Save the produced plot as a PNG file with the name `mahal_pca_illustration.png`.

Include the produced plot in `assignment2.pdf` .

Notice: Use the template `script5.m` or `script5.py` depending on if you are using MATLAB or Python. Do not use any MDS or PCA functions built into MATLAB or SciPy (i.e., you must write your own from basic matrix factorizations, such as `svd` / `np.linalg.svd` or `eig` / `np.linalg.eig`, and what you know about these techniques).

Problem 6

Using the template `script6.py` or `script6.m` complete the following. In this problem you will write an implementation of PCA that is more efficient for data with many features. This implementation skips the computation of the covariance matrix and uses the following steps:

1. The template will load a data matrix `X` whose rows correspond to images of a spinning bunny, a vector `theta` that corresponds to the angles of the bunny, and a vector `sz` that describes the dimensions of the bunny images. An example of how to reshape the rows of `X` using `sz` to visualize the bunny images is included in the template.
2. Average over the columns to `X` to compute the mean or average row for `X`. This mean row corresponds to the mean bunny image.
3. Plot an image of the mean bunny.
4. Save the image as `mean_bunny.png` .

5. Subtract the mean bunny from each row of `X` to center the data.
6. Compute the eigenvectors and eigenvalues of the covariance matrix (***indirectly***) using the Singular Value Decomposition (SVD) of `X` .[*Hint*: to avoid memory problems, the SVD needs to be run on `'econ'` mode in MATLAB or with `full_matrices=False` in Python.]
7. Project `X` on to the first three principal components.
8. Create a 3-dimensional scatter plot of these coordinates colored by the angle `theta` .
9. Add a title and save your PCA scatter plot as `pca_coordinates.png` .
10. Using (classical) Multidimensional Scaling (MDS), compute coordinates for the bunny images in $\mathbb{R}^3$.
11. Plot a 3-dimensional scatter plot of the MDS coordinates colored by the angle. `theta` [Hint: How should your two scatter plots compare in light of problem 3?]
12. Add a title and save your MDS scatter plot as `mds_coordinates.png` .

Include the three produced plot in `assignment2.pdf` .

Notice

Use the template `script6.m` or `script6.py` depending on if you are using MATLAB or Python. Do not use any MDS or PCA functions built into MATLAB or SciPy (e.g., you must write your own from basic matrix factorizations, such as `svd` / `np.linalg.svd`, and what you know about these techniques).

References

[1] "The scipy stack specification." [Online]. Available: https://www.scipy.org/stackspec.html

[2] Wikipedia, "Lp space — wikipedia, the free encyclopedia," 2017, [Online; accessed 18-September-2017]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Lp_space&oldid=795629487