UNIVERSIDAD DE ANTIOQUIA
1803

Lab I - Dimensionality Reduction

Machine Learning II

Prerequisites

Python

Python

- Data structures (properties of lists, tuples, dicts, built-in modules…)
- Classes
- Packages and modules

NumPy

- Arrays
- Inner product
- Vector - Matrix product
- Distances

Linear Algebra

Class Concepts

- Vectors and matrices
- Properties of matrices
- Eigendecomposition
- Dimensionality reduction

Workshop I

1. Simulate any random rectangular matrix A.
    - What is the rank and trace of A?
    - What is the determinant of A?
    - Can you invert A? How?
    - How are eigenvalues and eigenvectors of A'A and AA' related? What interesting differences can

you notice between both?
    - See [https://sites.cs.ucsb.edu/~mturk/Papers/jcn.pdf]
2. Add a steady, well-centered picture of your face to a shared folder alongside your classmates.
    - Edit your picture to be 256x256 pixels, grayscale (single channel)
    - Plot your edited face
    - Calculate and plot the average face of the cohort.
    - How distant is your face from the average? How would you measure it?
3. Let's create the *unsupervised* Python package
    - Same API as scikit-learn: fit(), fit_transform(), transform(), hyperparams at init
    - Manage dependencies with Pipenv or Poetry
    - Implement SVD from scratch using Python and NumPy
    - Implement PCA from scratch using Python and NumPy [https://github.com/rushter/MLAlgorithms/blob/master/mla/pca.py, https://github.com/patchy631/machine-learning/blob/main/ml_from_scratch/PCA_from_scratch.ipynb]
    - Implement t-SNE from scratch using Python and NumPy [https://nlml.github.io/in-raw-numpy/in-raw-numpy-t-sne/]
4. Apply SVD over the picture of your face, progressively increasing the number of singular values used. Is there any point where you can say the image is appropriately reproduced? How would you quantify how different your photo and the approximation are?
5. Train a naive logistic regression on raw MNIST images to distinguish between 0s and 8s. We are calling this our baseline. What can you tell about the baseline performance?
6. Now, apply dimensionality reduction using all your algorithms to train the model with only 2 features per image.
    - Plot the 2 new features generated by your algorithm
    - Does this somehow impact the performance of your model?
7. Repeat the process above but now using the built-in algorithms in the Scikit-Learn library. How different are these results from those of your implementation? Why?
8. What strategies do you know (or can think of) in order to make PCA more robust? (Bonus points for implementing them) [https://nbviewer.org/github/fastai/numerical-linear-algebra/blob/master/nbs/3.%20Background%20Removal%20with%20Robust%20PCA.ipynb]
9. What are the underlying mathematical principles behind UMAP? What is it useful for?
10. What are the underlying mathematical principles behind LDA? What is it useful for?
11. Use your unsupervised Python package as a basis to build an HTTP server that receives a record as input and returns the class of the image. Suggestions: MNIST digit classifier, Iris classifier…