Northeastern University
CS6220 – Data Mining Techniques
Fall 2017, Derbinsky

Homework 3, Part 1

Submit all parts of the assignment in a **single ZIP file**, containing ...

- A single PDF (`written.pdf`) with typset responses to *all* non-coded problems
- The notebook (`starter-3.2.ipynb`) from problem 2
- The notebook (`starter-3.3.ipynb`) from problem 3

1 PCA Eigenvector Orthogonality

The base approach to PCA entails the eigendecomposition of the covariance matrix. We discussed that this yields a set of orthogonal eigenvectors – a fact that derives from the symmetry of the covariance matrix. Your task is to prove that the eigenvectors of any symmetric matrix are orthogonal. Begin with the following ...

$$A\vec{x} = \lambda_1\vec{x}$$
$$A\vec{y} = \lambda_2\vec{y}$$

where A is a symmetric matrix, $\vec{x}$ and $\vec{y}$ are the eignvectors that correspond, respectively, to eigenvalues λ_1 and λ_2. Show mathematically that $\vec{x}$ and $\vec{y}$ must be orthogonal if the eigenvalues are different.

2 Eigenvalues/Eigenvectors via NumPy & SymPy

NumPy (Numerical Python) is one of the foundational packages for scientific computation. The major element is the `ndarray`, an efficient array representation, upon which many other libraries are built. NumPy also includes functions for linear algebra, random number generation, etc. If you are unfamiliar with NumPy, it is highly recommended you walk through their Quickstart tutorial[1].

SymPy is a pure-Python package for symbolic mathematics. If you are unfamiliar with SymPy, it is highly recommended you skim their Tutorial[2].

You will do all the work inside the `starter-3.2.ipynb` Jupyter notebook file. Make sure to complete all items flagged as `TODO`.

[1] `https://docs.scipy.org/doc/numpy-dev/user/quickstart.html`
[2] `http://docs.sympy.org/latest/tutorial/index.html`

3 Applying PCA with Pandas & SciKit-Learn

One of the packages that integrates well with NumPy is `scikit-learn` (one of the many SciPy kits[3]). This library includes efficient functionality for classification, regression, clustering, dimensionality reduction, data preprocessing, and more. If you are unfamiliar with scikit-learn, it is highly recommended you walk through their Quick Start[4].

Pandas is another library that integrates well with NumPy. It brings two additional data structures (`Series` for vectors and `DataFrame` for a table of values), as well as functions to import/export, explore/query/analyze, and visualize (with easy Matplotlib) data. If you are unfamiliar with Pandas, it is highly recommended you walk through their short introduction[5].

You will do all the work inside the `starter-3.3.ipynb` Jupyter notebook file. Make sure to complete all items flagged as `TODO`.

[3] `http://scikits.appspot.com/scikits`
[4] `http://scikit-learn.org/stable/tutorial/basic/tutorial.html`
[5] `http://pandas.pydata.org/pandas-docs/stable/10min.html`