# Machine Learning Toolbox for Feature Extraction and Inference

**Yo Sup "Joseph" Moon, Dana Modzelewski, William Chambers, Claudia Friedsam**

CS51 Final Project March 25 Draft
Due 11:59PM, March 25th, 2012.

## 1 Brief Overview

We want to implement two different methods for unsupervised learning with the goal to extract information from high dimensional data. The algorithms we chose are principal component analysis and k-means clustering. We want to implement the basic algorithms, test and compare them on a set of diverse data and investigate their performance. We plan to address the limitations we identified by extending the algorithms with straightforward solutions like e.g. heuristics or optimization methods. In a last step we try to apply our algorithms to address a real world problem like handwriting or object recognition.

## 2 Feature

1. K-means
   *Fundamental Features:*
   - K-means algorithm
   - Test functions
   - Evaluation methods to examine performance

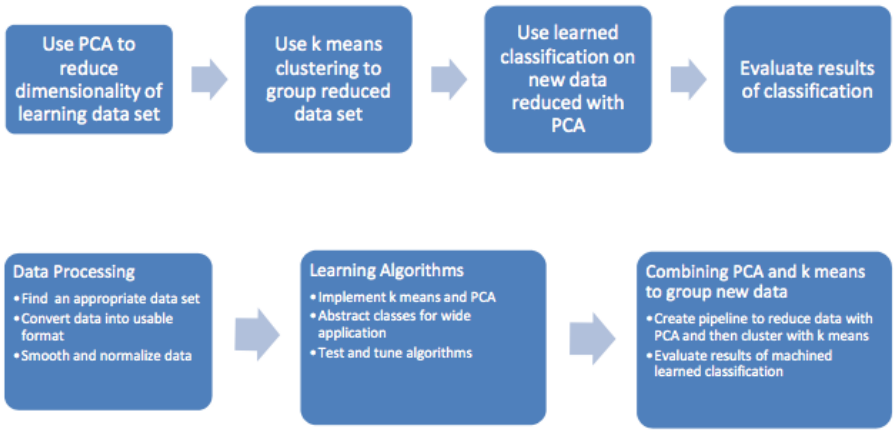
   *Possible Extensions:*
   - Heuristic or genetic algorithm for optimization
   - Explore variations of k-means, e.g. density based k- means, soft k-means
2. Principal Component Analysis (PCA)
   *Fundamental Features:*
   - PCA algorithm
   - Eigenvector Finding: can we find suitable time-efficient algorithm that suits our needs?
   - Test functions: sanity check on small-dimensional data. Test robustness of feature extraction: how much of the "useful" features of the data preserved?
3. Common Features:
   *Fundamental Features:*
   - Method to load and prepare data to be used
   - GUI to load data, pick the method, adjust input parameters, visualize results and performance for both methods
   - Testing performance of model: k-fold cross validation, logistic regression

   Possible Extensions:
   - Extension for handling a variety of different types of data
   - GUI elements for real world problems: end goal is to produce a standalone, platform-independent executable program in Windows/UNIX environment

## 3 Draft Technical Specification

We will be implementing our algorithms in Python, using vim and standard UNIX terminal environment to run our code. Our code will be shared and kept up-to-date via a Github repository. The modularity of our project arises naturally in the separate implementations of k-means and Principal Component Analysis algorithms. Both algorithms will be a modular component, and can be used for any part of the inference step in application to the data.

- K-means is a clustering method that aims at dividing data sets of n data points into k clusters which consist of similar points. This is done by assigning each data point to the cluster with the nearest mean.
- Another algorithm that we will implement is Principal Component Analysis (PCA). PCA takes a high dimensional dataset and compresses it by an orthogonal projection of the data and to the principal subspace, a lower dimensional linear space, such that the variance of the projected data is maximized. The PCA algorithm begins by finding the covariance matrix, S, of the dataset. We then find the M eigenvectors of S which correspond to the M largest eigenvalues (if the data is of dimensionality $D$, then $M < D$). The eigenvectors are the principal components onto which we project the data.




### 3.1 Read Data Module

*Functions:*

- Read file into array
- Perform data clean up (remove outliers, smoothing, etc)
- Export Data

*Exceptions:*

- Can't read/find file
- File contains ill-defined data
- Cleanup failed
- Can't write file

### 3.2 Description of K-means Module

Description of K-means algorithm

- Create initial random partition of $n$ data points in $k$ clusters

- Calculate centroids $C$ of $k$ clusters
- Initialize $C_{old}$ with empty list
- While $C \neq C_{old}$ **do**:
  - $C_{old} \leftarrow C$
  - Reassign data points to closest cluster centroid
  - Calculate $C$ for new clusters
- Return $C$ and $k$.

*K-means Functions:*

- Initialize Clusters
- Get centroids of partitioned data
- Calculate distances of points to centroids
- Reassign points to closest clusters
- Export data

*Exceptions:*

- Number of clusters too big
- Empty clusters
- Can't write file

### 3.3 K-means Evaluation Module

As a starting point we look at two criteria to investigate how well the clustering works. The first step is to look at the density of the clusters to see how compacted the clusters are. As a second criteria we will determine the silhouette, which is describing how distant the points in a specific cluster are from the other clusters to investigate the separation of the clusters.

*Functions:*

- Calculate Density
- Silhouette

### 3.4 Description of PCA Module

*PCA Functions:*

- Calculate covariance matrix, $S$
- Find $M$ largest eigenvalues of $S$
- Find eigenvectors corresponding to eigenvalues
- Project data onto eigenvectors
- Export Data

*Exceptions:*

- Can't compute covariance matrix
- Can't find eigenvalues for covariance matrix
- Can't write file

### 3.5 PCA Evaluation Module

As a preliminary criterion for determining the success of our PCA algorithm, we will look at the variances from projecting the data onto the principal components. *Functions*

- Calculate variances

### 3.6 Test Module

*Functions:*

- Test functions in k-means module
- Test functions in PCA module

## 4 What is next?

Since our project will be implemented in Python, we will need to take additional time to familiarize ourselves with the language. To do this we will use the Python manual as one of our main resources. In addition, we will also read through examples of well-written machine learning Python code such as that found at `http://norvig.com/sudoku.html`. We plan to use Vim in a standard Unix environment when writing our code and we will be sharing our code with one another using GitHub.

In order to begin working on our project it will be very important for us to identify a dataset. Without this we will not be able to test our algorithms in any meaningful way nor compare the results of one algorithm with the results of another. As a preliminary dataset, we would like to use that provided in the second assignment of CS 181, which is perfect for our needs. This is a subset of the MNIST Database, a large dataset of handwritten numbers (available at `http://yann.lecun.com/exdb/mnist/`). This dataset is ideal as it contains training, validation, and test sets which will allow us begin working with the data right away. As we get further along with our project we will want to expand our dataset as we see fit.