

Data Exploring:

Unsupervised Learning

Continuum Analytics



PCA/SVD

- Dimensionality reduction algorithm
 - Plot 100 or 1000 dimensional axis
- Linear algebra for Data Cleansing
- Unsupervised learning
 - Predict relationships
- Data Exploration

PCA/SVD Assumptions

- Linearity (covariance/change of basis)
- Variance: structurally important
- Orthogonality: Principal Components are Non-Degenerate

PCA

- Covariance matrix
 - How are features related to one another (linear)
 - Rows: all measurements of one feature
 - Columns: one measurement across all features
- Diagonalizing matrix
 - Find dimensions of greatest variance

PCA (LinAlg)

- Demean Data
- Calculate Covariance Matrix

$$C_x = \frac{1}{n} XX^T$$

- Calculate EigenValues/Vectors

$$\det(C_x - \lambda I) = 0$$

SVD

- Closely related to PCA
 - People often use the terms interchangeably
 - Another diagonalization method
- Math is a little less intuitive than PCA
- Often easier to compute

$$A = USV^T$$

- A is the input data (de-meaned)
- U, S, V
 - Rows of V are the eigenvectors
 - S^2 are eigenvalues

Numpy/Scipy Methods

- `numpy.cov`
- `numpy.linalg.eig`
- `numpy.linalg.svd`

Functions also exist in SciPy

- `scipy.linalg...`



SciKits-Learn

- Cleaner interface to PCA
- Uses `scipy.linalg.svd`
- Perform PCA via `sklearn.decomposition.PCA`
 - For large data sets use `sklearn.decomposition.RandomizedPCA`
 - `RandomizedPCA` does not center (de-mean)



Demo PCA



KMeans Clustering

- Iteratively minimize the distance between each point and a centroid to calculate center

$$J(X, C) = \sum_{i=0}^n \min_{\mu_j \in C} (||x_j - \mu_i||^2)$$

- Assign point to the closest centroid
- Recalculate position of centroid based on cluster's center of mass

$$x_c = \sum_{i=0}^n x_i$$



Demo Clustering

