Unsupervised learning: clustering algorithm

|  |
| --- |
| **K-Means Algorithm** |
| A training data    Initialize cluster centroid    Training data close to the cluster centroid will be classified as that centroid group    Take the average of points that are colored the same color, and re-group, repeat    Repeat until 2 clear cluster formed    Pseudocode    Assigning to cluster: Min((x-mu) .^ 2) *get index*  Eliminate the cluster, if no training example assigned to it |
| **Optimize K-Mean** |
| **Cost Function**  J = 1/m \* sum(||cluster group – cluster centroid|| .^ 2)  **Random initialization**  Initialize k onto k random training example  Try initializing different set of k’s to avoid local optimum, choose min(J) |
| **Choose number of clusters** |
| Choose the elbow |

Unsupervised Learning - Dimensionality Reduction

|  |
| --- |
| **Data Compression** |
|  |
| **Data visualization** |
| Summarize 50 features into <4 to visualize data |
| **Algorithm: Principal component analysis** |
| **Goal**: find a lower dimension surface/vector  onto which to project the data to minimize the distance between data and the surface (projection error)  **Projection error**      Feature normalization are needed before applying PCA |
| Algorithm - Pseudocode |
| (n\*1) \* (1\*n)  (singular value decomposition)  U = reduce  S = data retained after reconstruction sum(S(1:k, 1:k)) / sum(S(1:n, 1:n)) should > 99%    z= U.T (k \* n) \* X.T (n\*m) |
| **Reconstruct compressed data** |
| 🡪 |
| **Choosing the number of principal components (choose k)** |
| Such that 99% of the data is retained |
| **Use PCA to speed up learning algorithm** |
| PCA only apply to training set  Train ML with x🡪z data, get theta with size k, then when prediction test set, map x(test) 🡪 z(test) and predict  Do not use PCA for avoiding overfitting  Use PCA for data visualization, less space, speed up ML |