Clustering

CS/DSA 5703: Machine Learning Practice

Unsupervised Learning

- Building models that capture the distribution of samples in some high-dimensional space
- So far, we have focused on projecting these feature vectors into some lower-dimensional space
  - Non-linear case: attempted to translate the non-linear manifolds into linear ones
- Useful for better understanding the underlying data & as a basis for preprocessing data

Clustering

- Fundamental idea: we want to infer which samples are similar enough to be considered the same as one-another
- Like classification, this allows us to assign a discrete label to each of our samples
- But: the clustering algorithm determines the labels automatically

Clustering

Clustering algorithms/hyper-parameter choices vary:

- What do we mean by ***similar***?
  - How do we measure distance / similarity?
- How similar do two things need to be so that they are considered to be in the same cluster (class)?
- Is the number of clusters fixed or variable?

Clustering

A couple perspectives:

- Represents a form of dimensionality reduction: translating some N-dimensional feature vector into a single enumerated value
- In the simplest cases, we are identifying zero-dimensional manifolds (blobs) in the feature space
  - More advanced methods look at more interesting manifolds

Clustering

- K-Means:
  - Euclidean distance
  - Fixed number of clusters
  - Each cluster effectively has the same shape
- Mixture Models:
  - Use a probability density function as a similarity metric
  - Fixed number of clusters
  - When the PDF includes covariance, then we can handle interesting (local) manifold shapes

K-Means Clustering

- Predefine the number of clusters (K)
- Each cluster is parameterized by its center location in the N-dimensional feature space
- Initialize the centers of each cluster in some way
- Repeat:
  – Measure the distance (or similarity) between each sample and each cluster center
  – Assign membership of each sample to a cluster
    - Membership can be hard or soft
  – Update the cluster centers to reflect the member samples

Initialization

Variety of initialization options for the cluster centers:

- Distribution-based: pick centers randomly from within the feature space
  - Uniform sampling
  - Construct a Gaussian distribution over the training set and sample from this distribution
- Sample-based: pick K samples uniformly from the training set

Hard-Boundary Classification

- Each sample is assigned to the cluster that is closest to it
- Even if it is far away from the cluster center…

,

Soft-Boundary K-Means Clustering

CS/DSA 5703: Machine Learning Practice

Soft K-Means Clustering

Hard boundaries:

- Label is all-or-nothing
- For cluster mean updates: samples near the boundary are just as important as samples far away from the boundary
  - Though, we may be less sure about their "true flavor"
- Easy for the learning algorithm to get into a cycle where a repeatedly pops from one side of the boundary to another

Soft Bounary K-Means Clustering

- Model each sample as probabilistically belonging to each class
  - Probabilistic labels!
- Each sample then contributes to the cluster mean proportionally to this probability

Soft Boundary K-Means Clustering

- A sample near the boundary between two clusters contributes to both cluster means
  - The balance does not change very much as the sample crosses the boundary
- More stable learning
- Hyper-parameter: beta
  - Small: all classes have interesting probabilities for a given sample
  - Large: one class gets most of the probability

Example 1: K-Means Clustering

CS/DSA 5703: Machine Learning Practice

,

Example 2: K-Means Clustering

CS/DSA 5703: Machine Learning Practice

Example 2: K-Means Clustering

Arrow data set:

- Different parts of the feature space have very differently shaped manifolds
- Variation in the dimensionality (1 vs 2)
- Variation in the sparsity

,

Multi-Dimensional Gaussian Probability Density Functions

CS/DSA 5703: Machine Learning Practice

Representing Clusters

- K-Means Models: similarity metric is spherical:
  - All feature dimensions are treated in the same way
  - No acknowledgement of covariance of features

- What we want:
  - Cluster shapes that acknowledge local manifold structure
  - Some features may vary more than others
  - Some features may covary with others

Multi-Dimensional Gaussian Probability Density Functions

- Density function: given a sample in an N-dimensional space, what is the likelihood of this sample?
- Point in question is N-dimensional
- Output is still a scalar (it is a likelihood)
- We can explicitly capture:
  - Different variances for the different features
  - Covariance across features

,

Mixture Distributions

CS/DSA 5703: Machine Learning Practice

Mixture Distributions

- Gaussian distributions:
  - All samples are centered around a single mean
  - Likelihood of observing samples drops as we move away from the mean
  - Allow us to represent a single cluster of samples
- But:
  - Many data sets have distinct clusters
  - Gaussian distribution does not capture these situations well

,

Learning Gaussian Mixture Distributions

CS/DSA 5703: Machine Learning Practice

Learning Gaussian Mixture Distributions

- Given a data set, we need to estimate:
  - Means for all K clusters
  - Covariance matrices for all K clusters
  - Weights
- There is no closed form solution
- But, like soft boundary K-means, we can take an iterative approach

Learning Gaussian Mixture Distributions

Algorithm outline:

1. Guess at the mixture model parameters
2. Probabilistically assign samples to each cluster
3. Re-estimate the mixture model parameters given the sample assignment
4. Repeat starting with #2

,

Expectation Maximization Learning Example

CS/DSA 5703: Machine Learning Practice

,

Example 1: EM and Mixture Models

CS/DSA 5703: Machine Learning Practice

Example 1: EM and Mixture Models

Five-cluster data set…

▪

Example 2: EM and Mixture Models

CS/DSA 5703: Machine Learning Practice

Example 2: EM and Mixture Models

Arrow data set

Clustering Wrap-Up

CS/DSA 5703: Machine Learning Practice

Clustering Notes

- Soft Boundary & Mixture Model approaches: use probability density functions to describe the clusters
- Soft Boundary K-Means: models cluster location
  - Circular clusters
- Mixture Model: add scaling and covariance
  - Ellipsoidal clusters

Clustering Notes

- Both methods are iterative in nature
  - Lots of local maxima in our likelihood space
- Final solution depends on what our initial guess is
  - Quality of the final solutions can also vary a lot!
- Typical approach: perform the learning process multiple times and keep the best one

Clustering Notes

Hyper-parameters:

- How to make the initial guesses
- Soft-boundary: beta
- Gaussian mixture model: estimates its own shapes
- All require us to specify the number of clusters ahead of time

Clustering Notes

Picking the number of clusters:

- We typically try multiple values
- Regularized cost function:
  - Want to maximize the likelihood of our learned model generating our training data
  - Want to also minimize the number of model parameters that we use (so, keep K small!)
  - Common scorer choices: Bayesian Information Criterion (BIC) or the Akaike Information Criterion (AIC)
    - GaussianMixture class provides these!

Mixture Models

- We can move beyond Gaussian distributions (any PDF can be used!)
- Allows us to better match the manifold shapes of our data set
- Can also work in other metric spaces!
  - For example: PDFs describing 3D orientations