Chapter 9: Unsupervised learning

- Background
  - Unsupervised learning is the cake
  - Clustering: group similar instances
  - Anomaly detection: what does normal data look like? What is not normal
  - Density estimation: estimating probability density function of random process that generated data
- Clustering
  - Background:
    - Applications:
      - Customer segmentation
      - Data analysis
      - Dimensionality reduction technique: cluster affinities
      - Anomaly detection: low affinity
      - Semi-supervised learning: few labels → propagate labels to all instances in same cluster
      - Search engines
      - Image segmentation
    - No universal definition of what a cluster is:
      - Centroids
      - Continuous regions
      - Hierarchical
  - K-means
    - Label is index instance is assigned to, not like classification label
    - Algo does not behave well when blobs have different diameters: only cares about distance to centroid when assigning instance to cluster
    - Hard vs. soft clustering
      - Hard: assign each instance to single cluster
      - Soft: give each instance a score per cluster: distance, similarity (rbf)
    - Algorithm
      - Randomly choose k centers, label instances, update centroids
      - Guaranteed to converge, but may converge only to local optimum
    - Centroid initialization methods
      - Can set init hyperparameter if know roughly where centroids should be
      - Run algo many times with different random initializations: n_init
      - Inertia: performance metric used to judge: mean squared distance between each instance and closest centroid
      - Kmeans ++: default for sklearn Kmeans
        - Selects more distant centroids, suboptimal solutions less likely, additional computation in initialization more than compensates for # of times algo needs to run
        - Take centroid chosen at random

            - Take new centroid choosing instance with probability that is sum squared distances between instance and already chosen centroids
            - Repeat until all k centroids chosen
            - Ensures instances farther away from already chosen centroids are more likely to be centroids
        - Accelerated K-means and mini-batch K-means
            - Accelerated uses triangle inequality: straight line is shortest distance btwn two points
                - Keeps track of lower and upper bounds of distances btwn instances and centroids
                - Default in sklearn
            - Mini-batch
                - If dataset doesn't fit in memory use memmap or pass one mini-batch as time to partial_fit()
                - Faster than regular, but inertia worse, esp. as clusters increase
        - Finding optimal number of clusters
            - Choosing lowest inertia not good performance measure: keeps going lower
            - Elbow method: inflection point, but coarse
            - Silhouette score: mean of silhouette coefficient over all instances
                - Silhouette coefficient: (b-a)/max(a,b)
                    - b: mean nearest cluster distance → mean distance to instances of next closest cluster
                    - a: mean distance to other instances in same cluster
                    - Can vary between -1 and 1: 1 is well inside cluster, 0 close to cluster boundary, -1 assigned to wrong cluster
            - Silhouette diagram: one knife shape per cluster
                - Shape's height: number of instances in cluster
                - Width: coefficients in cluster
                - Dashed line: mean silhouette coefficient
                - Trade-off even sized clusters for higher silhouette coefficients
    - Limits of K-means
        - Must run algo several times, need to specify clusters
        - Poor behavior if cluster size varies, density differs, or shapes vary
        - Important scale features before run k-means
    - Using clustering for image segmentation
        - Image segmentation: Partitioning an image into multiple segments

        - Semantic segmentation: all pixels part of same object type get assigned to same segment
        - Instance segmentation: all pixels part of same individual object assigned to same segment
        - Color segmentation
    - Using clustering for preprocessing
        - Efficient approach to dimensionality reduction
    - Using clustering for semi-supervised learning
        - Plenty of unlabeled and few labeled instances
        - Label propagation: label all other instances as the same in same cluster
            - Problem with mislabeling if instances close to cluster boundaries
            - Propagate to x% of instances closest to centroid
        - Active learning
            - Human interacting with algo
                - Train on labeled instances
                - Give most uncertain instances to expert to label
                - Iterate until improvement stops being worth it
    - DBSCAN
        - Defines clusters as continuous regions of high density
        - Algo counts how many instances located within small distance epsilon: called epsilon-neighborhood
        - If instance has at least min_samples instances in its epsilon-neighborhood, then core instance; core instances in dense regions
        - All instances in neighborhood of core instance belong to same cluster. May include other core instances, ergo, long sequence of neighboring core instances forms a single cluster
        - Any instance not a core, does not have one in its neighborhood is an anomaly
        - Works well if all clusters are dense enough and well separated by low-density regions
    - Other clustering algorithms
        - Agglomerative clustering: built from bottom-up,
        - Birch: good for large datasets, assuming features < 20,
        - Mean-shift: circle centered on each instance, then computes mean of all instances within circle
        - Affinity propagation: Voting system
        - Spectral clustering: takes similarity matrix btwn instances, creates low-d embedding it, then uses another clustering; doesn't scale well or behave well if clusters have different sizes
- Gaussian Mixtures
    - Background: probabilistic model that assumes instances generate from mixture of several Gaussian distributions of unknown parameters
    - Generating process:

    - For each instance, cluster picked randomly. Probability of choosing any particular cluster (j) defined by cluster weight (phi). Index of cluster for instance called z
    - If z index equals jth cluster (ith instance assigned to jth cluster) location of $x^{(i)}$ is sampled randomly from Gaussian distribution
  - Expectations Maximization algorithm estimates
    - Similar to k-means
    - Initializes cluster parameters randomly
    - Assigns instances to clusters (expectation)
    - Updates clusters (maximization)
    - Generalization of k-means
      - Finds cluster centers mu and size, shape and, orientation sigma as well as relative weights
      - Uses soft cluster assignments
    - Each step estimates probability instance belongs to a cluster
    - At end each cluster updated using all instances: each instance weighted by estimated probability it belongs to that cluster
    - Just like k-means may converge on poor solution, so needs to run several times
    - Once have estimate of location, size, shape, orientation, and relative weight of each cluster can assign each instance to most likely cluster or estimate probability it belongs to a cluster
  - Generative model, so can sample new instances
  - Higher dimensions, many clusters, few instances EM may have difficulty converging to optimal solution:
      - Can impose constraints on covariance matrix to reduce difficult of task by limiting range of shapes and orientations of clusters
  - Anomaly detection using Gaussian mixtures
    - Any instance located in low-density region considered an anomaly
    - Set density threshold: if too many false positives, lower threshold, too many false negatives raise it
    - Novelty detection: algo assumed to be trained on clean dataset
    - GMM try to fit all data → too many outliers will bias model's view of normality.
      - Solution: fit model once, remove most extreme outliers, fit on cleaned dataset
  - Selecting the number of clusters
    - K-means uses silhouette score or inertia to select
    - Metrics not reliable on non-spherical or different sized clusters
    - Use model that minimizes theoretical information criterion
      - m: instances, p: parameters, L-hat: maximized value of likelihood
      - Akaike: 2p – 2log(L-hat)
      - Bayesian: log(m)p – 2log(L-hat)
    - AIC and BIC penalize more parameters, tend to pick same model

- Bayesian Gaussian Mixture models
    - Rather than manually search for number of clusters, remove clusters that are likely to be unnecessary
    - Requires some knowledge of data
    - Set n_components to number greater that what believe is optimal
    - Cluster parameters not fixed, but random like cluster assignments
    - Beta distribution used to model random variables in a fixed range
    - Stick Breaking Process: determines which instances assigned to which cluster
    - Wishart Distribution: used to sample covariance matrices and control shape
    - Set prior through weight_concentraion_prior
    - Bayes Theorem: $p(z|X) = p(X|z)p(z)/p(X)$
        - $p(X)$ intractable for GMM since requires considering all possible values of z, which requires all possible combos of cluster parameters and assignments
    - Variational inference picks family of distributions with own variational parameters, optimizes parameters to approximate $p(z|X)$
- GMM work great on ellipsoidal shaped clusters
    - Try it on two moon data set: get 8 clusters instead of 2!
- Other algorithms for anomaly and novelty detection
    - PCA
    - Fast-MCD (minimum covariance determinant)
    - Isolation Forest
    - Local Outlier Factor
    - One-class SVM