Big Picture for Today

Quick coverage of kNN

↳ recap from early in semester (distance/similarity based classification)

Clustering

- hierarchical
- k-means
- mixture models / EM

Nearest Neighbor Classification

- idea: similar/close items should are likely to have the same label
- basically a "likeness" lookup table
  - ↳ for new example X find closest and choose its class

K-nearest neighbor

- find k closest points to X
- choose class with most votes

OR

choose class with most more than L

↳ otherwise say unknown

Practical Considerations

- need distance metric
  - Euclidean distance

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_k \left(x_i^k - x_j^k\right)} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$$

  - Manhattan distance

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_k \left|x_i^k - x_j^k\right| = \|\mathbf{x}_i - \mathbf{x}_j\|_1$$

  - Hamming Distance

    sum over all features:

    0 if feature values are same

    1 otherwise

    (categorical)

- Different scales typically don't play well with distance

Ex: clustering houses

features = price
↳ ≃ 100,000
↳ dominates distance

beds
↳ ≈ 1

baths
↳ ≈ 1

Solution: normalize/rescale

- Different scales and variances → Whitening
transform to unit mean and identity covariance

Compute

$$z_i = \Lambda^{(-1/2)} U^T (x_i - \text{mean}(X))$$

where U = eigenvectors of covmat (X)

Λ = diagonal matrix of eigenvalues of covmat (X)

- Costly to find nearest neighbor for lots of points in high dimensions
  - ↳ Solution: approximate nearest neighbor