k-Nearest Neighbours

General Overview & Formulas

Lecture 5: Part 1

Marina Santini

Acknowledgements
Slides borrowed from Joakim Nivre, 2013

Lecture 5: Required Reading

- Handout
- Daume' III (2015: 26-32, excl. 2.4)
- Witten et al. (2011:131-138)

Outline

- Key components
- A geometric view of learning
- Eager and lazy learning
- The k parameter
- Distance metrics

Nearest Neighbor Classification

- An old idea

  This "rule of nearest neighbor" has considerable elementary intuitive appeal and probably corresponds to practice in many situations. For example, it is possible that much medical diagnosis is influenced by the doctor's ***recollection*** *of the subsequent history of an earlier patient whose symptoms* ***resemble*** *in some way those of the current patient.* (Fix and Hodges, 1952)

- Key components:
  - Storage of old instances
  - Similarity-based reasoning to new instances

k-Nearest Neighbour

- Learning:
  - Store training instances in memory
- Classification:
  - Given new test instance **x**,
    - Compare it to all stored instances
    - Compute a distance between **x** and each stored instance $\mathbf{x}^t$
    - Keep track of the k closest instances (nearest neighbors)
  - Assign to **x** the majority class of the k nearest neighbours
- A geometric view of learning
  - Proximity in (feature) space → same class
  - The smoothness assumption

Eager and Lazy Learning

- Eager learning (e.g., decision trees)
  - Learning – induce an abstract model from data
  - Classification – apply model to new data
- Lazy learning (a.k.a. memory-based learning)
  - Learning – store data in memory
  - Classification – compare new data to data in memory
  - Properties:
    - Retains all the information in the training set – no abstraction
    - Complex hypothesis space – suitable for natural language?
    - Main drawback – classification can be very inefficient

Dimensions of a k-NN Classifier

- Distance metric
  - How do we measure distance between instances?
  - Determines the layout of the instance space
- The k parameter
  - How large neighborhood should we consider?
  - Determines the complexity of the hypothesis space

Distance Metric 1

- Overlap = count of mismatching features

$$\Delta(\mathbf{x},\mathbf{z}) = \sum_{i=1}^{m} \delta(x_i, z_i)$$

$$\delta(x_i, z_i) = \begin{cases} \left| \dfrac{x_i - z_i}{\max_i - \min_i} \right| & if\, numeric, else \\ 0 & if\, x_i = z_i \\ 1 & if\, x_i \neq z_i \end{cases}$$

Distance Metric 2

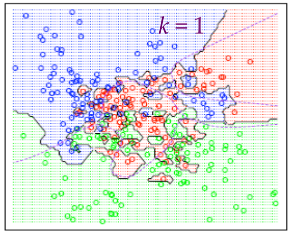
- MVDM = Modified Value Difference Metric

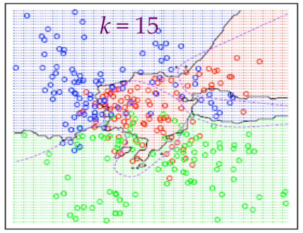
$$\Delta(\mathbf{x},\mathbf{z}) = \sum_{i=1}^{m} \delta(x_i, z_i)$$

$$\delta(x_i, z_i) = \sum_{j=1}^{K} \left| P(C_j \mid x_i) - P(C_j \mid z_i) \right|$$

The k parameter

- Tunes the complexity of the hypothesis space
  - If k = 1, every instance has its own neighborhood
  - If k = N, all the feature space is one neighborhood





$$\hat{E} = E(h \mid \mathcal{V}) = \sum_{t=1}^{M} 1\left(h\left(\mathbf{x}^t\right) \neq r^t\right)$$

A Simple Example

Training set:

1. (a, b, a, c) → A
2. (a, b, c, a) → B
3. (b, a, c, c) → C
4. (c, a, b, c) → A

New instance:

5. (a, b, b, a)

Distances (overlap):

- Δ(1, 5) = 2
- Δ(2, 5) = 1
- Δ(3, 5) = 4
- Δ(4, 5) = 3

k-NN classification:

- 1-NN(5) = B
- 2-NN(5) = A/B
- 3-NN(5) = A
- 4-NN(5) = A

$$\Delta(\mathbf{x},\mathbf{z}) = \sum_{i=1}^{m} \delta(x_i, z_i)$$

$$\delta(x_i, z_i) = \begin{cases} \left|\dfrac{x_i - z_i}{\max_i - \min_i}\right| & if\, numeric, else \\ 0 & if\, x_i = z_i \\ 1 & if\, x_i \neq z_i \end{cases}$$

Further Variations on k-NN

- Feature weights:
  - The overlap metric gives all features equal weight
  - Features can be weighted by IG or GR
- Weighted voting:
  - The normal decision rule gives all neighbors equal weight
  - Instances can be weighted by (inverse) distance

Properties of *k*-NN

- Nearest neighbor classification is appropriate when:
  - Features can be both categorical and numeric
  - Disjunctive descriptions may be required
  - Training data may be noisy (missing values, incorrect labels)
  - Fast classification is not crucial
- **Inductive bias of *k*-NN:**
  1. **Nearby instances should have the same label (smoothness assumption)**
  2. **All features are equally important (without feature weights)**
  3. **Complexity tuned by the *k* parameter**

The end