Classification: K-Nearest Neighbors - Instance Based Learning

Lifu Huang

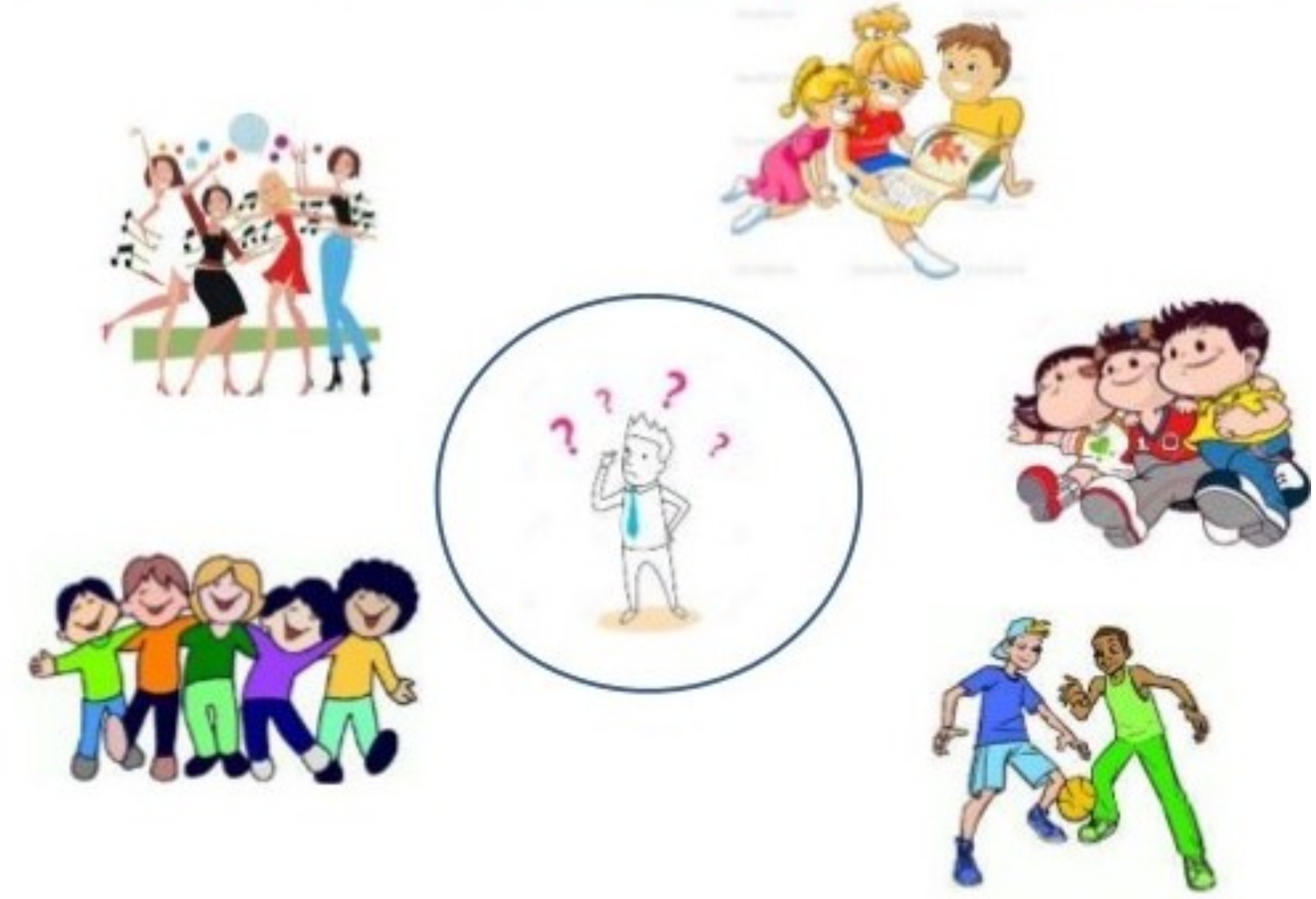
Computer Science, Virginia Tech

Feb. 9, 2021

Slides adapted from Luke Zettlemoyer, David Sontag, Bert Huang

Simple Analogy

Tell me about your friends(*who your neighbors are*) and *I will tell you who you are*.



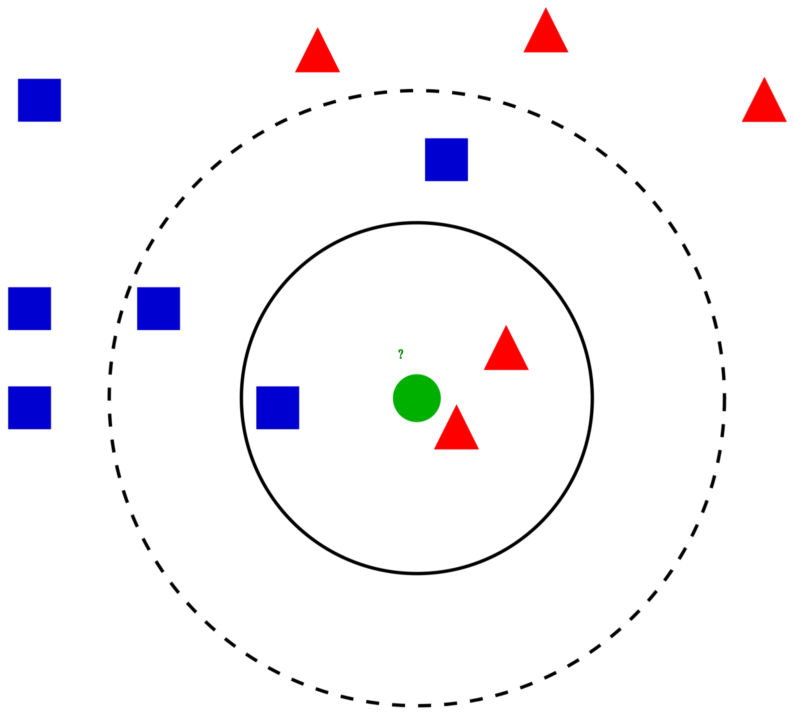
The closer you are, the more characteristics you share

Instance based learning – K-Nearest Neighbors (KNN)

- Idea:
  - Similar examples have similar label.
  - Classify new examples like similar training examples.
- Algorithm:
  - Given some new example x for which we need to predict its class y
  - Find most similar training examples
  - Classify *x* “like” these most similar examples
- Questions:
  - How to determine similarity?
  - How many similar training examples to consider?
  - How to resolve inconsistencies among the training examples?

What is KNN?

- A powerful classification algorithm used in pattern recognition
- One of the top data mining algorithms used today
- A non-parametric lazy learning algorithm
- An object (a new instance) is classified based on majority votes for its neighbor classes
- The object is assigned to the most common class amongst its K nearest neighbors



Distance (Similarity) Metric

- Given a data instance with p features

$$x^i = (x_1^i, x_2^i, \ldots, x_p^i)$$

- Most common distance metric is Euclidean Distance

$$d_E(x^i, x^j) = \left(\sum_{k=1}^{p} (x_k^i - x_k^j)^2 \right)^{\frac{1}{2}}$$

- Euclidean Distance makes sense when different data instances are with the same feature attributes
  - e.g., length and weight are not comparable

Distance Metrics

Minkowsky:
$$D(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{m} |x_i - y_i|^r\right)^{1/r}$$

Euclidean:
$$D(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$$

Manhattan / city-block:
$$D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{m} |x_i - y_i|$$

Camberra:
$$D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{m} \frac{|x_i - y_i|}{|x_i + y_i|}$$

Chebychev:
$$D(\boldsymbol{x},\boldsymbol{y}) = \max_{i=1}^{m} |x_i - y_i|$$

Quadratic:
$$D(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}-\boldsymbol{y})^T Q(\boldsymbol{x}-\boldsymbol{y}) = \sum_{j=1}^{m}\left(\sum_{i=1}^{m}(x_i - y_i)q_{ji}\right)(x_j - y_j)$$

Q is a problem-specific positive definite $m \times m$ weight matrix

Mahalanobis:
$$D(\boldsymbol{x},\boldsymbol{y}) = [\det V]^{1/m}(\boldsymbol{x}-\boldsymbol{y})^{\mathrm{T}} V^{-1}(\boldsymbol{x}-\boldsymbol{y})$$

V is the covariance matrix of $A_1..A_m$, and A_j is the vector of values for attribute j occuring in the training set instances $1..n$.

Correlation:
$$D(\boldsymbol{x},\boldsymbol{y}) = \frac{\sum_{i=1}^{m}(x_i - \overline{x_i})(y_i - \overline{y_i})}{\sqrt{\sum_{i=1}^{m}(x_i - \overline{x_i})^2 \sum_{i=1}^{m}(y_i - \overline{y_i})^2}}$$

$\overline{x_i} = \overline{y_i}$ and is the average value for attribute i occuring in the training set.

Chi-square:
$$D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{m} \frac{1}{sum_i}\left(\frac{x_i}{size_x} - \frac{y_i}{size_y}\right)^2$$

sum_i is the sum of all values for attribute i occuring in the training set, and $size_x$ is the sum of all values in the vector $\boldsymbol{x}$.

Kendall's Rank Correlation:
$$D(\boldsymbol{x},\boldsymbol{y}) = 1 - \frac{2}{n(n-1)} \sum_{i=1}^{m}\sum_{j=1}^{i-1} \mathrm{sign}(x_i - x_j)\,\mathrm{sign}(y_i - y_j)$$

sign(x)=-1, 0 or 1 if $x < 0$, $x = 0$, or $x > 0$, respectively.

Figure 1. Equations of selected distance functions.
($\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors of m attribute values).

KNN Example

	Food (3)	Chat (2)	Fast (2)	Price (3)	Bar (2)	BigTip
1	great	yes	yes	normal	no	yes
2	great	no	yes	normal	no	yes
3	mediocre	yes	no	high	no	no
4	great	yes	yes	normal	yes	yes

Similarity metric: Number of matching attributes (k=2)

- New examples:
  - Example 1 (great, no, no, normal, no)
  - Example 2 (mediocre, yes, no, normal, no)

KNN Example

	Food (3)	Chat (2)	Fast (2)	Price (3)	Bar (2)	BigTip
1	great	yes	yes	normal	no	yes
2	great	no	yes	normal	no	yes
3	mediocre	yes	no	high	no	no
4	great	yes	yes	normal	yes	yes

Similarity metric: Number of matching attributes (k=2)

•New examples:

- Example 1 (great, no, no, normal, no) Yes

  → most similar: number 2 (1 mismatch, 4 match) → yes

  →Second most similar example: number 1 (2 mismatch, 3 match) → yes

- Example 2 (mediocre, yes, no, normal, no)

KNN Example

	Food (3)	Chat (2)	Fast (2)	Price (3)	Bar (2)	BigTip
1	great	yes	yes	normal	no	yes
2	great	no	yes	normal	no	yes
3	mediocre	yes	no	high	no	no
4	great	yes	yes	normal	yes	yes

Similarity metric: Number of matching attributes (k=2)

- New examples:
  - Example 1 (great, no, no, normal, no) Yes

    → most similar: number 2 (1 mismatch, 4 match) → yes

    →Second most similar example: number 1 (2 mismatch, 3 match) → yes

  - Example 2 (mediocre, yes, no, normal, no) Yes/No

    → Most similar: number 3 (1 mismatch, 4 match) → no

    →Second most similar example: number 1 (2 mismatch, 3 match) → yes

Selecting the number of neighbors

- Increase k:
  - Makes KNN less sensitive to noise

- Decrease k:
  - Allows capturing finer structure of space

- ➔Pick k not too large, but not too small (depends on data)

KNN Feature Weighting and Normalization

- Feature Weighting
  - Scale each feature by its important for classification

$$D(a,b) = \sqrt{\sum_k w_k (a_k - b_k)^2}$$

  - Can use our prior knowledge about which features are more important
  - Can learn the weights using cross-validation
- Feature Normalization
  - Distance between neighbors could be dominated by some attributes with large values
    - e.g., income of customers
  - Normalize features: mapping values to 0-1

$$a_i = \frac{v_i - \min(v_i')}{\max(v_i') - \min(v_i')}$$

Advantages and Disadvantages of KNN

- Advantages
  - Very simple and intuitive
  - Requires little tunning
  - Often performs quite well!
- Disadvantages
  - Takes more time to classify a new example
    - Need to calculate and compare distance from new example to all other examples
  - Choosing K may be tricky
  - Need large number of samples to achieve good performance
  - Easily fooled by irrelevant attributes