Applied Machine Learning

Lecture 9
K-Nearest Neighbors (K-NN)

Ekarat Rattagan, Ph.D.

K-Nearest Neighbor Algorithm

- Assumption: Similar Inputs have similar outputs
- To classify a new input vector x, examine the k-closet training data points to x and assign the object to the most frequently occurring class

kNN Algorithm

0. Look at the data

Say you want to classify the grey point into a class. Here, there are three potential classes - lime green, green and orange.

1. Calculate distances

Start by calculating the distances between the grey point and all other points.

2. Find neighbours

Next, find the nearest neighbours by ranking points by increasing distance. The nearest neighbours (NNs) of the grey point are the ones closest in dataspace.

3. Vote on labels

Vote on the predicted class labels based on the classes of the k nearest neighbours. Here, the labels were predicted based on the k=3 nearest neighbours.

Diagram source: Cambridge Coding

K-NN

Euclidian Distance

$$d(i,j) = \sqrt{\sum_{k=1}^{n} (x_{i,k} - x_{j,k})^2}$$

Manhattan Distance

$$d(i,j) = \sum_{k=1}^{n} |x_{i,k} - x_{j,k}|$$

Minkowski Distance

$$d(i,j) = \left(\sum_{k=1}^{n} |x_{i,k} - x_{j,k}|^q \right)^{1/q}$$

$q=1$

$q=2$

$q=3$

Instance-based Learning

θ $J(\theta_1, \theta_2, \ldots \theta)$

- Non-parametric models
  KNN Classifier KNN Regressor
- Works for classification or regression problems
- Learning amounts to simply storing training data
- Instance-based methods are sometimes referred to as **lazy learning** methods because they delay processing until a new instance must be classified.

Euclidean

KNN Example

	Food (3)	Chat (2)	Fast (2)	Price (3)	Bar (2)	BigTip
1	great 1	yes 0	yes 0	normal 1	no 1	yes
2	great 1	no 1	yes 0	normal 1	no 1	yes
3	mediocre 0	yes 0	no 1	high 0	no 1	no
4	great 1	yes 0	yes 0	normal 1	yes 0	yes

K=2

3
4
2
2

Hamming dist

Ex1 5 great no no normal no

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

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

1 Yes
2 yes
5

KNN Example

	Food (3)	Chat (2)	Fast (2)	Price (3)	Bar (2)	BigTip
1	great 0	yes 1	yes 0	normal 1	no 1	yes
2	great	no	yes	normal	no	yes
3	mediocre 1	yes 1	no 1	high 0	no 1	no
4	great	yes	yes	normal	yes	yes

3

4

K=2

Ex. 5 M y n no n

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

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

tie

- Example 2 (mediocre, yes, no, normal, no) Yes/No ถ้าเท่า
  - → Most similar: number 3 (1 mismatch, 4 match) → no
  - →Second most similar example: number 1 (2 mismatch, 3 match) → yes

k=6
k=5
k=4

Ties

k=3

binary classification

- ties may occur in a classification problem when $K > 1$ & k is even number
  - for binary classification: choose K odd to avoid ties (วิธีแก้)
  - for multi-class classification: (วิธีแก้ไข)
    - decrease the value of K until the tie is broken
    - if that doesn't work, use the class given by a 1NN classifier

Credit: Mengye Ren

K-Nearest Neighbors

$K = 2 \rightarrow$ ties
$K = 3$

How do we choose k?

10

- Larger k may lead to better performance
- But if we set k too large we may end up looking at samples that are not neighbors (are far away from the query) outliers หมายเป็นเป็นต้น ถ้า k too large
- We can use cross-validation to find k
- Rule of thumb is $k < sqrt(n)$, where n is the number of training examples

$k < \sqrt{10}$ $k < 3 \approx$

[Slide credit: O. Veksler]

K-NN Decision Boundary

overfit

good fit

Credit: https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

Issues and Remedies (1/2)

- If some attributes (coordinates of **x**) have larger ranges, they are treated as more important

  x_1 x_2 x_3
  10000 30 3

  - normalize scale
    - Simple option: Linearly scale the range of each feature to be, e.g., in range [0,1]

      uniformly dist
    - Linearly scale each dimension to have 0 mean and variance 1 (compute mean μ and variance σ^2 for an attribute x_j and scale: $(x_j - m)/\sigma$)

      normal z-score

$\frac{x - \mu}{\sigma}$

$\frac{x - m}{\sigma}$

- Irrelevant, correlated attributes add noise to distance measure
  - eliminate some attributes

Credit: Zemel, Urtasun, Fidler (UofT)

Issues and Remedies (2/2)

bis 0

- Expensive at test time: To find one nearest neighbor of a query point **x**, we must compute the distance to all N training examples. Complexity: $O(kdN)$ for kNN

  feature selection x_1 x_2 x_3 x_4

  $k = 2, 5, 100$

  - Use subset of dimensions
  - Pre-sort training examples into fast data structures (e.g., kd-trees)

    $2 \times 5 \times 100$

    kd-trees

- Storage Requirements: Must store all training data
  - Remove redundant data (e.g., condensing)

- High Dimensional Data: "Curse of Dimensionality"

  "As the number of features or dimensions grows, the amount of data we need to generalize accurately grows exponentially." Charles Isbell

[Slide credit: David Claus]

Example: Digit Classification

- Decent performance when lots of data

- Yann LeCunn – MNIST Digit Recognition
  - Handwritten digits
  - 28x28 pixel images: $d = 784$
  - 60,000 training samples
  - 10,000 test samples
- Nearest neighbour is competitive

	Test Error Rate (%)
Linear classifier (1-layer NN)	12.0
K-nearest-neighbors, Euclidean	5.0
K-nearest-neighbors, Euclidean, deskewed	2.4
K-NN, Tangent Distance, 16x16	1.1
K-NN, shape context matching	0.67
1000 RBF + linear classifier	3.6
SVM deg 4 polynomial	1.1
2-layer NN, 300 hidden units	4.7
2-layer NN, 300 HU, [deskewing]	1.6
LeNet-5, [distortions]	0.8
Boosted LeNet-4, [distortions]	0.7

https://www.kaggle.com/cdeotte/mnist-perfect-100-using-knn

Example: Where on Earth is this Photo from?

- Problem: Where (eg, which country or GPS location) was this picture taken?
  - Get 6M images from Flickr with gps info (dense sampling across world)
  - Represent each image with meaningful features
  - Do kNN (large k better, they use $k = 120$)!

kNN

k=120

[Paper: James Hays, Alexei A. Efros. im2gps: estimating geographic information from a single image. CVPR'08. Project page: http://graphics.cs.cmu.edu/projects/im2gps/]

Example: Use of KNN for the Netflix Prize

https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-1-knn-item-based-collaborative-filtering-637969614ea

http://cs229.stanford.edu/proj2006/HongTsamis-KNNForNetflix.pdf