Northeastern University
College of Engineering

IE6600

k – *Nearest Neighbors*

Zhenyuan Lu

1. Introduction

Introduction

Can you imagine how a diner experiences the unseen food?

Upon first bite, the senses are overwhelmed. What are the dominant flavors?

Does the food taste savory or sweet?

Does it taste similar to something eaten previously?

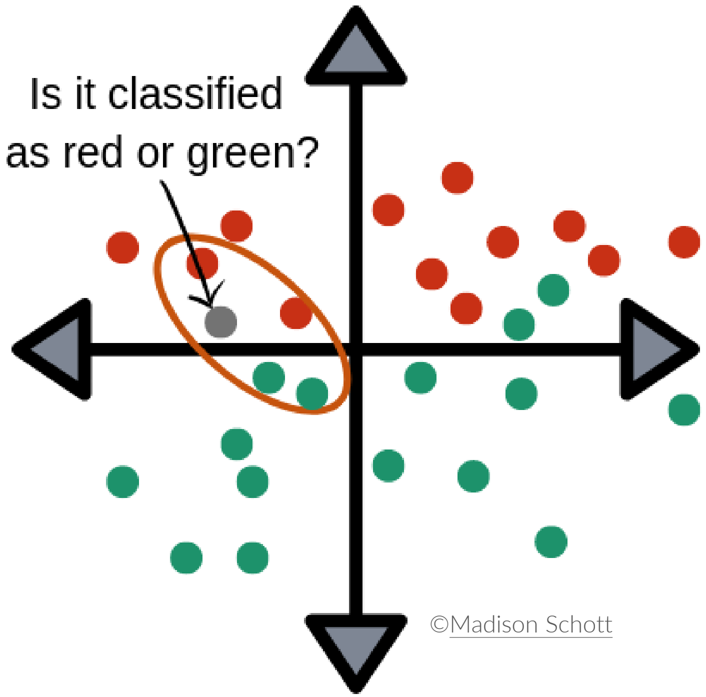
Personally, I imagine this process of discovery in terms of a slightly modified adage:

“If it walks like a duck, quacks like a duck, looks like a duck, and tastes like a duck, then it’s probably a duck.”

Introduction

kNN is a model that classifies data points based on the points that are most similar to it. It uses test data to make an "educated guess" on what an unclassified point should be classified as.

Introduction

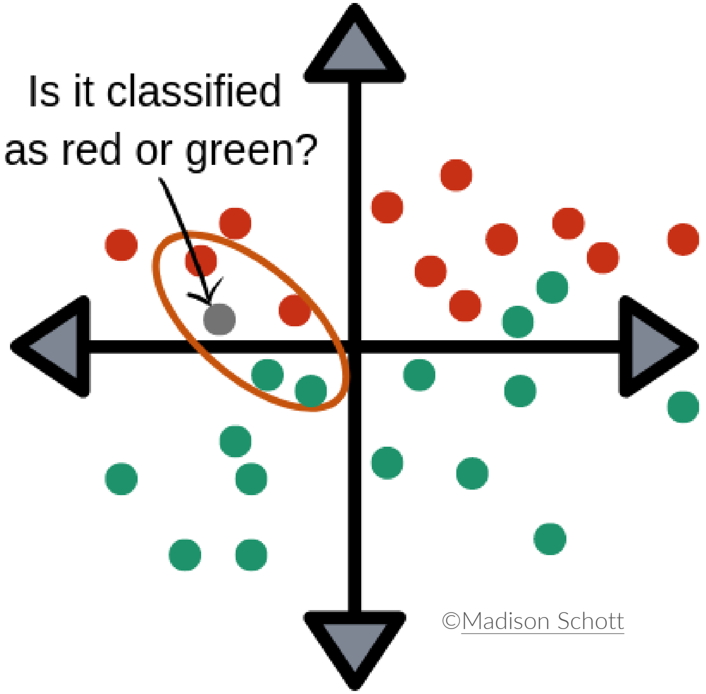


kNN is a model that classifies data points based on the points that are most similar to it. It uses test data to make an "educated guess" on what an unclassified point should be classified as.

Introduction

kNN is an algorithm that is considered both non-parametric and an example of lazy learning.

1. Non-parametric means that it makes no assumptions. The model is made up entirely from the data given to it
2. Lazy learning means that there is little training involved when using this method. Because of this, all of the training data is also used in testing when using kNN



©Madison Schott

2. Mathematics Behind kNN

From the general $L_p - norm$ we can define the corresponding $L_p - distance$ function, given as follows

$$\sigma_p(a, b) = ||a - b||_p$$

Mathematics *Distance*

1. Euclidean Distance*, L_2
2. Manhattan Distance, L_1
3. Hamming Distance, L_0
4. Minkowski Distance, L_p
5. Cosine Distance(similarity)

Euclidean Distance* is the most popular method

Euclidean Distance

$$L_2 - norm: \left|\left|x_i - y_i\right|\right|_2 = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Mathematics *Manhattan distance*

Manhattan Distance

$$L_1 - norm: ||x_i - y_i||_1 = \sum_{i=1}^{n} |x_i - y_i|$$

Hamming Distance

$$L_0 - norm: \left|\left|x_i - y_i\right|\right|_0 = \sum_{i=1}^{n} |x_i - y_i|$$

$$x = y \ \rightarrow 0$$
$$x \neq y \ \rightarrow 1$$

Minkowski Distance

$$L_p - norm: \left\|x_i - y_i\right\|_2 = \left(\sum_{i=1}^{n} (x_i - y_i)^2\right)^{\frac{1}{p}}$$

Mathematics *Cosine distance*

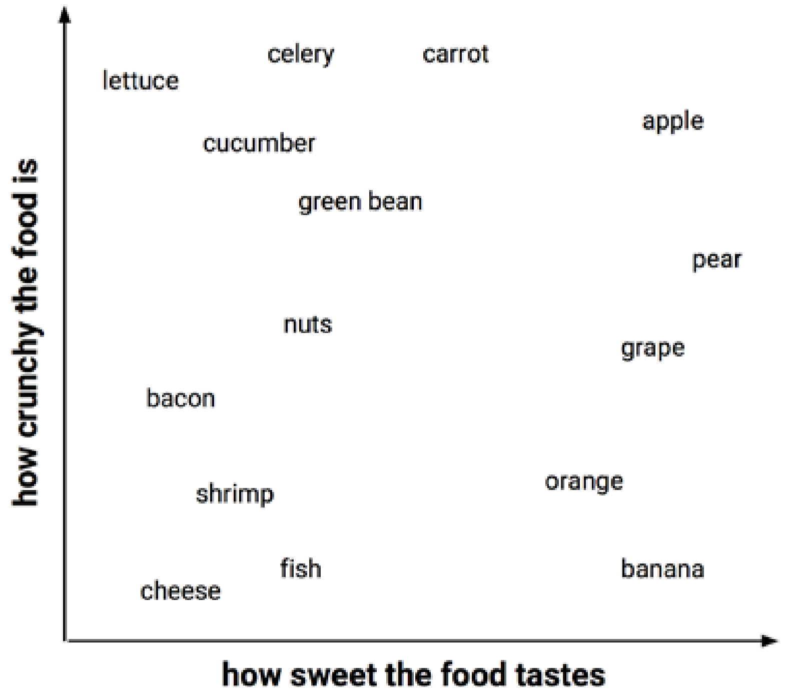
Cosine Distance

$$\cos(\theta) = \frac{a^T b}{||a||||b||}$$

3.Example

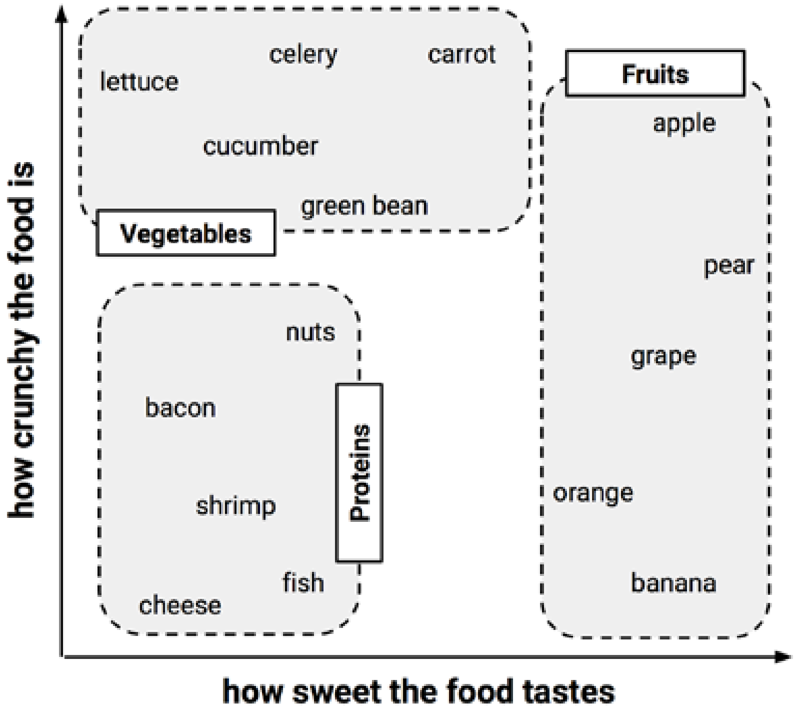
Food Example Again…

Ingredient	Sweetness	Crunchiness	Food Type
apple	*10*	*9*	*Fruit*
bacon	*1*	*4*	*Protein*
banana	*10*	*1*	*Fruit*
carrot	*7*	*10*	*Vegetable*
celery	*3*	*10*	*Vegetable*
cheese	*1*	*1*	*Protein*



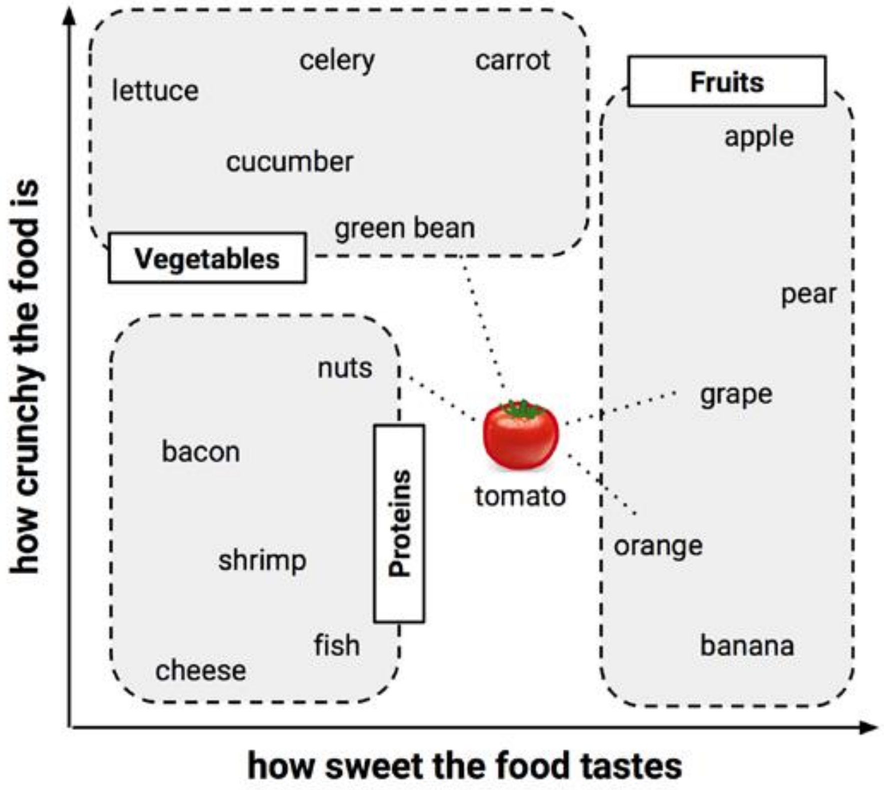
Food Example Again…

Ingredient	Sweetness	Crunchiness	Food Type
apple	*10*	*9*	*Fruit*
bacon	*1*	*4*	*Protein*
banana	*10*	*1*	*Fruit*
carrot	*7*	*10*	*Vegetable*
celery	*3*	*10*	*Vegetable*
cheese	*1*	*1*	*Protein*



Food Example Again…

Ingredient	Sweetness	Crunchiness	Food Type
apple	*10*	*9*	*Fruit*
bacon	*1*	*4*	*Protein*
banana	*10*	*1*	*Fruit*
carrot	*7*	*10*	*Vegetable*
celery	*3*	*10*	*Vegetable*
cheese	*1*	*1*	*Protein*



Food Example Again…

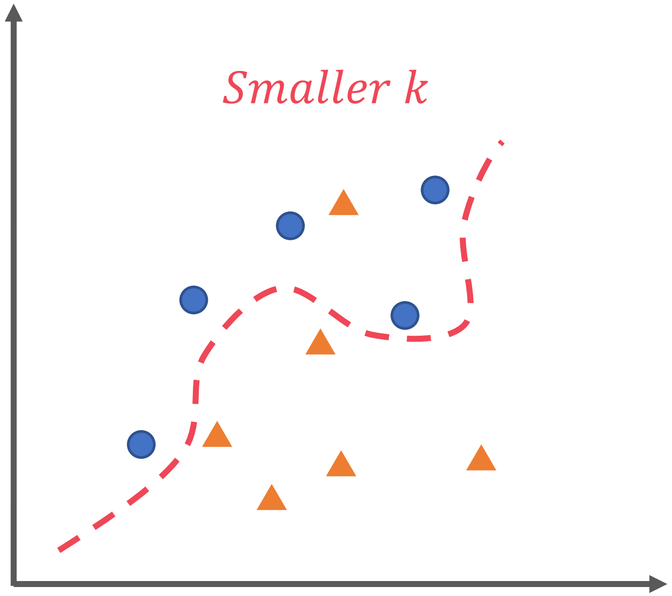
Euclidean Distance

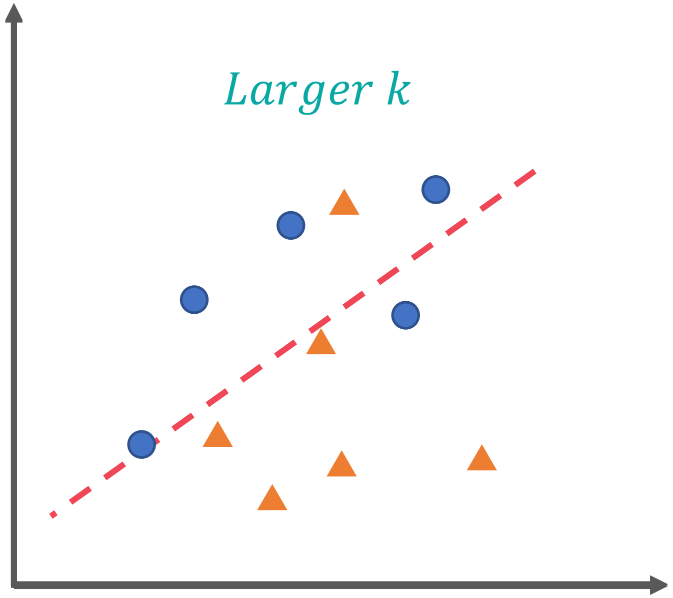
$$L_2 - norm: ||x_i - y_i||_2 = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Ingredient	Sweetness	Crunchiness	Food Type	Distance to tomato (r code)
grape	*8*	*5*	*Fruit*	*sqrt((6 – 8)^2 + (4 – 5)^2) = 2.2*
green bean	*3*	*7*	*Vegetable*	*sqrt((6 - 3)^2 + (4 - 7)^2) = 4.2*
nuts	*3*	*6*	*Protein*	*sqrt((6 - 3)^2 + (4 - 6)^2) = 3.6*
orange	*7*	*3*	*Fruit*	*sqrt((6 - 7)^2 + (4 - 3)^2) = 1.4*

4.Algorithm and others

Appropriate k





Normalization

Min-max normalization

$$x = \frac{x - \min(x)}{\max(x) - \min(x)}$$

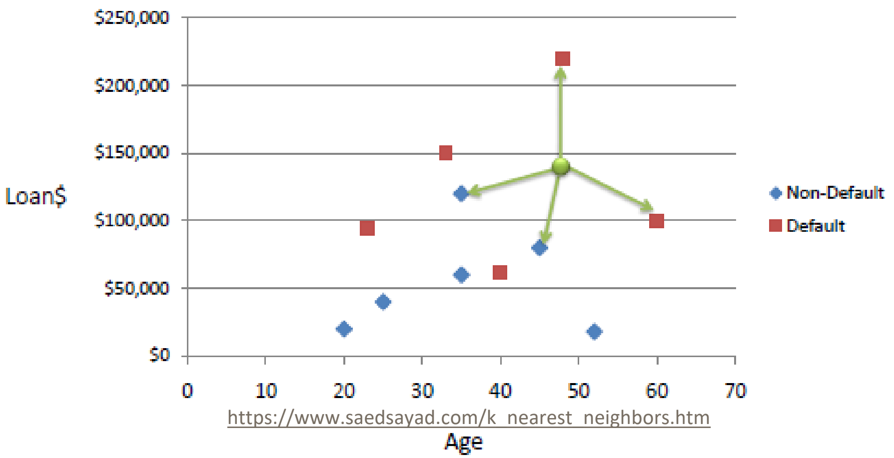
Z-score normalization

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

Why normalization?

Min-max normalization

$$x = \frac{x - \min(x)}{\max(x) - \min(x)}$$



https://www.saedsayad.com/k_nearest_neighbors.htm

Age	Loan	Default	Distance	
25	$40,000	N	102000	
35	$60,000	N	82000	
45	$80,000	N	62000	
20	$20,000	N	122000	
35	$120,000	N	22000	2
52	$18,000	N	124000	
23	$95,000	Y	47000	
40	$62,000	Y	80000	
60	$100,000	Y	42000	3
48	$220,000	Y	78000	
33	$150,000	Y	8000	1
48	$142,000	?		

Euclidean Distance

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

Age	Loan	Default	Distance
0.125	0.11	N	0.7652
0.375	0.21	N	0.5200
0.625	0.31	N	0.3160
0	0.01	N	0.9245
0.375	0.50	N	0.3428
0.8	0.00	N	0.6220
0.075	0.38	Y	0.6669
0.5	0.22	Y	0.4437
1	0.41	Y	0.3650
0.7	1.00	Y	0.3861
0.325	0.65	Y	0.3771
0.7	0.61	?	

Standardized Variable

$$X_s = \frac{X - Min}{Max - Min}$$

Algorithm

Let k be the number of nearest neighbors and D be the set of training examples.
for each test example $z = (\mathbf{x}', y')$ **do**
    Compute $d(\mathbf{x}', \mathbf{x})$, the distance between z and every example, $(\mathbf{x}, y) \in D$.
    Select $D_z \subseteq D$, the set of k closest training examples to z.
    $y' = \underset{v}{\operatorname{argmax}} \sum_{(\mathbf{x}_i, y_i) \in D_z} I(v = y_i)$
end for

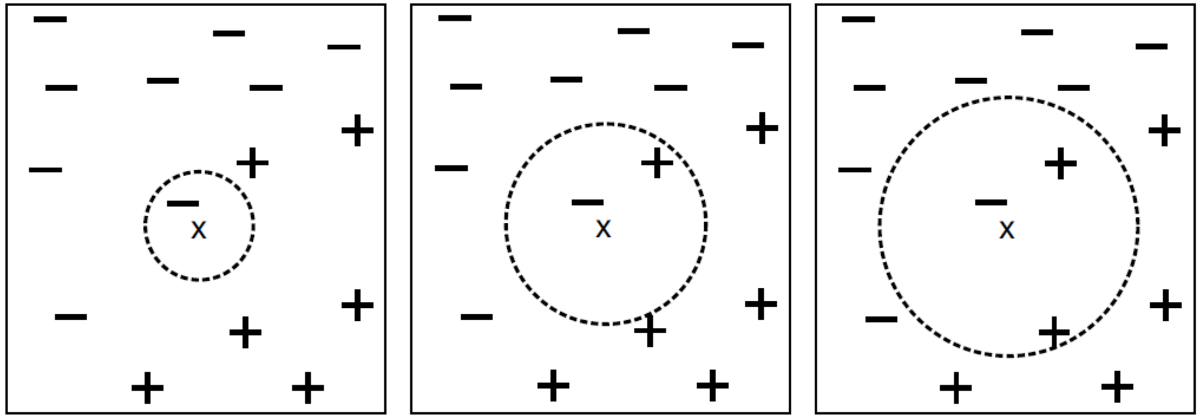
Algorithm *Voting*

Let k be the number of nearest neighbors and D be the set of training examples.
for each test example $z = (\mathbf{x}', y')$ **do**
  Compute $d(\mathbf{x}', \mathbf{x})$, the distance between z and every example, $(\mathbf{x}, y) \in D$.
  Select $D_z \subseteq D$, the set of k closest training examples to z.
  $\boxed{y' = \underset{v}{\operatorname{argmax}} \sum_{(\mathbf{x}_i, y_i) \in D_z} I(v = y_i)}$
end for

$$\text{Majority Voting:} \quad y' = \underset{v}{\operatorname{argmax}} \sum_{(\mathbf{x}_i, y_i) \in D_z} I(v = y_i),$$

where v is a class label, y_i is the class label for one of the nearest neighbors, and $I(\cdot)$ is an indicator function that returns the value 1 if its argument is true and 0 otherwise.

Algorithm *Voting*



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

$$\text{Majority Voting:} \quad y' = \underset{v}{\operatorname{argmax}} \sum_{(\mathbf{x}_i, y_i) \in D_z} I(v = y_i),$$

In the majority voting approach, every neighbor has the same impact on the classification. This makes the algorithm sensitive to the choice of k

Algorithm *Voting*

One way to reduce the impact of k is to weight the influence of each nearest neighbor x_i according to its distance: $w_i = 1/d(x', x_i)^2$. As a result, training examples that are located far away from z have a weaker impact on the classification compared to those that are located close to z.

$$\text{Distance-Weighted Voting:} \quad y' = \underset{v}{\operatorname{argmax}} \sum_{(\mathbf{x}_i, y_i) \in D_z} w_i \times I(v = y_i).$$

4.Pros and Cons

Pros and Cons

Pros:

1. Simple and effective
2. Makes no assumptions about the underlying data distribution
3. Fast training phase

Cons:

1. Does not produce a model, limiting the ability to understand how the features are related to the class
2. Requires selection of an appropriate k
3. Slow classification phase
4. Nominal features and missing data require additional processing

Resources

Resource

Textbook:

Galit Shmueli, Peter C. Bruce, Inbal Yahav, Nitin R. Patel, Kenneth C. Lichtendahl Jr., Data Mining for Business Analytics: Concepts, Techniques, and Applications in R (DMBA), Wiley, 1st Edition, ISBN-10: 1118879368, ISBN-13: 978-1118879368.

Additional Textbooks:

R For Data Science (open license, R4DS), Wickham, Hadley, and Garrett Grolemund

R Markdown (open license, RMD), Xie, Yihui, et al.

James, Gareth, et al. An Introduction to Statistical Learning: with Applications in R. Springer, 2017. (open license, ISL)

Mohammed J. Zaki, Wagner Meira, Jr., Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, May 2014. ISBN: 9780521766333.

David Hand, Heikki Mannila, Padhraic Smyth. Principles of Data Mining, The MIT Press, 2001, ISBN-10: 026208290X, ISBN-13: 978-0262082907.

Tan, Pang-Ning, et al. Introduction to Data Mining (DM). Pearson Education, 2006.