Machine Learning for Business

Module 11: Similarity and nearest neighbor

Day 6, 9.00 – 12.00

Asst. Prof. Dr. Santitham Prom-on

Department of Computer Engineering, Faculty of Engineering
King Mongkut's University of Technology Thonburi

BIG DATA EXPERIENCE CENTER

BIG DATA EXPERIENCE

g·able

Similarity

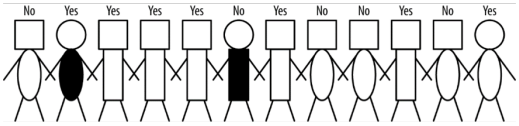
Part 1

BIG DATA EXPERIENCE

g·able

Similarity

- Similarity underlies many data science methods and solutions
- If two things (people, companies, products) are similar in some ways, they often share other characteristics as well.



BIG DATA EXPERIENCE

g·able

Tasks involving with similarity

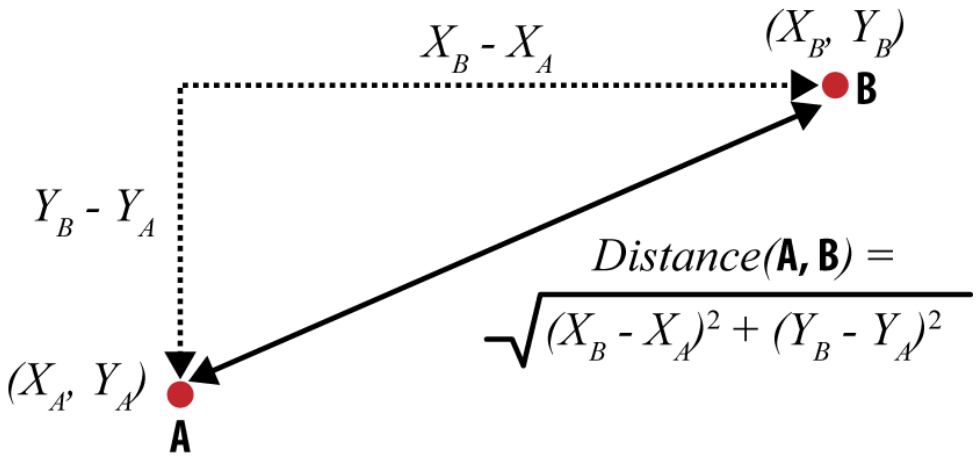
- Retrieve similar things directly
  - IBM wants to find companies that are similar to their best business customers
- Use in classification or regression
- Group similar items together into clusters
  - To see whether customer base contains group of similar customers and what they have in common
- Provide recommendations of similar products
  - Amazon, Netflix
- Provide reasoning from similar cases
  - Knowledge management, troubleshooting

Example: credit card application

Attribute	Person A	Person B
Age	23	40
Years at current address	2	10
Residential status (1=Owner, 2=Renter, 3=Other)	2	1

- Multiple attributes
- No single best method for reducing them to a single distance measurement

Geometric approach Euclidean distance



Equation 6-1. General Euclidean distance

$$\sqrt{(d_{1,A} - d_{1,B})^2 + (d_{2,A} - d_{2,B})^2 + \ldots + (d_{n,A} - d_{n,B})^2}$$

Example: credit card application

Attribute	Person A	Person B
Age	23	40
Years at current address	2	10
Residential status (1=Owner, 2=Renter, 3=Other)	2	1

$$d(A,B) = \sqrt{(23-40)^2 + (2-10)^2 + (2-1)^2}$$
$$\approx 18.8$$

- No unit, no meaningful interpretation
- Useful for comparing the similarity of one pairs to others

R: Euclidean distance

```
library(proxy)
x <- c(age = 23, year = 2,  resident = 2)
y <- c(age = 40, year = 10, resident = 1)
dist(rbind(x,y))

#          x
# y 18.81489
```

BIG DATA EXPERIENCE

Distance Functions

Equation 6-2. Euclidean distance (L2 norm)

$$d_{\text{Euclidean}}(\mathbf{X},\mathbf{Y}) = \| \mathbf{X} - \mathbf{Y} \|_2 = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots}$$

Equation 6-3. Manhattan distance (L1 norm)

$$d_{\text{Manhattan}}(\mathbf{X},\mathbf{Y}) = \| \mathbf{X} - \mathbf{Y} \|_1 = |x_1 - y_1| + |x_2 - y_2| + \cdots$$

Equation 6-4. Jaccard distance

$$d_{\text{Jaccard}}(X, Y) = 1 - \frac{|X \cap Y|}{|X \cup Y|}$$

Equation 6-5. Cosine distance

$$d_{cosine}(\mathbf{X},\mathbf{Y}) = 1 - \frac{\mathbf{X} \cdot \mathbf{Y}}{\| \mathbf{X} \|_2 \cdot \| \mathbf{Y} \|}$$

g·able

BIG DATA EXPERIENCE

R: Euclidean vs Manhattan distance

```
# Euclidean distance is suitable for most of
# numeric data
dist(bankData[1:5,c("age","balance")],
     method = "Euclidean")

# Manhattan distance is less sensitive to outlier
dist(bankData[1:5,c("age","balance")],
     method = "Manhattan")
```

g·able

Similarity calculation Heterogeneous attributes

- Sometime, attributes may be numeric and categorical
- Also, values may not have the same range
- The variable thus need to be scaled/normalized by its range before calculating similarity

Attribute	Person A	Person B
Sex	Male	Female
Age	23	40
Years at current address	2	10
Residential status (1=Owner, 2=Renter, 3=Other)	2	1
Income	50,000	90,000

Binary/nominal variable Simple matching

Object i \ Object j	X=1	X=0
X=1	a	b
X=0	c	d

a+b+c+d = Number of variables

Simple matching coefficient

$$d(i,j) = \frac{b+c}{a+b+c+d}$$

Proportion of variables, in which people disagree

BIG DATA EXPERIENCE

Gower distance function

- Idea: Use distance measure between 0 and 1 for each variable: $d_{ij}^{(f)}$
- Aggregate: $d(i,j) = \frac{1}{p}\sum_{i=1}^{p} d_{ij}^{(f)}$

- Binary (a/s), nominal: Use methods discussed before
- Interval-scaled: $d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{R_f}$
  x_{if}: Value for object i in variable f
  R_f: Range of variable f for all objects
- Ordinal: Use normalized ranks; then like interval-scaled based on range

g·able

BIG DATA EXPERIENCE

R: Gower distance

```
dist(bankData[1:5,])
```

```
          1         2         3         4
2 0.3024425
3 0.3529137 0.2269418
4 0.2147588 0.2428507 0.2558020
5 0.3729730 0.2180662 0.3329368 0.2844463
```

g·able

Nearest neighbor reasoning

- We could use distance to
  - find the companies most similar to our best corporate customers
  - find the online consumers most similar to our best retail customers
- Once we have found these, we can take whatever action is appropriate in the business context
- The most similar instance is called nearest neighbor

Example: Wine comparison Attribute list

1.	**Color:** *yellow, very pale, pale, pale gold, gold, old gold, full gold, amber,* etc.	(14 values)
2.	**Nose:** *aromatic, peaty, sweet, light, fresh, dry, grassy,* etc.	(12 values)
3.	**Body:** *soft, medium, full, round, smooth, light, firm, oily.*	(8 values)
4.	**Palate:** *full, dry, sherry, big, fruity, grassy, smoky, salty,* etc.	(15 values)
5.	**Finish:** *full, dry, warm, light, smooth, clean, fruity, grassy, smoky,* etc.	(19 values)

68 binary attributes

Example: Drink Comparison
Most similar to *Bunnahabhain*

	Distance	Descriptors
Bunnahabhain	—	*gold; firm,med,light; sweet,fruit,clean; fresh,sea; full*
Glenglassaugh	0.643	gold; firm,light,smooth; sweet,grass; fresh,grass
Tullibardine	0.647	gold; firm,med,smooth; sweet,fruit,full,grass,clean; sweet; big,arome,sweet
Ardbeg	0.667	sherry; firm,med,full,light; sweet; dry,peat,sea;salt
Bruichladdich	0.667	pale; firm,light,smooth; dry,sweet,smoke,clean; light; full
Glenmorangie	0.667	p.gold; med,oily,light; sweet,grass,spice; sweet,spicy,grass,sea,fresh; full,long

Dataset are available at http://adn.biol.umontreal.ca/~numericalecology/data/scotch.html

Look-alike customers

- Finding look-alike customers is one of the practical way to identify potential customers
- The steps include

1. Find the target group
2. Calculate distance from all customer to ..
   - All customers in the target group, or
   - The centroid of the group
3. Rank the customers based on the distance

BIG DATA EXPERIENCE

g·able

R: identifying target group

```
library(rpart)
library(rpart.plot)
tree <- rpart(y ~ ., bankData)
# plot the tree
rpart.plot(tree)

# see the ids of leaf nodes
# in binary tree format
table(tree$where)
```

BIG DATA EXPERIENCE

g·able

R: Get the target group

```
library(dplyr)
set.seed(555)
target_cond <- tree$where == 6
bankData.target <- sample_n(
  bankData[target_cond,],5)
bankData.1 <- sample_n(
  bankData[-which(target_cond),],20)
```

BIG DATA EXPERIENCE

g·able

R: Calculate distance and get the most similar (least distance)

```
library(proxy)
d <- dist(bankData.1, bankData.target)
cust_d <- apply(d,1,mean)
print(sort(cust_d))

# if the ids are needed, use “names”
```

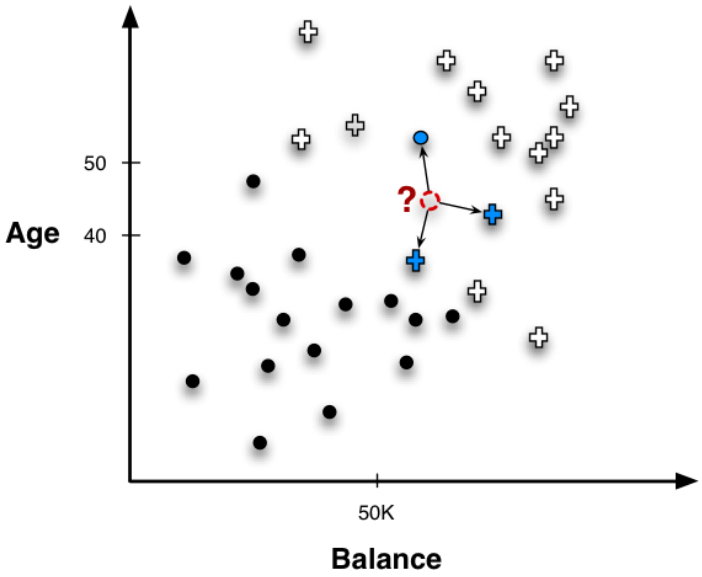
BIG DATA EXPERIENCE

g·able

Nearest neighbor for predictive modeling

- Given a new example whose target variable we want to predict, we scan through all the training examples and choose several that are most similar to the new examples
- Then we predict the new example’s target value, based on the nearest neighbors’ (known) target values.
- The prediction is based on a combining function

Classification



What should be our combining function?

A simple combining function would be majority vote

The predicted class would be positive

Example: similarity in prediction

Table 6-1. Nearest neighbor example: Will David respond or not?

Customer	Age	Income (1000s)	Cards	Response (target)	Distance from David
David	*37*	*50*	*2*	*?*	0
John	35	35	3	Yes	$\sqrt{(35-37)^2+(35-50)^2+(3-2)^2} = 15.16$
Rachael	22	50	2	No	$\sqrt{(22-37)^2+(50-50)^2+(2-2)^2} = 15$
Ruth	63	200	1	No	$\sqrt{(63-37)^2+(200-50)^2+(1-2)^2} = 152.23$
Jefferson	59	170	1	No	$\sqrt{(59-37)^2+(170-50)^2+(1-2)^2} = 122$
Norah	25	40	4	Yes	$\sqrt{(25-37)^2+(40-50)^2+(4-2)^2} = 15.74$

How many neighbors should we use?
Should they have equal weights in the combining function?

Probability estimation

- Probability estimation is as important as Yes/No decision
- Consider again the previous example of deciding whether David will be a responder or not

Customer	Age	Income (1000s)	Cards	Response (target)	Distance from David
David	*37*	*50*	*2*	*?*	0
John	35	35	3	Yes	$\sqrt{(35-37)^2+(35-50)^2+(3-2)^2} = 15.16$
Rachael	22	50	2	No	$\sqrt{(22-37)^2+(50-50)^2+(2-2)^2} = 15$
Ruth	63	200	1	No	$= 152.23$
Jefferson	59	170	1	No	$\sqrt{(59-37)^2+(170-50)^2+(1-2)^2} = 122$
Norah	25	40	4	Yes	$\sqrt{(25-37)^2+(40-50)^2+(4-2)^2} = 15.74$

Score = 2/3 = 0.667

Regression

- Once we can retrieve nearest neighbors, we can use them for any predictive mining task by combining them in different ways
- Note that, in retrieving neighbors, we are not using the target variable in prediction.

Customer	Age	Income (1000s)	Cards	Response (target)	Distance from David
David	*37*	*50*	*2*	*?*	0
John	35	35	3	Yes	$\sqrt{(35-37)^2+(35-50)^2+(3-2)^2} = 15.16$
Rachael	22	50	2	No	$\sqrt{(22-37)^2+(50-50)^2+(2-2)^2} = 15$
Ruth	63	200	1	No	$\sqrt{(200-50)^2+(1-2)^2} = 152.23$
Jefferson	59	170	1	No	$\sqrt{(59-37)^2+(170-50)^2+(1-2)^2} = 122$
Norah	25	40	4	Yes	$\sqrt{(25-37)^2+(40-50)^2+(4-2)^2} = 15.74$

Average = (35+50+40)/3 = 42
Median = 40

BIG DATA EXPERIENCE

g·able

How many neighbors and how much influence?

- No simple answer to how many neighbors should be used
- Nearest neighbor algorithms are often referred by *k*-NN, where *k* is the number of neighbors
- The greater *k* is, the more estimates are smoothed out among neighbors
- If $k = n$ (number of training samples), the majority is the class probability
- But the answer is sensitive to the number *k*

BIG DATA EXPERIENCE

g·able

Weighted voting

- Nearest neighbor method often uses weighted voting or similarity moderated voting by scaling neighbor contribution by distance

Name	Distance	Similarity weight	Contribution	Class
Rachael	15.0	0.004444	0.344	No
John	15.2	0.004348	0.336	Yes
Norah	15.7	0.004032	0.312	Yes
Jefferson	122.0	0.000067	0.005	No
Ruth	152.2	0.000043	0.003	No

BIG DATA EXPERIENCE

R: kknn

```
library(kknn)
library(caret)
train_pos <- sample(nrow(bankData), 0.3*nrow(bankData))
trainingData <- bankData[train_pos,]
testingData <- bankData[-train_pos,]
result <- kknn(y ~ ., trainingData, testingData, k = 11)
```

g·able

BIG DATA EXPERIENCE

R: kknn results

```
confusionMatrix(result$fitted.values,
testingData$y, positive = 'yes',
mode = 'prec_recall')
```

```
Confusion Matrix and Statistics

          Reference
Prediction    no   yes
       no  27171  2547
       yes   750  1180

               Accuracy : 0.8958
                 95% CI : (0.8924, 0.8992)
    No Information Rate : 0.8822
    P-Value [Acc > NIR] : 1.276e-14

                  Kappa : 0.3663
 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.61140
                 Recall : 0.31661
                     F1 : 0.41718
```

g·able

BIG DATA EXPERIENCE

g·able

Activity: similarity for classification

- Use kknn to classify the the credit approval data

BIG DATA EXPERIENCE

g·able

Thank you

Question?

BIG DATA EXPERIENCE CENTER