Discovering Knowledge in Data

Daniel T. Larose, Ph.D.

Chapter 6
Preparing to model the data

Prepared by James Steck and Eric Flores

Supervised vs Unsupervised Methods

- Data mining methods may be categorized as either *supervised* or *unsupervised*
- In *unsupervised methods*, there is no specific target variable
- The most common unsupervised method is *clustering* (Chapters 10 and 11)
  - For example, political consultants might use clustering to uncover voter profiles based on income, gender, race, etc., for fund-raising and advertising purposes
  - When used for market basket analysis (which products are bought together), *association mining* is also considered as a *unsupervised method* (Chapter 12)
- Most data mining methods are *supervised*
  - There is a particular pre-specified target variable
  - The algorithm is given many examples where the value of the target variable is provided
  - The algorithm learns which values of the **target variable** are associated to which values of the **predictor variables**
  - Regression, from Chapter 5, is a supervised method
  - All classification methods from Chapters 7 to 9 (decision trees, neural networks, k-nearest neighbors) are supervised methods too
- Important: Supervised and Unsupervised are just data mining terms
  - Unsupervised methods do not mean that they require no human involvement!

Discovering Knowledge in Data

Daniel T. Larose, Ph.D.

Chapter 7
k-Nearest Neighbor Algorithm

Prepared by James Steck and Eric Flores

Classification Task

- Classification is probably the most common data mining task
- Examples
  - Banking - determine whether a mortgage application is a good or bad credit risk.
  - Education - place a student into a particular track with respect to special needs
  - Medicine - diagnose whether a disease is present
  - Law - determine if a will is fraudulent
  - Security - identify whether a certain financial transaction represents a terrorist threat

Classification Task (*cont'd*)

- In classification, there is a categorical target variable with is partitioned into two or more classes
- For example, the target variable *income_bracket* may include the categories "Low", "Middle", and "High"
- The algorithm learns by examining relationships between the values of the predictor (input) fields and target values
- Suppose, we want to classify a person's income bracket based on the age, gender, and occupation values given in a database

Table 7.1

Subject	Age	Gender	Occupation	Income Bracket
001	47	F	Software Engineer	High
002	28	M	Marketing Consultant	Middle
003	35	M	Unemployed	Low
…	…	…	…	…

Classification Task (*cont'd*)

- First, the classification algorithm examines the data set values for the predictor and the already classified target variables in the *training set*
- This way, the algorithm "learns" which values of the predictor variables are associated with values of the target variable
- For example, older females may be associated with *income_bracket* values of "High"
- Now that the data model is built from the *training set*, the algorithm examines *new records* for which *income_bracket* is unknown
- According to the classifications in the training set, the algorithm classifies the new records
- For example, a 63 year-old female might be classified in the "High" income bracket

k-Nearest Neighbor Algorithm

- The *k*-Nearest Neighbor algorithm is an example of instance-based learning (memory-based learning).
- In instance-based learning, training set records are first stored, and then, the classification of a new unclassified record is performed by comparing it to the records in the training set it is most similar to.
- *k*-Nearest Neighbor is most often used for classification, although it is also applicable to estimation and prediction tasks

- Example: Patient 1
  - Recall from Chapter 1 that we were interested in classifying the type of drug a patient should be prescribed.
  - The training set consists of 200 patients with Na/K ratio, age, and drug prescribed.
  - Our task is to classify the type of drug a new patient should be prescribed that is 40-years-old and has a Na/K ratio of 29 (represented as the circle labeled as Patient 1 in Figure 7.1)

k-Nearest Neighbor Algorithm (*cont'd*)

- This scatter plot of Na/K against Age shows the records in the training set that patients 1, 2, and 3 are most similar to
- A "drug" overlay is shown where Light points = drug Y, Medium points = drug A or X, and Dark points = drug B or C

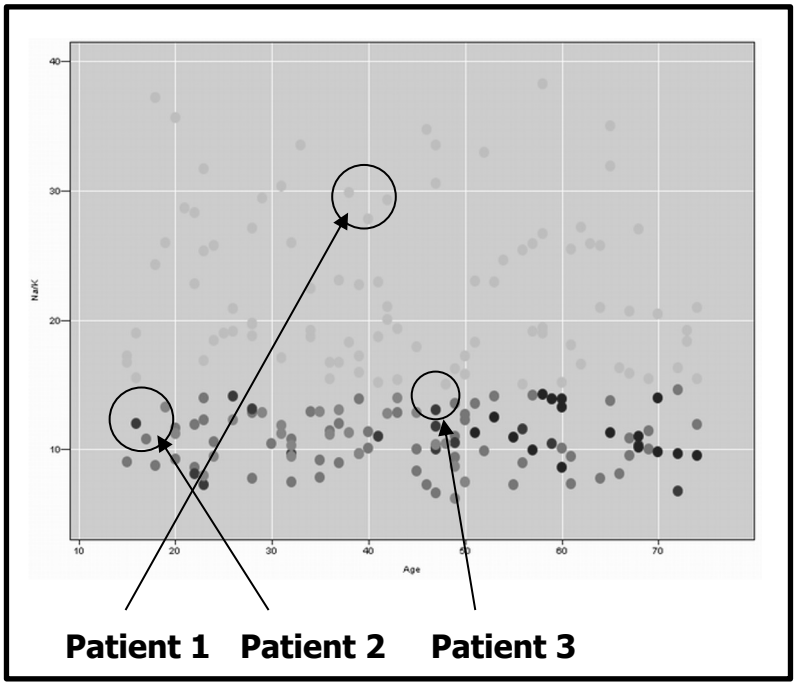

Figure 7.1

k-Nearest Neighbor Algorithm (*cont'd*)

- Which drug should Patient 1 be prescribed?
- Since Patient 1's profile places them in the scatter plot near patients prescribed drug Y, we classify Patient 1 as drug Y
- All points near Patient 1 are prescribed drug Y, making this a straightforward classification

- Example: Patient 2
  - Next, we classify a new patient who is 17-years-old with a Na/K ratio = 12.5. A close-up (Figure 7.2) shows the neighborhood of training points in close proximity to Patient 2

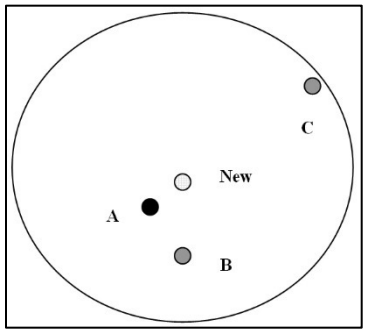

Figure 7.2

k-Nearest Neighbor Algorithm (*cont'd*)

- Suppose we let $k = 1$ for our *k*-Nearest Neighbor algorithm
- This means we classify Patient 2 according to whichever single point in the training set it is closet to
- In this case, Patient 2 is closest to the Dark point, and therefore we classify them as drug B or C

- Suppose we let $k = 2$ and reclassify Patient 2 using *k*-Nearest Neighbor
- Now, Patient 2 is closest to a Dark point and Medium point
- How does the algorithm decide which drug to prescribe?
- A simple voting scheme does not help. There is one vote for each of the two classes.

k-Nearest Neighbor Algorithm (*cont'd*)

- However, with $k = 3$, voting determines that two of the three closet points to Patient 2 are Medium
- Therefore, Patient 2 is classified as drug A or X
- Note that the classification of Patient 2 differed based on the value chosen for *k*

- Example: Patient 3
  - Patient 3 is 47-years-old and has a Na/K ratio of 13.5. A close-up shows Patient 3 in the center, with the closest 3 training data points

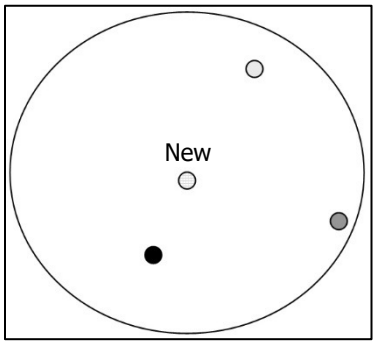

Figure 7.3

k-Nearest Neighbor Algorithm (*cont'd*)

- With $k = 1$, Patient 3 is closest to the Dark point, based on a distance measure
- Therefore, Patient 3 is classified as drug B or C
- Using $k = 2$ or $k = 3$, voting does not help since each of the three nearest training points have different target values

- Considerations when using *k*-Nearest Neighbor
  - How many neighbors should be used? $k = ?$
  - How is the distance between points measured?
  - How do we combine the information from more than one observation?
  - Should all points be weighted equally, or should some points have more influence?

Distance Function

- How is similarity defined between an unclassified record and its neighbors?
  - Example: For a 50-year-old male, which patient is more similar, a 20-year-old male or a 50-year-old female
- A distance metric is a real-valued function d, such that for any coordinates x, y, and z:

1. $d(x,y) \geq 0$, and $d(x,y) = 0$ if and only if $x = y$

   Distance is always non-negative

2. $d(x,y) = d(y,x)$

   Commutative, distance from "A to B" is equal to distance from "B to A"

3. $d(x,z) \leq d(x,y) + d(y,z)$

   Triangle inequality holds, introducing a third point can never shorten the distance between two other points

Distance Function (*cont'd*)

- The Euclidean Distance function is commonly-used to measure distance

$$d_{Euclidean}(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$$

where $\mathbf{x} = x_1, x_2, \ldots, x_m$, and $\mathbf{y} = y_1, y_2, \ldots, y_m$ represent the m attribute values of two records

- Example
  - Suppose Patient A is 20-years-old and has a Na/K ratio = 12, and Patient B is 30-years-old and has a Na/K ratio = 8
  - What is the Euclidean distance between these instances?

Distance Function (*cont'd*)

Solution

- Using the function for Euclidean distance:

$$d_{\text{Euclidean}}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2} = \sqrt{(20-30)^2 + (12-8)^2} = 10.77$$

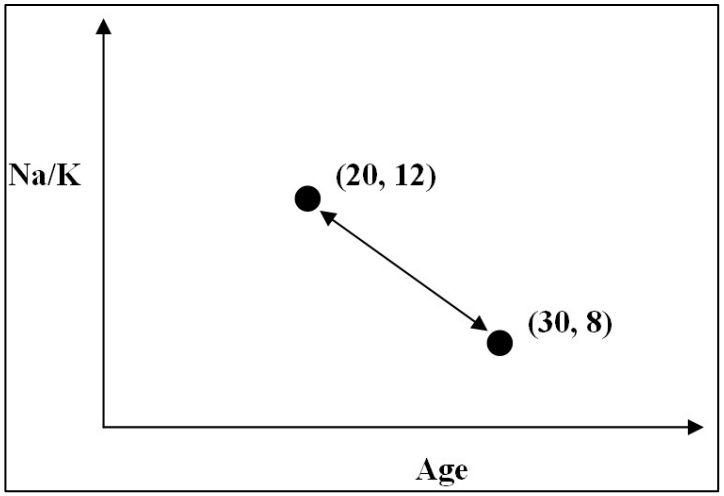

Figure 7.1

Distance Function (*cont'd*)

Normalization

- When measuring distance, one or more attributes can have very large values, relative to the other attributes
- For example, *income* may be scaled 30,000-100,000, whereas *years_of_service* takes on values 0-10
- In this case, the values of *income* will overwhelm the contribution of *years_of_service*
- To avoid this situation, we use normalization.
- Continuous data values should be normalized using Min-Max Normalization or Z-Score Standardization
  - As discussed in Chapter 2

$$\text{Min - Max Normalization} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

$$\text{Z - Score Standardization} = \frac{X - \text{mean}(X)}{\text{standard deviation}(X)}$$

Distance Function (*cont'd*)

- For categorical attributes, the Euclidean Distance function is not appropriate.
- Instead, we define a function called "different"

$$\text{Different}(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{otherwise} \end{cases}$$

- We substitute *different*(x_i, y_i) for the *i*th term in the Euclidean distance metric above

- Example
  - Which patient is more similar to a 50-year-old male: a 20-year-old male or a 50-year-old female?

Distance Function (*cont'd*)

- Let Patient A = 50-year-old male, Patient B = 20-year-old male, and Patient C = 50-year-old female
- Suppose that the *Age* variable has a range = 50, minimum = 10, mean = 45, and standard deviation = 15
- The table contains original, Min-Max Normalized, and Z-Score Standardized values for *Age*

Table 7.2

Patient	Age	Age_{MMN}	Age_{Zscore}	Gender
A	50	$\frac{50-10}{50}=0.8$	$\frac{50-45}{15}=0.33$	Male
B	20	$\frac{20-10}{50}=0.2$	$\frac{20-45}{15}=-1.67$	Male
C	50	$\frac{50-10}{50}=0.8$	$\frac{50-45}{15}=0.33$	Female

Distance Function (*cont'd*)

- Age not normalized
  - Assume we do not normalize *Age* and calculate the distance between Patient A and Patient B, and Patient A and Patient C

$$d(A,B) = \sqrt{(50-20)^2 + 0^2} = 30$$

$$d(A,C) = \sqrt{(50-50)^2 + 1^2} = 1$$

  - We determine, although perhaps incorrectly, that Patient C is nearest to Patient A
  - Is Patient B really 30 times more distant than Patient C is to Patient A?
  - Perhaps neglecting to normalize the values of *Age* is creating this discrepancy.

Distance Function (*cont'd*)

- Age Normalized using Min-Max
  - *Age* is normalized using Min-Max Normalization. Values lie in the range $[0, 1]$
  - Again, we calculate the distance between Patient A and Patient B, and Patient A and Patient C

$$d_{MMN}(A,B) = \sqrt{(0.8-0.2)^2 + 0^2} = 0.6$$

$$d_{MMN}(A,C) = \sqrt{(0.8-0.8)^2 + 1^2} = 1.0$$

  - In this case, Patient B is now closer to Patient A

Distance Function (*cont'd*)

- Age Standardized using Z-Score
  - This time, *Age* is standardized using Z-Score Standardization

$$d_{Zscore}(A,B) = \sqrt{(0.33-(-1.67))^2 + 0^2} = 2.0$$

$$d_{Zscore}(A,C) = \sqrt{(0.33-0.33)^2 + 1^2} = 1.0$$

  - Using Z-Score Standardization, most values are typically contained in the range **[-3, 3]**. (99.7% confidence interval for Normal distribution is [Mean-3*SD, Mean+3*SD]. Z-Score Standardization distribution is N[0,1]).
  - Now, Patient C is nearest to Patient A. This is different from the results obtained using Min-Max Normalization

Distance Function (*cont'd*)

- Conclusion
  - The use of different normalization techniques resulted in Patient A being nearest to different patients in the training set
  - This underscores the importance of understanding which normalization technique is being used
  - Note that the *different(x,y)* and Min-Max Normalization functions produce values in the range [0, 1]
  - Perhaps, when calculating the distance between records containing both numeric and categorical attributes, the use of Min-Max Normalization is preferred

Database Considerations

- Instance-based learning methods benefit from having access to learning examples composed of many attribute value combinations
- The data set should be balanced to include a sufficient number of records with common, as well as less-common, classifications
- One approach to balancing the data set is to reduce the proportion of records with more common classifications
- Restrictions on main memory space may limit the size of the training set used
- The training set may be reduced to include only those records that occur near a classification "boundary". For example, in Figure 7.1, all records with Na/K ratio value greater than, say, 19 could be omitted from the database without loss of classification accuracy, since all records in this region are classified as light gray. New records with Na/K ratio >19 would therefore be classified similarly.

k-Nearest Neighbor Algorithm for Estimation and Prediction

- *k*-Nearest Neighbor algorithm may be used for estimation and prediction of continuous-valued target variables
- A method used to accomplish this is Locally Weighted Averaging

- Example
  - We will estimate the systolic blood pressure for a 17-year-old patient with Na/K ratio equal to 12.5, using $k = 3$
  - The predictors are *Na/K* and *Age* and the target variable is *BP*
  - The three neighbors (A, B, and C) from the training set are shown below

Table 7.4

Record	Age	Na/K	BP	Age_{MMN}	Na/K_{MMN}	Distance
New	17	12.5	?	0.05	0.25	--
A	16.8	12.4	120	0.0467	0.2471	0.009305
B	17.2	10.5	122	0.0533	0.1912	0.176430
C	19.5	13.5	130	0.0917	0.2794	0.097560

k-Nearest Neighbor Algorithm for Estimation and Prediction (*cont'd*)

- Assume BP has a range = 80, and minimum = 90
- We also stretch the axes for the Na/K ratio, to reflect its importance in estimating BP. In addition, we use the inverse square of the distances for the weights

$$\hat{y}_{new} = \frac{\sum_i w_i y_i}{\sum_i w_i} \qquad \text{where } w_i = \frac{1}{d(new, x_i)^2} \text{ for existing records } x_1, x_2, \cdots, x_k$$

- The estimated systolic blood pressure for the new record is:

$$\hat{y}_{new} = \frac{\sum_i w_i y_i}{\sum_i w_i} = \frac{\frac{120}{.009305^2} + \frac{122}{.17643^2} + \frac{130}{.09756^2}}{\frac{1}{.009305^2} + \frac{1}{.17643^2} + \frac{1}{.09756^2}} = 120.0954$$

- Since Record A is closest to the new record, its BP value of 120 makes a significant contribution to the estimated BP value

Choosing *k*

- What value of *k* is optimal?
- There is not necessarily an obvious solution

- Smaller *k*
  - Choosing a small value for *k* may lead the algorithm to overfit the data
  - Noise or outliers may unduly affect classification
- Larger *k*
  - Larger values will tend to smooth out distinctive or uncertain (doubtful) data values in the training set
  - If the values become too large, locally interesting behavior will be overlooked
- Cross-validation alternative
  - Iterating over different values of k for finding the k value that minimizes classification/estimation error