CS 513
Knowledge Discovery & Data Mining

k-Nearest Neighbor Algorithm

Khasha Dehnad

Supervised vs. Unsupervised Methods

- Data mining methods are categorized as either Unsupervised or Supervised
- Unsupervised Methods
  - A target variable is not specified
  - Instead, the algorithm searches for patterns and structure among the variables
  - Clustering is the most common unsupervised method
  - For example, political consultants analyze voter clusters in congressional districts that may be responsive to their particular candidate
  - Important variables such as gender, age, income, and race are input to the clustering algorithm
  - Voter profiles for fund-raising and advertising are created

Supervised vs. Unsupervised Methods (*cont'd*)

- Supervised Methods
  - A target variable is <u>specified</u>
  - The algorithm "learns" from the examples by determining which values of the predictor variables are associated with different values of the target variable
  - For example, the regression methods discussed in Chapter 4 are supervised. The observed values of the response (target) variable are read by the least-squares algorithm, while it attempts to minimize the prediction error
  - All classification methods in Chapters 5 – 7 are supervised methods including: <u>Decision Trees</u>, <u>Neural Networks</u>, and <u>*k*-Nearest Neighbors</u>

Methodology for Supervised Modeling

- Supervised data mining methods use Training, Test, and Validation data sets as part of the model building and evaluation process
- Training
  - The Training Set includes records with predictor variables and pre-classified values for the target variable
  - This is the initial stage where a provisional data mining model is built using the training set
  - The model "learns" from the examples in the training set
  - What happens if the model blindly applies all patterns learned from the training set to future data?

Methodology for Supervised Modeling (*cont'd*)

- For example, suppose every customer in a training set named "David" happens to be in the high-income bracket
- A data mining model that "memorizes" this idiosyncrasy in the training set is actually <u>overfitting</u> the data
- Most likely we would not want our model to apply this rule to future or unseen data
- Therefore, the next step in the process is to examine the performance of the provisional data model using a <u>different set of data</u>

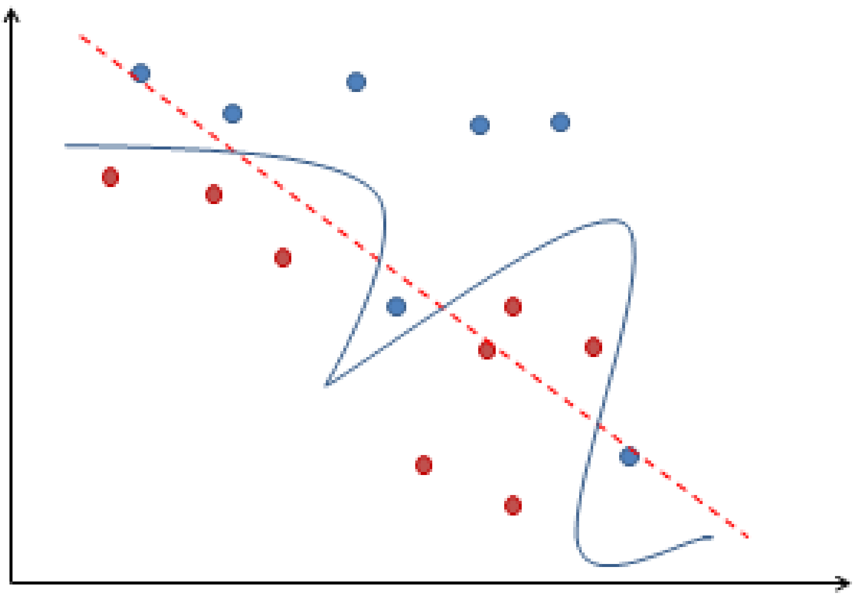
Methodology for Supervised Modeling (*cont'd*)

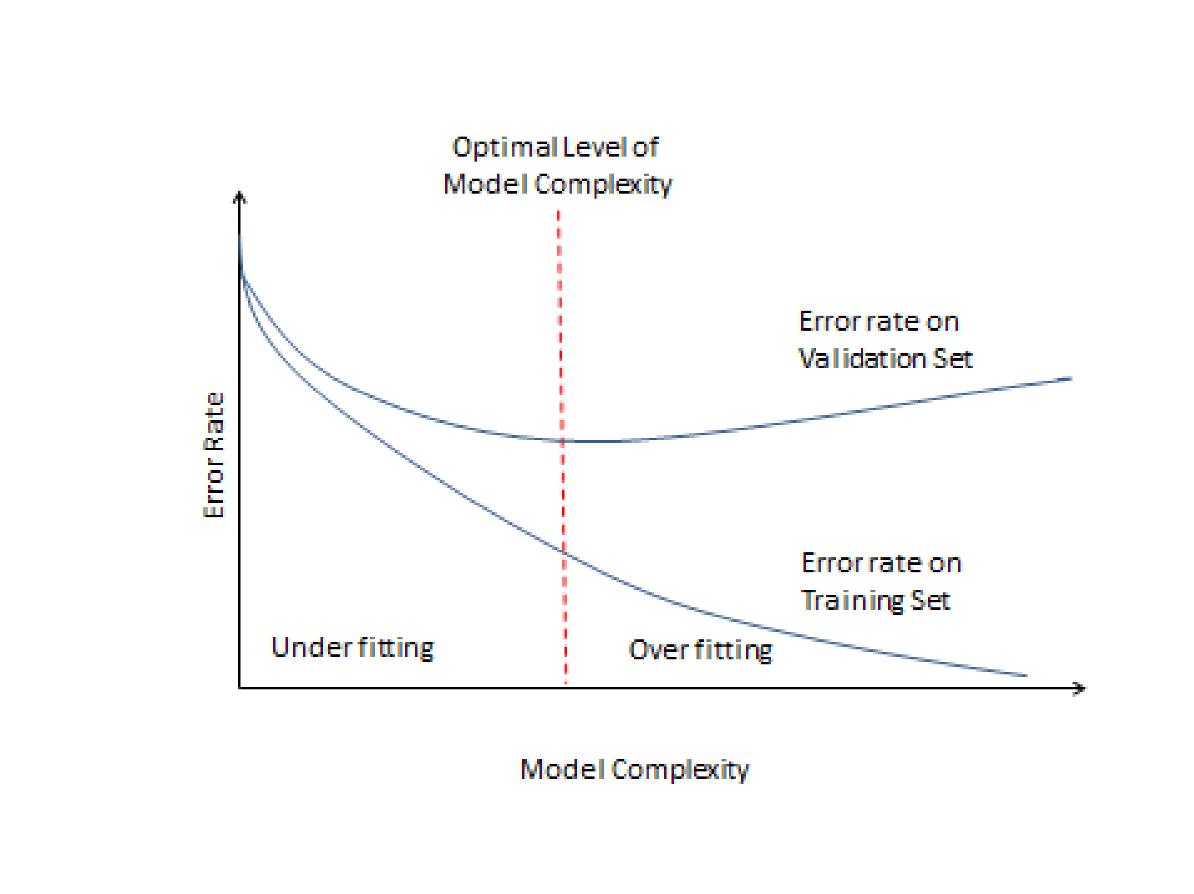
- Testing
  - The Test Set is a "holdout" set of data independent from the training set that was used to build the provisional data model
  - The true values of the target variable in the test set are hidden temporarily from the provisional data model
  - The provisional data model simply classifies the records in the test set according to the rules and patterns it learned from the records in the training set
  - The performance of the provisional data model is evaluated by comparing its classifications against the actual values of the target variable
  - The provisional data model is then adjusted in an effort to minimize the error rate on the test set

Methodology for Supervised Modeling (*cont'd*)

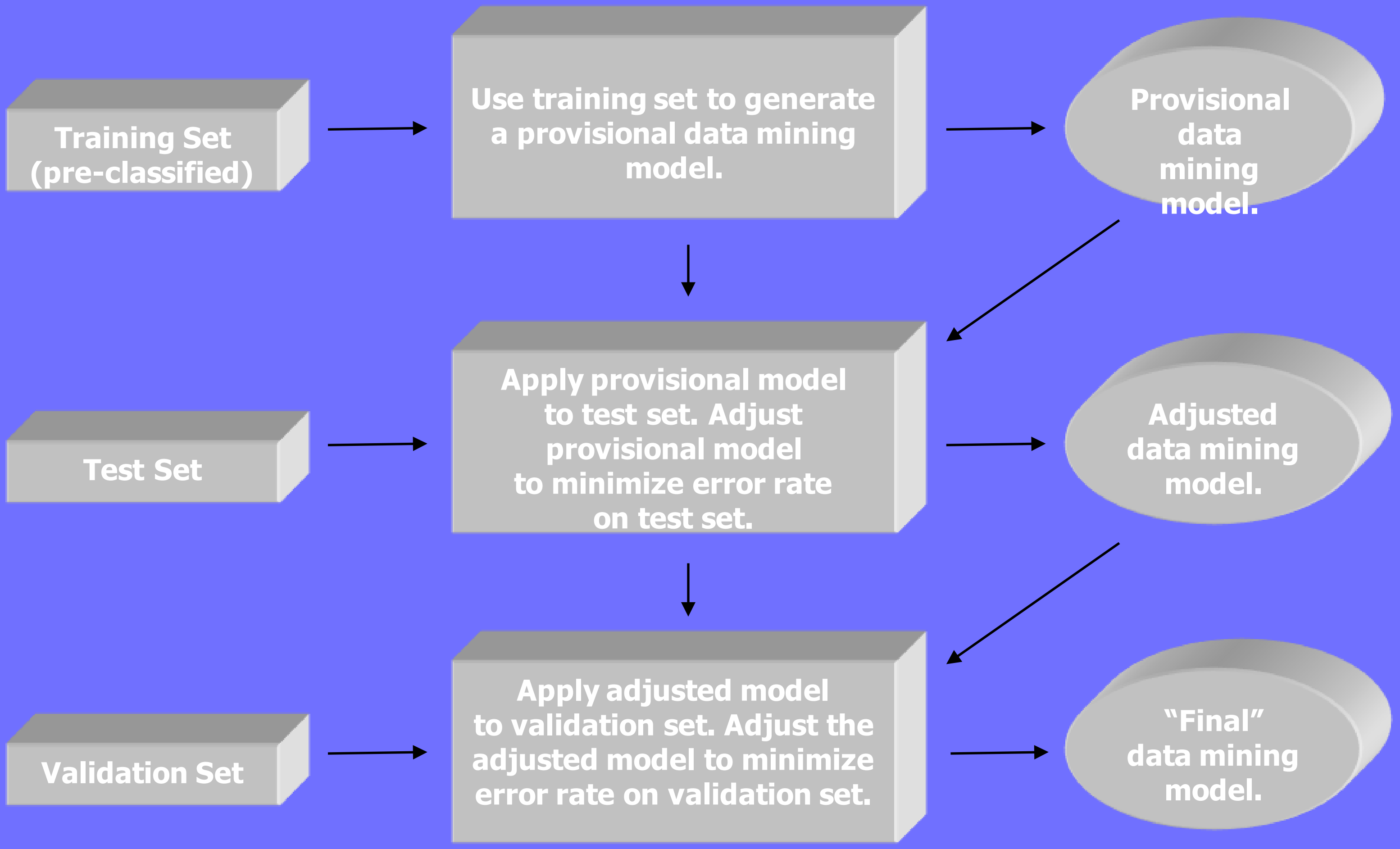
- Validation
  - Next, the adjusted data model is applied to another set of data called the Validation Set
  - The validation set is another "holdout" set of data independent of the training and test sets
  - The performance of the adjusted data model is evaluated against the validation set
  - If required, the adjusted data model is modified to minimize the error rate on the validation set
  - Estimates of data model performance for future, unseen data are computed using evaluative measures applied to results obtained when classifying the validation set

BIAS-Variance Trade off




Optimal Level of
Model Complexity
Error rate on
Validation Set
Error Rate
Error rate on
Training Set
Under fitting
Over fitting
Model Complexity

Methodology for Supervised Modeling (*cont'd*)



Data Mining View

- Traditional Analytics:
  - Have techniques that can be applied to different problems
- Data mining view
  - Needing a solution for a given problem no matter where it comes from
    - Example: How to treat a patient

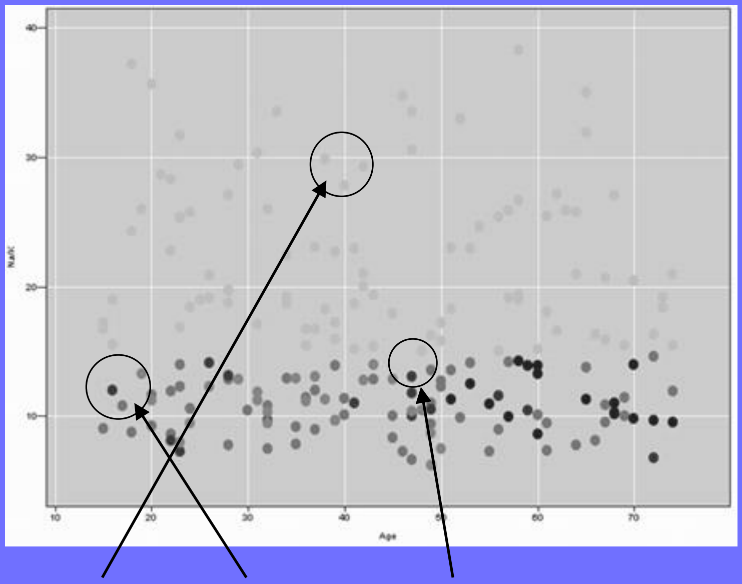
k-Nearest Neighbor Algorithm

- The *k*-Nearest Neighbor algorithm is an example of instance-based learning where training set records are first stored
- Next, the classification of a new unclassified record is performed by comparing it to records in the training set it is most similar to
- *k*-Nearest Neighbor is used most often 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 attributes
  - Our task is to classify the type of drug new a patient should be prescribed that is 40-years-old and has a Na/K ratio of 29

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

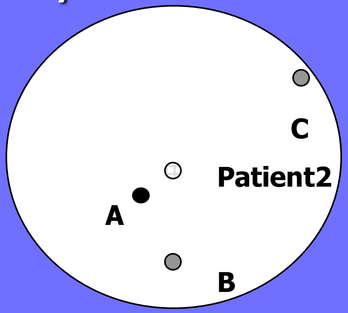
- This scatter plot of Na/K against Age shows the records in the training set that patients 1, 2, and 3 are most similar to
- A "drug" overlay is shown where Light points = drug Y, Medium points = drug A or X, and Dark points = drug B or C



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 shows the neighborhood of training points in close proximity to Patient 2



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

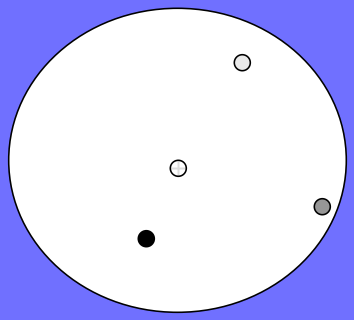
- Suppose we let $k = 1$ for our *k*-Nearest Neighbor algorithm
- This means we classify Patient 2 according to whichever <u>single point</u> in the training set it is closet to
- In this case, Patient 2 is closest to the Dark point, and therefore we classify them as drug B or C

- Suppose we let $k = 2$ and reclassify Patient 2 using *k*-Nearest Neighbor
- Now, Patient 2 is closest to a <u>Dark point and Medium point</u>
- How does the algorithm decide which drug to prescribe?
- A simple voting scheme does not help

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



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 <u>different target values</u>
- Considerations when using *k*-Nearest Neighbor
  - How many neighbors should be used? $k = ?$
  - How is the distance between points measured?
  - How is the information from two or more neighbors combined when making a classification decision?
  - 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?
- A distance metric is a real-valued function d used to measure the similarity between coordinates x, y, and z with properties:

$$
\begin{array}{ll}
1. & d(x,y) \geq 0, \text{and } d(x,y) = 0 \text{ if and only if } x = y \\
2. & d(x,y) = d(y,x) \\
3. & d(x,z) \leq d(x,y) + d(y,z)
\end{array}
$$

- Property 1: Distance is always non-negative
- Property 2: Commutative, distance from "A to B" is distance from "B to A"
- Property 3: Triangle inequality holds, distance from "A to C" must be less than or equal to distance from "A to B to C"

Distance Function (*cont'd*)

- The <u>Euclidean Distance</u> function is commonly-used to measure distance

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

where $\mathbf{x} = x_1, x_2, \ldots, x_m$, and $\mathbf{y} = y_1, y_2, \ldots, y_m$ represent the m attributes

- 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?

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

Distance Function (*cont'd*)

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

$$d_{\text{Minkowski}}(\mathbf{x}, \mathbf{y}) = \left(\sum_i (x_i - y_i)^p\right)^{1/p}$$

where $\mathbf{x} = x_1, x_2, \ldots, x_m$, and $\mathbf{y} = y_1, y_2, \ldots, y_m$

represent the m attributes

Distance Function (*cont'd*)

- Example 1: What is the distance between customer A and B?
  - Customer A age= 20-years-old Income= 10,000
  - Customer B age= 30-years-old Income= 20,000
- Example 2: What is the distance between customer A and B?
  - Customer A age= 20-years-old Income= 10K
  - Customer B age= 30-years-old Income= 20K

Distance Function (*cont'd*)

	Age	Income
A	20	10,000
B	30	20,000

	Age	Income
A	20	10
B	30	20

sqrt((10,000^2)+(10^2)) = 10,000

sqrt((10^2)+(10^2)) = 14.14

Distance Function (*cont'd*)

- Example 3: What is the distance between customer A and B?
  - Customer A income=$100k   Assets= $1 Million
  - Customer B income=$110k   Assets= $.7 Million
- Example 4: What is the distance between customer A and c?
  - Customer A income=$100k   Assets= $1 Million
  - Customer C income=$70k Assets= $1.1 Million

Remember to Normalize the data!!!

Distance Function (*cont'd*)

- 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

- Normalization
  - Continuous data values should be normalized using Min-Max Normalization or Z-Score Standardization

$$\text{Min-Max Normalization} = \frac{X - \min(X)}{\max(X) - \min(X)} \qquad \text{Z-Score Standardization} = \frac{X - \text{mean}(X)}{\text{standard deviation}(X)}$$

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,y)* for each categorical attribute in the Euclidean Distance function

- 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*

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 <u>do not normalize</u> *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 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
- Age Standardized using Z-Score
  - This time, *Age* is standardized using Z-Score Standardization

Distance Function (*cont'd*)

$$d_{Zscore}(A, B) = \sqrt{(0.33 - (-1.67))^2 + 0^2} = 2.0$$
$$d_{Zscore}(A, C) = \sqrt{(0.33 - 0.33)^2 + 1^2} = 1.0$$

- Using Z-Score Standardization, most values are typically contained in the range [-3, 3]
- Now, Patient C is nearest Patient A. This is different from the results obtained using Min-Max Normalization

- 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 technique is being used

Distance Function (*cont'd*)

- Note that the *distance(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

Combination Function

- The Euclidean Distance function determines the similarity of a new unclassified record to those in the training set
- How should the most similar (*k*) records combine to provide a classification?

- Simple Unweighted Voting
  - This is the most simple combination function
  - Decide on the value for *k* to determine the number of similar records that "vote"
  - Compare each unclassified record to its *k* nearest (most similar) neighbors according to the Euclidean Distance function
  - Each of the *k* similar records vote

Combination Function (*cont'd*)

- Recall that we classified a new patient 17-years-old with a Na/K ratio = 12.5, using *k* = 3
- Simple unweighted voting determined that two of the three closet points to Patient 2 are Medium
- Therefore, Patient 2 is classified as drug A or X with a <u>confidence</u> of 2/3 = 66.67%

- We also classified a new patient 47-years-old that has a Na/K ratio of 13.5, using *k* = 3
- However, simple unweighted voting did not help and resulted in a tie
- Perhaps weighted voting should be considered?

Weighted Voting

- Weighted Voting
  - In this case, the closer the neighbor, the more influence it has in the classification decision
  - This method assumes a closer neighbor is more similar, and therefore its vote should be weighted more heavily, as compared that of more distant neighbors
  - The weight of particular record is inversely proportional to its distance to the unclassified record
  - A "tie" is unlikely to occur using this approach

Weighted Voting (*cont'd*)

- Example

Record	Age	Na/K	Age_{MMN}	Na/K_{MMN}
New Patient	17	12.5	0.05	0.25
A (Dark)	16.8	12.4	0.0467	0.2471
B (Med)	17.2	10.5	0.0533	0.1912
C (Med)	19.5	13.5	0.0917	0.2794

Weighted Voting (*cont'd*)

- Example
  - Again, recall that we classified a new patient 17-years-old with a Na/K ratio = 12.5, using $k = 3$
  - We determined, using unweighted voting, two of the closest points were Medium, and the third was Dark
  - However, the Dark point is the most similar to the new patient
  - Now, we reclassify the new patient using a weighted voting scheme using values from the table below

Record	Age	Na/K	Age_{MMN}	Na/K_{MMN}
New Patient	17	12.5	0.05	0.25
A (Dark)	16.8	12.4	0.0467	0.2471
B (Med)	17.2	10.5	0.0533	0.1912
C (Med)	19.5	13.5	0.0917	0.2794

Weighted Voting (*cont'd*)

– The distance of records A, B, and C to the new patient are:

$$d(new, A) = \sqrt{(.05 - .0467)^2 + (.25 - .2471)^2} = .004393$$

$$d(new, B) = \sqrt{(.05 - .0533)^2 + (.25 - .1912)^2} = .058893$$

$$d(new, C) = \sqrt{(.05 - .0917)^2 + (.25 - .2794)^2} = .051022$$

– Next, the votes of these records are weighted according to the inverse square of their distance to the new record

– Record A votes to classify the new patient as Dark (drug B or C)

$$Votes(Dark\ Gray) = \frac{1}{d(new, A)^2} = \frac{1}{.004393^2} \cong 51{,}818.$$

Weighted Voting (*cont'd*)

– Records B and C vote to classify the new patient as Medium (drug A or X)

$$Votes(Medium\ Gray) = \frac{1}{d(new, B)^2} + \frac{1}{d(new, C)^2} = \frac{1}{.058893^2} + \frac{1}{.051022^2} \cong 672.$$

– Convincingly (51,818 vs. 672) the weighted voting method classifies the new patient as Dark (drug B or C)
– Note that this procedure reverses our classification decision determined using unweighted voting, *k* = 3
– The inverse distance of 0 is undefined using weighted voting
– Theoretically, the value of *k* could be increased, such that all training records participate in voting; however, the computational complexity may result in poor performance

Quantifying Attribute Relevance: Stretching the Axes

- Not all attributes may be relevant to classification
- For example, Decision Trees only include attributes that contribute to improving classification accuracy
- In contrast, *k*-Nearest Neighbor's default behavior is to calculate distances using all attributes
- A relevant record may be proximate for important variables, while at the same time very distant for other, unimportant variables
- Taken together, the relevant record may now be moderately far away from the new record, such that it does not participate in the classification decision

Quantifying Attribute Relevance: Stretching the Axes (*cont'd*)

- Perhaps, we should consider restricting the algorithm to using the most important fields for classification
- However, rather than making this determination *a priori*, we can make attributes either <u>more</u>, or <u>less</u> important
- This is accomplished using cross-validation or applying domain knowledge expertise

- Stretching the Axes
  - <u>Stretching the Axes</u> finds the coefficient z_j by which to multiply the j_{th} axis. Larger values of z_j are associated with the more important variable axes
- Cross-validation
  - Cross-validation selects a random subset of data from the training set and determines the set of z_1, z_2, ..., z_m that minimize the classification error on the test set

Quantifying Attribute Relevance: Stretching the Axes (*cont'd*)

- Repeating the process leads to a more accurate set of values for $z_1, z_2, \ldots, z_m$

- Domain Expertise
  - Alternately, we may call upon domain experts to recommend values for $z_1, z_2, \ldots, z_m$
  - Using either approach the *k*-Nearest Neighbor algorithm may be made more precise

- Example
  - Suppose that the Na/K ratio was determined to be <u>3 times more important</u> than the Age attribute, for performing drug classification

Quantifying Attribute Relevance: Stretching the Axes (*cont'd*)

- The distance of the records A, B, and C to the new record are calculated as follows:

$$\text{where } z_{Na/K} = 3,\ z_{Age} = 1$$

$$d(new, A) = \sqrt{(.05 - .0467)^2 + ((3)(.25 - .2471))^2} = .009305$$

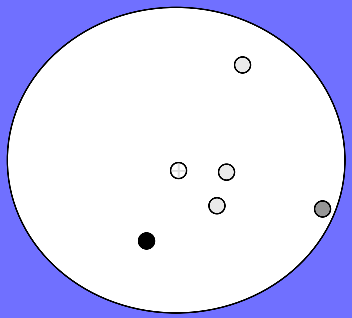
$$d(new, B) = \sqrt{(.05 - .0533)^2 + ((3)(.25 - .1912))^2} = .17643$$

$$d(new, C) = \sqrt{(.05 - .0917)^2 + ((3)(.25 - .2794))^2} = .097561$$

- The classification does not change by stretching the axes for Na/K ratio
- In many situations, stretching the axes leads to improved accuracy by quantifying the relevance of each variable used in the classification decision

Database Considerations

- Instance-based learning methods benefit from having access to learning examples composed of <u>many attribute value combinations</u>
- The data set should be <u>balanced</u> to include a sufficient number of records with common, as well as less-common, classifications
- One approach to balancing the data set is to <u>reduce</u> 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"



k-Nearest Neighbor Algorithm for Estimation and Prediction

- *k*-Nearest Neighbor 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

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

$$\hat{y}_{new} = \frac{\sum_i w_i y_i}{\sum_i w_i}$$

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

- 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 idiosyncratic or obscure data values in the training set
  - It the values become too large, locally interesting values will be overlooked

Choosing *k* (*cont'd*)

- Choosing the appropriate value for *k* requires balancing these considerations
- Using cross-validation may help determine the value for *k*, by choosing a value that minimizes the classification error

Reference

- Text:
  - Chapter 3: Exploring Categorical Variables
  - Chapter 4: Statistical approaches to estimation and prediction -- confidence interval estimation
  - Chapter 5: Entire chapter
- SAS and R
- Next topic
  - K-means chapter 8