slides originally by
Dr. Richard Burns,
modified by
Dr. Stephanie Schwartz

DATA PREPROCESSING

CSCI 452: Data Mining

Topics Covered

- Handling Missing Data
- Reclassifying Categorical Variables
- Binning Numerical Variables
- When to Remove Variables

Handling Missing Data

- *Motivation:* Will frequently encounter missing values, especially in big data
  - Lots of fields, lots of observations
- *Question:* how to handle missing data?

How much data is missing?

- Suppose…
  - Dataset of 30 variables
  - 5% of data is missing
  - Missing values are spread evenly throughout data
- Then 80% of records would have at least one missing value

What to do? The options:

1. Omit records with any missing values (bad)
2. Replace missing value with some constant specified by analyst
3. Replace missing value with mean, median, or mode
4. Replace missing value with value generated at random from the observed variable distribution
5. Replace missing value with *imputed values*, based on other characteristics of the record

Replacing missing value with imputed values

- *"The value of an item for which actual values are not available. Imputed values are a logical or implicit value for an item, or time set, wherein a "true" value has yet to be ascertained. It would be a best guess estimate, in order to accurately estimate a larger set of values or series of data points."*

Imputation of Missing Data

1. Impute the values of the variable with the *fewest* missing values.
   - Use only the variables with no missing values as predictors.
2. Impute the values of the variables with the next fewest missing values.
3. Repeat as necessary.

Reclassifying Categorical Variables

- Sometimes a categorical variable will contain *too many* factors to be easily analyzable
  - Logistic regression and decision trees sometimes yield poor predictive performance in this case
- *Example:* state field could contain 50 different values
  - *Solution #1:* reclassify state into its region: {*NorthEast, NorthWest, Central, West, …*}
  - *Solution #2:* reclassify state by its economic level: {*WealthyStates, MiddleStates, PoorestStates*}
  - Up to the analyst to appropriately reclassify

When to Remove Variables

- Variables that will not help the analysis should be removed
  - Unary variables: take on a single value
    - *Example:* gender variable for students at an all-girls school
  - Variables that are *nearly* unary
    - *Example:* gender of football athletes at elementary school
      - 99.95% of the players are male
    - Some data mining algorithms may treat the variable as unary
      - Not enough data to investigate the female players anyway…

When (Not) to Remove Variables

- 90% of the values are missing
  - Are the values that are present representative or not?
  - If the present values are representative, then either (1) remove the variable or (2) impute the values.
  - If the present values are non-representative, their presence adds value.
    - *Scenario: donation_dollars* field in a self-reported survey
    - *Possibility:* those who donate a lot are more inclined to report their donation
    - Could make the variable binary: *donation_flag*

When (Not) to Remove Variables

- Strong correlation between two variables
  - Inclusion of correlated variables may "double-count" a particular aspect of the analysis, depending on the machine learning technique used.
  - *Example: precipitation and people on a beach*
  - *Strategy #1:* remove one of the two correlated variables
  - *Strategy #2:* use PCA to transform the variables (Principal Component Analysis)

ID Fields

- ID fields have a different value for each record
- Won't be helpful in predictive analysis
  - If they are, the relationship is usually spurious
- *Recommended Approach:*
  - Don't include ID field in modeling
  - But keep it in the dataset to differentiate between records

Aggregation

- Combining two or more *attributes* into a single *attribute*
  - Or – combining two or more *objects* into a single *object*
- *Purpose:*
  1. Data reduction: reduce # of attributes/objects
  2. Change of scale: high-level vs. low-level
  3. More “stability”: aggregated data has less variability

Aggregation – Data Reduction

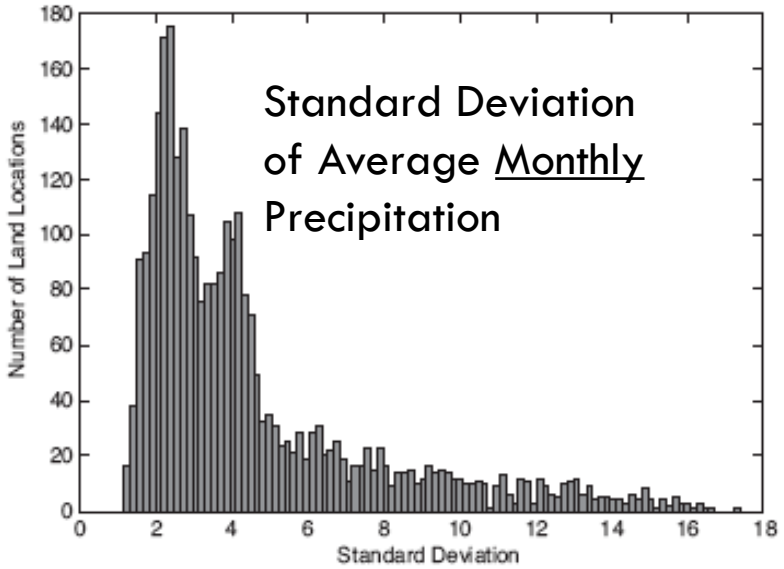
- "less is more" in some research questions
- *Example:*
  - Combining individual sales transactions into aggregate sales
  - *Action:* remove "store location" and "item"
    - Sum sales for a day
- *Motivations:*
  - Higher-level view of data?
  - Less memory and processing time

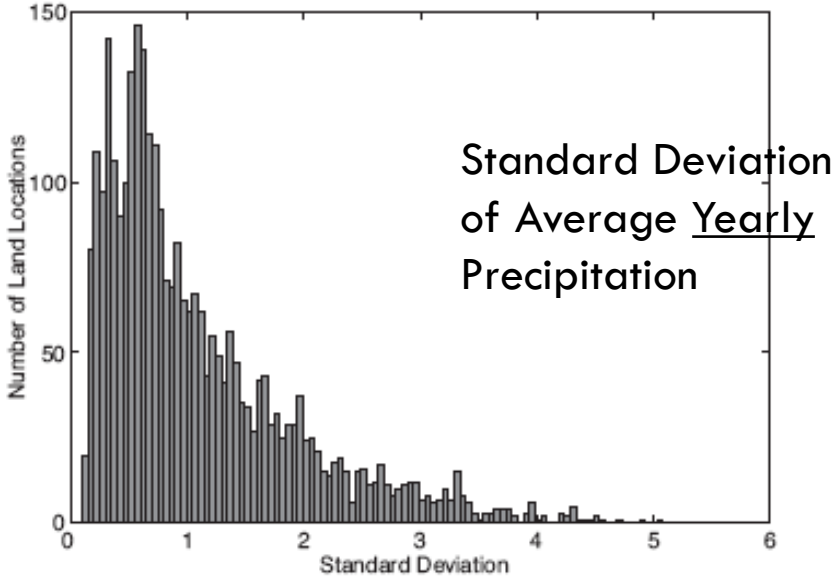
Aggregation – Data Reduction

Transaction ID	Item	Store Location	Date	Price	…
…	…	…	…	…	
101123	Watch	Chicago	09/06/04	$25.99	…
101123	Battery	Chicago	09/06/04	$5.99	…
101124	Shoes	Minneapolis	09/06/04	$75.00	…
…	…	…	…	…	

Aggregation – Change in Variability

- Less variability at "higher-level" view





Variation of Precipitation in Australia

Binarization

- Mapping a *categorical* attribute to a set of attributes that are *binary*
- Assuming an *ordinal attribute*:
  - ({*Small, Medium, Large*})
  - Must maintain ordinal relationship

Binarization

Question: How many binary variables are required to represent 5 categorical values?

Categorical Value	Integer value	X_1	X_2	X_3
Awful	0	0	0	0
Poor	1	0	0	1
OK	2	0	1	0
Good	3	0	1	1
Great	4	1	0	0

Answer: 3 binary variables

$$\overline{\log_2(x)} = \text{ceiling of } \log_2(x) = \overline{\log_2(3)} = 3$$

Question: any issues with this approach?

- Unintended relationships
  - X_2 and X_3 are now correlated because "good" is encoded using both attributes

Binarization

Categorical Value	Integer value	X_1	X_2	X_3	X_4	X_5
Awful	0	1	0	0	0	0
Poor	1	0	1	0	0	0
OK	2	0	0	1	0	0
Good	3	0	0	0	1	0
Great	4	0	0	0	0	1

- Binary attributes, where only the presence of 1 is important
- One binary attribute for each categorical value
- *Be Careful:*
  - Number of resulting attributes may become too large

Discretization

- Discretization: Mapping / “binning” *continuous* attributes to *categorical* attributes

Binning Numerical Variables

- Some algorithms prefer *categorical* variables rather than *continuous* variables
  - *Example:* decision trees
- Need to discretize numerical variable into bins
  - *Example:* "binning" the numerical predictor variable *house value* into {Low, Medium, High}

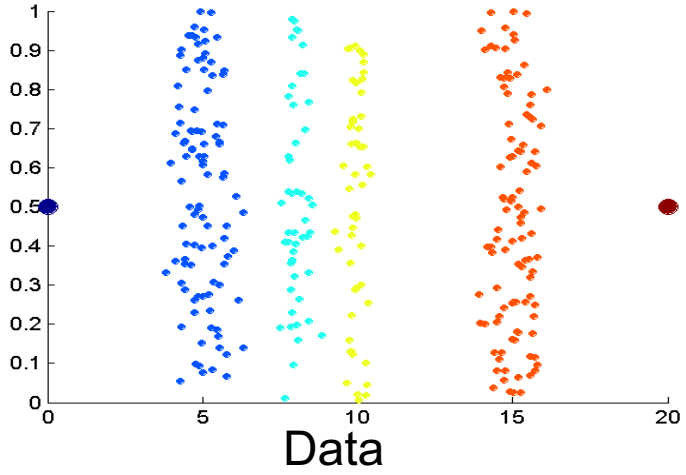
Methods for Binning Numeric Predictor Variables

1. Equal width binning
2. Equal frequency binning
3. Binning by clustering
4. Binning based on predictive value

Equal Width Binning

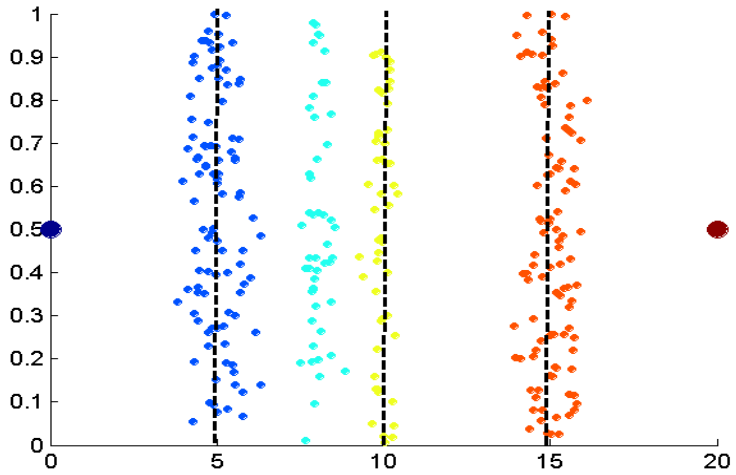
- Divides the numerical predictor into *k* categories of equal width
  - *k* is chosen by the analyst
- Not usually recommended
  - Can be greatly affected by outliers

Equal Width



Data

- *4* classes in target attribute (indicated by color)
- x-axis: value of some continuous predictor variable that we wish to discretize
- Two outliers (at x=0 and x=20)



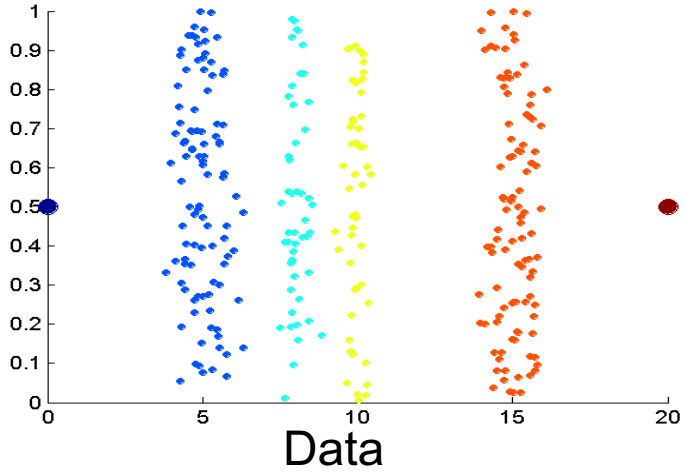
Equal width

- *k=4* chosen by analyst

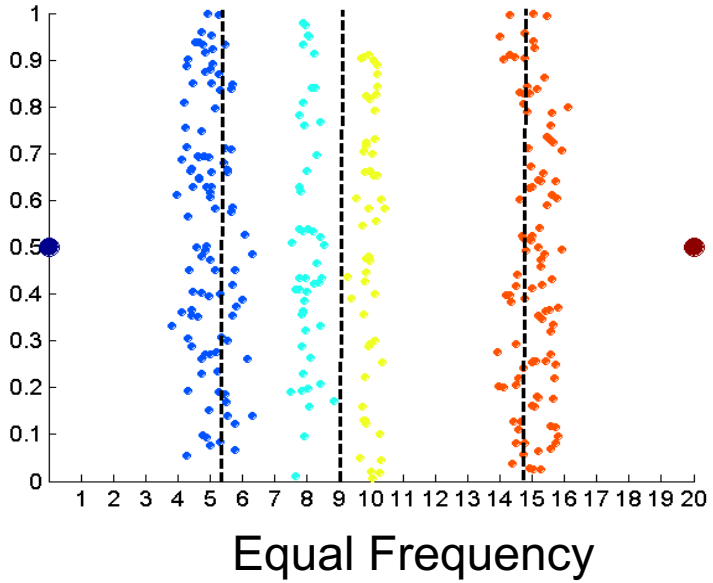
Equal Frequency Binning

- Divides the numerical predictor into k categories, each having n/k records,
  - Where there is n records
- Assumes that each category is equally likely
  - Assumption is usually not warranted

Equal Frequency



- 4 classes in target attribute (indicated by color)
- x-axis: value of some continuous predictor variable that we wish to discretize
- Two outliers (at x=0 and x=20)

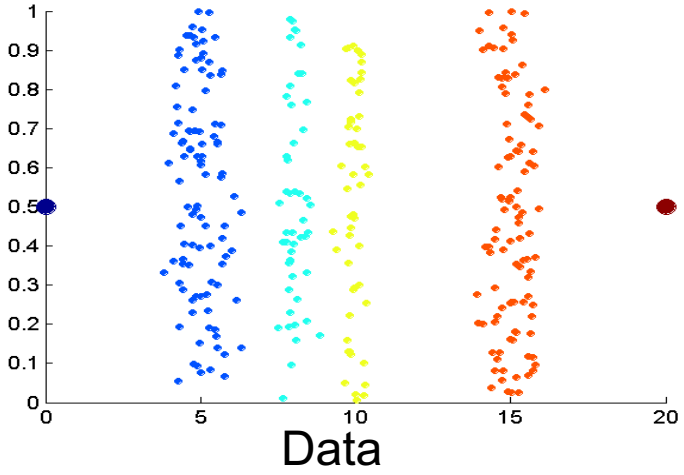


- k=4 chosen by analyst
- n/k observations in each bin
  - 100/4=25 observations in each bin

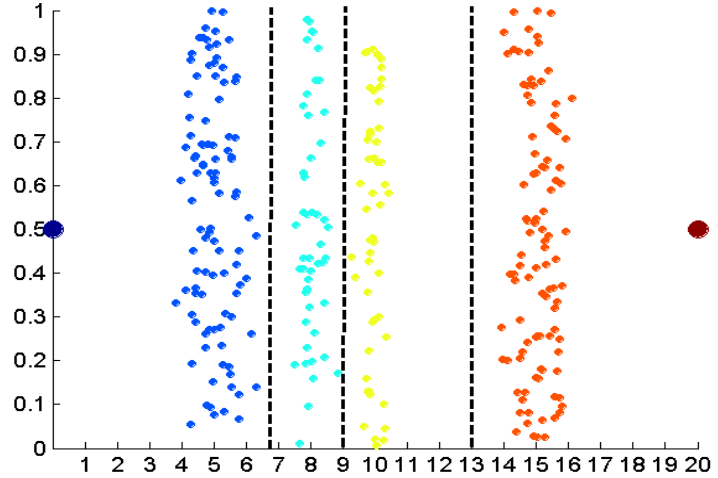
Binning by Clustering

- Using a “clustering” algorithm to learn “optimal” binning
  - *Example:* k-means clustering
- Will be covered later in course.

K-Means



- 4 classes in target attribute (indicated by color)
- x-axis: value of some continuous predictor variable that we wish to discretize
- Two outliers (at x=0 and x=20)



K-Means

- k=4 chosen by analyst
- "Seems to work better"

Binning based on Predictive Value

- Partitions the numerical predictor based on the effect each partition has on the value of the target variable
- *(Previous three methods ignored the target variable.)*
- Supervised vs. Unsupervised

Supervised vs. Unsupervised

- Unsupervised: discretization of observation based solely on data point
  - No knowledge of class label
- Supervised: class label is known and is used

- Unsupervised discretization usually better than no discretization.
- Supervised discretization sometimes produces better results.

Supervised Discretization

- Split *continuous* variable in such a way to maximize the "purity of the intervals"
- Entropy:

$$e_i = -\sum_{i=1}^{k} p_{ij} \log_2 p_{ij}$$

$$p_{ij} = \frac{m_{ij}}{m_i}$$

k = # of class labels

m_i = # of observations in the ith interval

m_{ij} = # of values of class j in interval i

e_i = entropy of interval i

Total entropy: weighted average of the individual interval entropies

Entropy

10 data points with classes: {A,B,C,A,A,B,B,C,C,C}

- 3 intervals
- 1st interval has {A,B,C,A,A}
- 2nd interval has {B,B}
- 3rd interval has {C,C,C}

- Entropy: measure of "purity of an interval"
- *Examples:*
  - Interval containing only values of one class
    - 2nd and 3rd intervals are perfectly pure

$$Entropy_{2nd\ \text{interval}} = \frac{0}{2}\log_2\frac{0}{2} + \frac{2}{2}\log_2\frac{2}{2} + \frac{0}{2}\log_2\frac{0}{2} = 0 + 0 + 0 = 0$$

  - Interval containing a mix of class labels

$$Entropy_{1st\ \text{interval}} = \frac{3}{5}\log_2\frac{3}{5} + \frac{1}{5}\log_2\frac{1}{5} + \frac{1}{5}\log_2\frac{1}{5} = -.442 - .464 - .464 = 1.37$$

Simple Supervised Discretization Algorithm

1. Bisect initial values so that resulting 2 intervals give minimal entropy
2. Repeat with another interval, (typically choosing the interval with the worst/highest entropy)
3. Stop when user-specified number of intervals is reached (or some other stopping criteria)

Supervised Discretization

Training

- Class labels (of target variable) are used for discretization of predictive variable
- Bins / "Bin definitions" are formed

Testing

- Discretize predictive variable into bin, as defined from training
- Use model learned from training data to predict class label

References

- *Discovering Knowledge in Data*, 2nd edition, Larose et al.
- *Introduction to Data Mining*, 1st edition, Tan et al.