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

FEATURE SUBSETS

CSCI 452: Data Mining

High Dimensionality

- … can be bad
- Datasets can have a large number of features
  - *Example:* stock prices (time series)
    - Each stock is individual instance
    - Features/variables are closing price on given day
      - Imagine 30 years worth of closing prices (30 x 365)

Why is it a Problem?

BAD: $p > n$,
p = # of features
n = # of instances

- Many times data mining algorithms work better if there is not an overwhelming number of attributes
  - The "dimensionality" is lower
- "The Curse of Dimensionality"
- As dimensionality increases (more features), the data becomes increasingly sparse in the "feature space" that it occupies.
  - Not enough data objects for the number of features that are present
    - Reduced classification model accuracy

Other Benefits to Dimensionality Reduction

1. More understandable models
   - Learned model may involve fewer attributes
2. Better visualizations
   - Fewer attributes = less variables to plot
3. Computational time
   - Fewer attributes = quicker model learning?
4. Elimination of irrelevant features

Techniques for Dimensionality Reduction

1. Linear Algebra Techniques
   - Automatic approaches
   - Project data from high-dimensional space into a lower-dimensional space
     1. Principal Components Analysis (PCA)
     2. Singular Value Decomposition (SVD)
   - Not necessarily interested in "losing information"; rather eliminate some of the sparsity

Techniques for Dimensionality Reduction

2. Feature Construction
   - *Example*: combining two separate features (# of full baths, # of half baths) into one feature ("total baths")
   - *Example*: combining features (mass) and (volume) into one feature (density), where density = mass / volume

Techniques for Dimensionality Reduction

3. Feature Subset Selection
   - Reducing number of features by only using <u>a subset of features</u>
     - How many should be in the subset?
   - Losing information if we only consider a subset of features?
     - Redundant features
       - *Example:* (1) purchase price and (2) sales tax
     - Irrelevant features
       - *Example:* student id numbers
   - By eliminating unnecessary features, we hope for a better model.

Eliminating Redundant and Irrelevant Features

1. Manually via Data Analyst
   - Intuition about problem domain
2. Systematic Approach
   - Try all possible combinations of feature subsets?
     - See which combination results in best model
   - For *n* features, there are 2^n possible combinations of subsets
     - Infeasible to try each of them

Three Systematic Approaches

1. Embedded Approaches
2. Filter Approaches
3. Wrapper Approaches

Embedded Approaches

- Algorithm specific
- Occurs naturally as part of the data mining algorithm
  - *Example:* present in decision tree induction
    - Only certain subset of features are used in final decision tree
  - *Example:* not present in linear regression
    - Fitted model contained coefficient for each predictor variable

Filter Approaches

- Features are selected <u>before</u> the data mining algorithm is run
- Filter approach is <u>independent</u> of the data mining task
- *Example: (trying to eliminate redundant features)*
  1. Look at pairwise correlation between variables
  - Pick subset of variables that each have low pairwise correlation
  2. Then use only that subset in Linear Regression model.

Wrapper Approaches

- Data mining algorithm is a “black box” for finding best subset of features
  - Tries different combinations of subsets
  - Typically will never enumerate all 2^n possible combinations
    - Will search a feature space that is much smaller
  - Final model uses the specific subset that evaluates the best

Top-Down Wrapper

- *Assuming n number of features…*
- Start with no attributes

1. Train classifier *n* times, each time with a different feature
   - Each classifier only has a single predictor
   - See which of the *n* classifiers <u>performs the best</u>
2. Add to the best classifier. Recursively use remaining attributes to find which attribute that improves performance the most
   - Keep including best attribute

- *Stopping criterion:* Stop if no improvement to classifier performance, or classifier performance is less than some threshold

Bottom-Up Wrapper

- *Assuming n number of features…*
- Start with all *n* attributes in model
- Create *n* models, each with a different predictor omitted.
  - Each classifier has *n-1* predictors
  - See which of the *n* classifiers <u>affects performance the least</u>
  - Throw that attribute out
- Recursively find the attribute that affects performance the least
- *Stopping criterion:* Stop if classifier performance begins to degrade

Other Wrappers

- Bi-Directional
  - Combining Top-Down and Bottom-Up
- Greedy Search with Backtracking
  - *(if you're familiar with AI)*
- …

Adjusted R^2 Statistic

Always <u>increases</u> as more variables are added to the model.

- Recall the R^2 statistic that we use in Linear Regression:
  - Measured the <u>proportion of variance</u> explained by the model
  - Always a value between 0 and 1
  - Higher is better

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

$$TSS = \sum (y_i - \bar{y})^2$$

$$RSS = \sum (y_i - \hat{y}_i)^2$$

Adjusted R^2 Statistic

- In contrast to R^2, Adjusted R^2 penalizes for unnecessary variables in the model.
- d = *number of predictors*
- n = *number of instances*

$$Adjusted\ R^2 = 1 - \frac{RSS/(n-d-1)}{TSS/(n-1)}$$

$$TSS = \sum (y_i - \bar{y})^2$$

$$RSS = \sum (y_i - \hat{y}_i)^2$$

References

- *Introduction to Data Mining*, 1st edition, Tam et al.
- *Discovering Knowledge in Data*, 2nd edition, Larose
- *An Introduction to Statistical Learning*, 1st edition, James et al.