CIS501 – Lecture 11

Woon Wei Lee
Fall 2013, 10am-11:15am,
Sundays and Wednesdays
(and occasionally, Tuesdays ;-))

For today:

- Administrative stuff
- Continuous Variables
- Missing values
- Decision Tree: Advantages and Disadvantages
- Presentations
  - Shoroye Zakariyah
  - Ahmed Nasser

Decision trees + continuous variables

- **For simplicity, we have so far only dealt with discrete/nominal variables**
  - Easy because discrete values provide natural granularity
- **However decision trees can also work with continuous variables**
  - Threshold on the variables
  - Infinite possible values... what should we do?
- **Continuous valued variables have infinite possible values but..**
  - we only have *N* values in training set
  - Treat this as *N* possible features, evaluate IG on each
  - Choose the best, etc. etc.

Golf data revisited

Outlook	Temp (°F)	Humidity (%)	Windy	Class
sunny	75	70	true	play
sunny	80	90	true	don't play
sunny	85	85	false	don't play
sunny	72	95	false	don't play
sunny	69	70	false	play
overcast	72	90	true	play
overcast	83	78	false	play
overcast	64	65	true	play
overcast	81	75	false	play
rain	71	80	true	don't play
rain	65	70	true	don't play
rain	75	80	false	play
rain	68	80	false	play
rain	70	96	false	play

Classes
play, don't play
Outlook
sunny, overcast, rain
Temperature
numerical value
Humidity
numerical value
Windy
true, false

Continuous feature

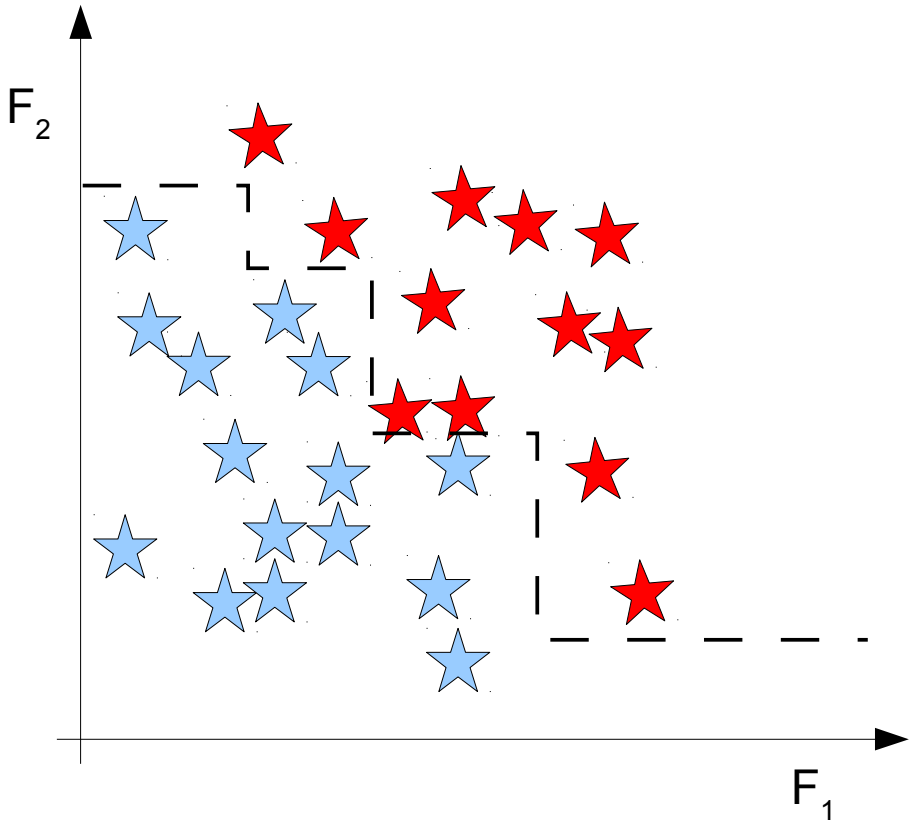
(Cont'd)

- **Focus on temperature**

64	65	68	69	70	71	72	72	75	75	80	81	83	85
Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No

- **Try splitting on alternative points, for e.g.**
  - Temperature < 71.5: yes/ 4, no/2; Temperature ≥ 71.5: yes/5, no/3
  - Info([4,2],[5,3]) = 6/14 info([4,2]) + 8/14 info([5,3]) = 0.939 bits
  - (Split all value pairs down the middle, and evaluate for each)
- **Further, split points which divide between points of same class cannot be optimal**
  - Therefore, only evaluate split points between points with different class labels
- **Note, that it is now possible to split on the same feature more than once in the same branch..**

Graphical depiction



- For decision tree, classification boundary lies along individual features at a time
- On one hand – high flexibility, potentially nonlinear decision boundary
- On other hand, complicated, awkward and unnatural..
- Curse of dimensionality: lines → hypersurfaces, compounds problem..

Missing Values

- **A major problem in data mining is that of *missing data***
- **Causes:**
  - Data removed due to corruption
  - Incomplete data collected
  - Equipment failure, etc..
  - (It's a fact of life!)
- **Several strategies for coping:**
  1. Discard offending instances
     - Probably easiest strategy
     - Might work well in simple cases
     - Impractical in general → too wasteful!

Outlook	Windy	Class
Sunny	TRUE	don't play
Sunny	FALSE	don't play
?	FALSE	don't play
Rain	TRUE	don't play
Rain	TRUE	don't play
Sunny	TRUE	play
Sunny	FALSE	play
Overcast	TRUE	play
Overcast	?	play
Overcast	?	play
Overcast	FALSE	play
Rain	FALSE	play
?	FALSE	play
?	FALSE	play

Missing Values (Cont'd)

2. Allow “?” values
    - i.e. “Don't know” value
    - Treat as a separate value, and split accordingly
    - Inappropriate where we “don't know” may not be a suitable description!

      Q: Can you name some examples?
3. Assign most probable value
    - Two variants, (1) Simply substitute attribute with most common value for that attribute at that node
    - (2) Calculate the probabilities for each of the possible attribute values
    - Propagate these down the respective branches on the tree.
    - (used by C4.5)

C4.5 Decision Tree algorithm

- **ID3 is venerable, but for a variety of reasons is impractical for many realistic datasets.**
  - The C4.5 algorithm attempts to address many of the shortcomings of ID3.
- **In particular:**
  - Choosing an appropriate attribute selection measure.
  - Avoiding overfitting the data
    - Determining how deeply to grow a decision tree.
    - Reduced error pruning.
  - Handling continuous attributes
  - Handling training data with missing features.
- **(Exact details vary amongst different implementations but broad principles apply)**

CART Decision Tree algorithm

- **Stands for "Classification And Regression Trees",**
- **Similar objectives as C4.5 (reduce over-fitting, handle missing values, etc..**)
- **But, different set of techniques:**
  - GINI index for split selection
  - Cost-complexity pruning
  - Binary only splits
  - Supports *regression*
    - (Splitting criterion is the mean squared error)
  - Missing data handled using combination of techniques:
    - During training, instances with missing attributes are discarded
    - *Surrogate splits* used for handling instance during classification
    - Use features which are highly correlated to the missing feature

Rule induction using decision trees

- **Want to generate rules of the form:**
  - *if A and B and C … then class X*
- **C4.5 can be used to auto-generate rules from unpruned tree:**
  - Each path from root to leaf gives possible simplified rule
  - A, B, C, from above are simply conditions on path and X is class at leaf
  - Prune using estimated error
  - Organize rules according to class (to form “rulesets”)
  - Order class rulesets by number of false positives
  - Find “optimal” subsets of rules with class rulesets (using MDL principle)
- **To classify case using ruleset:**
  - Check class rulesets in turn
  - If case satisfies any rule in ruleset, assign case to that class
  - If no rulesets match → assign case to default class

Decision Trees: advantages and disadvantages

- **Advantages:**
  - Able to simultaneously handle numerical/categorical data in the same classifier
  - Conceptually simple – contrast with “black box” approaches like kNN, Neural Nets..
  - Easy to set-up – no tricky hyperparameters to select, etc.
  - Can be very efficient if well designed – only a subset of parameters may be used for each clasification
- **Disadvantages**:
  - Trees may not partition input space efficiently
  - Does not take into account closeness to boundary, confidence, etc..
  - Learning process is heuristic and can often result in overfitting of data..
  - Difficult to intepret in Bayesian Framework