CSI-777
Principles of Knowledge Mining

Class 9
Data Transformations & Project Management

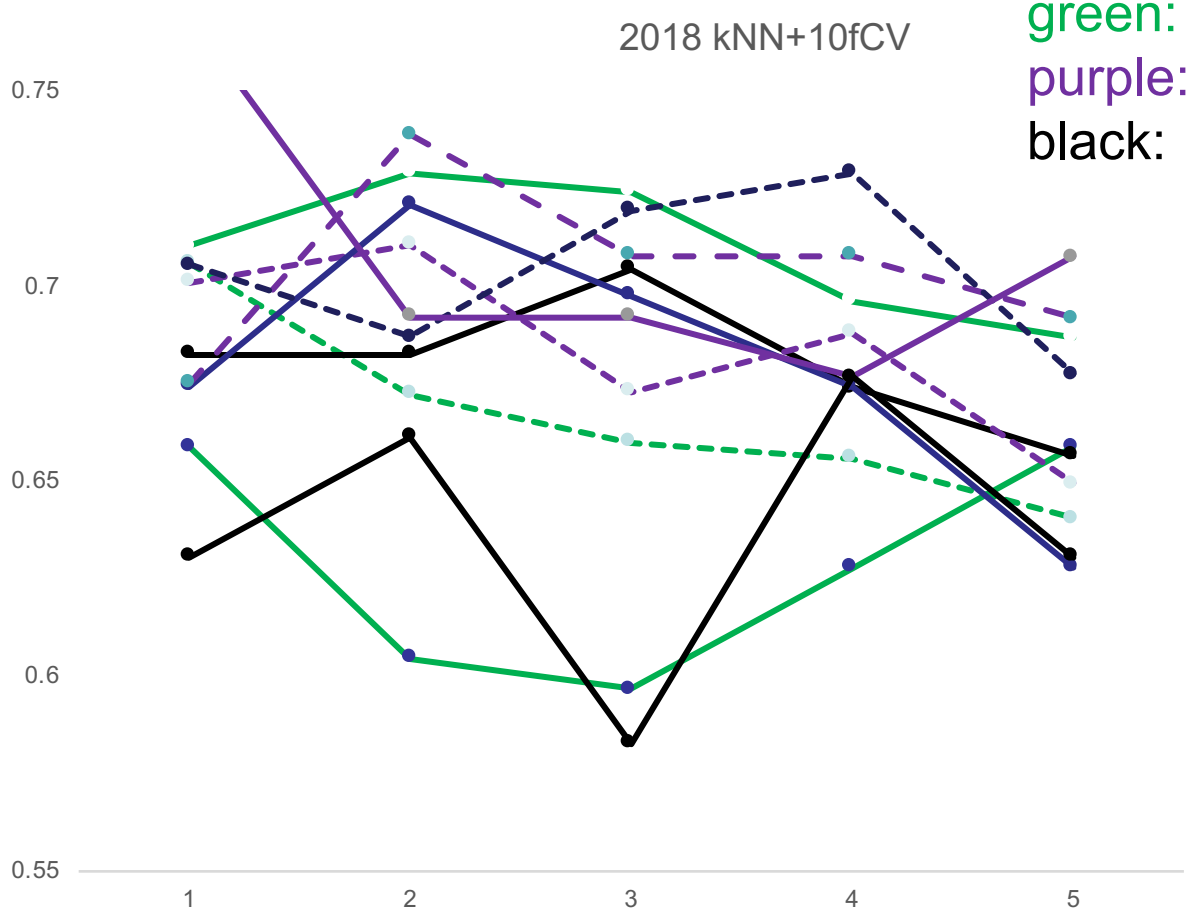
William G. Kennedy,
PhD, CAPT, USN (Ret.)
Center for Social Complexity
Computational and Data Sciences Dept.
College of Science

Review of Previous Class

Review of Homework

HW#3 kNN & 10 fold CV

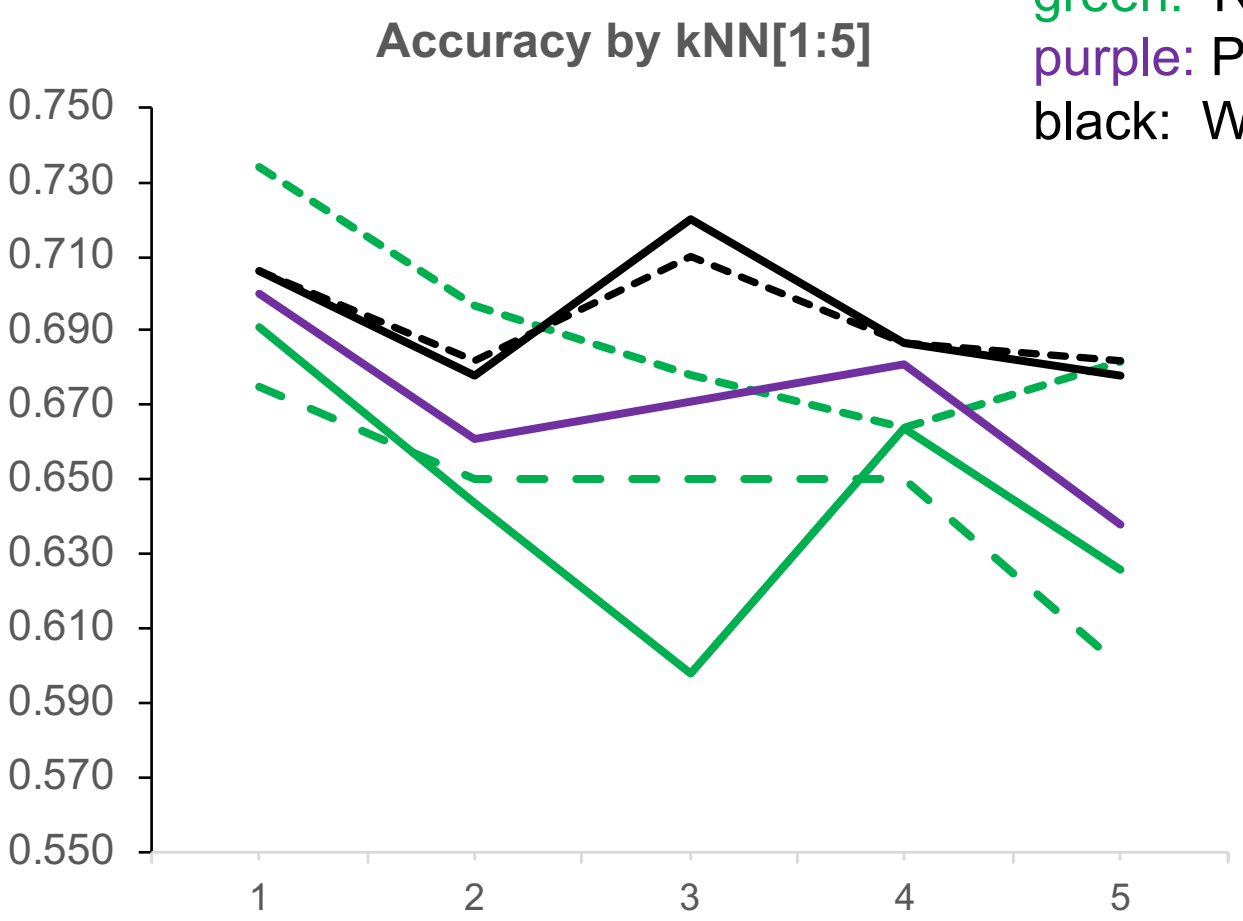
2018 Results

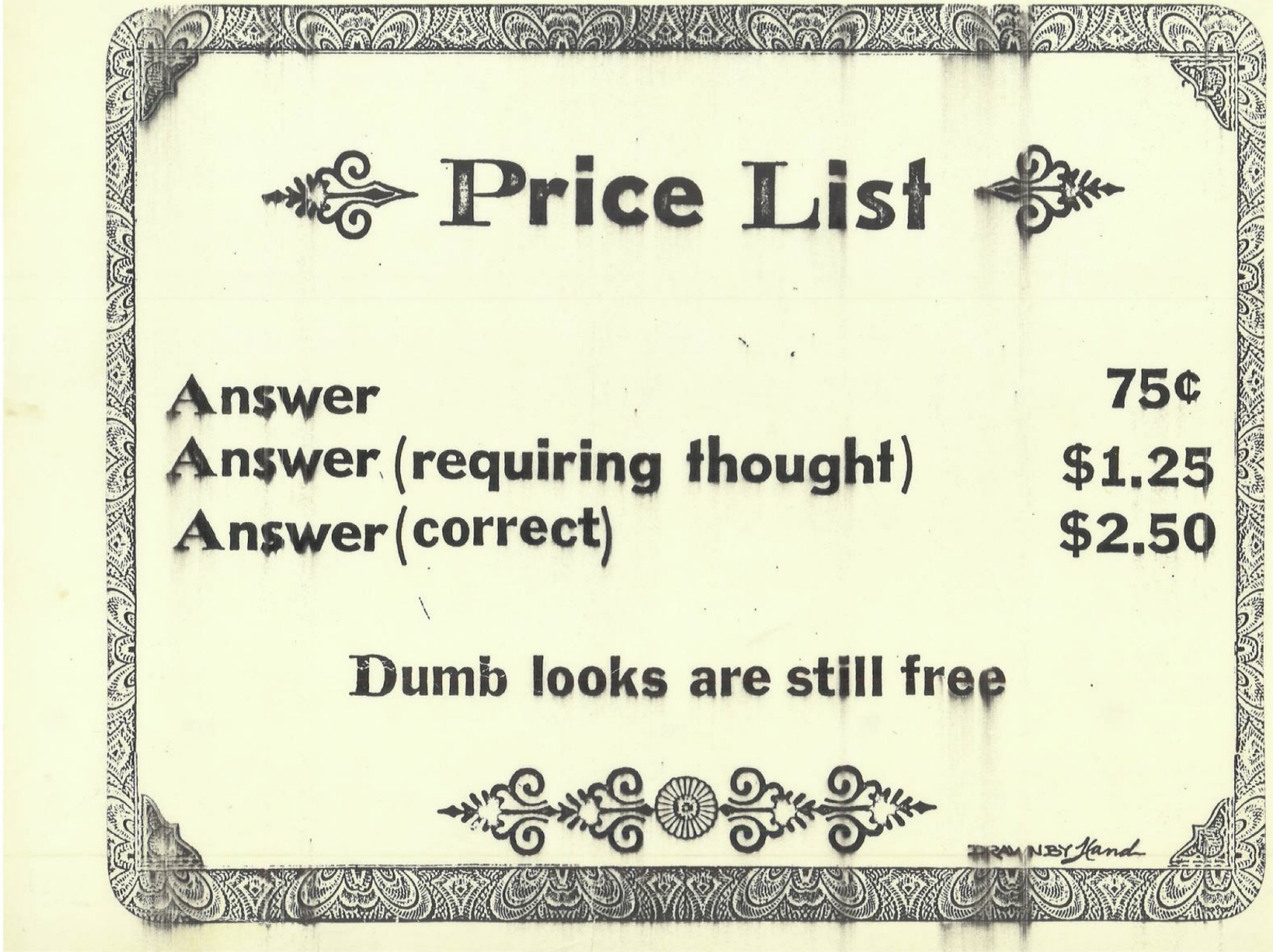


green: R,R-Studio
purple: Python
black: Weka

2017 Results

green: R,R-Studio
purple: Python
black: Weka




Price List
Answer 75¢
Answer (requiring thought) $1.25
Answer (correct) $2.50
Dumb looks are still free
DRAWN BY Hand

Advanced Approaches 2

1. Instance-based Learning
2. Extending Linear Models
3. Local Linear Models

Instance-Based Learning

- Practical problems with basic k-NN:
  - Slow for large datasets, one pass through for each test instance
  - Doesn’t handle noisy data well
  - Presumes balanced impact of attributes
  - Doesn’t explicitly generalize

Instance-Based Learning

Which exemplars to keep?

- Keep all… unnecessary
- Discard correctly classified?
  - Ideal: 1 exemplar per class
  - But may need “errors” to adapt, but …
- Keep errors?
  - If noisy, keeping noise…

Instance-Based Learning

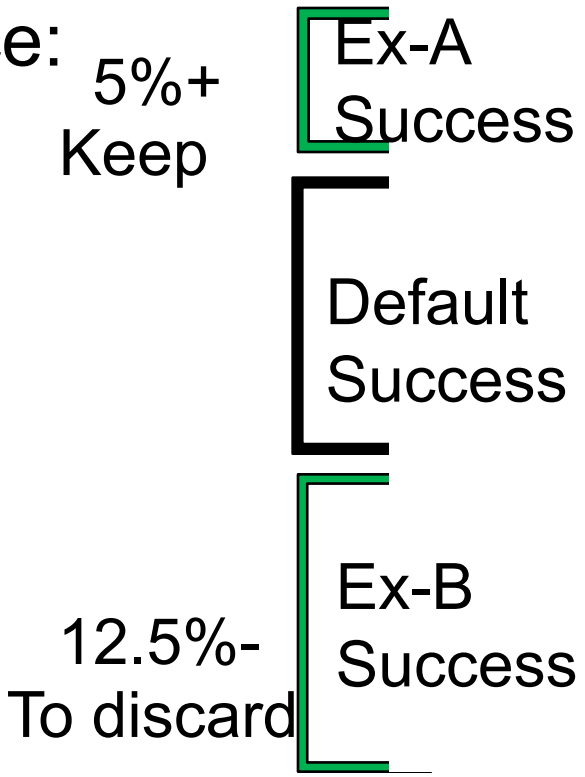
Noisy exemplars

- Keeping them causes problems
- 1st option: Keep k NN, rather than 1, and vote
- Noise ↑ need k ↑, explore several values…

Instance-Based Learning

Noisy exemplars

- Keep stats on performance:
  - Too low, discard
  - High, use
  - Between, keep around
- Confidence limits
  - Level & bounds
  - IB3



Instance-Based Learning

Weighting Attributes

- Basic: same distance measure for all but…
- Learn relevance of each attribute (+/-)
- Updated with each training instance

Instance-Based Learning

Generalized Attributes

- Rectangular exemplars (“hyperrectangles”)
- +: Generalized by merging with nearest
- -: Shrunk away from new instance
- Overgeneralize by growing allowed Y/N

Instance-Based Learning

Distance Functions

- With generalized exemplars, simplify measure
- 0 or dist. to closest part of rectangle
- Generalized distance measure: use all transformations to a similar class

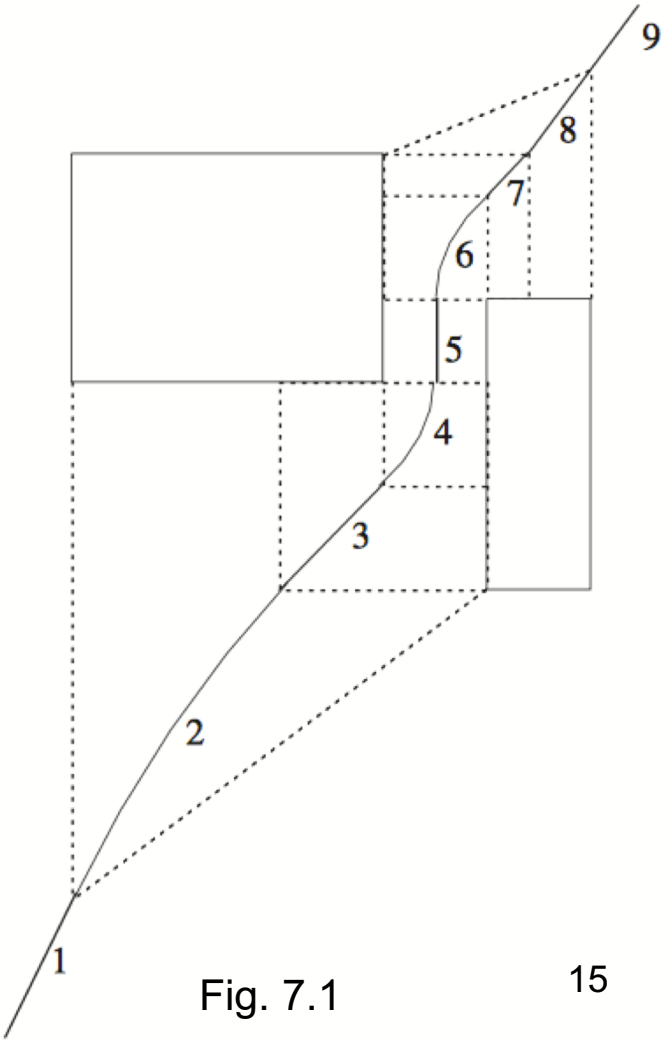


Fig. 7.1

Extending Linear Models

- LM weakness: only linear boundaries
- SVM use LM to implement non-linear boundaries (non-linear mapping of input):

$$x = w_1 a_1^3 + w_2 a_1^2 a_2 + w_3 a_1 a_2^2 + w_4 a_2^3$$

(2 attributes, 4 weights)

- But …Computational complexity: 10a w/5 factors=>2,000 coef.
- and … Overfitting

Extending Linear Models

- SVM address both weaknesses
- Max margin hyperplane = max separation
- Set of SVs define max margin hyperplane
- Finding SVs for training instances solved optimization problem
- Text describes how SVM solve both weaknesses

Extending Linear Models

Support Vector Regression: from classification to prediction

- Find function that minimizes prediction error (predicting training data)
- Tradeoff prediction error ε and tube's flatness

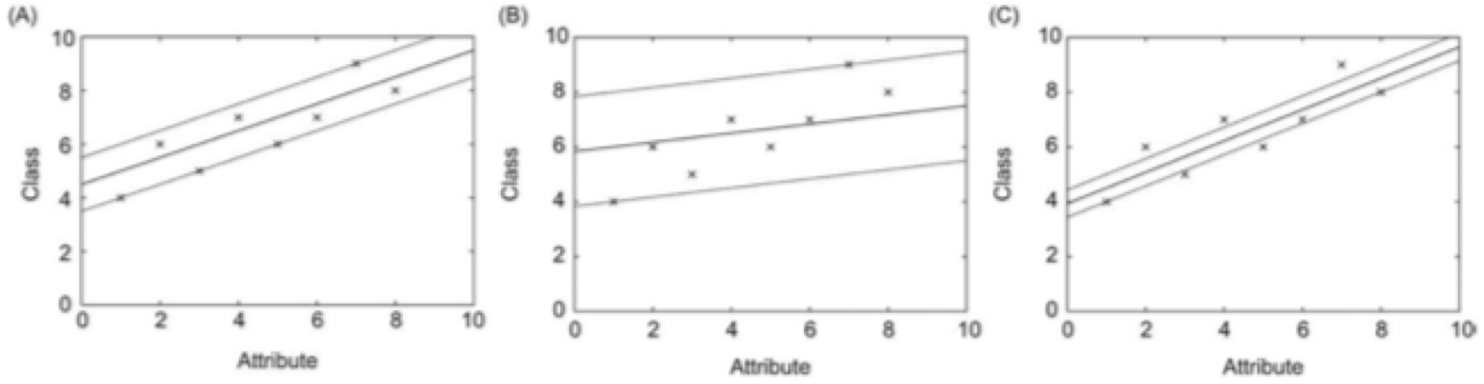
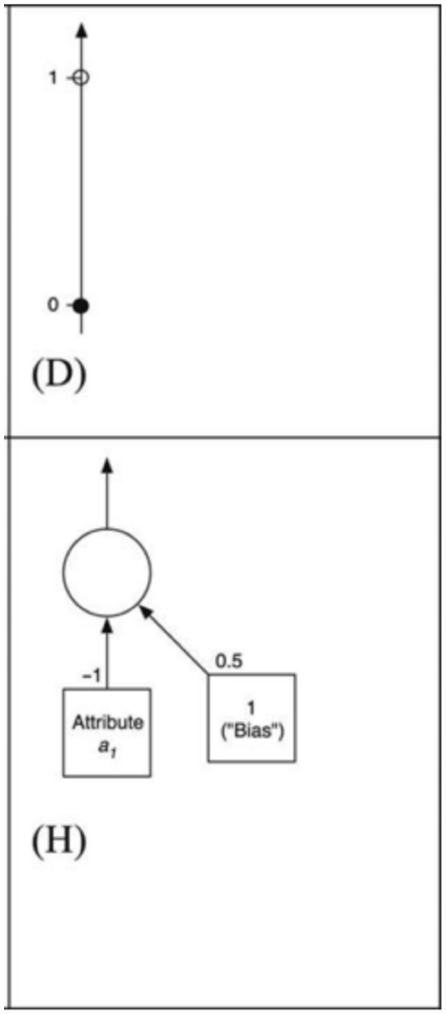


FIGURE 7.3 Support vector regression: (A) ε=1; (B) ε=2; (C) ε=0.5.

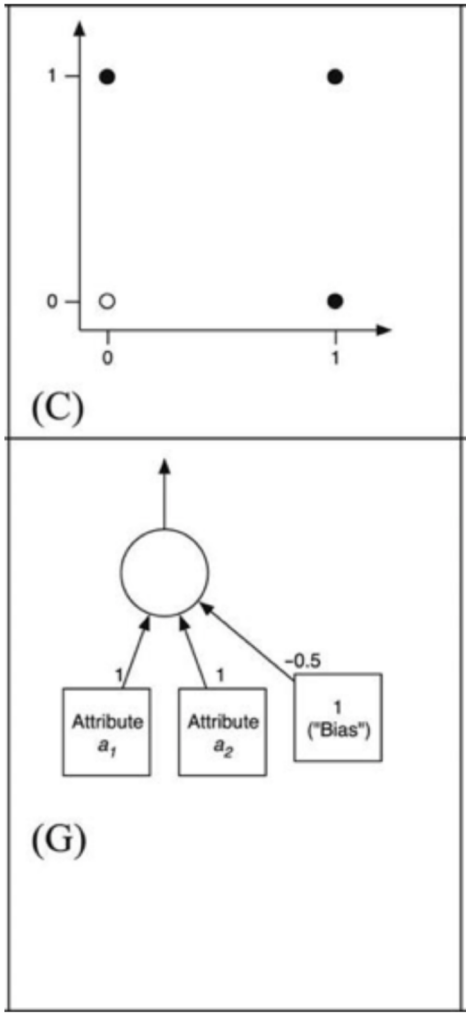
Extending Linear Models

Multi-layer Perceptrons



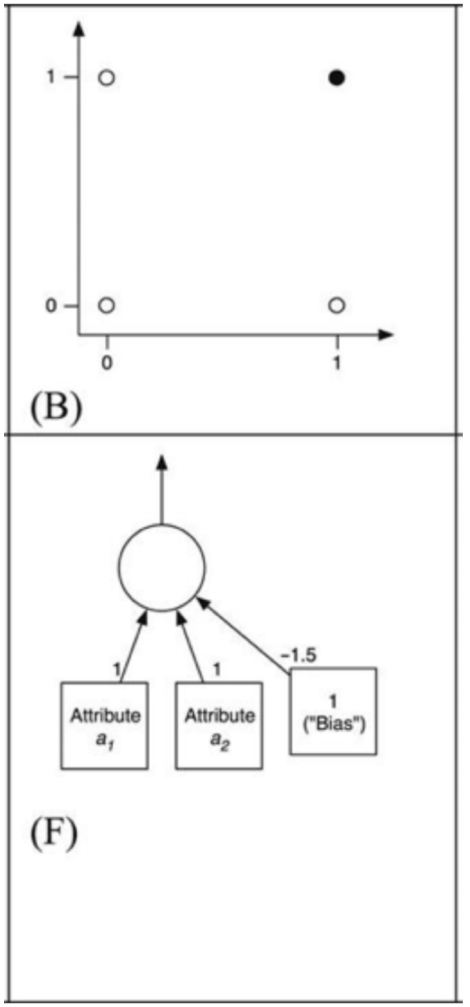
Extending Linear Models

Multi-layer Perceptrons



Extending Linear Models

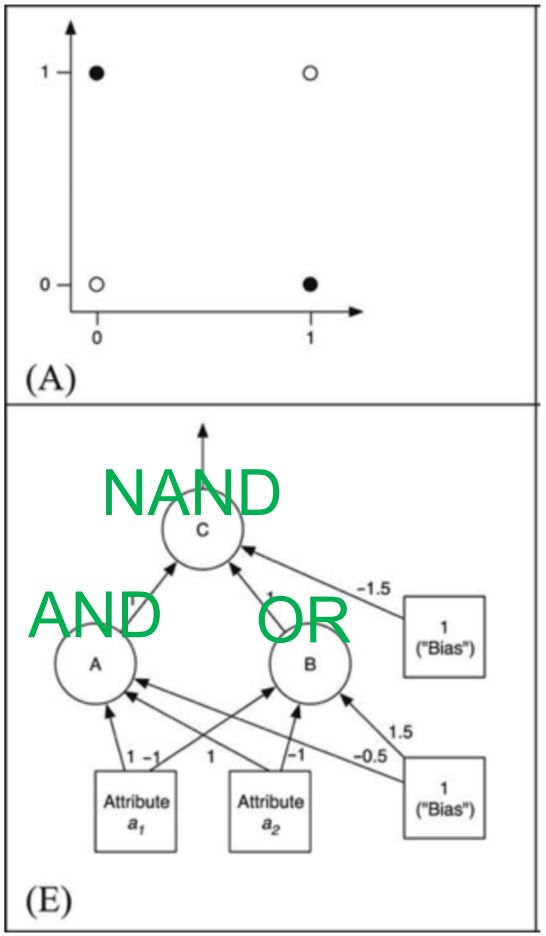
Multi-layer Perceptrons



Extending Linear Models

Multi-layer Perceptrons

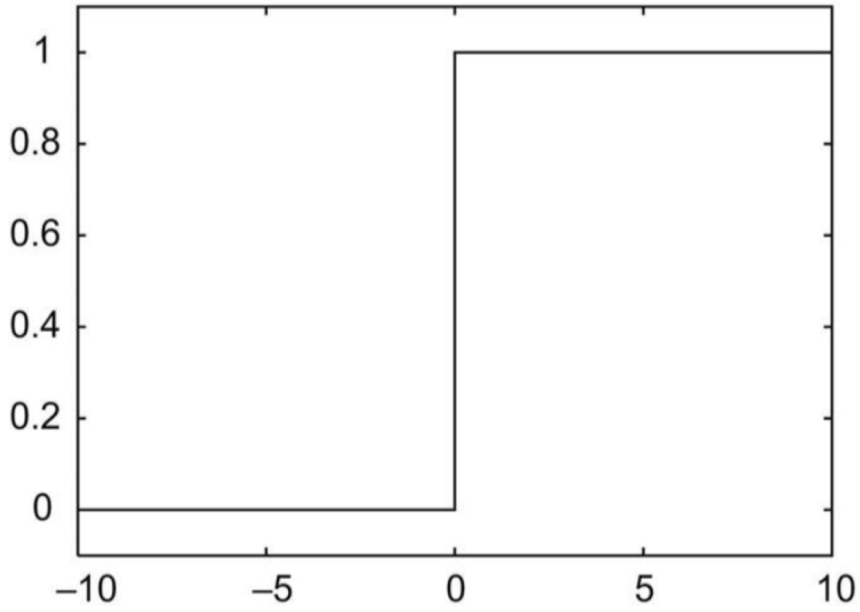
- Discussed XOR difficulty
- Multi-layer fixes XOR
- Train adjusting weights with “BackProp”



Extending Linear Models

Multi-layer Perceptrons

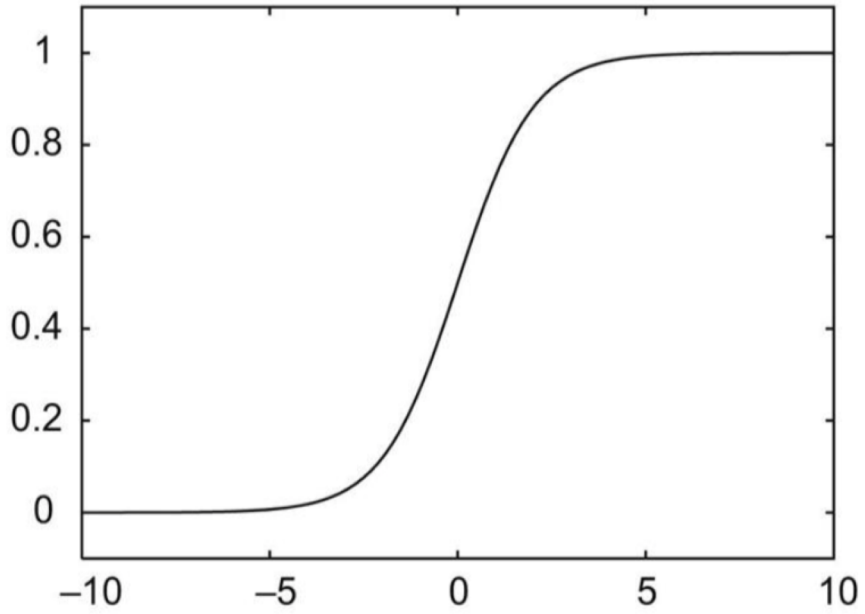
- Discussed XOR difficulty
- Multi-layer fixes XOR
- Train adjusting weights with “BackProp”



Extending Linear Models

Multi-layer Perceptrons

- Discussed XOR difficulty
- Multi-layer fixes XOR
- Train adjusting weights with “BackProp”



$$f(x) = \frac{1}{1 + e^{-x}}$$

Extending Linear Models

Stochastic Gradient Decent

- Gradient decent general purpose optimization
- Can learn linear models such as SVM & logistic regression
- Not at all what I expected…

Numeric Prediction with Local Linear Models

- Trees with numeric prediction: leaves either a class or ave. value of instances.
- “Regression” tree or “model” tree
- Decision tree down to class or linear model
- Pruning involved in tree

Numeric Prediction with Local Linear Models

Model Tree:

- Smooth over linear models to cover gaps
- Back up to the root through LMs at each node

$$p' = (np + kq) / (n + k)$$

p' : passed up
q : prediction at this node
p : prediction from below
n : number of instances covered below
k : smoothing factor

Numeric Prediction with Local Linear Models

Pruning a Model Tree:

- Estimated error rate at each node adjusted by number of instances and parameters in LM
- Simplify (reducing attributes) until est. error increases
- Prune nodes below if error of lower nodes exceeds current node

Numeric Prediction with Local Linear Models

Rules from Model Trees:

- Just like other trees: pick leaf, build rule, remove instances, repeat
- Rules have LM on RHS (then clause)

Numeric Prediction with Local Linear Models

Locally Weighted Linear Regression:

- For subdivided instance space, create LM for each
- Increase wt of instances close test instance
- Select smoothing factor, controls # instances to consider (like *k*NN)
- Locally weighted naïve Bayes algorithm > naïve Bayes and kNN

Advanced Approaches 2

1. Instance-based Learning
2. Extending Linear Models
3. Local Linear Models

The Project

The Project

Knowledge Mining project: 25%

The knowledge mining modeling project is intended to have students apply their knowledge of the subject by developing an analysis of a large dataset their choosing. Projects will be done individually or in teams of no more than 2 and presented to the class near the end of the classes. Students will propose a project (5 pts) and the instructor will provide feedback on scope and projected level of difficulty. Presented projects will be graded demonstrated knowledge mining operation (7 pts), usefulness of patterns reported (7 pts), and explanation of results, (6 pts).

The Project

Knowledge Mining project proposal: (5pts)

1. A title
2. Research question(s) (ends with “?”)
3. Identification of the source dataset (source, size, complexity)
4. Identification of methods & tools to be applied
5. Expected results

The Project

Presented projects will be graded on:

- demonstrated knowledge mining operation (7 pts),
- usefulness of patterns reported (7 pts), and
- explanation of results, (6 pts).

New Material

Data Transformations (Chap 8)

- Attribute Selection
- Discretizing Numeric Attributes
- Projections (Transformations)
- Sampling
- Cleaning (”Cleansing”)
- Transforming Multiple Classes into Binary
- Calibrating Class Probabilities

Data Transformations & Project Management

- *The best performance on the training data – not necessarily the best for the full dataset*
- Chapter's processes can "materially" improve successful application of machine learning
- "Data engineering"
- Chapter: "Bag of tricks"

Attribute Selection

- Theory & practice…

  “There is no difference between theory and practice, in theory, but in practice there is.”

- Irrelevant or distracting attributes confuse ML

Attribute Selection

- Decision-tree learners sensitive to noise (errors, randomness) at some depth (when noise > support) (“Fragmentation problem”)
- Instance-based learners worse with noise
- Naïve Bayes learners ignores irreverent attributes, but bad with correlated attributes
- So, pre-select attributes… best: manually, based on understanding problem and attributes

Attribute Selection

Scheme-Independent Selection

- Two approaches/methods
- ***Filter:*** Select based on attribute characteristics

  Such as: min set that distinguishes training instances

- ***Wrapper:***

  Try ML on subset of attributes

  Tree learning (e.g., 1R) use some (nec'y?) attributes only

  Instance-based near-hits/near-misses

  Use 1 ML technique to identify attributes, another to exploit them

  … Try ML, explore … guided manually

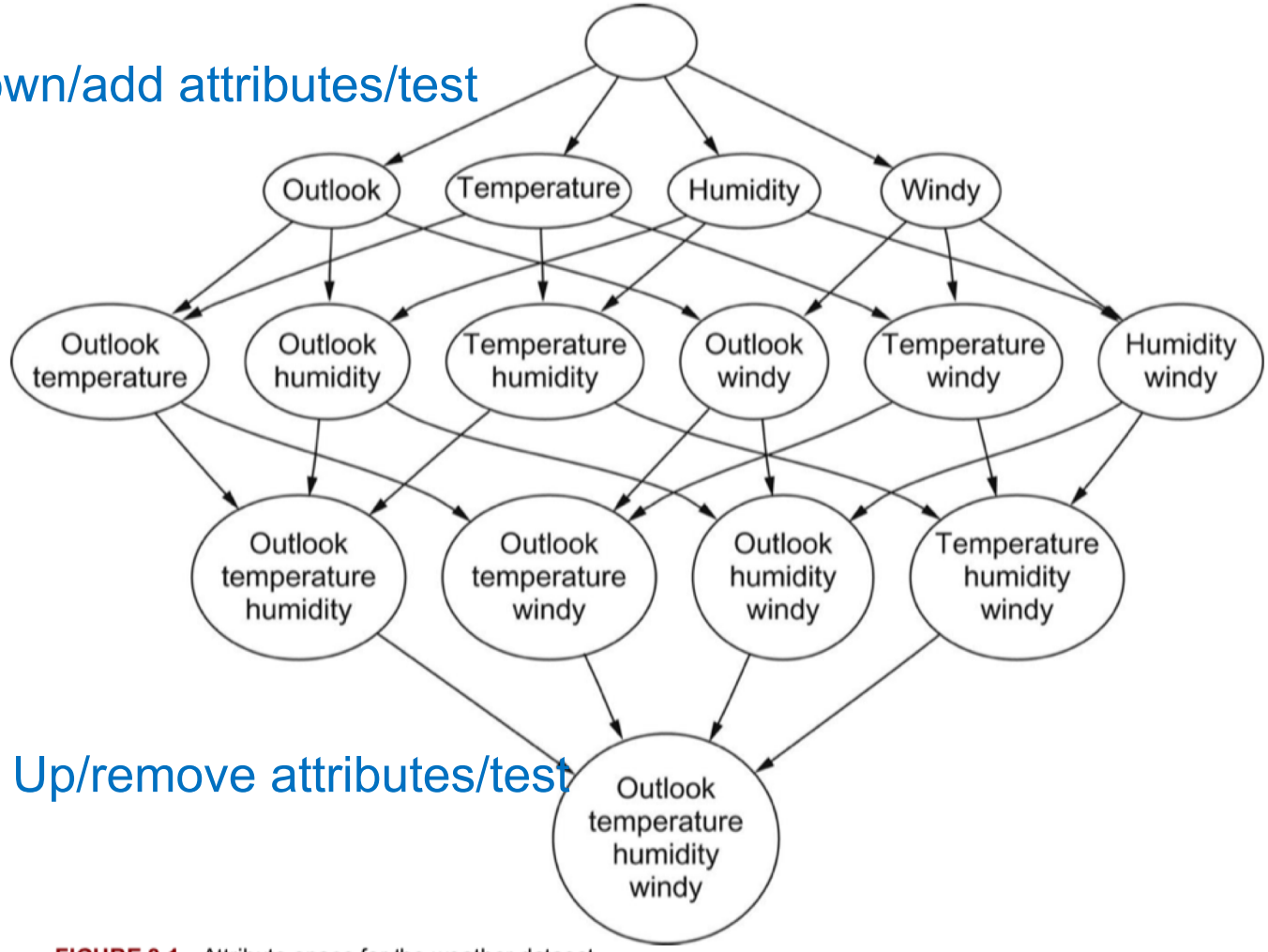
Attribute Selection

Searching attribute space for effective subset

- Space exponentially large in # attributes
- Downwards/forward selection adding attributes
- Upwards/backwards elimination
- Evaluate performance with each change

Attribute Selection

Top down/add attributes/test



Bottom Up/remove attributes/test

FIGURE 8.1 Attribute space for the weather dataset.

Attribute Selection

Scheme-specific selection

- Use ML technique's classification performance
- |backward attribute sets| > |forward|, but is more accurate
- Sophisticated searches generally not justified
- Simple table of instances tested cheaply
- Good results using forward selection to detect redundant attributes with target method is naïve Bayes:
  - Works well with random attribute selection
  - Can be misled by redundancy in attributes
  - Good results with forward selection approach
  - Less likely to overfit
  - ("Selective Naïve Bayes" handles tree- or rule-based classifiers)

Discretizing Numeric Attributes

Supervised discretization

- Discretize before learning
- Global: sort attribute's values & divide ranges
- Local: decision-trees split at each node
- Global before learning:
  - Treat new attributes as others
  - Treat new attributes differently: ordered
  - (If can't handle "ordered", make set of attributes with binary (bar LEDs): FTTTT,FFTTT,FFFTT,…)

Discretizing Numeric Attributes

Unsupervised discretization

- Without info on classes
- Discretize during collection
- Consider *equal-frequency binning*
  - "histogram equalization", goal: flat (equal freq.)
  - Works well with naïve Bayes learning when # bins choice is data-dependent (sqrt of # instances) called "proportional k-interval discretization"

Discretizing Numeric Attributes

Entropy-based discretization: divide at lowest

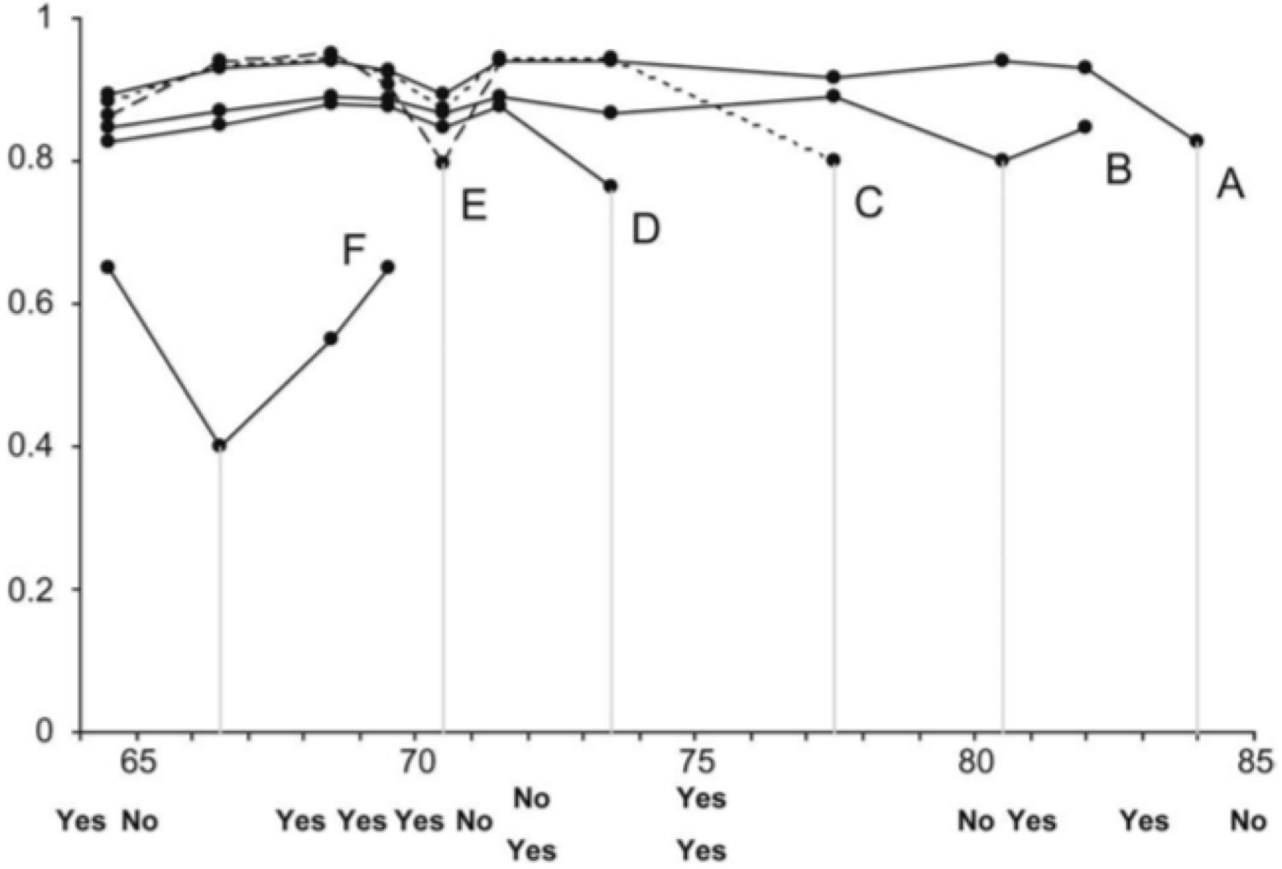


FIGURE 8.2 Discretizing the *temperature* attribute using the entropy method.

Discretizing Numeric Attributes

Entropy-based discretization: divide at lowest

Results:

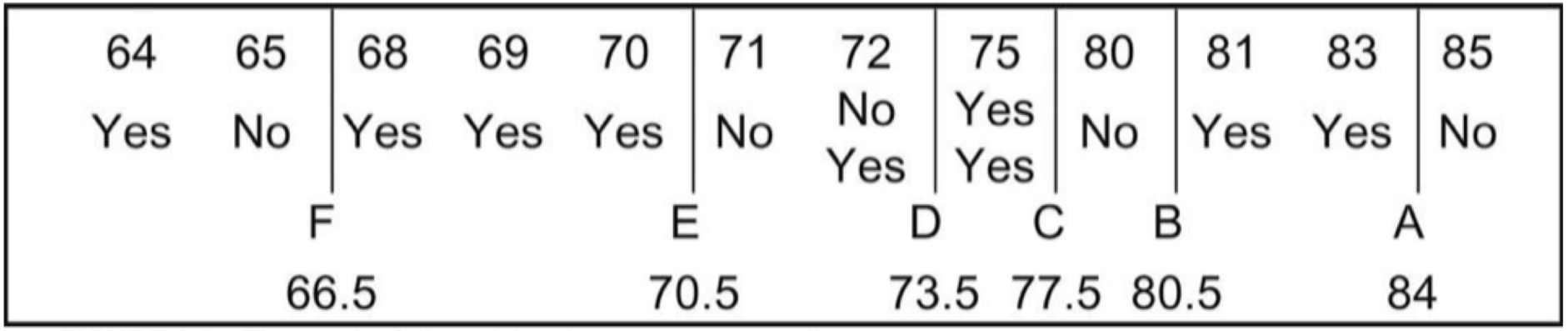


FIGURE 8.3 The result of discretizing the *temperature* attribute.

Discretizing Numeric Attributes

Entropy-based discretization

- When to stop? MDL-based
- Minimize theory & specification of data
- (in this example, wouldn’t split and temperature wouldn’t be selected as an attribute…)

Discretizing Numeric Attributes

Other methods of discretization

- MDL best for supervised
- Top down, bottom up, statistical test,…
- Error-based (1R, k-intervals) vs. entropy-based
- Dynamic programming partition N instances into k intervals (in time ~ kN^2)
- Error-based minimize error count (linear in N)

Discretizing Numeric Attributes

Entropy-based vs. Error Based

- Why not just use efficient error based?

Discretizing Numeric Attributes

Entropy-based vs. Error Based

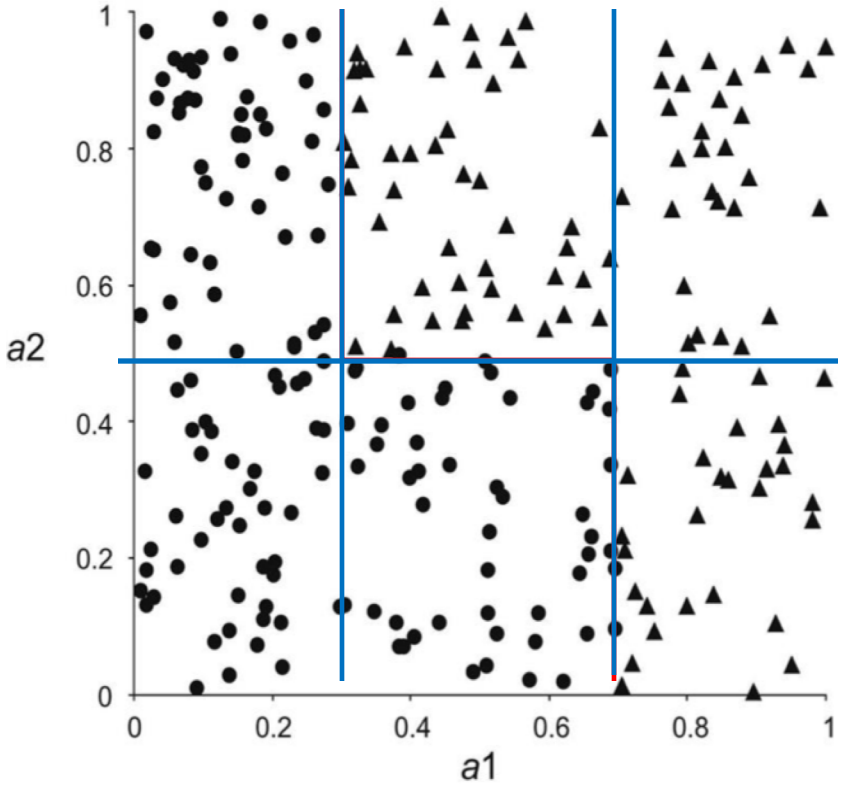


FIGURE 8.4 Class distribution for a two-class, two-attribute problem.

a2 ok
a1 …

Error methods not sensitive to distribution.

Discretizing Numeric Attributes

Converting discrete attributes to numeric

- Attribute nominal values, same=0, diff=1
- Attribute transformation:
  - Replace k-valued numeric with k synthetic binary attribute
- More subtle diff for attributes with values
- Orderable? More options

Projections (Transforms)

- Semantic transformation, e.g., birth date to age
- Linear relationship between attributes: ratio
- Days of the week, holidays
- Concatenate two nominal attributes to one attribute with n1 X n2 values
- Height & weight to BMI
- Clusters to p(membership in cluster)
- Sparse to dense

Projections (Transforms)

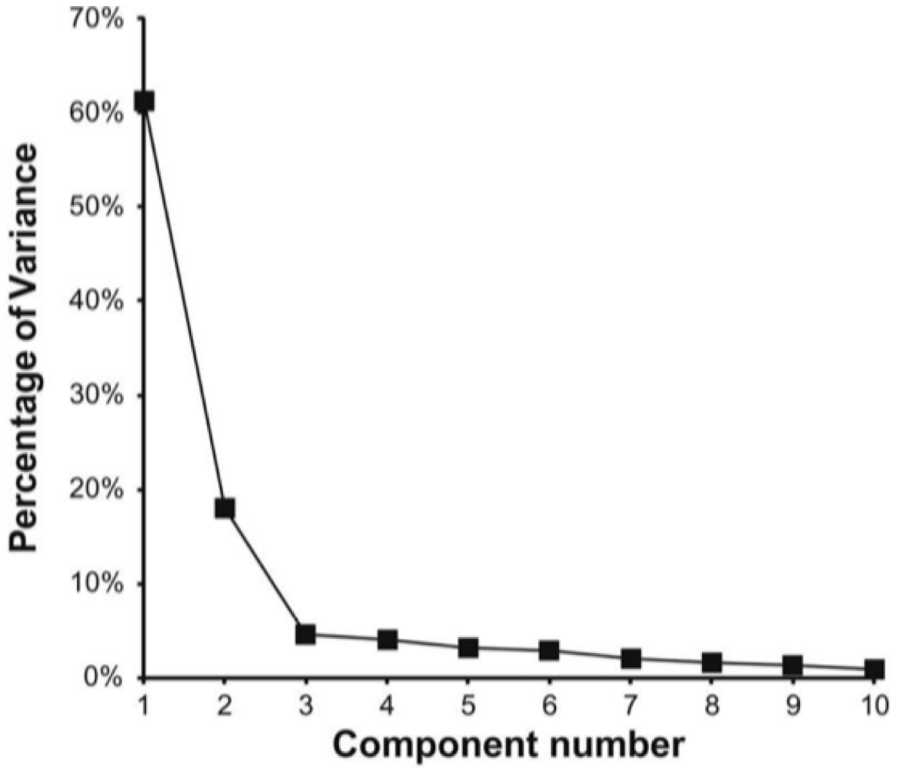
Principal Component Analysis

- Up/down vs. North, South, East, West
- Any perpendicular references
  - First along major variance, second perpendicular
- Polar coordinates?
- PCA: normal to standardize to zero mean, unit variance (SD=var=1)
- (if done at each node in tree, “multivariate decision tree”)

Projections (Transforms)

Axis	Variance	Cumulative
1	61.2%	61.2%
2	18.0%	79.2%
3	4.7%	83.9%
4	4.0%	87.9%
5	3.2%	91.1%
6	2.9%	94.0%
7	2.0%	96.0%
8	1.7%	97.7%
9	1.4%	99.1%
10	0.9%	100.0%

(A)



(B)

FIGURE 8.5 Principal component transform of a dataset: (A) variance of each component; (B) variance plot.

Projections (Transforms)

Random Projections

- Random projection matrix into specified dim:
  - First, transform to reduce dimensions
  - Then, build a tree for transformed space
- Preserves distance relationships well (on average)
- Perform worse than careful PCA, but not by much, AND cheaper

Projections (Transforms)

Partial least-squares regression

- PCA before linear regression, result: “principle components regression”
- If used all components, effectively applying least-squares regression on original data
- Partial least-squares takes class attributes into account as well as predictor attributes
- Simple iterative method involves dot product

Projections (Transforms)

Partial least-squares regression

- Start with standardized input attributes
- The attribute coefficients for **first** partial leas-squares direction found by dot product of each attribute vector and class vector
- **Second** original attributes replaced by diff between attribute's value and prediction from simple univariate regression that uses previous predictor

Projections (Transforms)

Text to attribute vectors

- Attribute values can be full source text
- Models check for equality, not internal structure
- Word most useful unit (? with # occurrences/freq.)
- “tokenization” conversion of text to words
- Term frequency, inverse document frequency

Projections (Transforms)

Time Series

- Attribute values at each time step (think weather)
- Sometimes series of differences (“deltas”)
- If not regular timed samples, data with timestamps

Sampling Data

- Random sample of large volume of data
- (Each instance has equal chance of use)
- From N instances, create any number of instances with rnd sample w/ replacement
- Reservoir Sampling:

Sampling Data

- **Reservoir Sampling**
  - Instances arrive sequentially, but # unknown
  - Sample input stream without repeating sampling
  - Need reservoir size r:

```
1.keep all until have r, then
2.with p(keeping next)=r/(r+1), replace a
  random instance with new instance
```

Cleaning Data

Improving decision trees

- Can improve, often w/o loss of accurately, by eliminating misclassified training instances
- Difference rarely significant
- Due to nature of pruning: unjustified gone
- Local effect
- Presumes no systematic errors (intentional or not)
- If learns from noisy attributes, keep noise

Cleaning Data

Robust Regression

- "Show me the data!"
- Outliers obvious, surprising
- Statistical methods addressing outliers: "robust"
- Auto handling:
  - Use absolute value v. square – outliers weaker
  - remove 10% farthest from the LM line
  - Minimize median v. mean from squares of divergences from line (deals with x&y outliers) ($$)

Cleaning Data

Detecting Anomalies

- Try different methods, if all say anomaly, must be…
- Effectively letting methods vote
- Then use resulting filtered data for training
- Could be the class that the methods “don’t like”
- However, automatic filtering poor substitute for getting good data in the first place!

Cleaning Data

One-Class Learning (target class or unknown)

- **Generating artificial data**:
  - Generate non-target data
  - Too much data can overwhelm
  - Goal to learn accurate class probability est., not minimize classification error
  - Diff. thresholds can serve as decision boundary
  - As #attributes goes up, more difficult to generate data: stay close to target – “reference” distribution

For Next Week

- Project Proposal

Your Questions?