Data Mining for Scientific & Engineering Applications

Robert Grossman, Laboratory for Advanced Computing, University of Illinois & Magnify

Chandrika Kamath, Lawrence Livermore National Laboratory

Vipin Kumar, Army High Performance Research Center, University of Minnesota

Chapter 2 – Classification & Regression

Robert Grossman, Laboratory for Advanced Computing, University of Illinois & Magnify

Goals of Chapter 2

- What are the three basic algorithms in data mining?
- Importance of cleaning data
- Importance of derived attributes
- What is a consistent classifier?

Chapter 2. Basic Ideas

2.1 Nearest Neighbor Learning

Classification

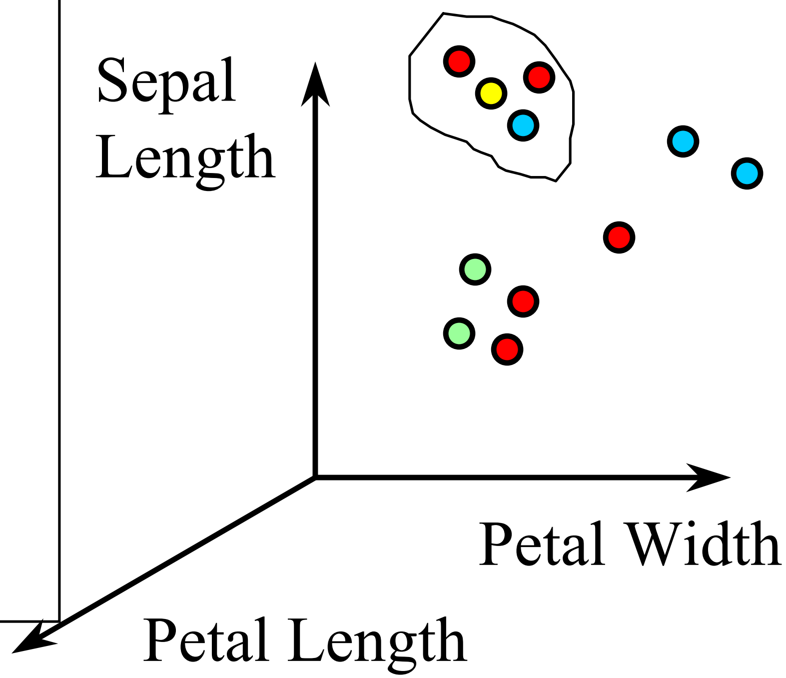
Petal Len.	Petal Width	Sepal Len.	Sepal Width	Species
02	14	33	50	A
24	56	31	67	C
23	51	31	69	C
13	45	28	57	B

- **Assume data is arranged into rows (records) and columns (attributes or features)**
- **Assume each row is classified A, B or C**
- **Goal: given unclassified record, to classify it.**

k-Nearest Neighbor Learning

1. Assume records have features.

2. Assume records are either or



To classify

1. find nearest three records

2. classify via majority vote

- View records as points in feature space
- Find k-nearest neighbors and take majority vote.
- Example of supervised learning.

(j, k) Nearest Neighbor Learning

- Choose j points from the test set to produce a model M[1]. Choose another j points to produce a Model[2], etc.
  - This gives an ensemble of models:

    {M[1], …, M[p]}
  - Selecting the j points can be done in many different ways.
- To classify a point,
  1. evaluate each of the k-nearest neighbor models in the ensemble
  2. use a majority vote to get an overall class

Learning -
Map from Data to Models

Petal Len.	Petal Width	Sepal Len.	Sepal Width	Species
02	14	33	50	A
24	56	31	67	C
23	51	31	69	C
13	45	28	57	B

Learning Sets (n data points)

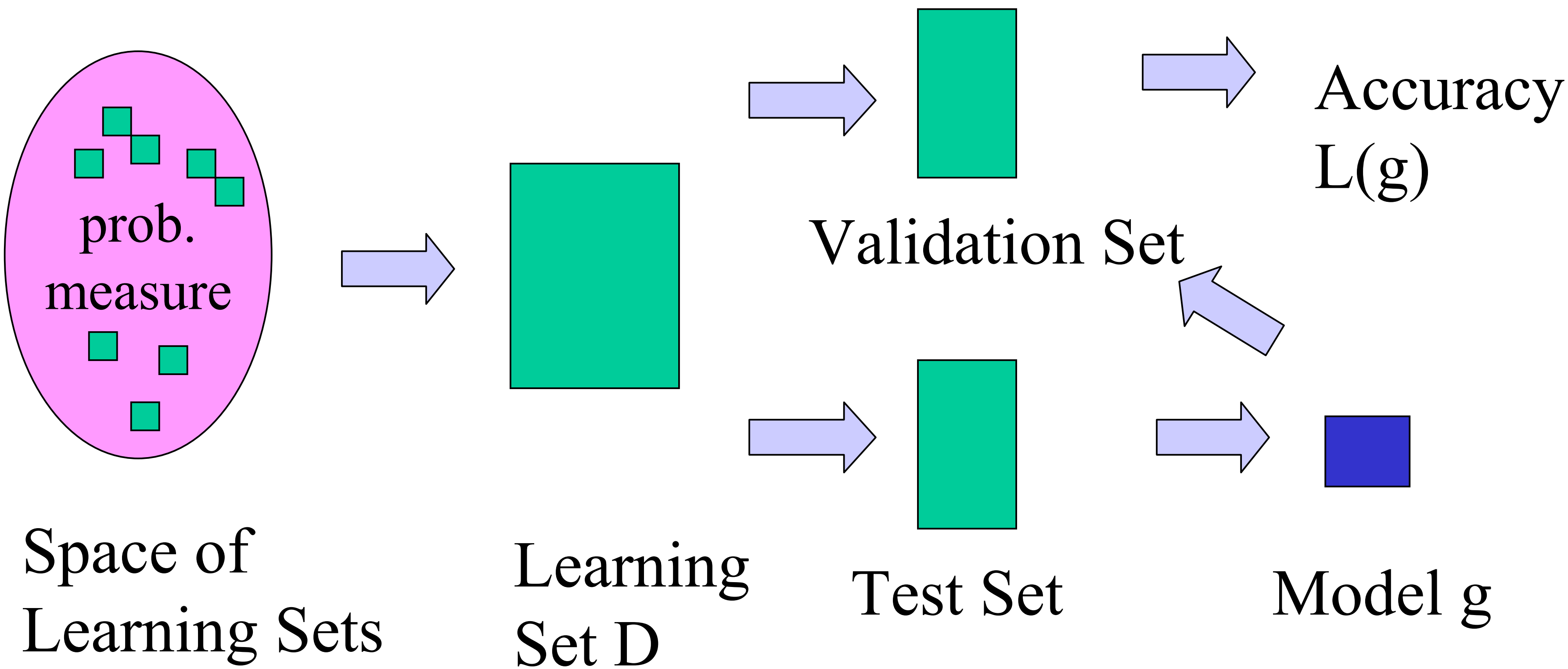
<pmml><nearest-neighbor>…

02	14	33	50	A
13	45	28	57	B

</nearest-neighbor></pmml>

Models or Rules (j points)

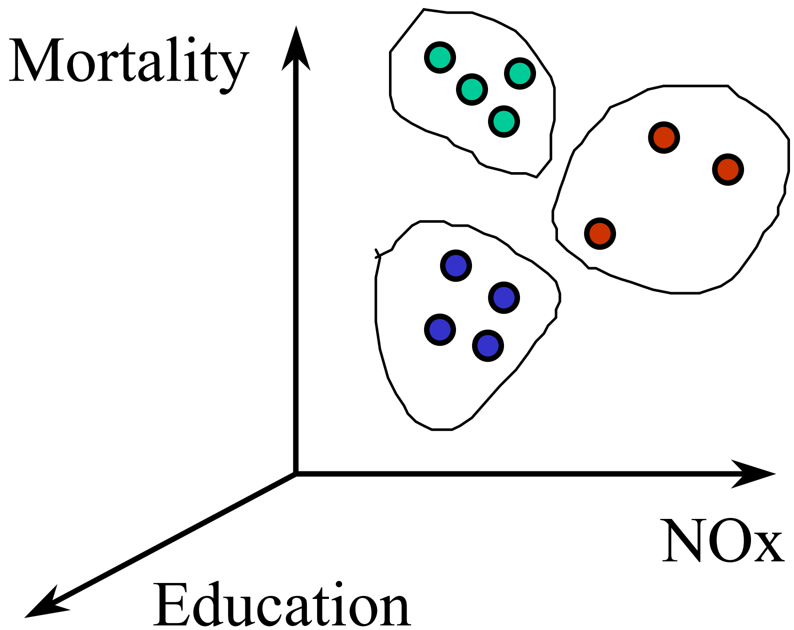
Does the Model Generalize?



- R^d x {0,1}-valued random pair (X,Y)
- L(g) = P (g(X) = Y), exp. accuracy E(L(g))

2.2 Cluster-based Learning

Learning via Clustering

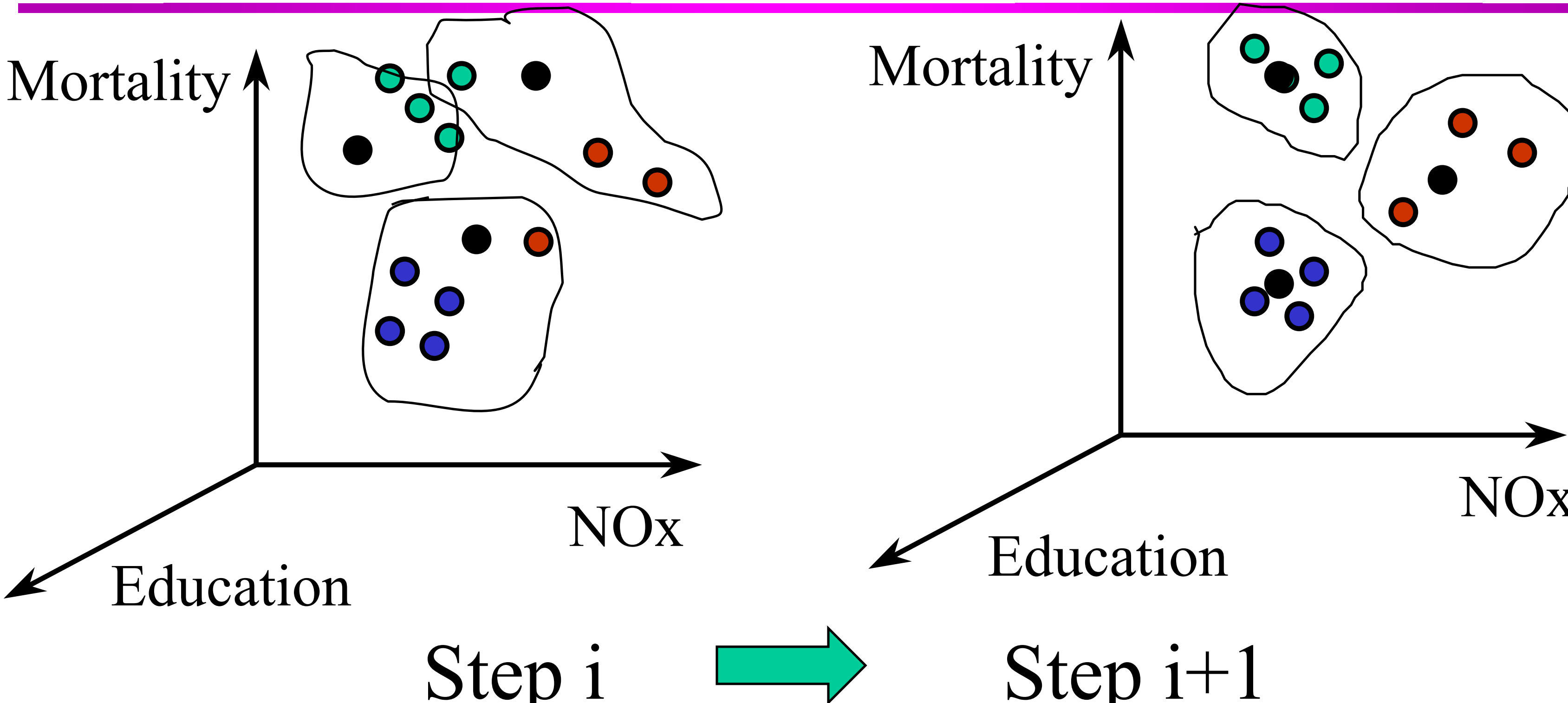


- Form the k=3 “best” clusters in feature space.
- Example of unsupervised learning
  – no prior knowledge needed about classification.

K-Means Clustering

1. Set i = 0. Choose k centroids a[i, 1], …, a[i, k] in feature space.
2. Assign each point in the test set to the nearest centroid (break ties using the lowest index) to form clusters C[1], …, C[k].
3. Compute the new centroid a[i+1, j] for each cluster C[j], j=1, …, k.
4. Repeat until the centroids converge.

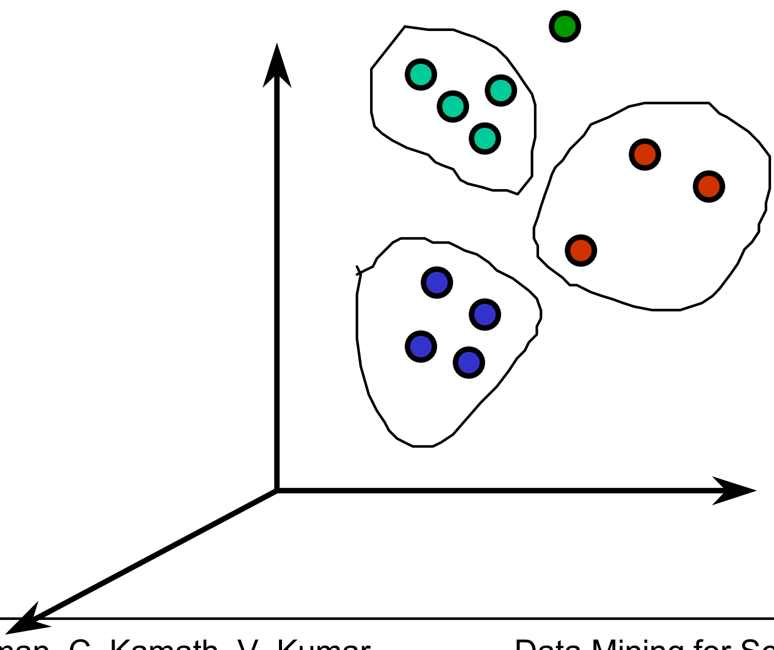
K-Means Clustering



- Centroids ● converge to the centroids of the final clusters

Learning via Clustering

- Form the three "best" clusters.
- Example of unsupervised learning
  - no prior knowledge is needed about the classification.
- Use as a basis for subsequent supervised learning.



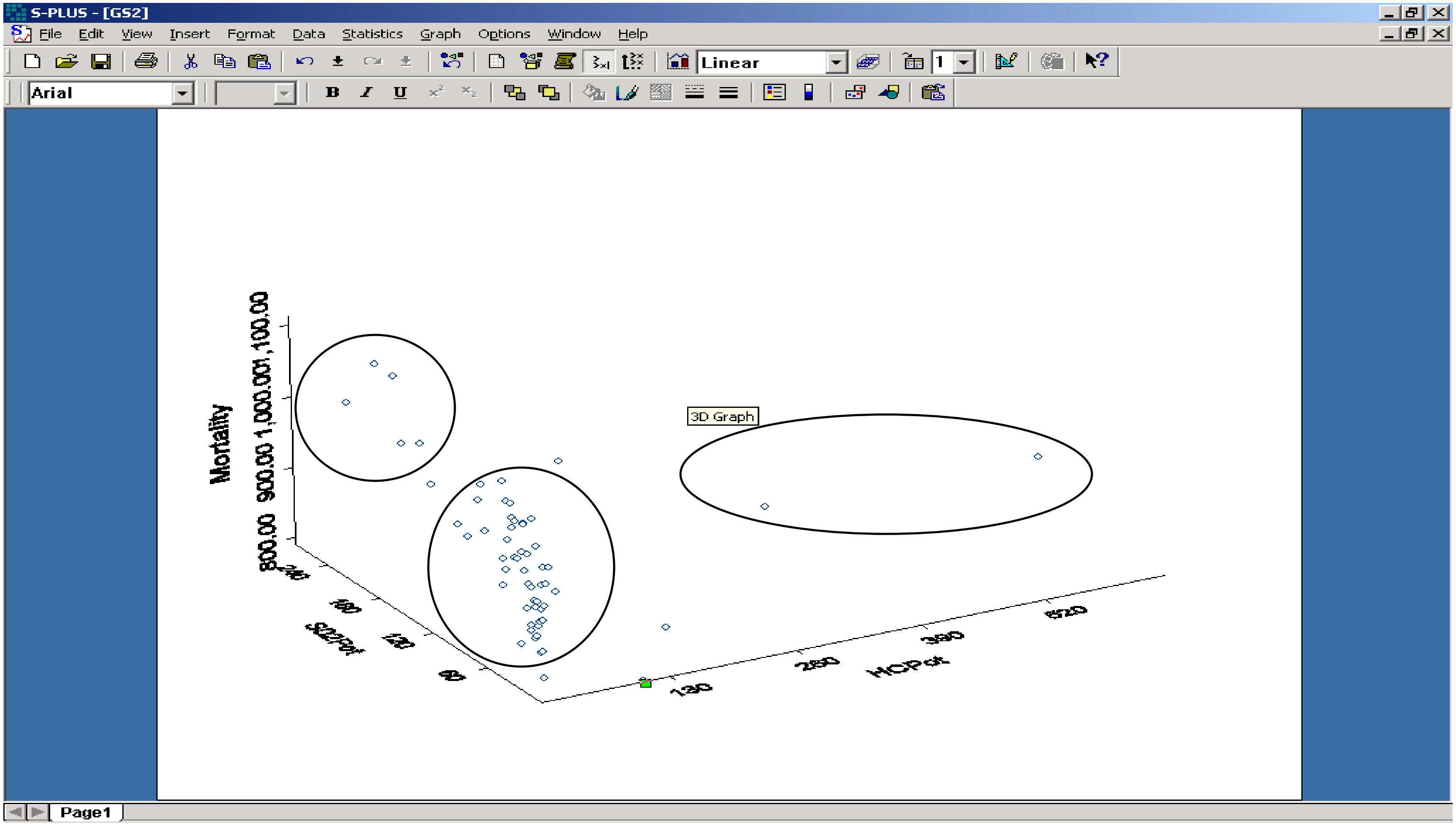
To classify ●

1. find nearest cluster

2. classify ●

using nearest cluster

Example: Polution vs. Mortality



2.3 Trees

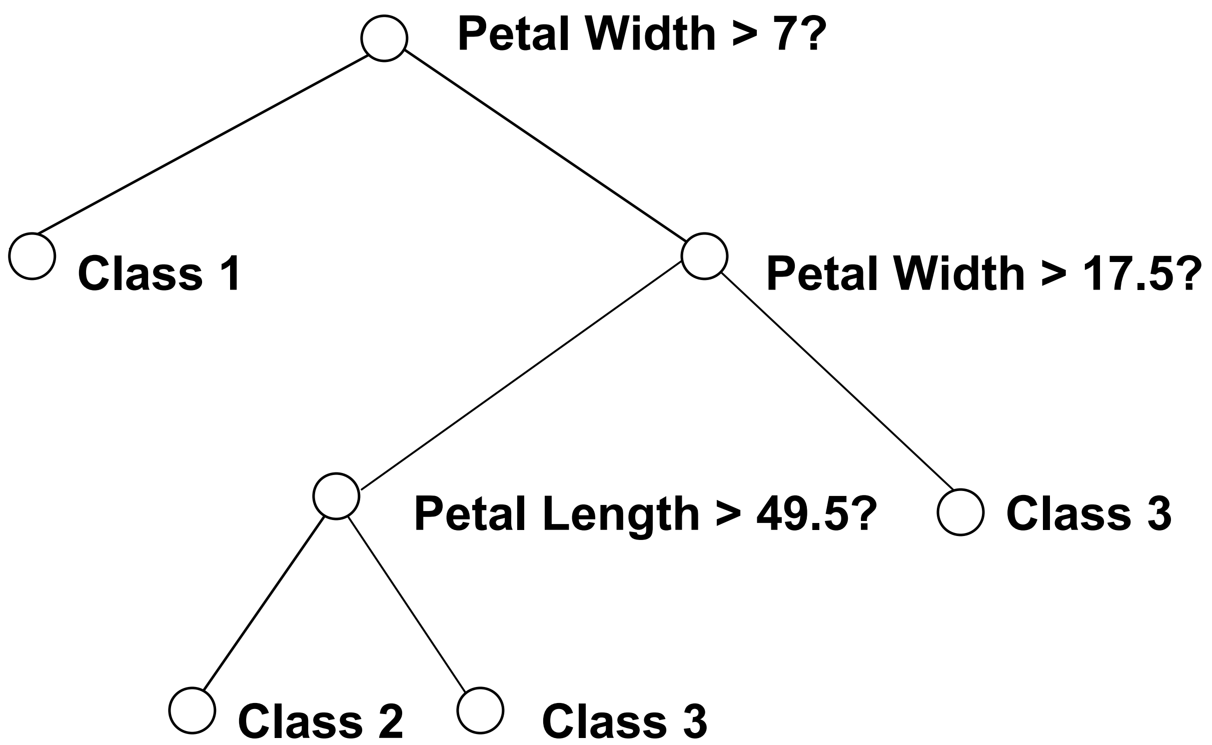
Following L. Breiman, J. Friedman, R. A. Olshen, C. J. Stone, Classification and Regression Trees, 1984, Chapman & Hall.

Classification Trees

Petal Len.	Petal Width	Sepal Len.	Sepal Width	Species
02	14	33	50	A
24	56	31	67	C
23	51	31	69	C
13	45	28	57	B

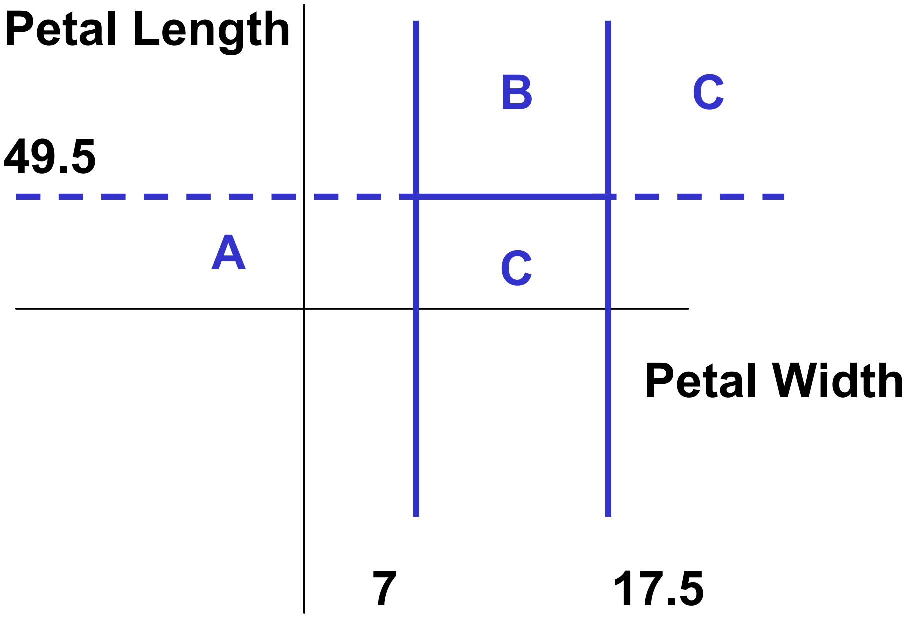
- **Want a function Y = g(X), which predicts the red variable Y using one or more of the blue variables X[1], …, X[4]**
- **Assume each row is classified A, B, or C**

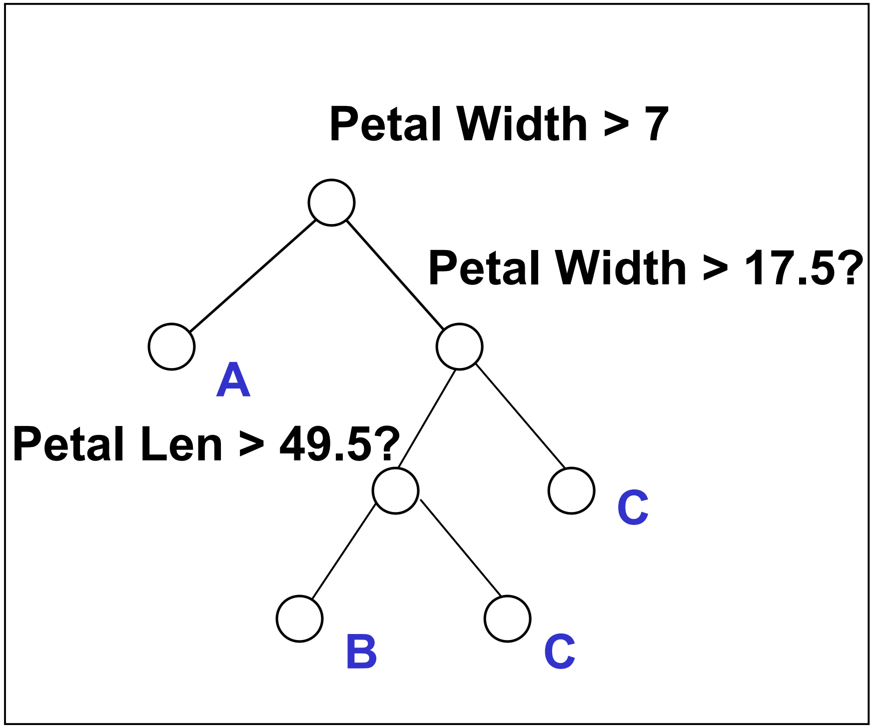
Simple Classification Tree



- Divide feature space into regions
- Use a majority vote to get class A, B, C, etc.

Trees Partition Feature Space





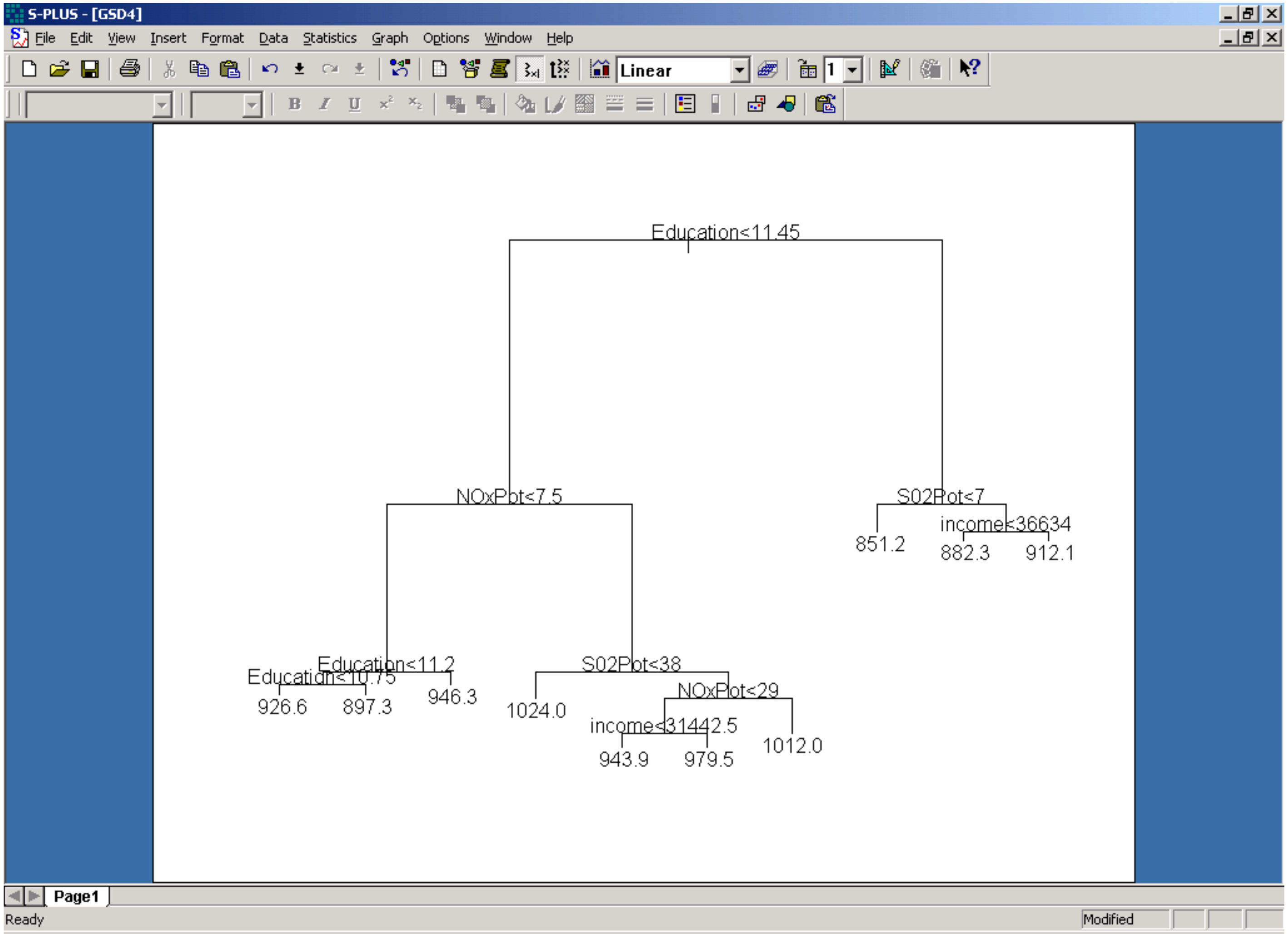
- Trees partition the feature space into regions by asking whether an attribute is less than a threshold.

Regression Trees

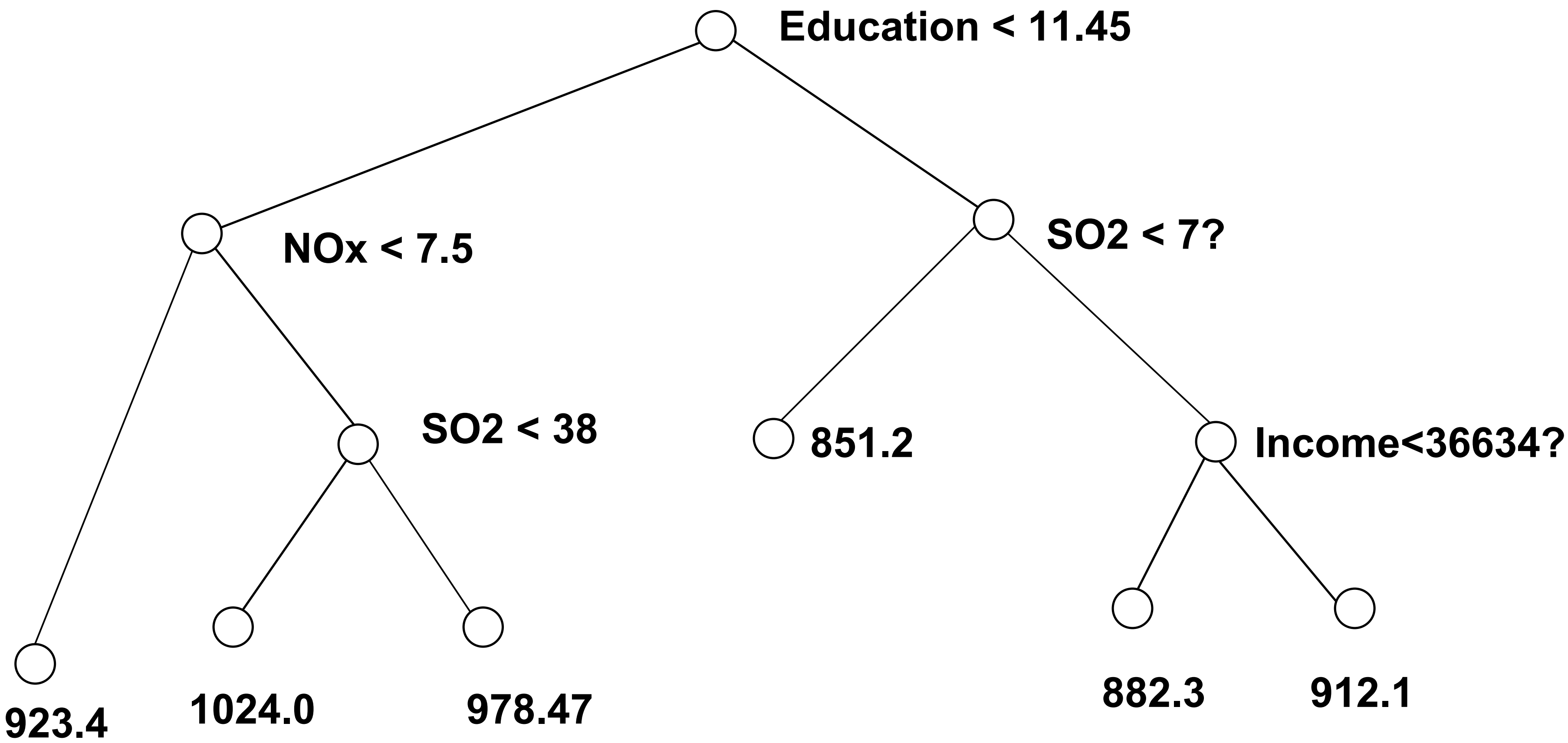
City	Education	NOx	SO2	Mortality
Akron	11.4	15	59	921.87
Boston	12.1	32	62	934.70
Chicago	10.9	63	278	1024.89
Dallas	11.8	1	1	860.10

- Want a function Y = g(X), which predicts the red variable Y using one or more of the blue variables X[1], …, X[14]

Regression Trees



Regression Trees



- Divide training sets into buckets.
- Average the dependent variable in each bucket.

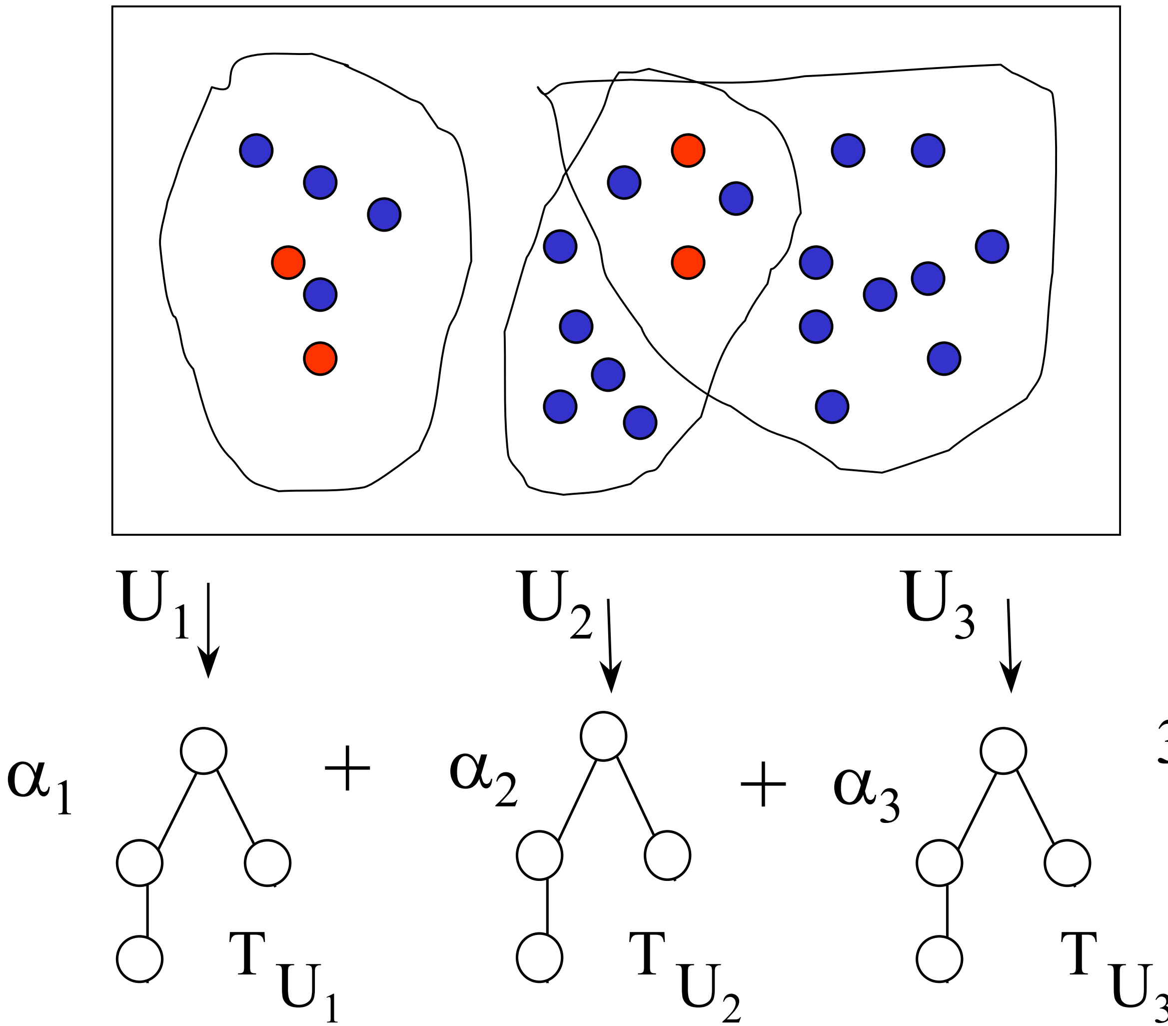
ART and ACT (Averaged Reg. & Class. Trees)

- **Define a Cover of the Data.** A cover *U* of the data x consists of a collection of sets U such that each record is in at least one U.
- **Build Trees.** Build a tree T_U as usual for the data assigned to each set U in *U*.
- **Average Trees.** Fix a finite probability measure α_U on *U*. Given an object x, ART uses the score:

$$\Sigma \alpha_U \, T_U(x),$$

- This defines an ensemble of trees.

Basic Idea: ART



1. Define a cover $\boldsymbol{U} = \{U_1, U_2, U_3\}$ of the data x.

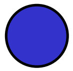
2. Construct a tree T_U on each set U of the cover.

3. Average the trees:

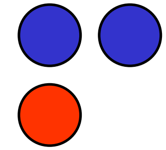
$$\Sigma\, \alpha_j = 1$$

$$\alpha_j > 0$$

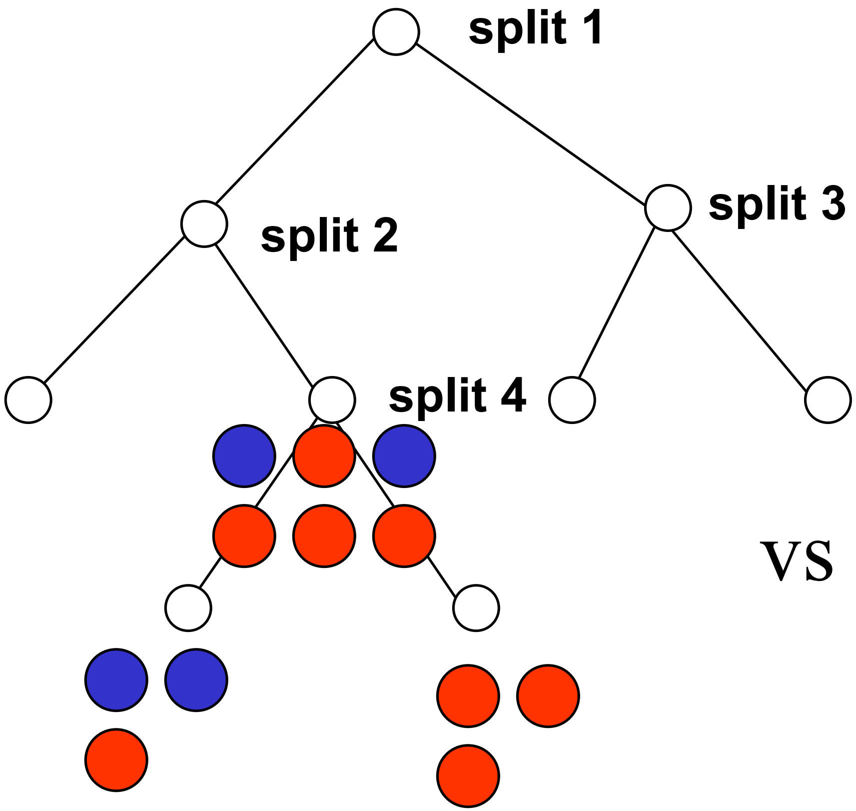
Split using Entropy



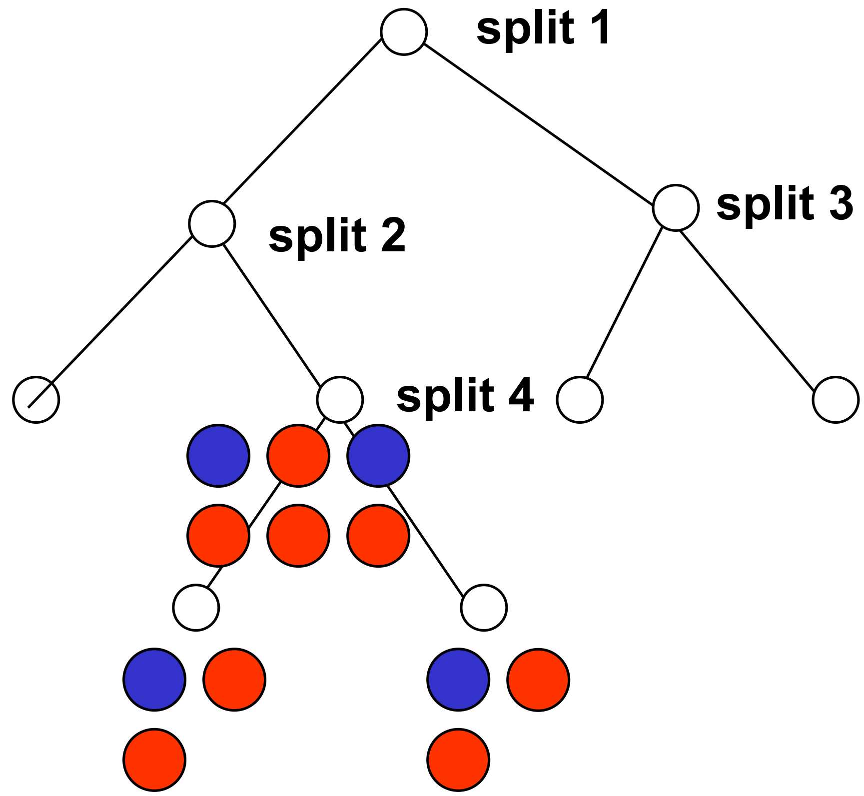
objects have attributes



information = 0.64



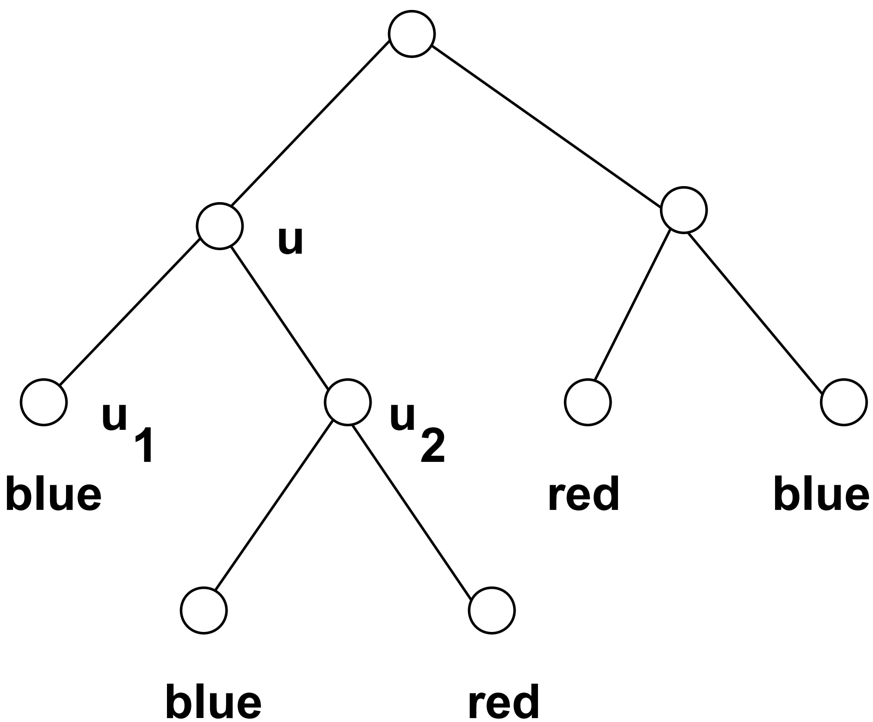
vs



increase in information = 0.32

increase in information = 0

Growing the Tree



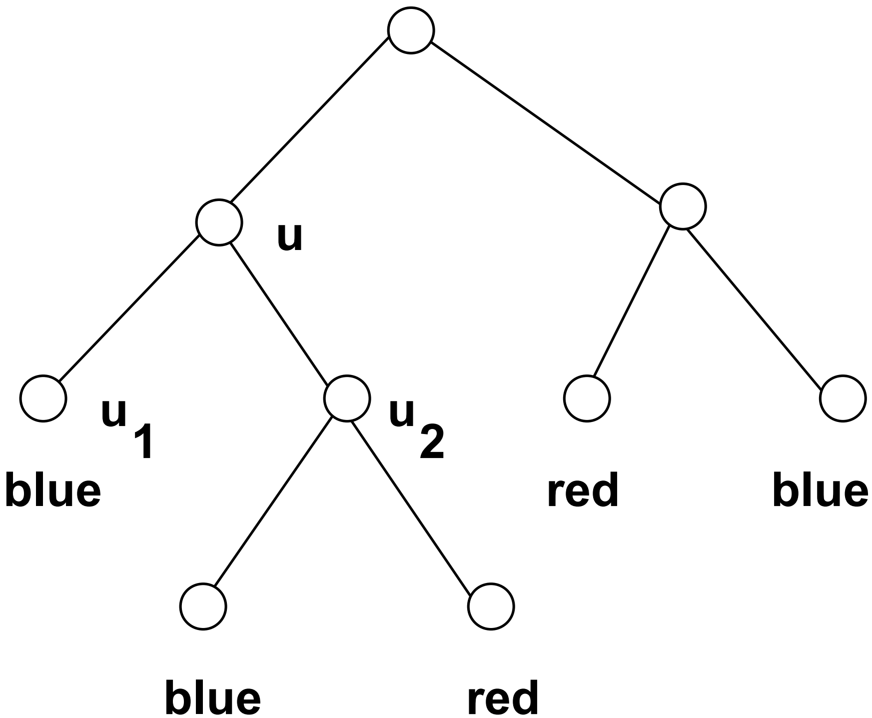
Step 1. Class proportions.
Node u with n objects
n_1 of class A (red)
n_2 of class B (blue), etc.

Step 2. Entropy
$I(u) = - \Sigma\, n_j/n \log n_j/n$

Step 3. Split proportions.
m_1 sent to child 1– node u_1
m_2 sent to child 2– node u_2

Step 4. Choose attribute to maximize
$\Delta = I(u) - \Sigma\ m_j/n\ I(u_j)$

Split Using GINI Impurity



Step 1. Class proportions.
Node u with n objects
n_1 of class 1 (red)
n_2 of class 2 (blue), etc.

Step 2. Compute Gini Index
$\text{Gini}(u) = 1 - \Sigma (n_j/n)^2$

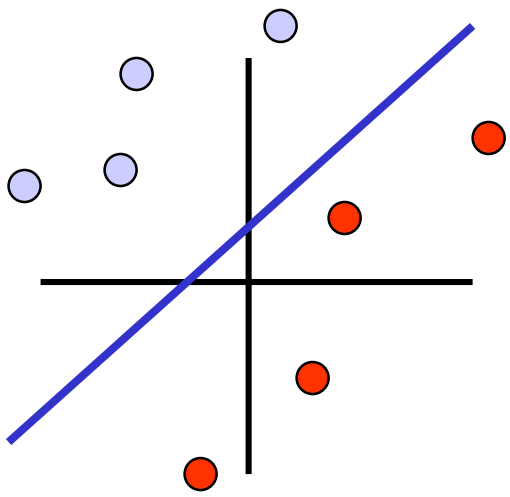
Step 3. Split proportions.
m_1 sent to child 1– node u_1
m_2 sent to child 2– node u_2

Step 4. Choose split to min
$\text{Gini of Split} = \Sigma\ m_j/n\ \text{Gini}(u_j)$

2.4 Neural Networks

Perceptron

- Inputs $x_1, x_2, \ldots, x_n,$
- Ouput +1 or -1
- Perceptron is determined by weights
  - $w_0, w_1, \ldots, w_n$ (define $x_0 = 1$)
- Output $\hat{y} = \mathrm{sgn}(w \bullet x)$
- Given a learning set $L = \{ (x,y) \}$
- Perceptron Training Rule
  - $w_i \leftarrow w_i + \Delta w_{i,}$
  - $\Delta w_i = \eta \, (y - \hat{y}) \, x_i$
- Weights are unchanged when $y = \hat{y}$, increased otherwise by factor $\eta \, x_i$

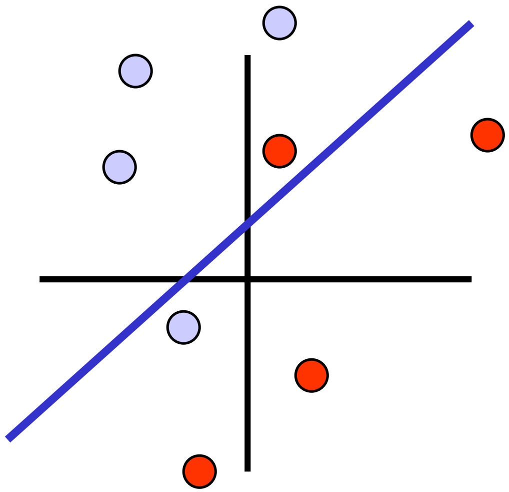


Gradient Descent

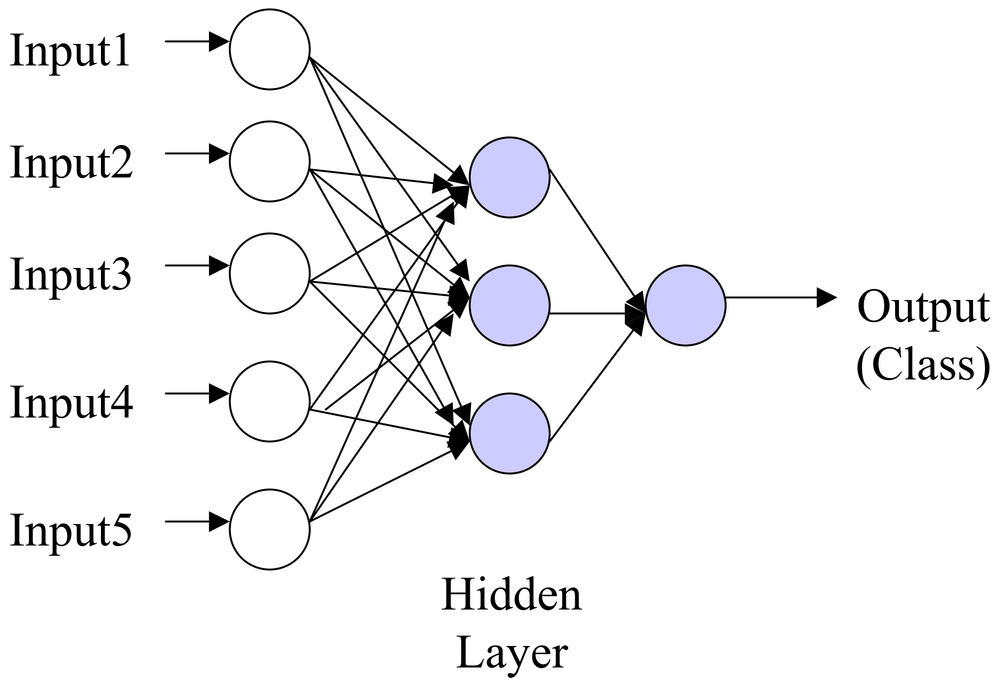
- Inputs $x_1, x_2, \ldots, x_n,$
- Output $\hat{y} = w \bullet x$
- Given a learning set $L = \{ (x,y) \}$, define the
- Training error

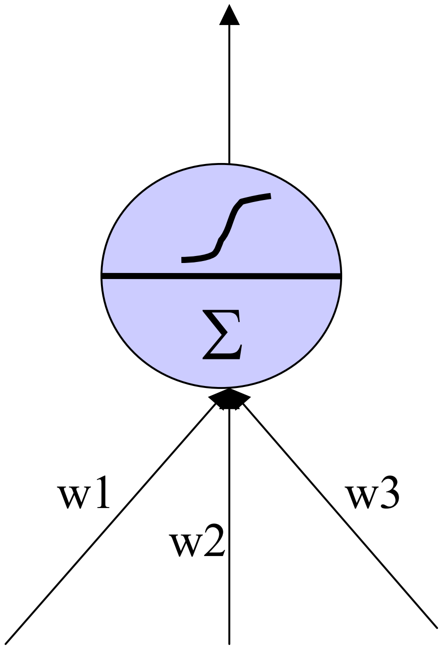
  $$E(w) = (1/2) \sum_L (y - \hat{y})^2$$

- Gradient Descent Rule
  - $w_i \leftarrow w_i + \Delta w_{i,}$
  - $\Delta w_i = - \eta \; \partial E \; / \; \partial w_i$



Multilayer Neural Networks





- **Inputs x_{ji} from node i to node j, with weight w_{ji}**
- **Output $\hat{y}_j = \sigma(\Sigma_i w_{ji} x_{ji})$, where threshold uses smooth logistic function $\sigma(y) = 1/ (1 + \exp(-y))$**

Back Propagation Algorithm for NN

1. Node i in the network may be input, hidden layer, or output; let w_{ji} denote the weight from node i to node j
2. Propagate an input x_{ij} forward through the NN
3. Propagate the errors from the output and the hidden layers backwards through the neural network; let E denote the total error as before
4. update the weights
   – $w_{ij} \leftarrow w_{ij} + \Delta w_{ij}$
   – $\Delta w_{ij} = -\eta \; \partial E \; / \; \partial w_{ij}$

2.5 Derived Attributes

Derived Attributes

- **In practice, statistical models are not computed from the raw data attributes, but rather from *derived* attributes computed from the data attributes.**
- **Derived attributes are often aggregated from multiple records**

Example: The Shuttle

- Problem: determining whether it is safe to launch the Space Shuttle
- Data consists thousands of tables, graphs, spreadsheets and reports

The Data

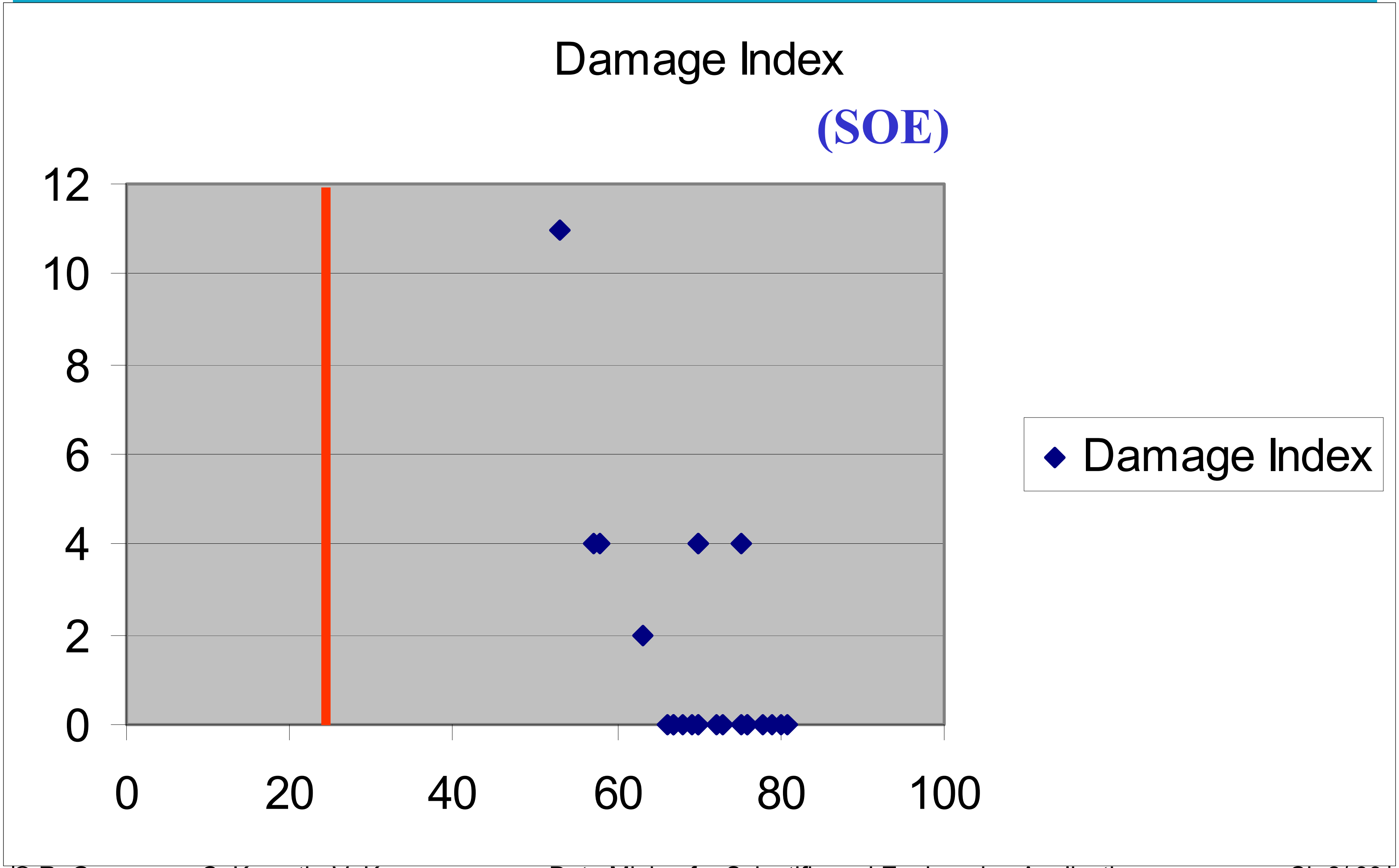
- Thousands of reports, charts, & graphs
- Data below
  - date, temperature, details of O-ring erosion

8	8/30/83	73							
9	11/28/83	70							
41-B	2/3/84	57	0.39	0.75			0.04	3	
41-C	4/6/84	63	0.034	1.8			some	some	
41-D	8/30/84	70	0.046	4			0.028	3	
41-G	10/5/84	78							
51-A	11/8/84	67							
51-C	1/24/85	53					0.01	4.25	0.038
51-D	4/12/85	67	0.68	6	0.011	2.12			
51-B	4/29/85	75	0.005	3.4	0.171	1.59			
51-G	6/17/85	70	0.023	0.88	0.013	1.12			
51-F	7/29/85	81							
51-I	8/27/85	76	0.064	13.5					

Key Idea

- Introduce severity of erosion (SOE) as *derived* attribute
- Ignore most of the details concerning the erosion
- Count number of eroded incidents per type
  - normal or blow by
- Take weighted average to obtain SOE attribute

The Results



2.6 Consistent Classifiers

Following L. Devroye, L. Gyorfi, and G. Lugosi, A Probabilistic Theory of Pattern Recognition, Springer-Verlag, New York, 1996.

What is a Classifier?

- A classifier is a map

  κ: Data Space $\longrightarrow$ Pattern Space

  where the pattern space is of lower dimension
- Example: binary classifier (model)

  $$g : R^d \longrightarrow \{0, 1\} \qquad x \mapsto y$$

- Because of uncertainty, consider $R^d \times \{0,1\}$-valued random pair (X,Y)
- Probability of error

  $$L(g) = P\ (\ g(X) \neq Y\ )$$

What is a Consistent Classifier?

- Supervised learning is a map from iid sequences

$$D_n = X_1, Y_1, X_2, Y_2, \ldots, X_n, Y_n$$

- to a classifier (model)

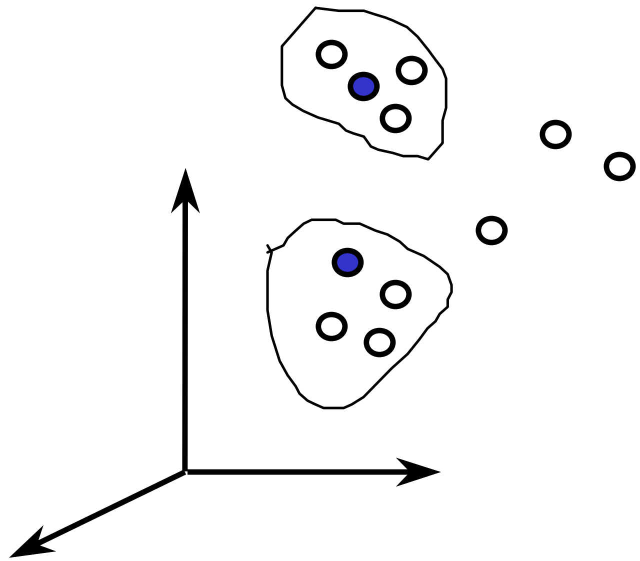
$$Y = g_n\,(X; X_1, Y_1, X_2, Y_2, \ldots, X_n, Y_n)$$

- conditional probability of error

$$L_n = L\,(g_n) = P\,(g_n(\,X; D_n\,) = Y \mid D_n\,)\,.$$

- rule consistent if $E(L_n) \longrightarrow L_{optimal}$.

Nearest Neighbor Learning



To classify ●

1. find nearest three objects

2. classify via majority vote

- Given data $D_n = X_1, Y_1, X_2, Y_2, \ldots, X_n, Y_n$ and a query point X, classify Y by Y_J, where X_J is the nearest point to X .
- Cover-Hart: $\lim\sup E\, L_n \lesssim 2\, L_{optimal}$.

References

Tom M. Mitchell, Machine Learning, WCB McGraw-Hill, Boston, 1997 - a concise introduction to machine learning from the AI perspective.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone, Classification and Regression Trees, Chapman & Hall, New York, 1984. - A classic introduction to trees and their applications to classification and prediction.

Luc Devroye, Laszlo Gyorfi, Gabor Lugosi, A Probabilistic Theory of Pattern Recognition, Springer-Verlag, New York, 1996 - proves probabilistic error estimates for many of the common data mining algorithms.

References

Trevor Hastie, Robert Tibshirani, and Jerome Friedman, The Elements of Statistical Learning, Springer-Verlag, New York, 2001 - data mining algorithms from a statistical point of view.

N. Vapnik, The Nature of Statistical Learning Theory, Second Edition, Springer-Verlag, 2000 - an technical book covering VC dimension and related concepts.

Nello Cristianini and John Shawe-Taylor, An Introduction to Support Vector Machines, Springer-Verlag, New York, 2000 - one of the first books on support vector machines, which are emerging as a powerful data mining technique.