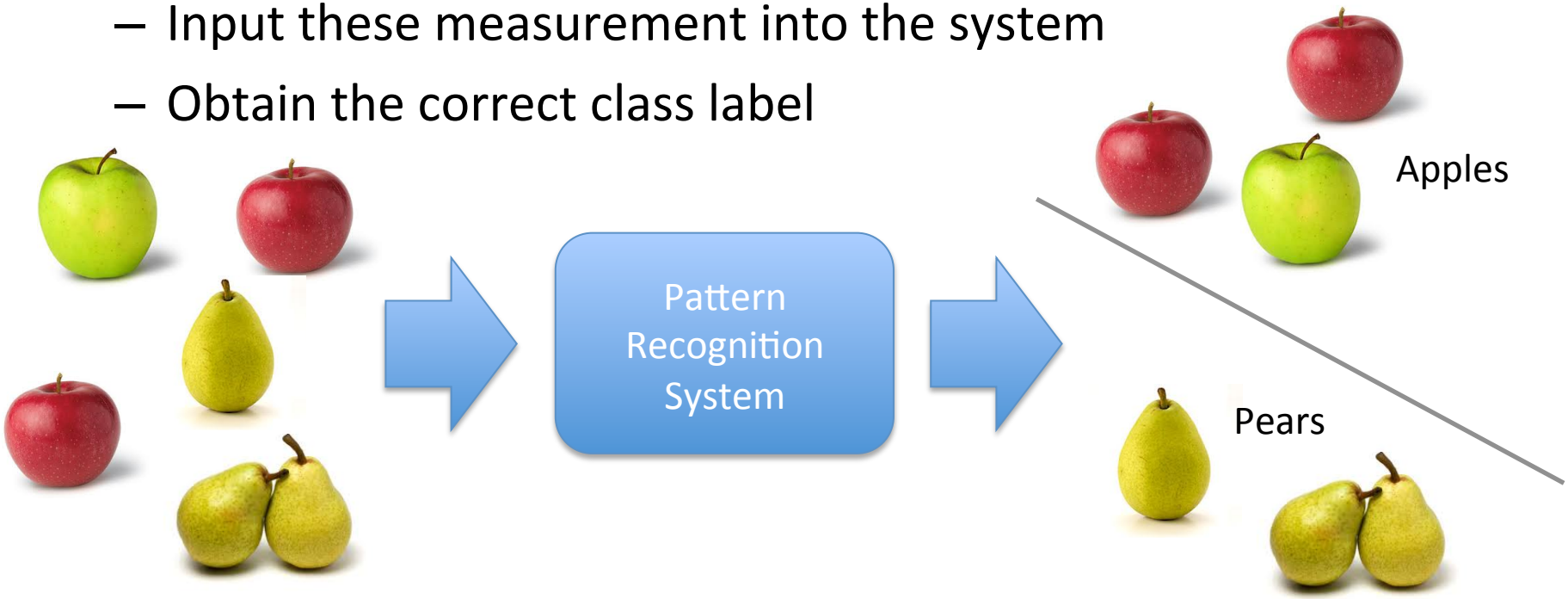
Intro to Pattern Recognition

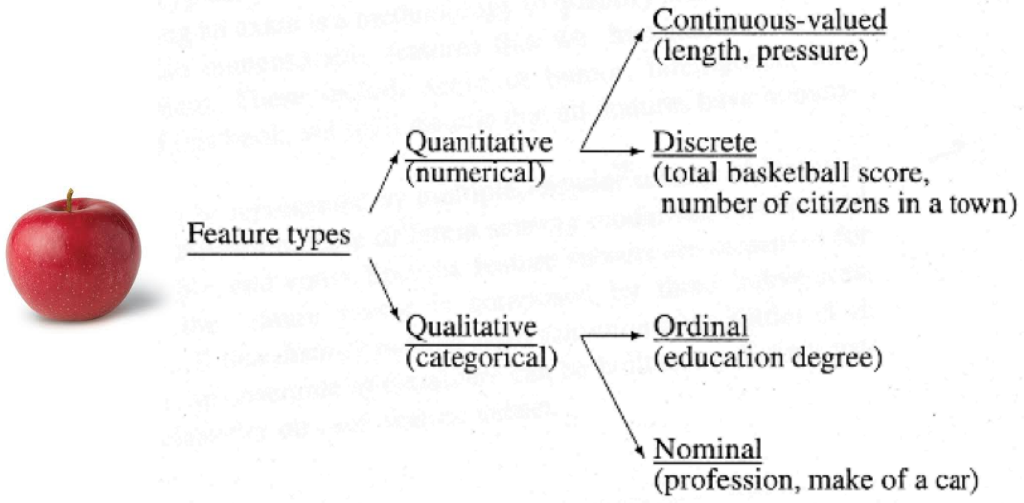
CSCI 8260 – Spring 2016
Computer Network Attacks and Defenses

What's Pattern Recognition?

- Target problem: build a system that automatically recognizes and categorizes objects into *classes*
  - Measure a number of *features* that characterize the object (e.g., color, size, shape, weight, etc.)
  - Input these measurement into the system
  - Obtain the correct class label



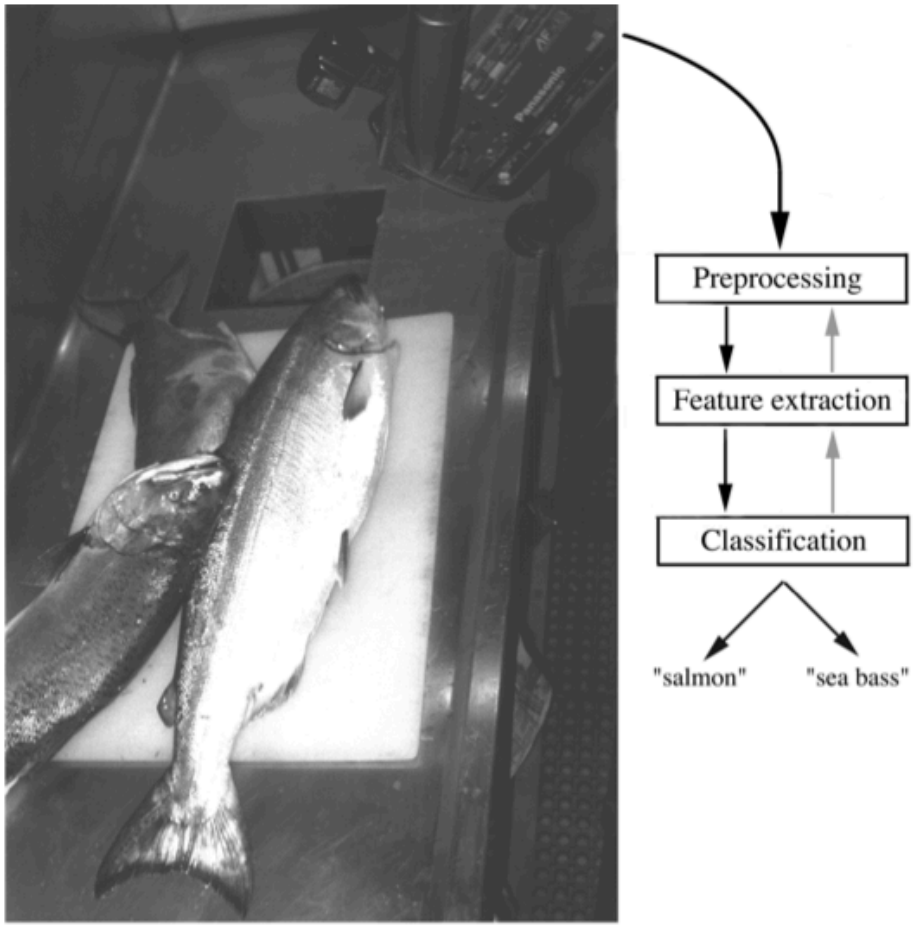
Types of Features



Example features for apples

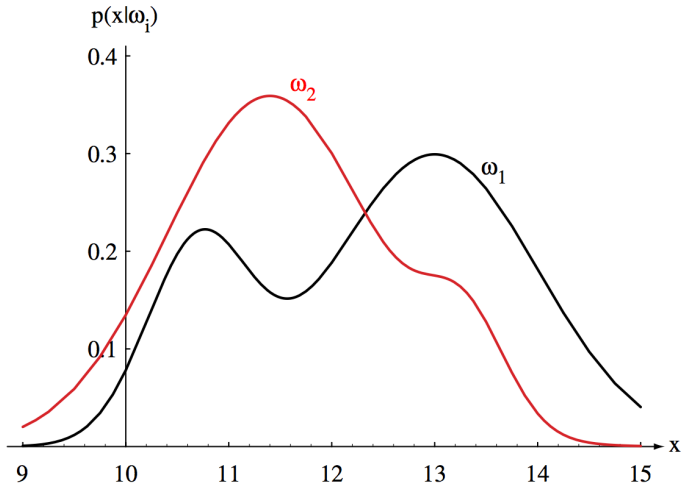
- Quantitative
  - Weight (continuous)
  - Darkness level, 1-255 (discrete)
- Qualitative
  - Color {red,yellow,green} (nominal)
  - Sweetness level {sour,sweet,very-sweet,extremely-sweet} (ordinal)

Real-World Application Example



Bayesian Decision Theory

- Notation
  - X = [length, lightness]
  - W1 = salmon
  - W2 = see bass



- Assume we know the following
  - P(X|W1) = P(length,lightness|salmon)
  - P(X|W2) = P(length,lightness|see bass)

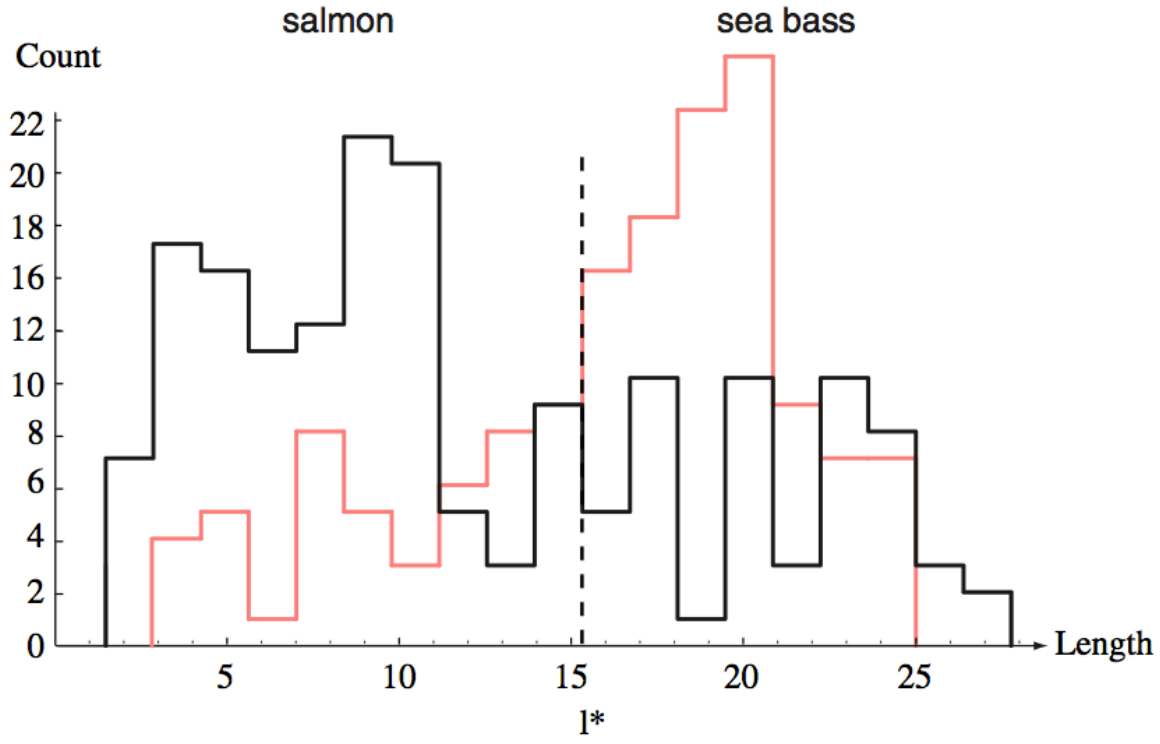
- Bayes Rule
  - P(W1|X) = P(X|W1)P(W1)/P(X)
  - P(W2|X) = P(X|W2)P(W2)/P(X)

- Optimum Decision rule for a new input X'
  - Decide W1 if P(W1|X') > P(W2|X'), otherwise decide W2
  - In other words: P(X'|W1)/P(X'|W2) > P(W2)/P(W1) : then W1, else W2

In reality, we do not know the true P(X|W1) and P(X|W2)!

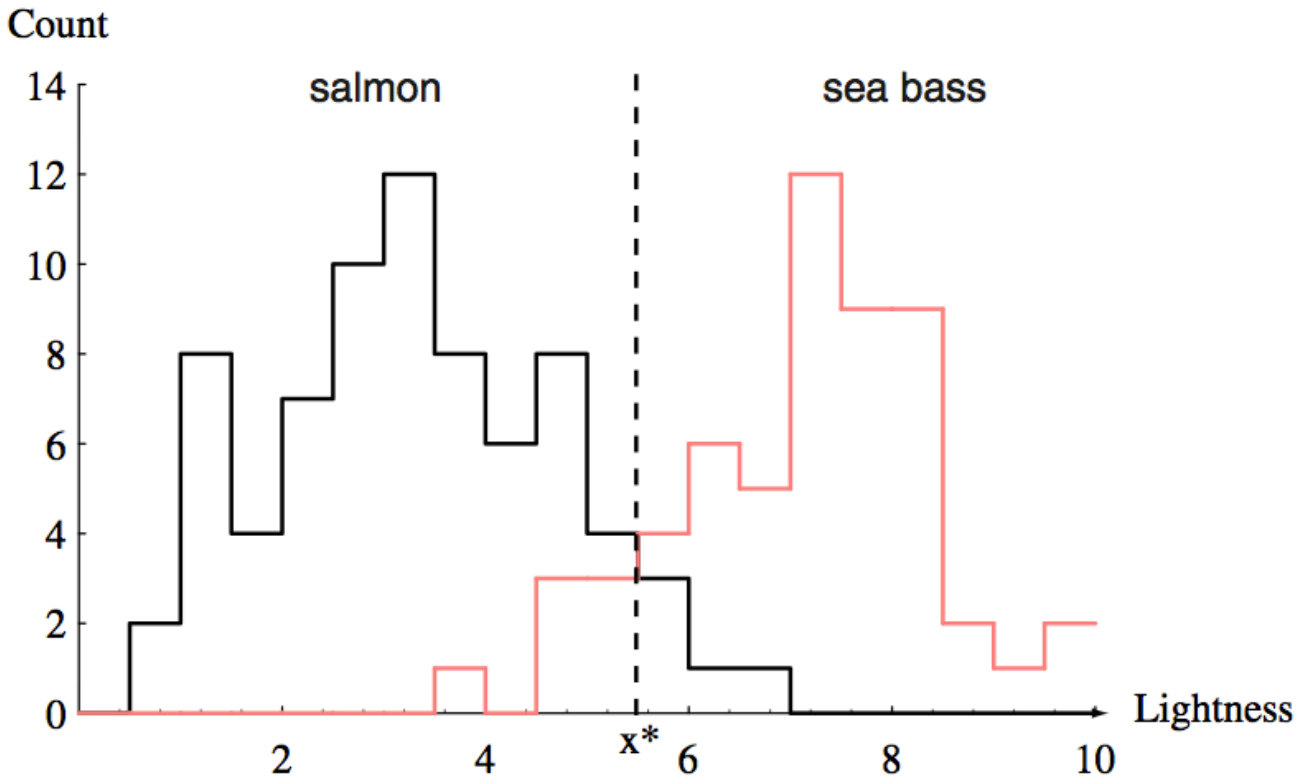
Approximate Class-conditional distributions

- ~P(length|W_i)
  - E.g., estimated from examples of labeled fish provided by a fisherman



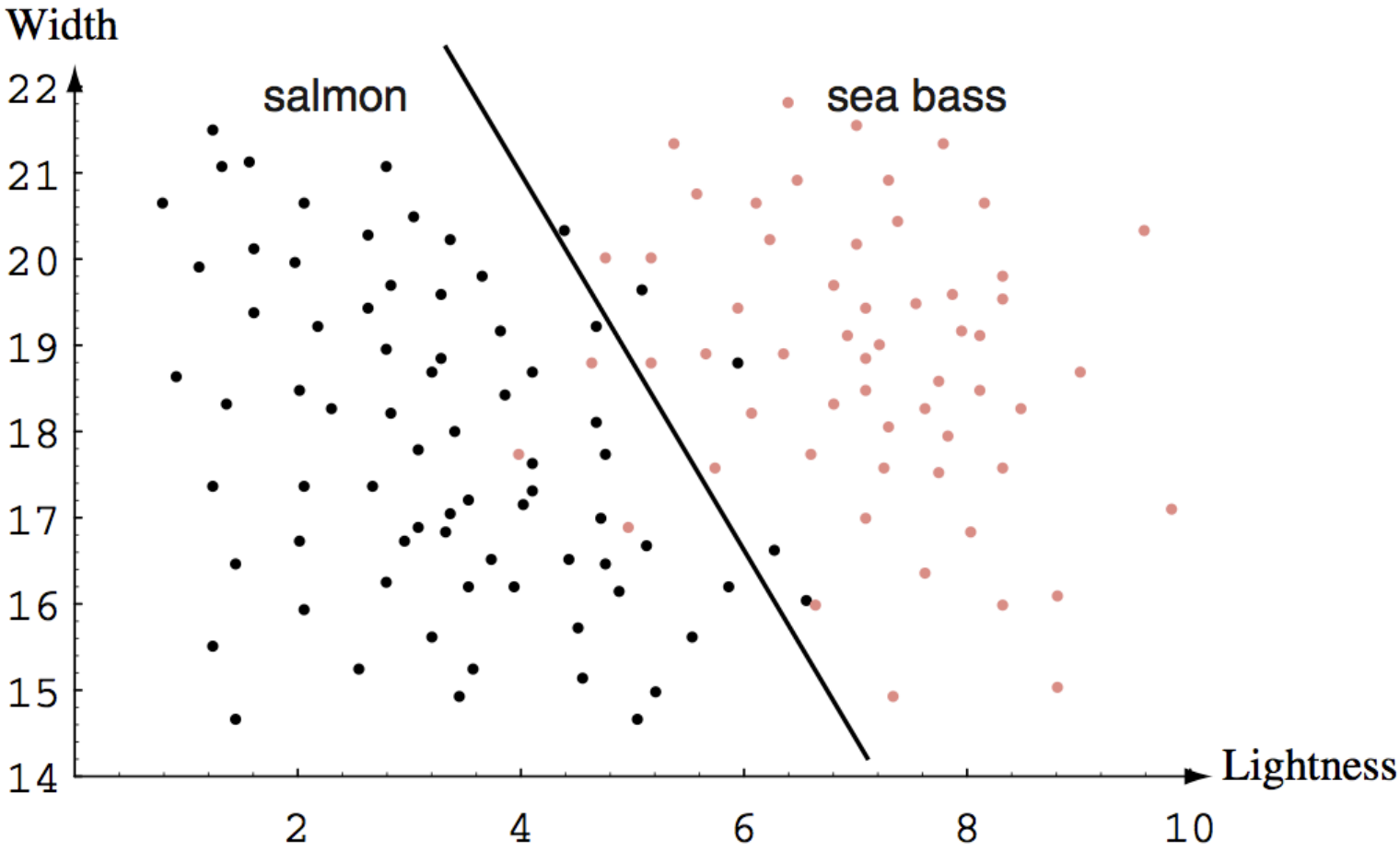
Approximate Class-conditional distributions

- ~P(lightness|W_i)
  - Estimated from examples



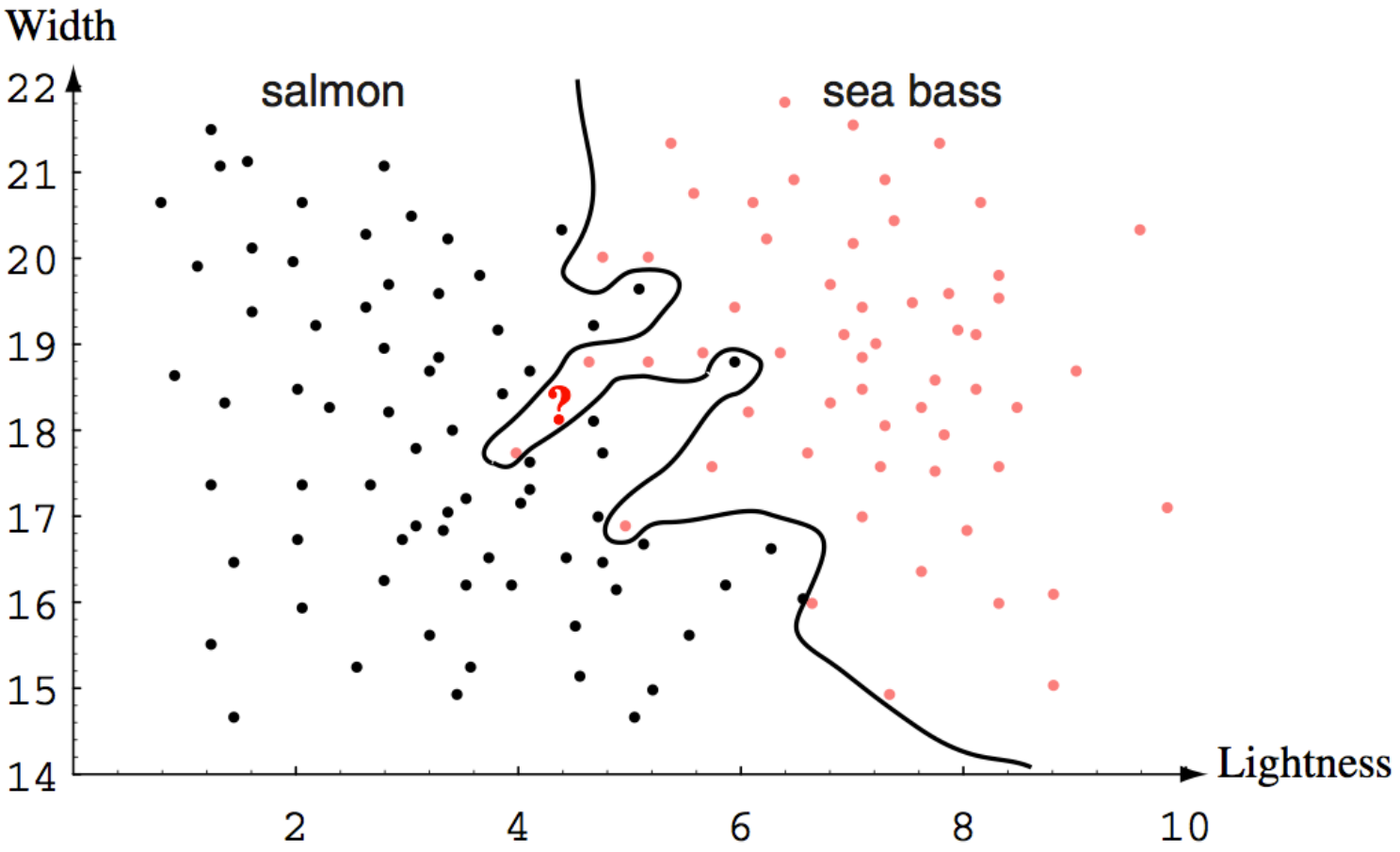
Learning a Decision Surface

- Linear classifier



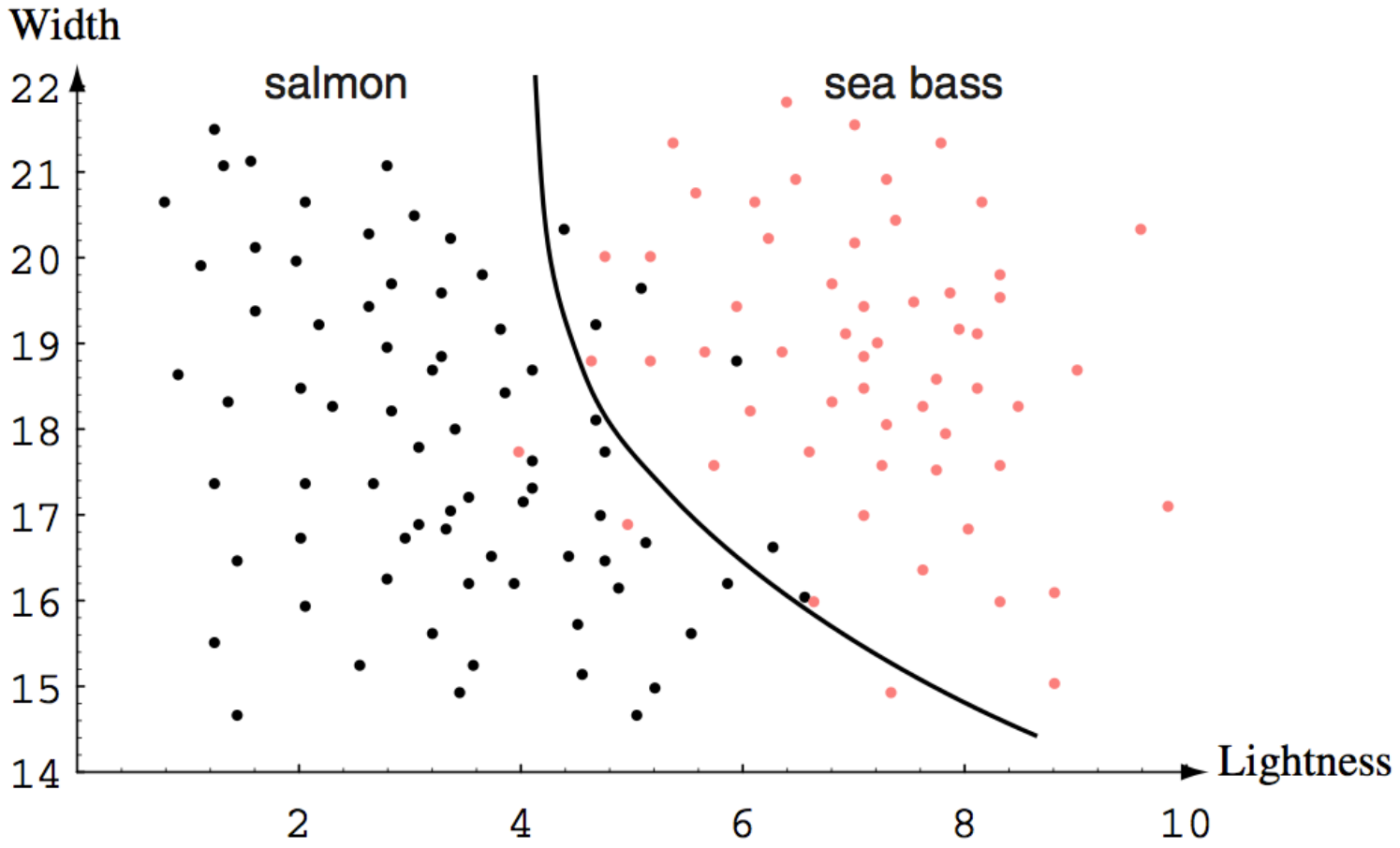
“Flexible” Classifier

- Very “flexible” classifier (e.g., Artificial Neural Nets)



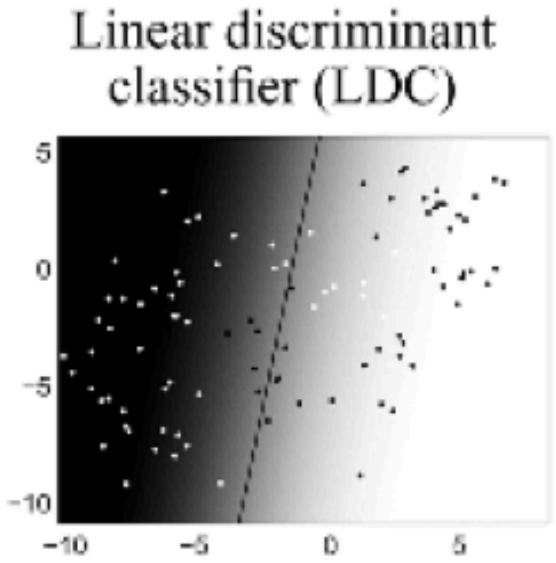
Learning a Decision Surface

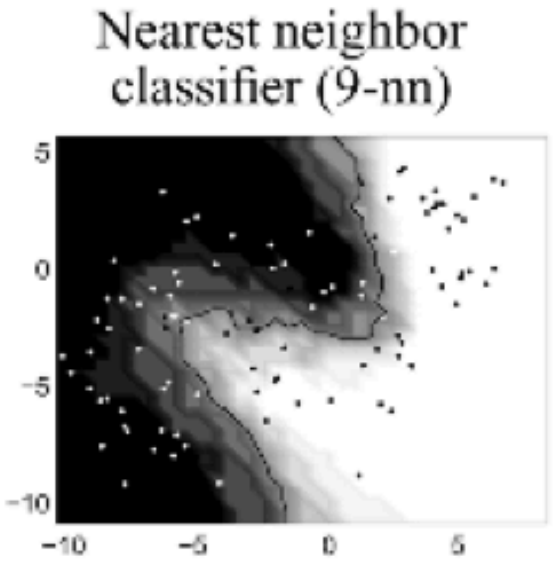
- Quadratic Classifier

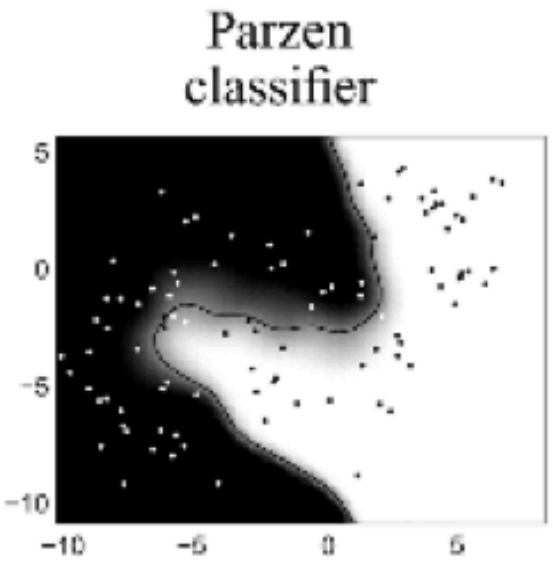


Decision Surface

- Different classifier learn different models
  - Different generalization ability
  - Different accuracy when testing on a separate dataset







Supervised Learning in practice

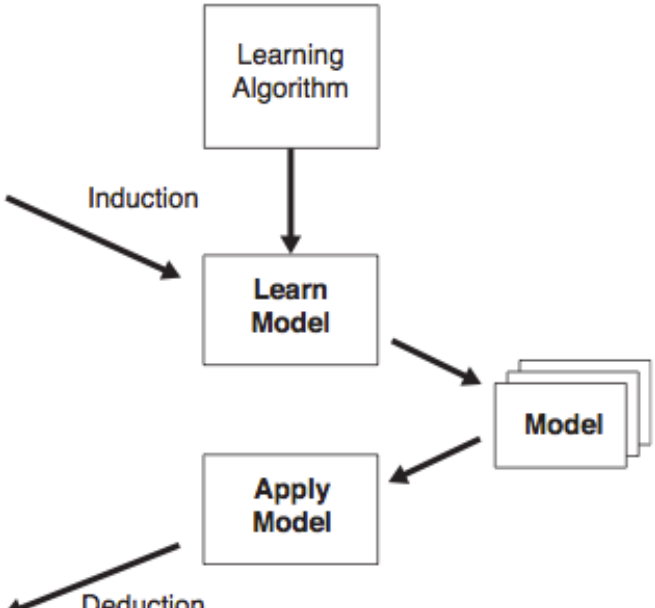
- Assume you have a large dataset of *labeled examples*
  - Each entry represents an objects (its features)
  - Each object is assigned a "ground-truth" label"
    - e.g., labeled fruits, labeled
- Split the dataset in two parts
  - Use a training set to automatically learn an object model
  - Use a test set to evaluate how your model is going to perform on never-before-seen data

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Test Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?



Another example

Table 4.1. The vertebrate data set.

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
human	warm-blooded	hair	yes	no	no	yes	no	mammal
python	cold-blooded	scales	no	no	no	no	yes	reptile
salmon	cold-blooded	scales	no	yes	no	no	no	fish
whale	warm-blooded	hair	yes	yes	no	no	no	mammal
frog	cold-blooded	none	no	semi	no	yes	yes	amphibian
komodo dragon	cold-blooded	scales	no	no	no	yes	no	reptile
bat	warm-blooded	hair	yes	no	yes	yes	yes	mammal
pigeon	warm-blooded	feathers	no	no	yes	yes	no	bird
cat	warm-blooded	fur	yes	no	no	yes	no	mammal
leopard shark	cold-blooded	scales	yes	yes	no	no	no	fish
turtle	cold-blooded	scales	no	semi	no	yes	no	reptile
penguin	warm-blooded	feathers	no	semi	no	yes	no	bird
porcupine	warm-blooded	quills	yes	no	no	yes	yes	mammal
eel	cold-blooded	scales	no	yes	no	no	no	fish
salamander	cold-blooded	none	no	semi	no	yes	yes	amphibian

Learned Decision Tree

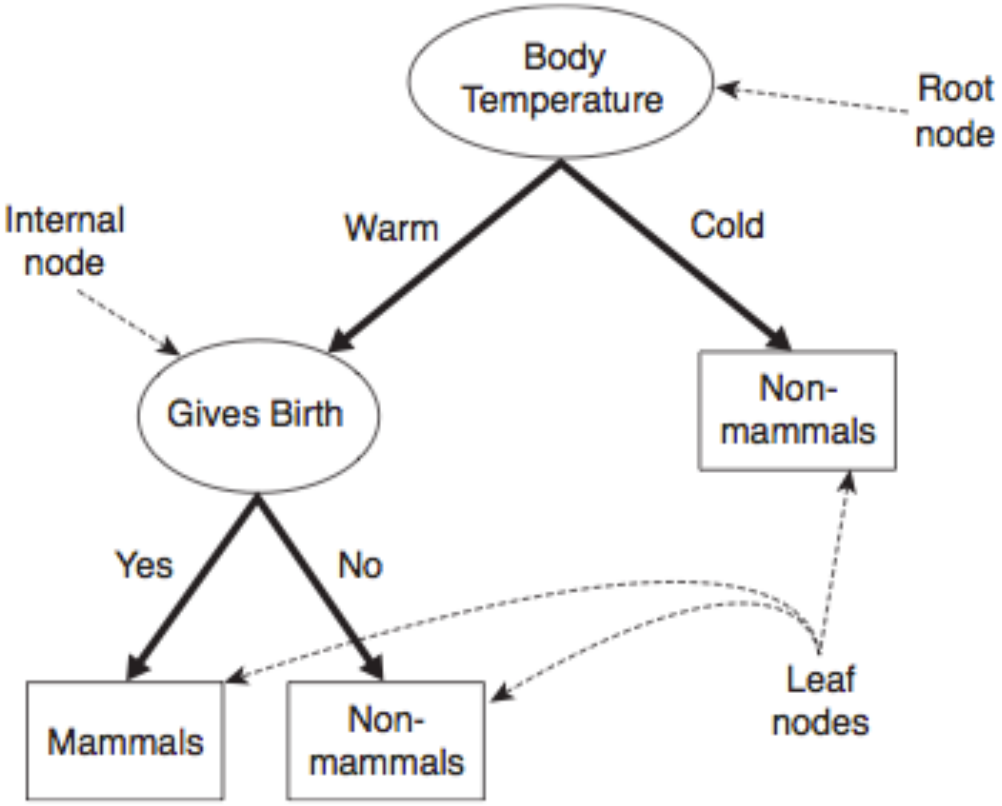


Figure 4.4. A decision tree for the mammal classification problem.

Evaluation Metrics

- Test results help estimate accuracy

Table 4.2. Confusion matrix for a 2-class problem.

		Predicted Class	
		$Class = 1$	$Class = 0$
Actual Class	$Class = 1$	f_{11}	f_{10}
	$Class = 0$	f_{01}	f_{00}

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

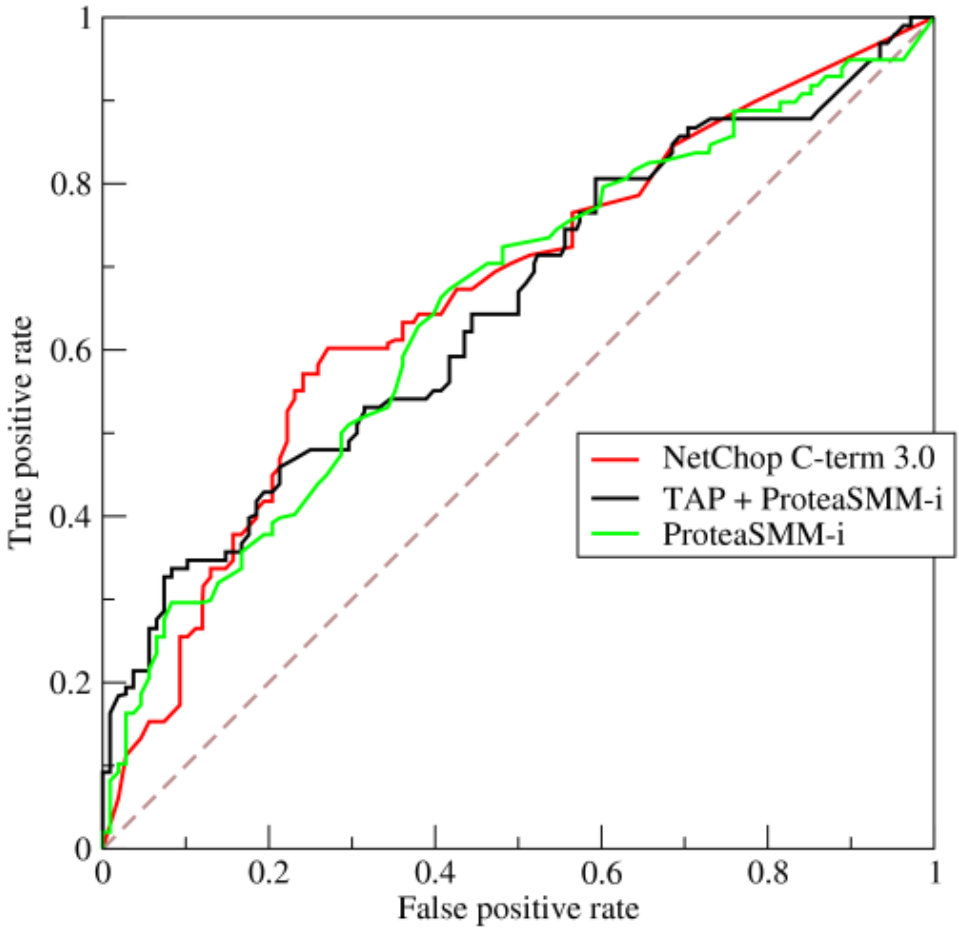
$$\text{Error rate} = \frac{\text{Number of wrong predictions}}{\text{Total number of predictions}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

False Positives vs. True Positives

- Let N be the total number of test instances (or patterns) in the test dataset
- Instances can belong to two possible classes
  - Positive (or target) class
  - Negative class
- TP = Number of correctly classified positive sampels
- FP = Num of misclassified negative samples

ROC and AUC

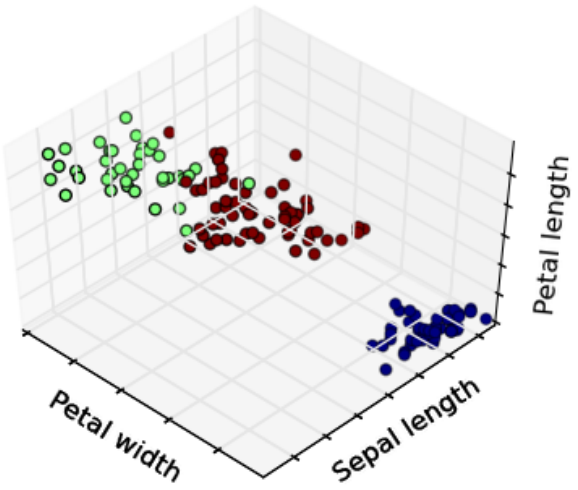
- ROC = Receiver Operating Characteristic Curve
  - Plots trade-off between FPs and TPs for varying detection thresholds
- AUC = Area under the ROC (the larger the better)



Unsupervised Learning

- Learn from unlabeled examples
  - Seriously???
  - Yes!

- Discover groups of similar objects in a multi-dimensional feature space
  - Provides new useful information
  - Discovers new “concepts” or previously unknown classes



Clustering

- Different clustering algorithms find different data clusters

Fig. 1.16 Banana data for the clustering example.

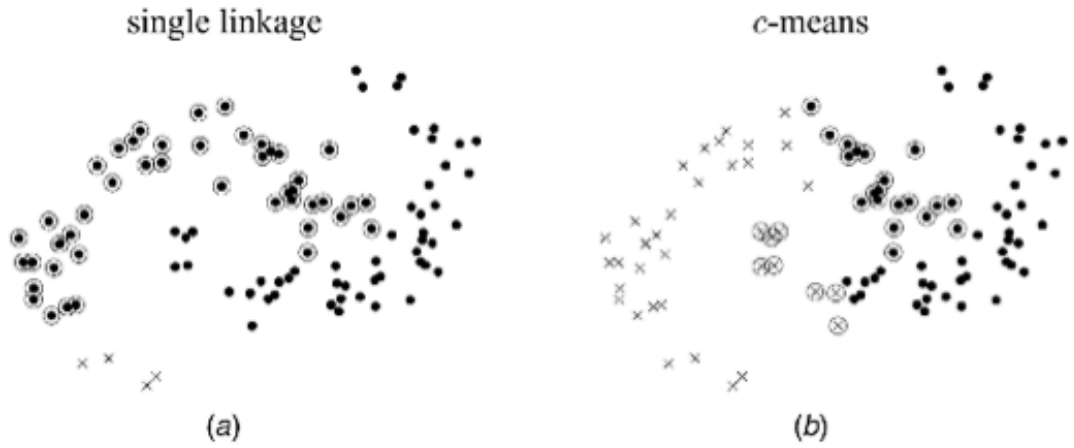
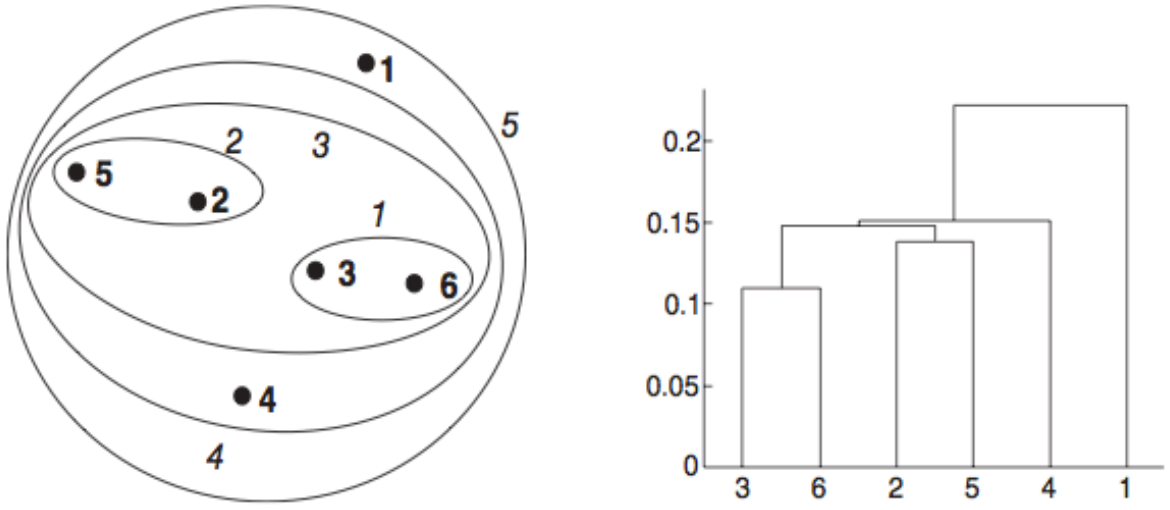


Fig. 1.17 Results from the single linkage (a) and c-means (b) clustering on a banana data set with 50 points on each banana shape. The "mislabeled" points are encircled.

Hierarchical Clustering

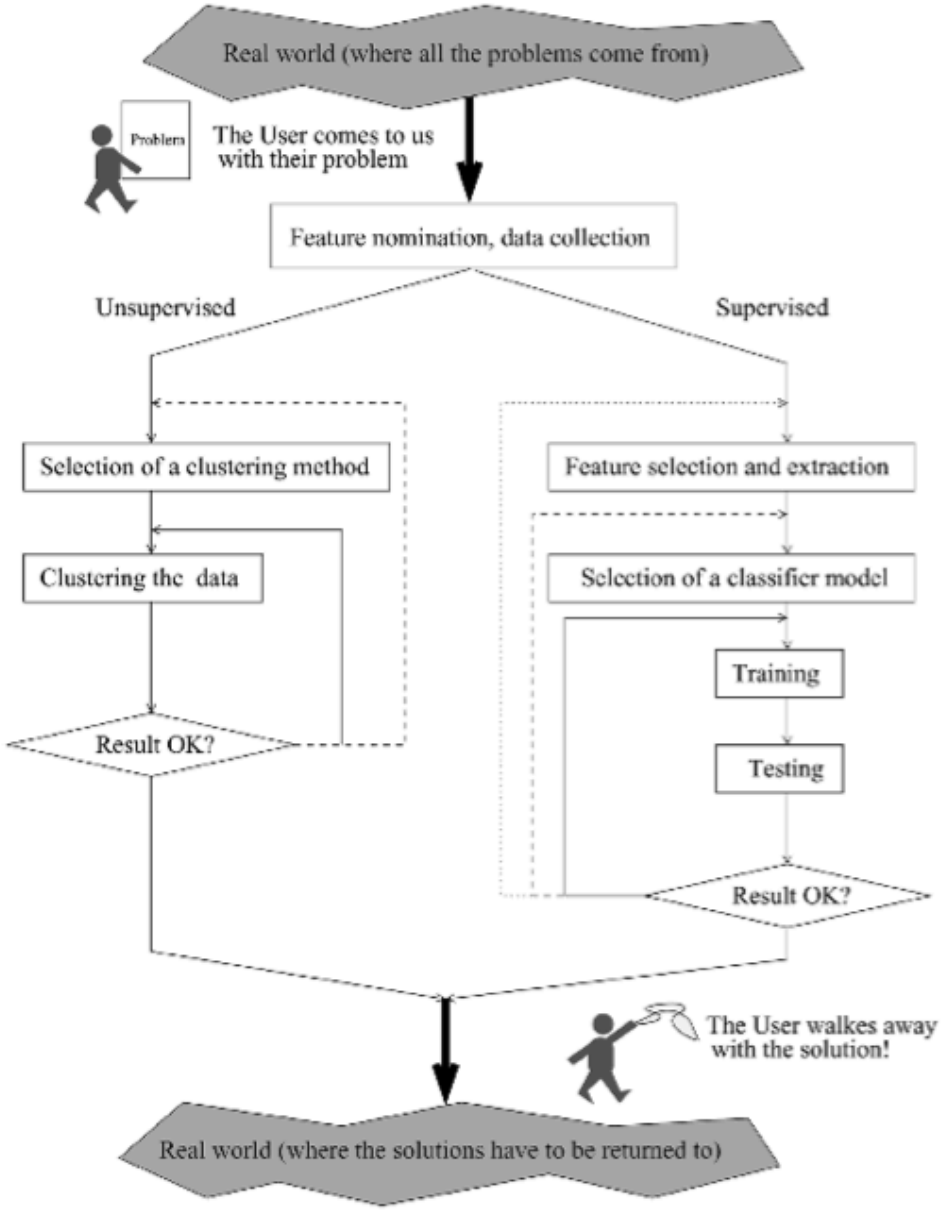


(a) Single link clustering.

(b) Single link dendrogram.

Figure 8.16. Single link clustering of the six points shown in Figure 8.15.

Pattern Recognition Process



Security Applications

- Network/Host-based Intrusion Detection
- Malware Detection
- Detecting Search Poisoning
- Detecting Malicious Domain Names
- Etc.

Example Security Application

- Given a PE file (e.g., an MS-Windows .exe file)
  - Decide if the file is "packed" without running it [1]

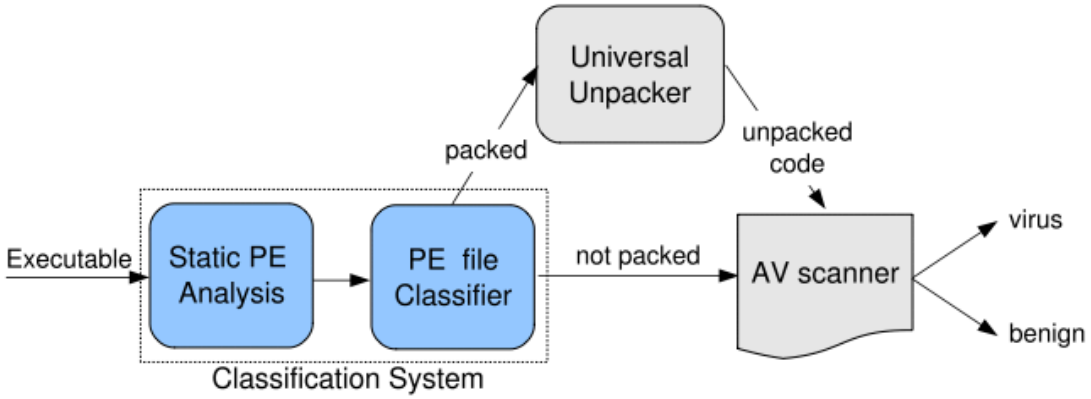


Fig. 1. Example of use of our classification system.

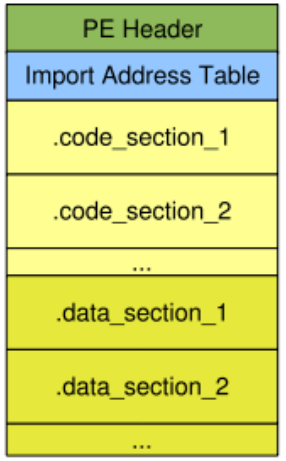


Fig. 2. PE file format.

[1] Roberto Perdisci, Andrea Lanzi, Wenke Lee. "Classification of Packed Executables for Accurate Computer Virus Detection." Pattern Recognition Letters, 29(14), 2008, pp. 1941-1946.