

CS 780

Data Mining for Multimedia Data

Dr. Jessica Lin

Outline

- Administrative stuff
- Multimedia Data – What is this class all about?
- Light Review on Data Mining

Administrative stuff

- Class website: http://www.cs.gmu.edu/~jessica/cs780_f13.html
- Prerequisite: INFS 755 or CS 659 (previously CS 750), or equivalent data mining/machine learning/pattern recognition course, or permission of instructor
- Textbooks:
 - ★ Recommended: Data Mining: Concepts and Techniques, 2nd Ed, Morgan Kauffmann Publishers
 - ★ Reading materials will be given in class
- Grading
 - ★ Class Participation: 5%
 - ★ Assignments: 25%
 - ★ Project: 30%
 - ★ Midterm: 20%
 - ★ Take-home Final: 20%

Homework/Project Submission

- <https://mymasonportal.gmu.edu/webapps/portal/frameset.jsp>
- Login with your GMU student account
- Click on the Courses tab on the upper right hand corner
- Choose CS 780
- Use Blackboard for:
 - ★ Electronic submission of assignments/projects
 - ★ Checking grades
 - ★ Getting course materials such as homework solutions

Honor Code System

- GMU honor Code

<http://academicintegrity.gmu.edu/honorcode/>

- In addition, the CS Department has specific honor code policies for programming projects, etc.:

<http://cs.gmu.edu/wiki/pmwiki.php/HonorCode/HomePage>

- For this class

- ★ Assignments are individual, unless specified otherwise.
Group discussions are encouraged but final solution and write up must be individual.
- ★ You may work in a team of 2 for the project.
- ★ Exams: individual effort

Email Policy

- I strongly prefer that you only email me from your official GMU email. If you must email me from another account, you must state your full name and your official GMU email address.
- We will use Piazza, a free online class Q&A platform. You should have received an invitation to sign up.
- Think before posting the question (e.g. is the answer in the book or the lecture slides?). You are encouraged to answer each other's questions, but make sure you don't just give away the answers to the assignments/projects. This will be considered a form of academic dishonesty. If your questions reveal your solutions to the assignments, send them to me as private posts.

Outline

- Administrative stuff
- Multimedia Data – What is this class all about?
- Light Review on Data Mining

Some of the World's Largest Databases

- Facebook: 100 PB of data; 955 Million users; 200 Billion images (2012)
- Google: processes 20 PB of data *per day* (2008)
- Google Maps: 20 PB of data
- YouTube: Over 4 billion videos viewed per day; 72 hours of video uploaded every minute (2012)
- Twitter: 200 Million active users sending 400 Million tweets per day in 2013 (compared to 360M tweets in 2012 and 200M tweets in 2011)
- Wikipedia (English): Over 4 million articles in 2013
- NOAA's National Climate Data Center: 6 PB

- **What do they have in common?**

Multimedia Data

- Any type of information medium that can be represented, processed, stored and transmitted over network in digital form
- Beyond alphanumeric data
- Usually very high dimensional
- Different data types: text, image, audio, speech, hypertext, video, etc.
- Typical applications:
 - ★ Social Media
 - ★ GIS (Geographical Information System)
 - ★ Entertainment
 - ★ Surveillance
 - ★ Scientific disciplines
 - ✓ biology, astronomy, earth sciences, meteorology
 - ★ Medicine
 - ★ Search Engine

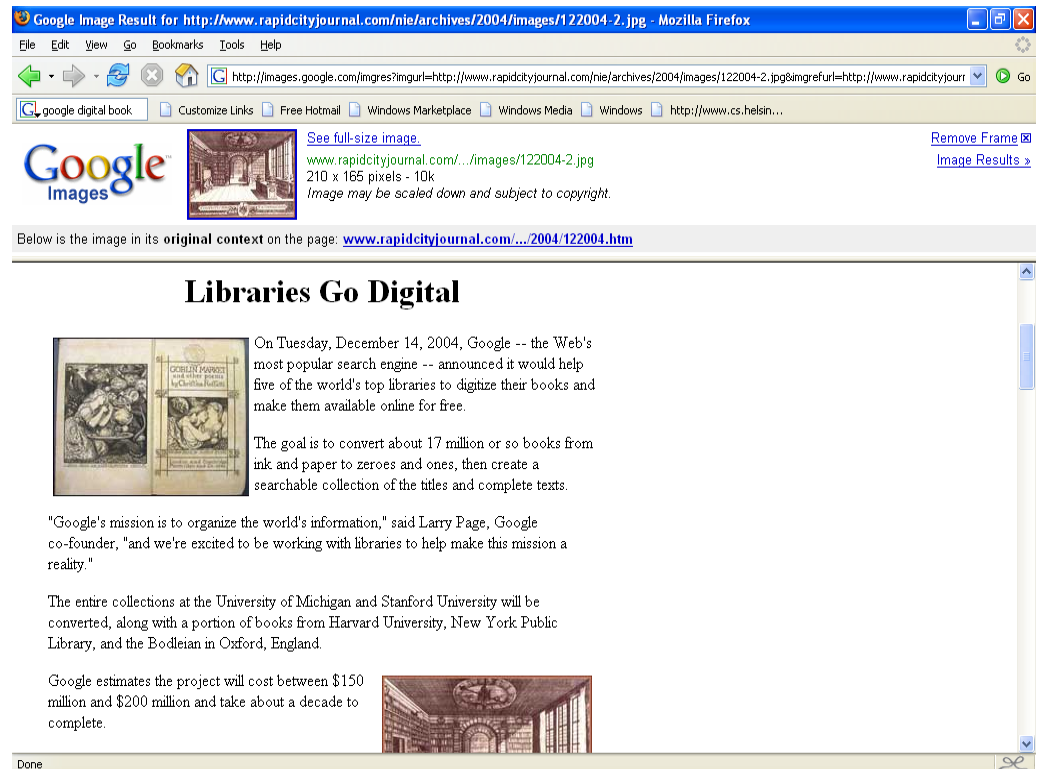
Text/Hypertext Data

- Search Engine
- Digital Library
- Newspaper Archive
- Emails

:

■ Challenges:

- ★ High Dimensional
- ★ Concept discovery
 - ✓ Synonym/Acronym
- ★ Finding previously unknown information

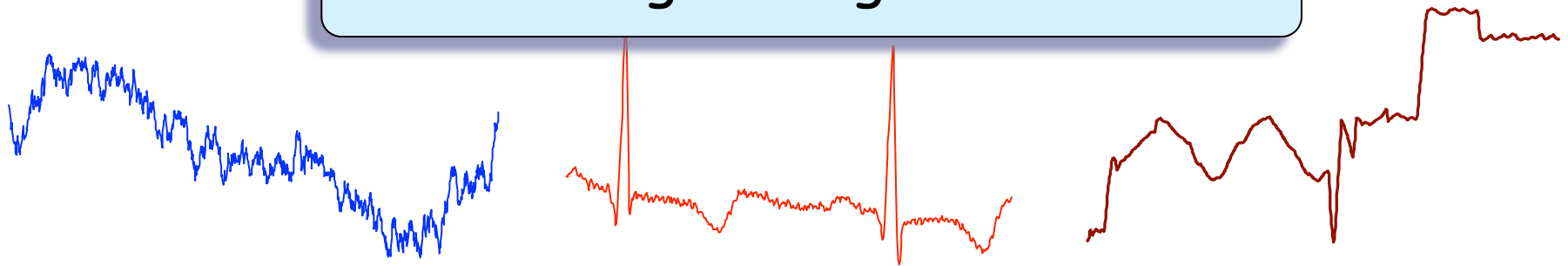


Time Series

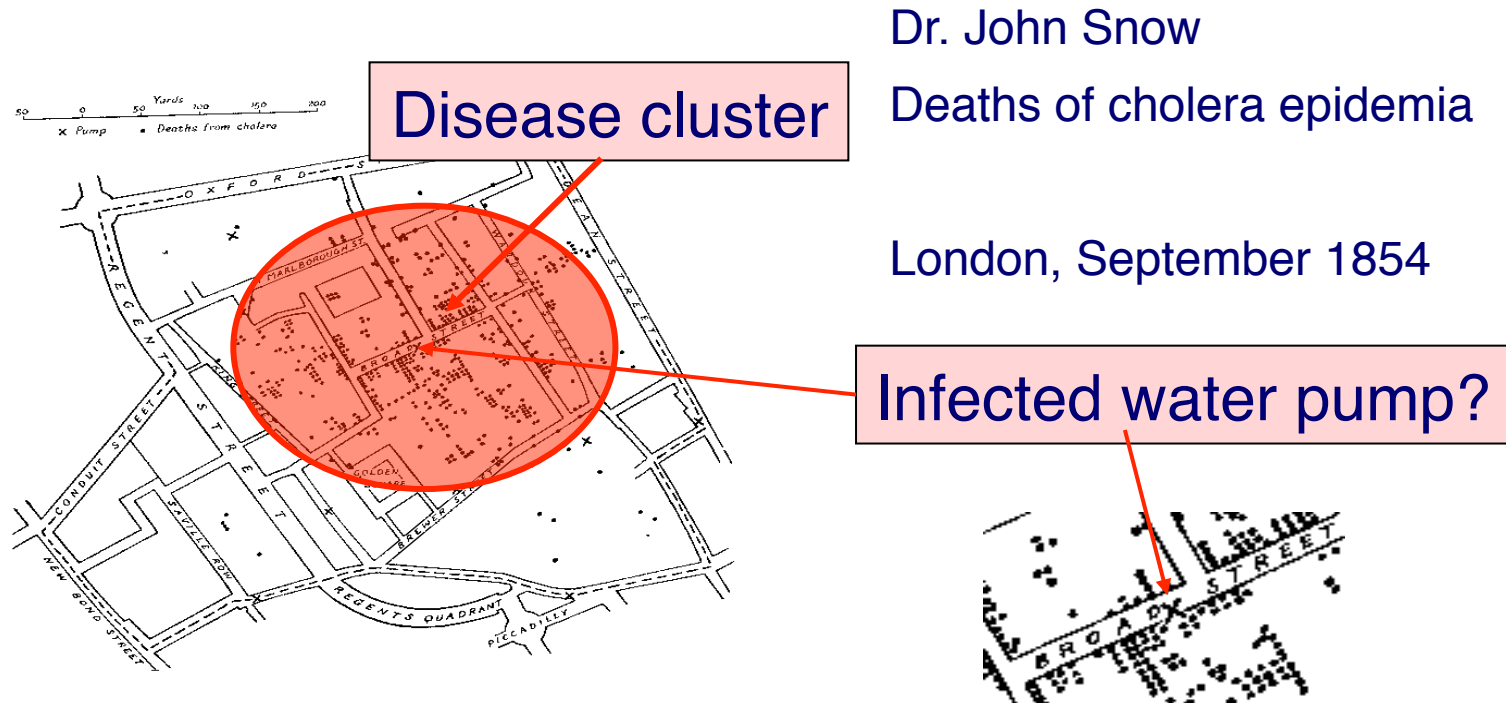
People measure things...

- *Their blood pressure*
- *President Obama's popularity rating*
- *The annual rainfall in Seattle*
- *The value of their Google stock*

...and things change over time...



Spatial Data



■ Modern examples of spatial patterns:

- ★ Cancer clusters to investigate environmental health hazards
- ★ Unusual warming of Pacific Ocean affecting U.S. weather
- ★ Crime hotspots to plan police patrol
- ★ West Nile virus spreading from Northeast U.S. to West and South

Spatial-Temporal Data

■ People Move

- ★ Cars, planes
- ★ Sometimes too fast...
- ★ Applications:
 - ✓ Identify congested area in the future
 - ✓ Mobile usage



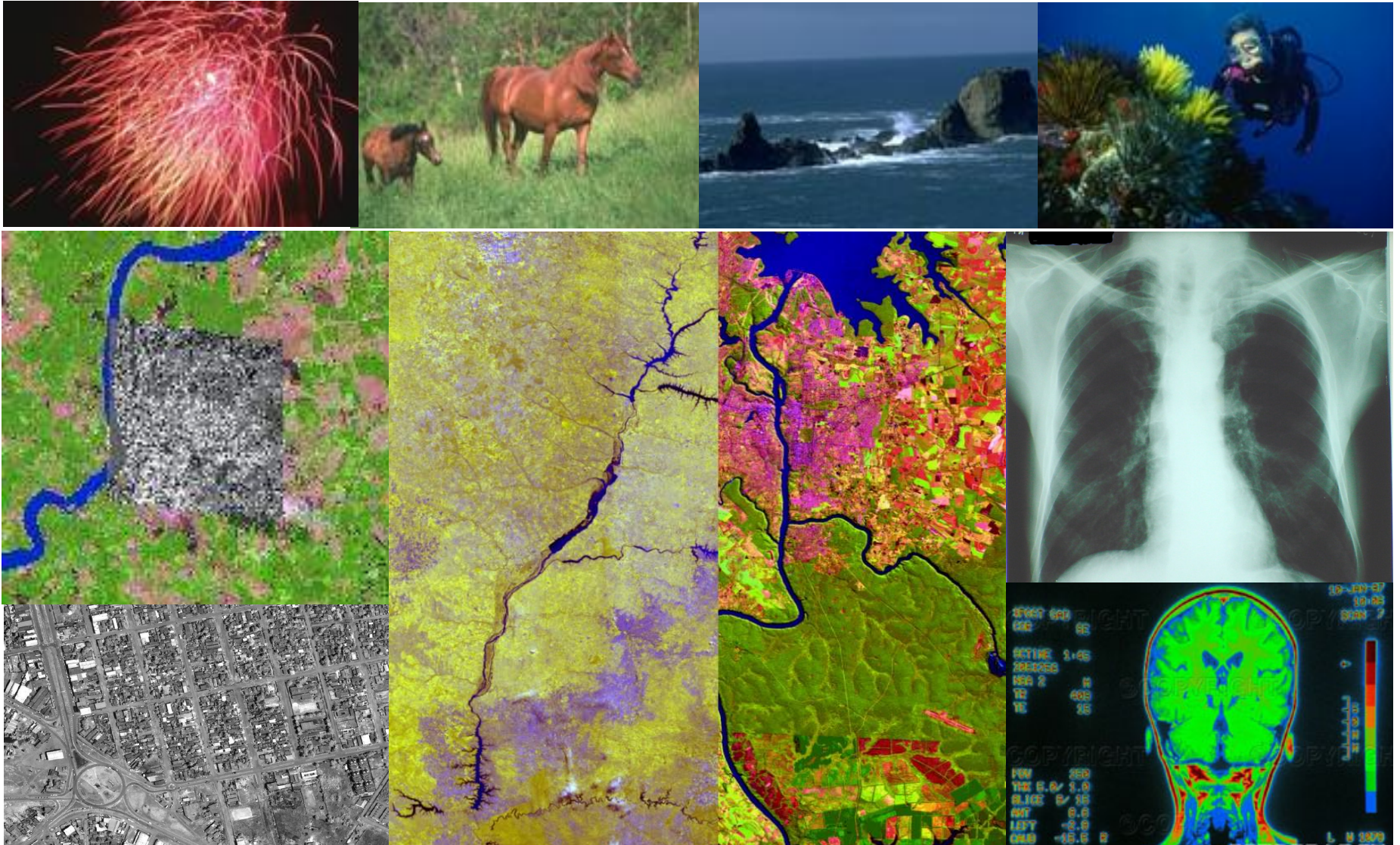
■ Data collected at different locations

- ★ Earth Sciences data
- ★ Applications:
 - ✓ Tracking/predicting hurricanes
 - ✓ Hazardous atmospheric release study



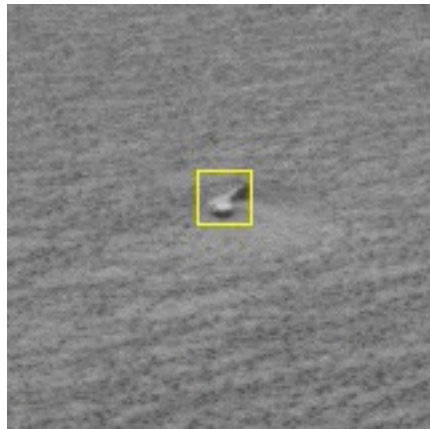
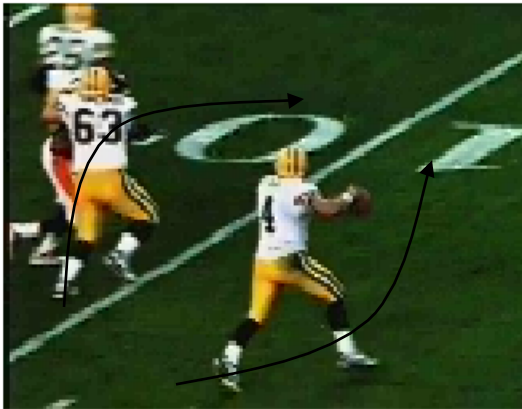
Image Data

■ Satellites/Medical/Photos/...



Video Sequences

- Event-based video sequences
- Target tracking
- Video Surveillance



Social Media

- Facebook
- Twitter
- Google+
- Foursquare
- Vine Video
- ...

Music

- Shazam (<http://www.shazam.com/>)
 - ★ commercial mobile phone based music identification service
- Pandora, Last.fm, Spotify, etc
 - ★ Music streaming and recommendation

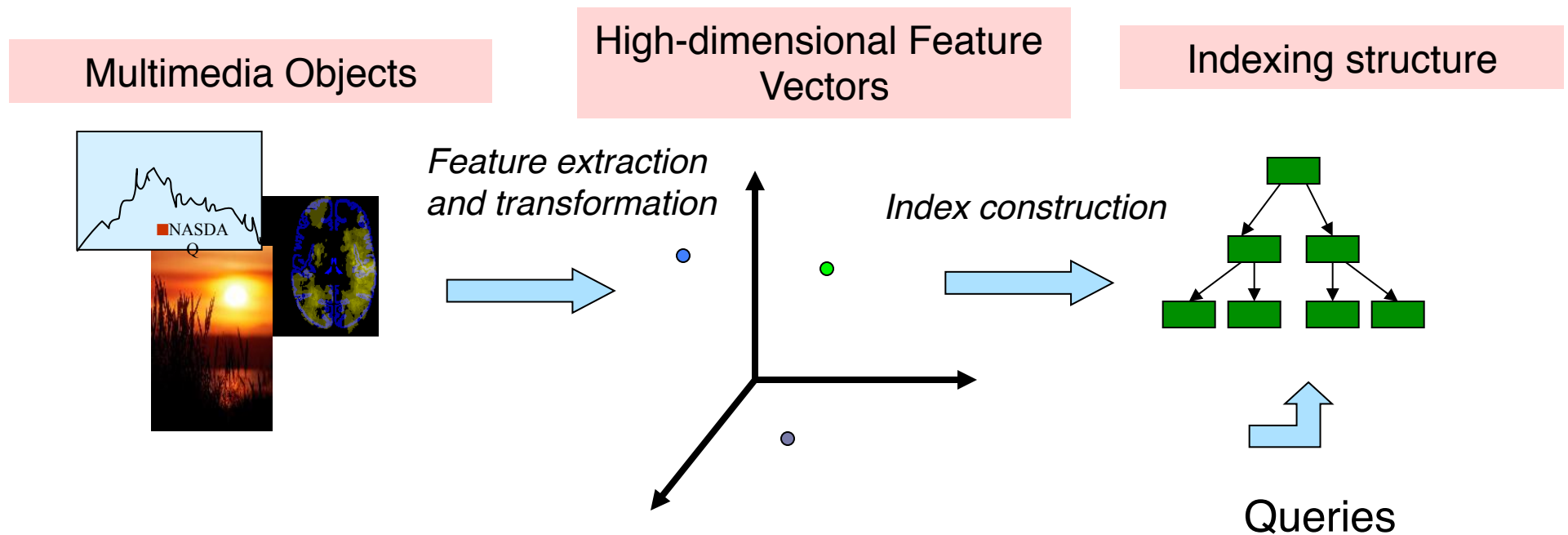
Multimedia Data

■ So far we've seen:

- ★ Text/Hypertext
- ★ Image, Video, Music
- ★ Spatial/Spatial-Temporal Data
- ★ Time Series
- ★ Multimedia Database: combination of different types of data
- ★ Social media

High-dimensional indexing

- Most multimedia data are high-dimensional
- Need efficient feature extraction and high-dimensional indexing



Light Review on Data Mining

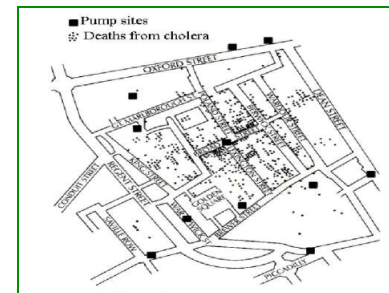
Some slides were created by Professor Eamonn Keogh

Large-scale Data is Everywhere!

- There has been enormous data growth in both commercial and scientific databases due to advances in data generation and collection technologies
- New mantra
 - Gather whatever data you can whenever and wherever possible.
- Expectations
 - Gathered data will have value either for the purpose collected or for a purpose not envisioned.



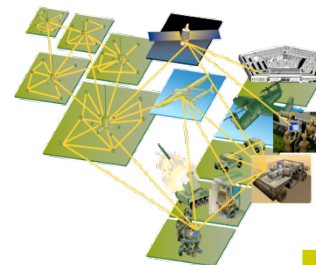
■ **Homeland Security**



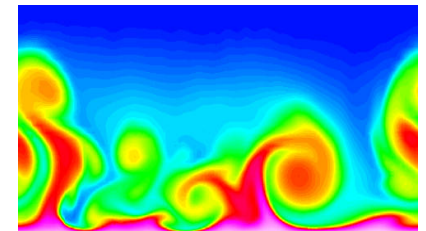
■ **Geo-spatial data**



■ **Business Data**



■ **Sensor Networks**



■ **Computational Simulations**

Why Data Mining? Commercial Viewpoint

■ Lots of data is being collected and warehoused

★ Web data

- ✓ Google processes 20 PB/day
- ✓ Facebook has 955M active users
- ✓ Twitter has more than 400M tweets/day

★ purchases at department/grocery stores, e-commerce

- ✓ Amazon has 42 TB of data

★ Bank/Credit Card transactions

■ Computers have become cheaper and more powerful

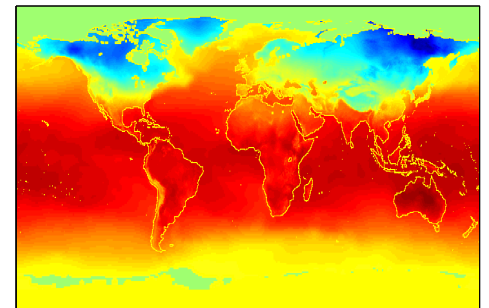
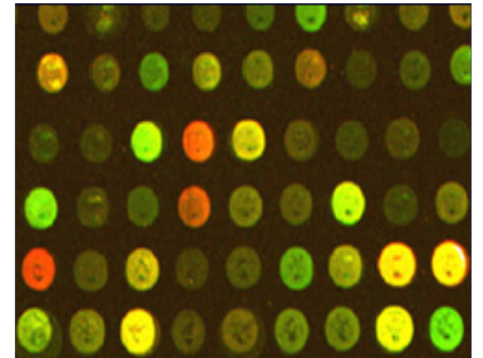
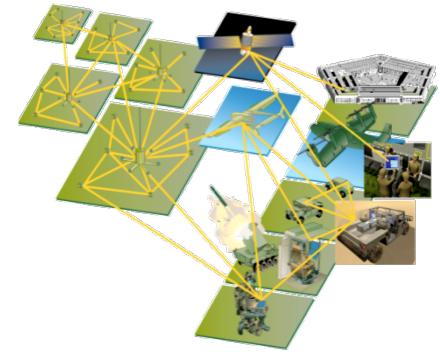
■ Competitive Pressure is Strong

- ★ Provide better, customized services for an edge (e.g. in Customer Relationship Management)



Why Data Mining? Scientific Viewpoint

- Data collected and stored at enormous speeds
 - ★ remote sensors on a satellite
 - ✓ NASA EOSDIS archives over 1-petabytes of earth science data / year
 - ★ telescopes scanning the skies
 - ✓ Sky survey data
 - ✓ LSST (Large Synoptic Survey Telescope) project: 20 PB science data & 100 PB image archive
 - ★ High-throughput biological data
 - ★ scientific simulations
 - ✓ terabytes of data generated in a few hours
- Data mining helps scientists
 - ★ in automated analysis of massive datasets
 - ★ in hypothesis formation



Mining Scientific Data - Fields

- Past decade has seen a huge growth of interest in mining data in a variety of scientific domains

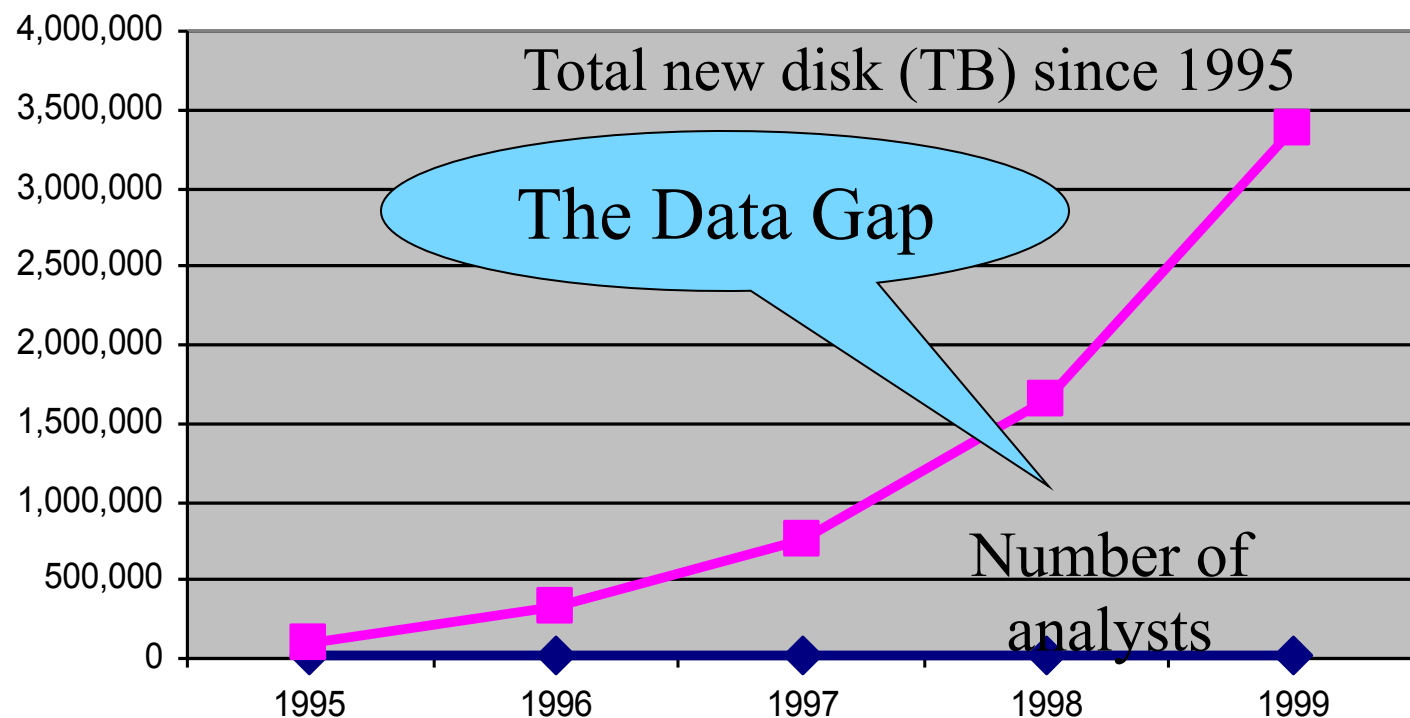
- Astroinformatics
- Neuroinformatics
- Quantum Informatics
- Health Informatics
- Evolutionary Informatics
- Veterinary Informatics
- Organizational Informatics
- Pharmacy Informatics
- Social Informatics
- Ecoinformatics
- Geoinformatics
- Chemo Informatics
- National Security/Defense
- Medicine
- Manufacturing
- And more

Some Data Mining Examples

- Amazon.com, Google, Netflix
 - ★ Personal Recommendations
 - ★ Profile-based advertisements
- Spam Filters/Priority Inbox
 - ★ Keep those efforts to pay us millions of dollars at bay
- Scientific Discovery
 - ★ Grouping patterns in sky
 - ★ Prediction of weather and natural disasters
 - ★ Prediction of solar radiation (solar power as an energy source)
- Security
 - ★ Phone Conversations, Network Traffic

Mining Large Data Sets - Motivation

- There is often information “hidden” in the data that is not readily evident
- Human analysts may take weeks to discover useful information
- Much of the data is never analyzed at all

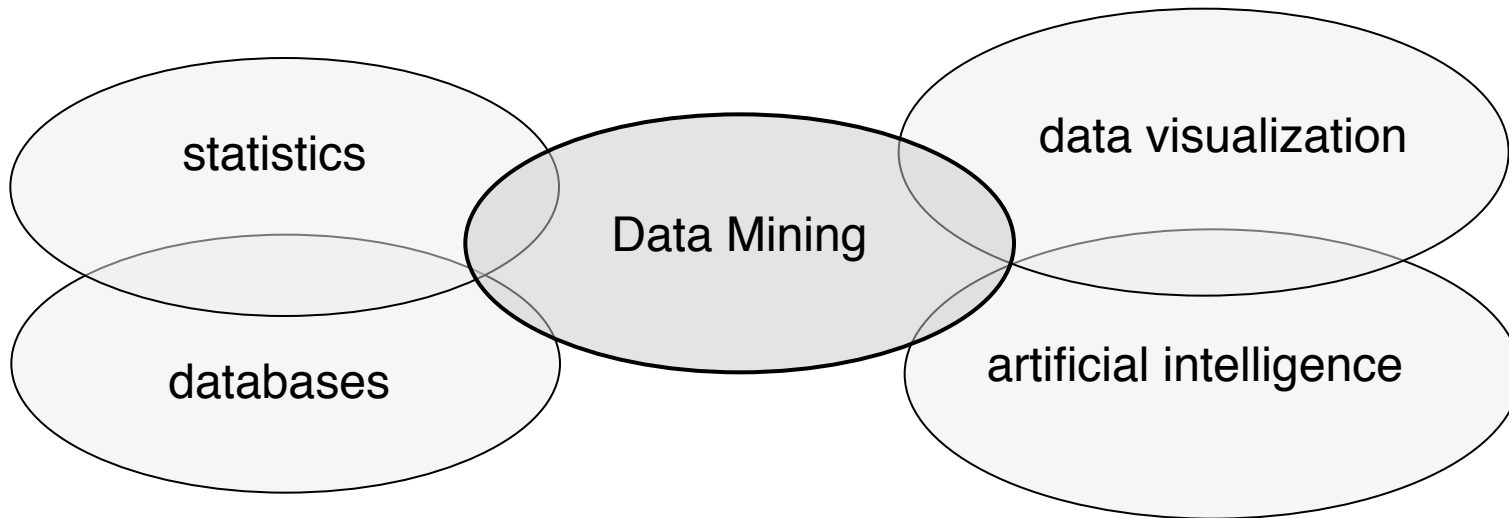


From: R. Grossman, C. Kamath, V. Kumar, “Data Mining for Scientific and Engineering Applications”

What is Data Mining?

Data Mining has been defined as

“The nontrivial extraction of implicit, previously unknown, and potentially useful information from data”.



Informally, data mining is the extraction of interesting knowledge (rules, regularities, patterns, constraints) from data in large databases.

Data Mining

- Broadly speaking, data mining is the process of semi-automatically analyzing large databases to find useful patterns
- Like knowledge discovery in artificial intelligence data mining discovers statistical rules and patterns
- Differs from machine learning in that it deals with large volumes of data stored primarily on disk.
- Some types of knowledge discovered from a database can be represented by a set of rules.
 - e.g.,: “Young women with annual incomes greater than \$50,000 are most likely to buy sports cars”
- Other types of knowledge represented by equations, or by prediction functions, or by clusters
- Some manual intervention is usually required
 - Pre-processing of data, choice of which type of pattern to find, postprocessing to find novel patterns

Data Mining Tasks

■ Prediction Methods

- ★ Use some variables to predict unknown or future values of other variables.

■ Description Methods

- ★ Find human-interpretable patterns that describe the data. Make good inferences from the data.

Data Mining Tasks...

- Classification [Predictive]
- Clustering [Descriptive]
- Association Rule Discovery [Descriptive]
- Sequential Pattern Discovery [Descriptive]
- Regression [Predictive]
- Deviation Detection [Predictive]

Applications of Data Mining

■ **Prediction** based on past history

- Predict if a credit card applicant poses a good credit risk, based on some attributes (income, job type, age, ..) and past history
- Predict if a customer is likely to switch brand loyalty
- Predict if a customer is likely to respond to “junk mail”
- Predict if a pattern of phone calling card usage is likely to be fraudulent

■ Some examples of prediction mechanisms:

- **Classification**
 - Given a training set consisting of items belonging to different classes, and a new item whose class is unknown, predict which class it belongs to
- **Regression** formulae
 - given a set of parameter-value to function-result mappings for an unknown function, predict the function-result for a new parameter-value

Applications of Data Mining (Cont.)

■ Descriptive Patterns

- **Associations**

- Find books that are often bought by the same customers. If a new customer buys one such book, suggest that he buys the others too.
- Other similar applications: camera accessories, clothes, etc.
- Associations may also be used as a first step in detecting **causation**
 - E.g. association between exposure to chemical X and cancer, or new medicine and cardiac problems

- **Clusters**

- Detection of clusters remains important in detecting epidemics
- E.g. typhoid cases were clustered in an area surrounding a contaminated well

Classification Example

<i>categorical</i>	<i>categorical</i>	<i>continuous</i>	<i>class</i>	
<i>Tid</i>	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Refund	Marital Status	Taxable Income	Cheat
No	Single	75K	?
Yes	Married	50K	?
No	Married	150K	?
Yes	Divorced	90K	?
No	Single	40K	?
No	Married	80K	?



Classification: Definition

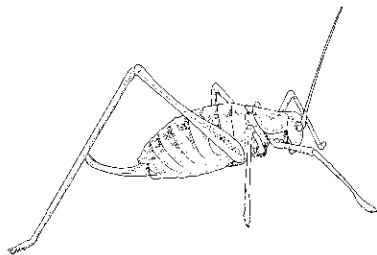
- Given a collection of records (*training set*)
 - ★ Each record contains a set of *attributes*, one of the attributes is the *class*.
- Find a *model* for class attribute as a function of the values of other attributes.
- Goal: previously unseen records should be assigned a class as accurately as possible.
 - ★ A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

Classification: Examples

- Document, music, video categorization
- Predict if a pattern of phone calling card usage is likely to be fraudulent
- Predict if a patient is healthy or has some medical condition
- Determine if an image contains nudity
- Intrusion detection, fraud detection

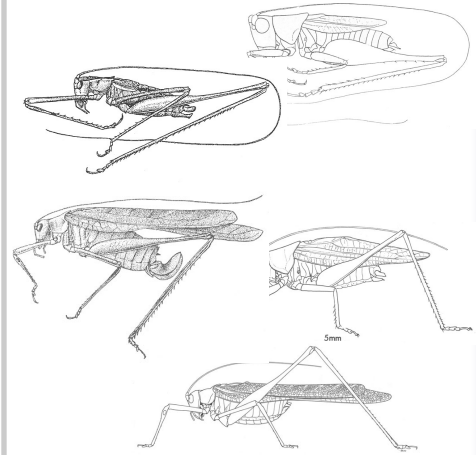
The Classification Problem (informal definition)

Given a collection of annotated data. In this case 5 instances **Katydids** of and five of **Grasshoppers**, decide what type of insect the unlabeled example is.

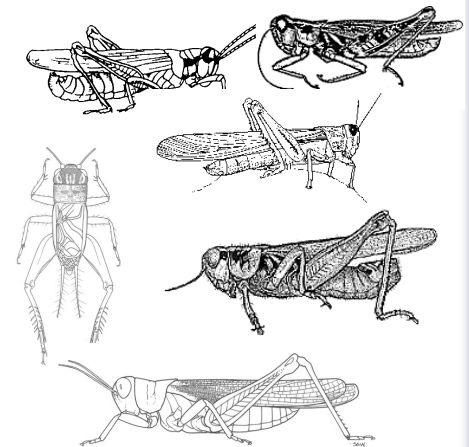


Katydid or **Grasshopper**?

Katydids



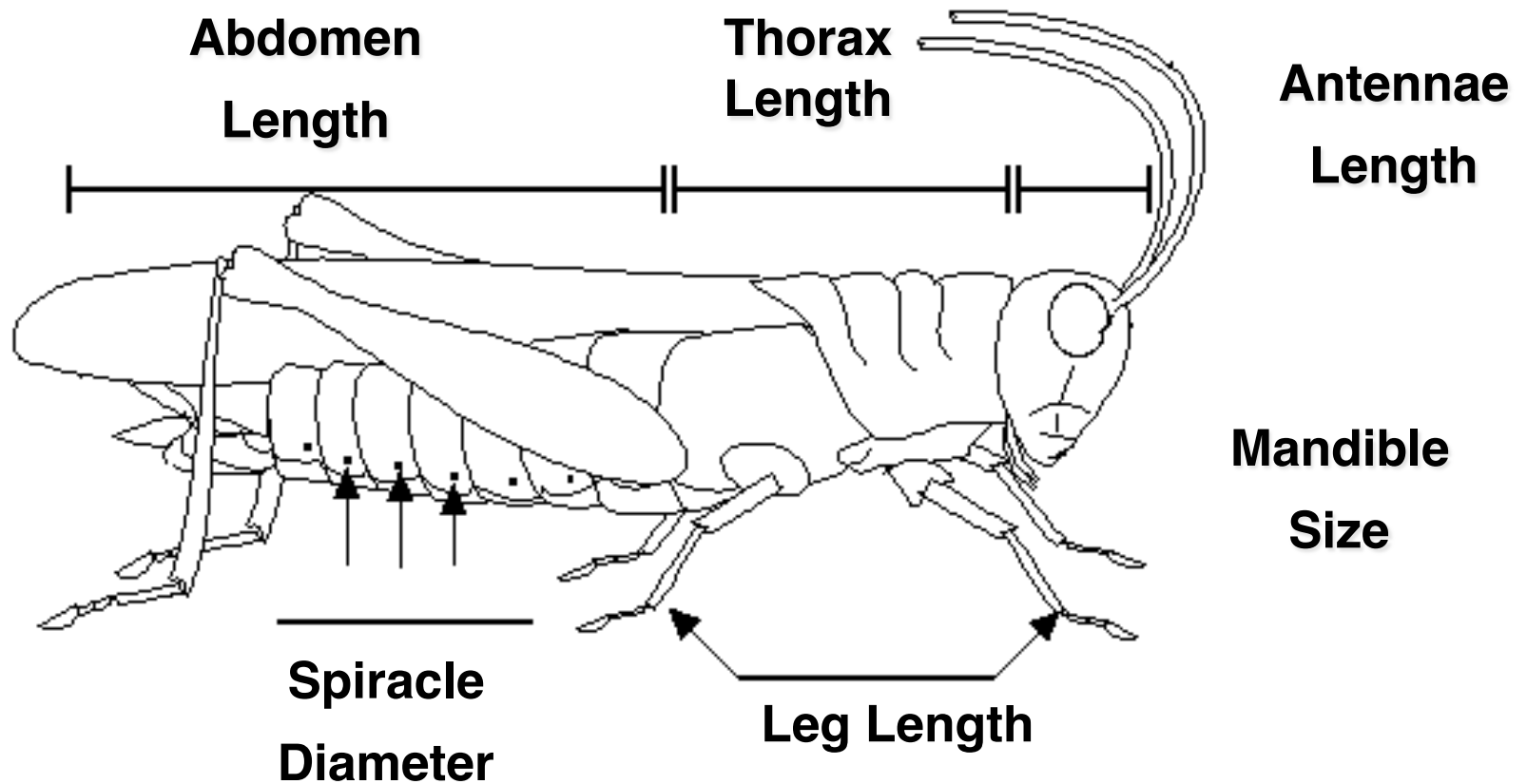
Grasshoppers



For any domain of interest, we can measure *features*

Color {Green, Brown, Gray, Other}

Has Wings?



We can store features in a database.

The classification problem can now be expressed as:

Given a training database (**My_Collection**), predict the **class** label of a previously unseen instance

My_Collection

Insect ID	Abdomen Length	Antennae Length	Insect Class
1	2.7	5.5	Grasshopper
2	8.0	9.1	Katydid
3	0.9	4.7	Grasshopper
4	1.1	3.1	Grasshopper
5	5.4	8.5	Katydid
6	2.9	1.9	Grasshopper
7	6.1	6.6	Katydid
8	0.5	1.0	Grasshopper
9	8.3	6.6	Katydid
10	8.1	4.7	Katydid

previously unseen instance =

11

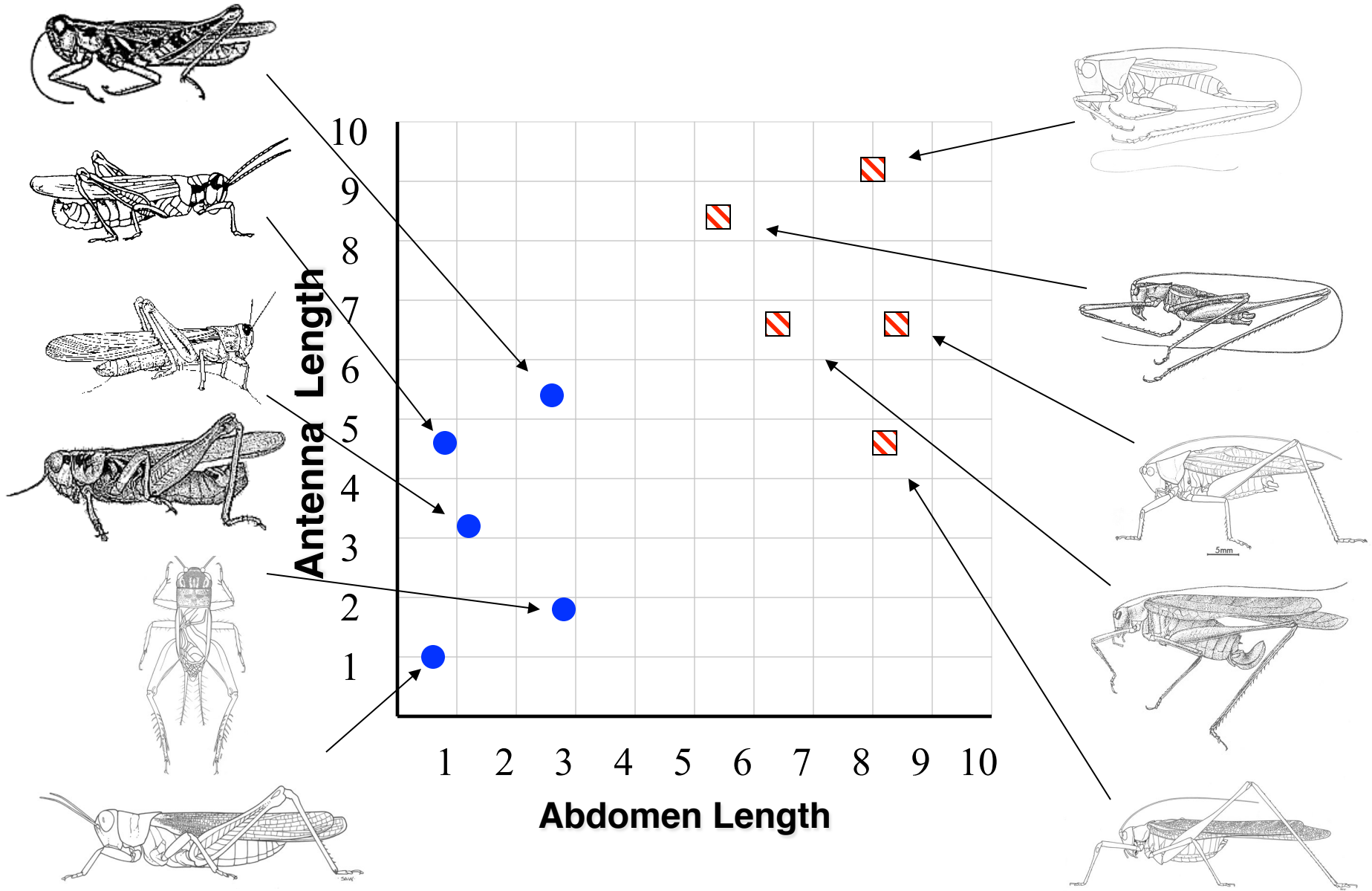
5.1

7.0

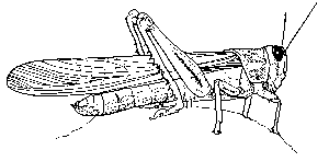
??????

Grasshoppers

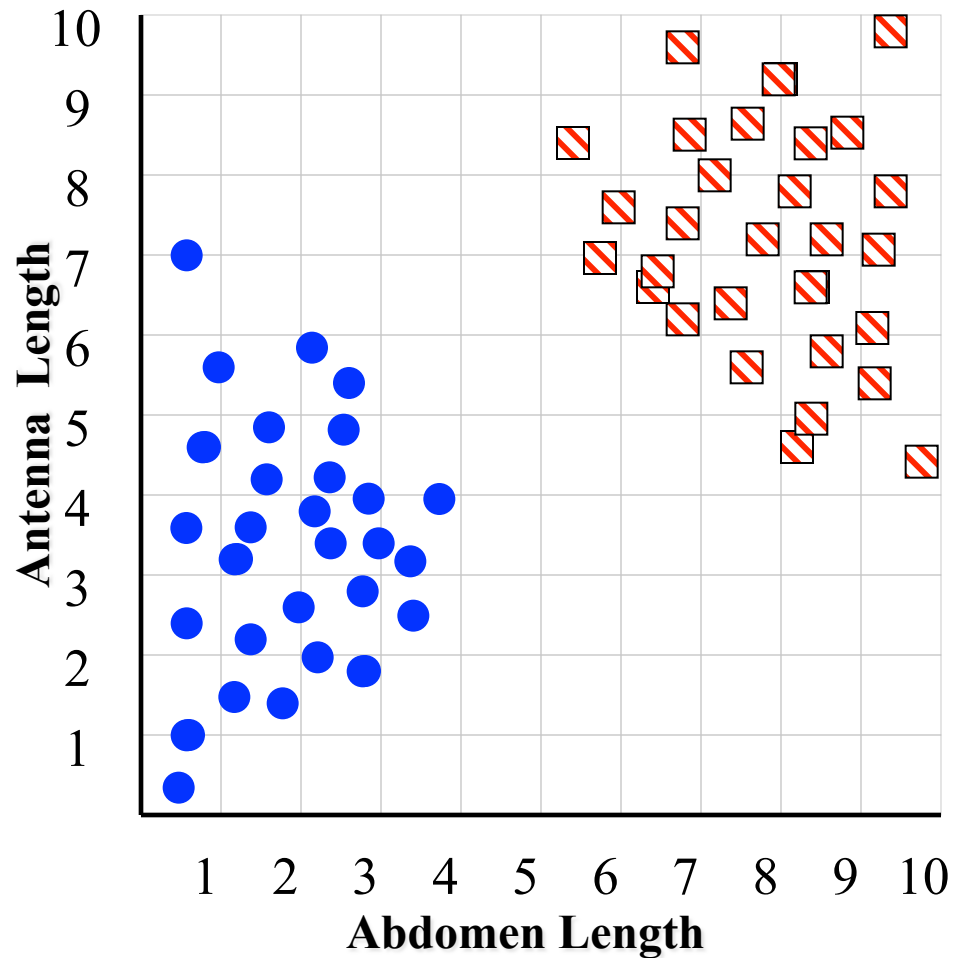
Katydid



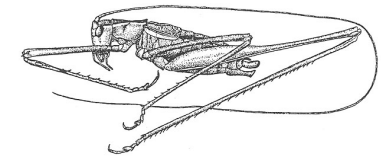
Grasshoppers



We will also use this larger dataset as a motivating example...

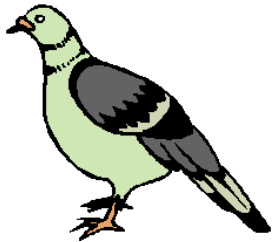


Katydidids



Each of these data objects are called...

- exemplars
- (training) examples
- instances
- tuples



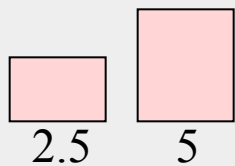
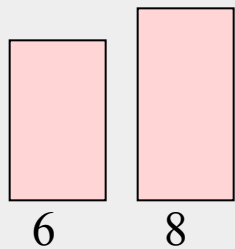
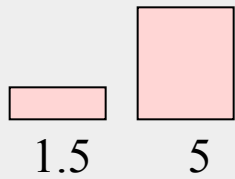
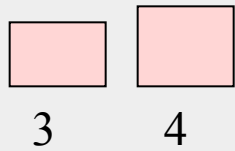
We will return to the previous slide in two minutes. In the meantime, we are going to play a quick game.

I am going to show you some classification problems which were shown to pigeons!

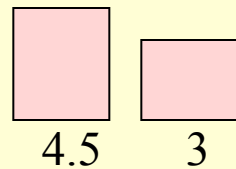
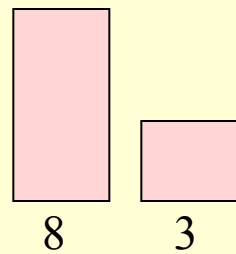
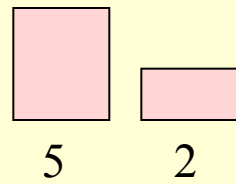
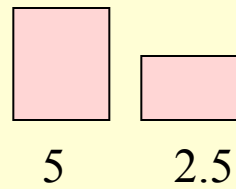
Let us see if you are as smart as a pigeon!

Pigeon Problem 1

Examples of class A

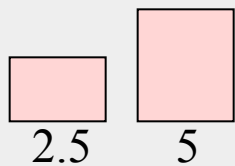
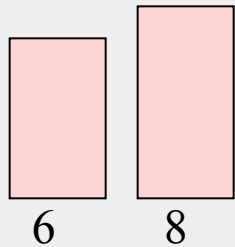
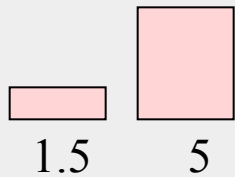
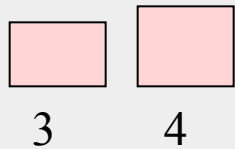


Examples of class B

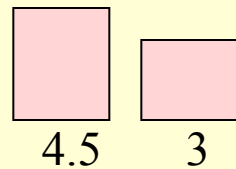
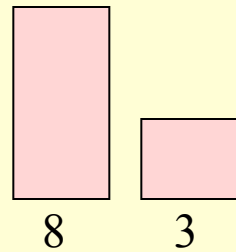
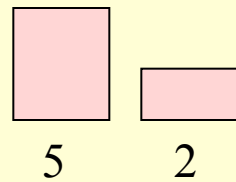
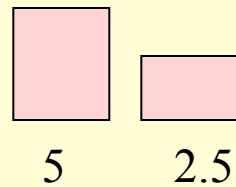


Pigeon Problem 1

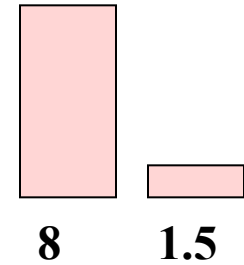
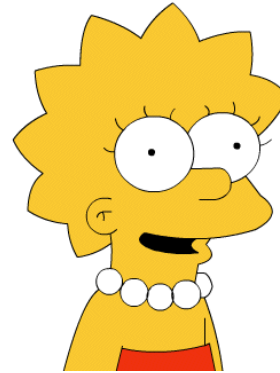
Examples of class A



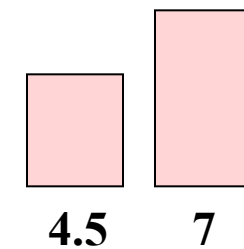
Examples of class B



What class is this object?

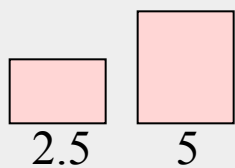
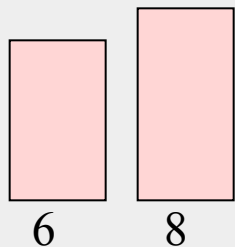
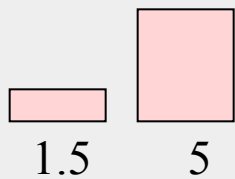
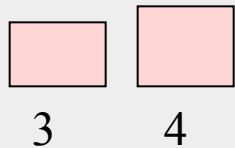


What about this one, **A** or **B**?

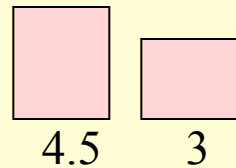
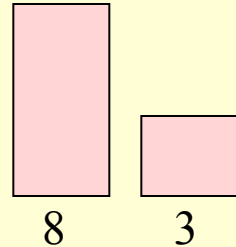
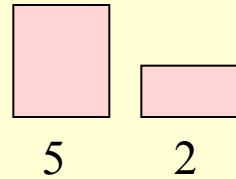
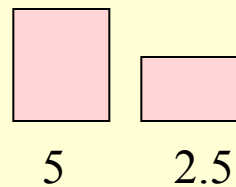


Pigeon Problem 1

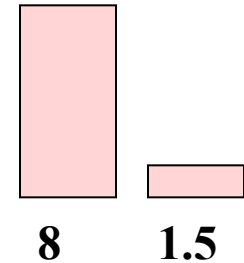
Examples of class A



Examples of class B



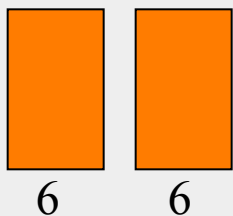
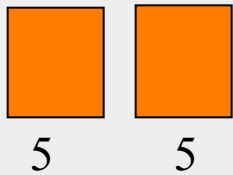
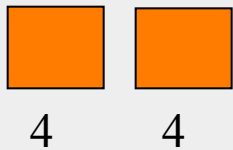
This is a **B**!



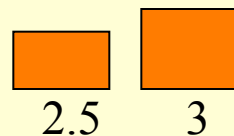
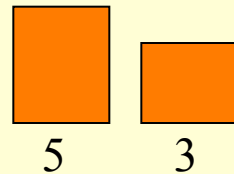
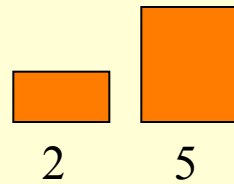
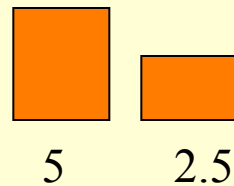
Here is the rule.
If the left bar is smaller than the right bar, it is an **A**, otherwise it is a **B**.

Pigeon Problem 2

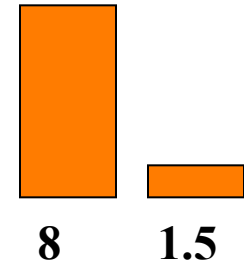
Examples of class A



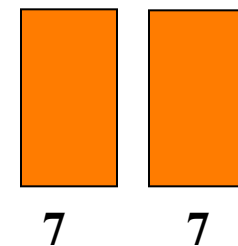
Examples of class B



Oh! This ones hard!

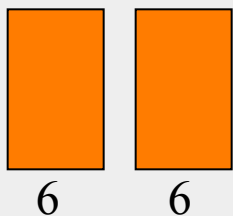
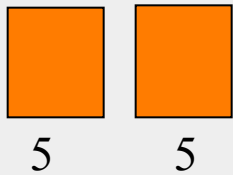


Even I know this one

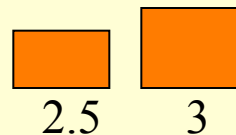
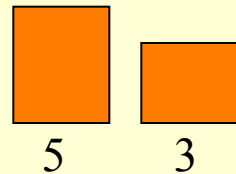
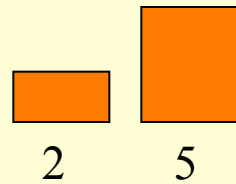
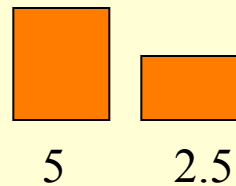


Pigeon Problem 2

Examples of class A

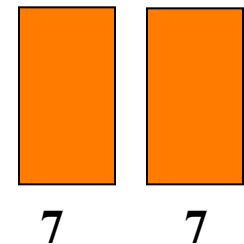


Examples of class B



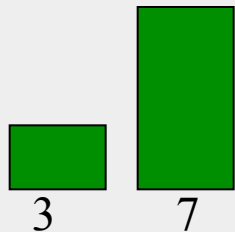
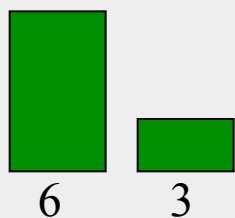
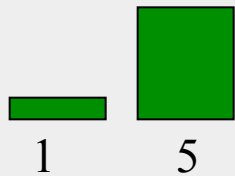
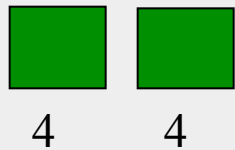
The rule is as follows,
if the two bars are
equal sizes, it is an **A**.
Otherwise it is a **B**.

So this one is an **A**.

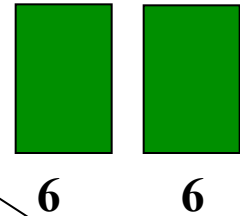
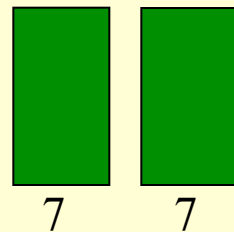
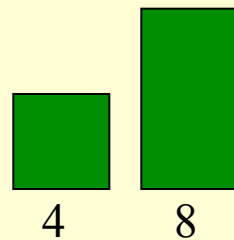
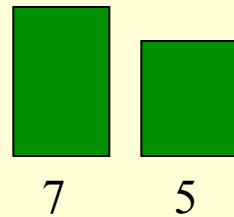
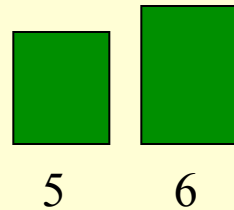


Pigeon Problem 3

Examples of class A



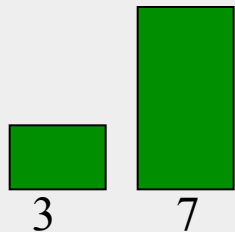
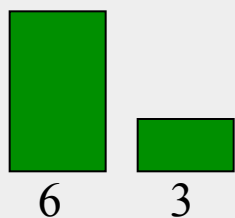
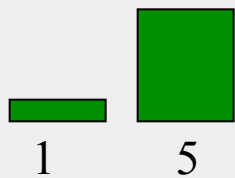
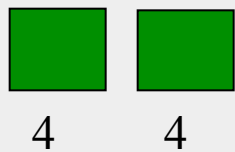
Examples of class B



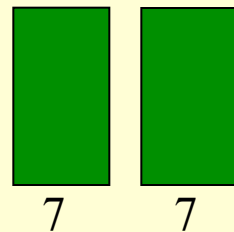
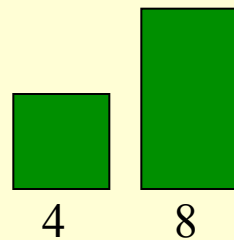
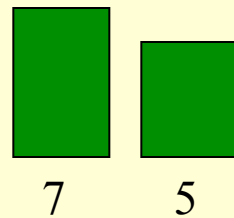
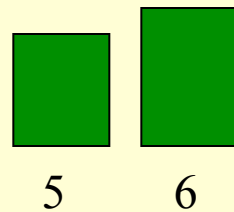
This one is really hard!
What is this, **A** or **B**?

Pigeon Problem 3

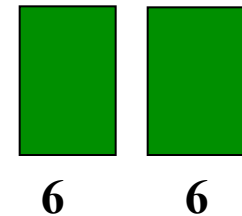
Examples of class A



Examples of class B



It is a **B**!

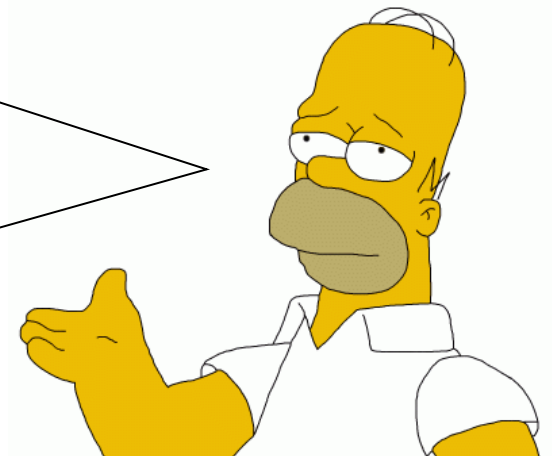


The rule is as follows,
if the square of the
sum of the two bars is
less than or equal to
100, it is an **A**.
Otherwise it is a **B**.



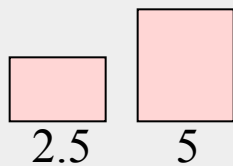
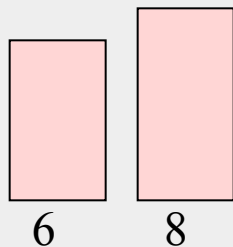
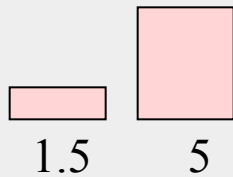
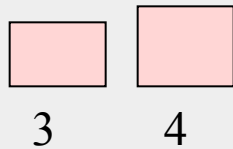
Why did we spend so much time with this game?

Because we wanted to show that almost all classification problems have a geometric interpretation, check out the next 3 slides...

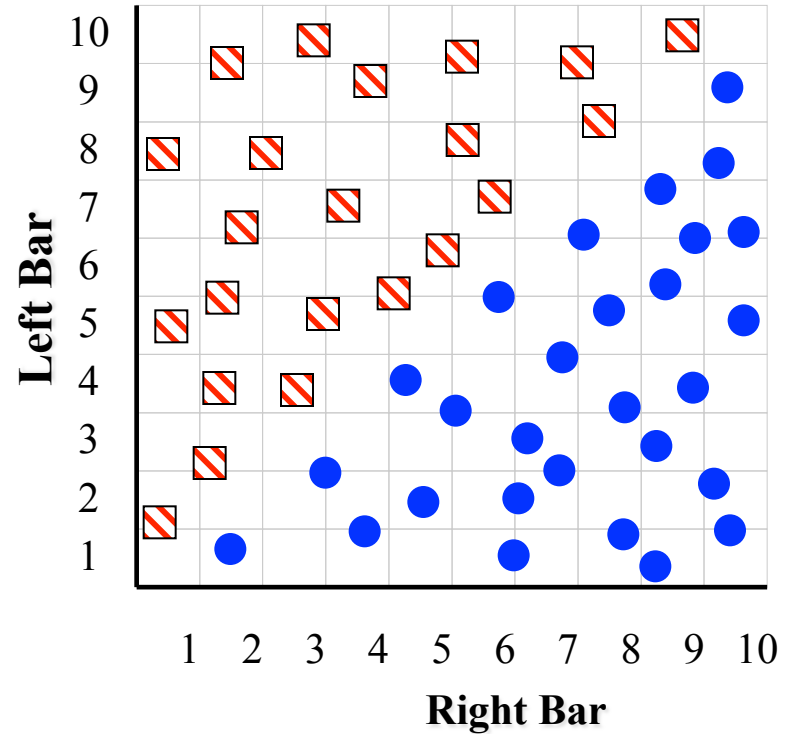
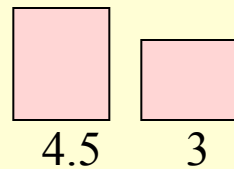
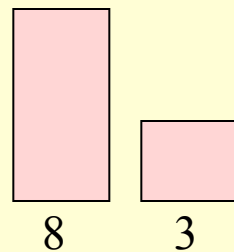
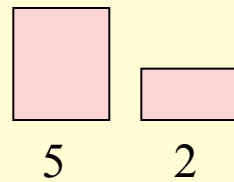
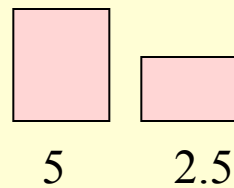


Pigeon Problem 1

Examples of class A



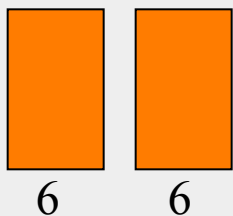
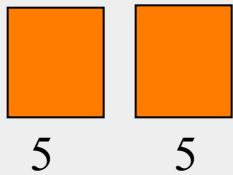
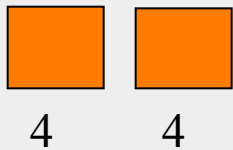
Examples of class B



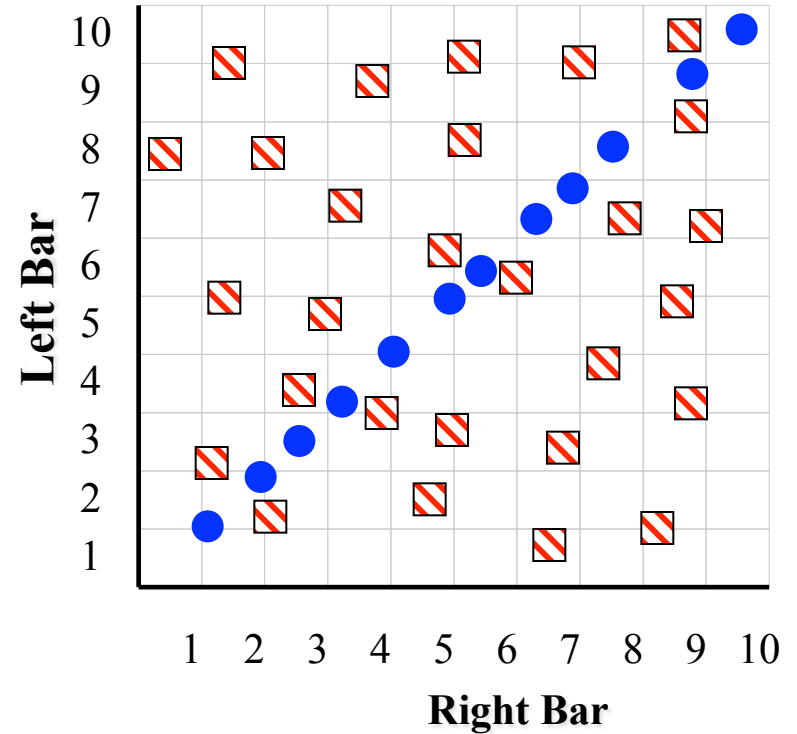
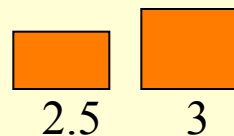
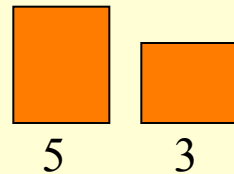
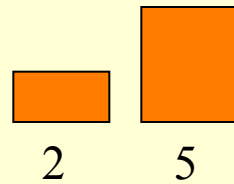
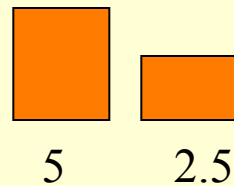
Here is the rule again.
If the left bar is smaller
than the right bar, it is
an **A**, otherwise it is a **B**.

Pigeon Problem 2

Examples of class A



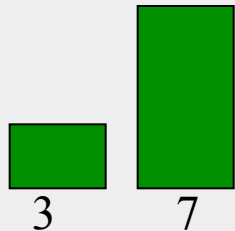
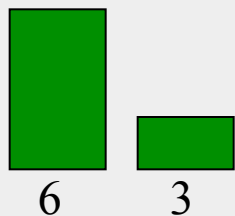
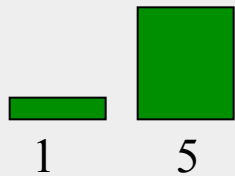
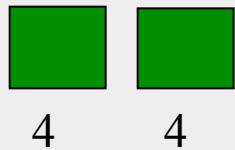
Examples of class B



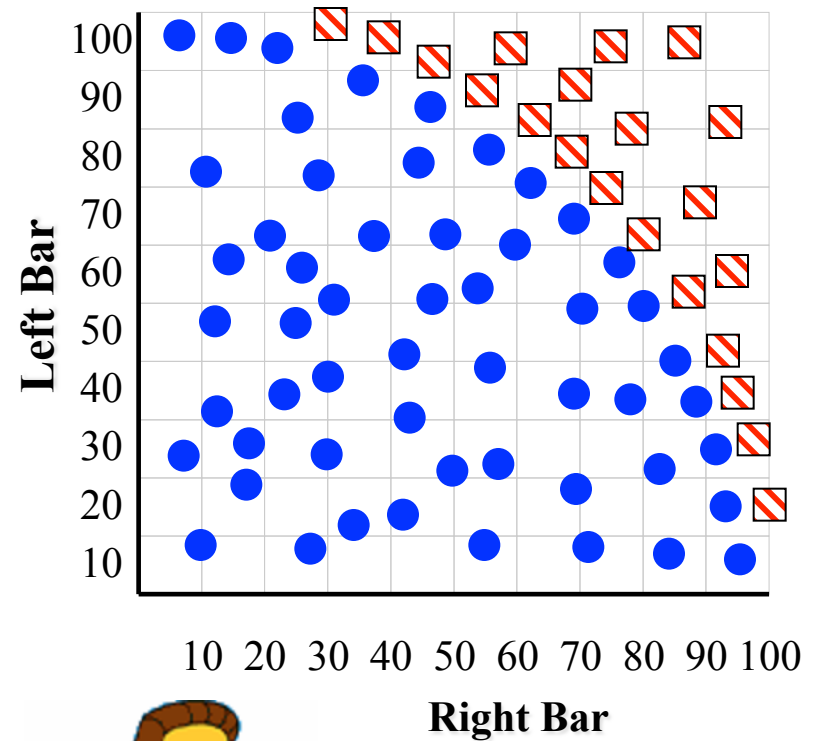
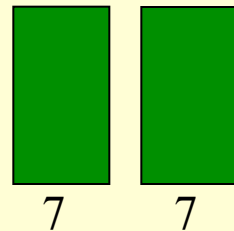
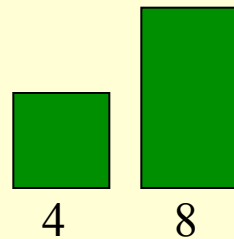
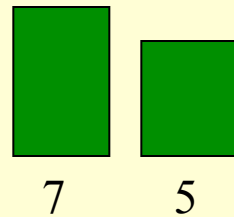
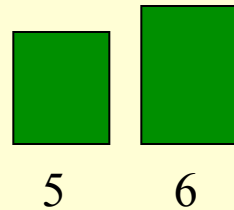
Let me look it up... here it is..
the rule is, if the two bars
are equal sizes, it is an **A**.
Otherwise it is a **B**.

Pigeon Problem 3

Examples of class A



Examples of class B

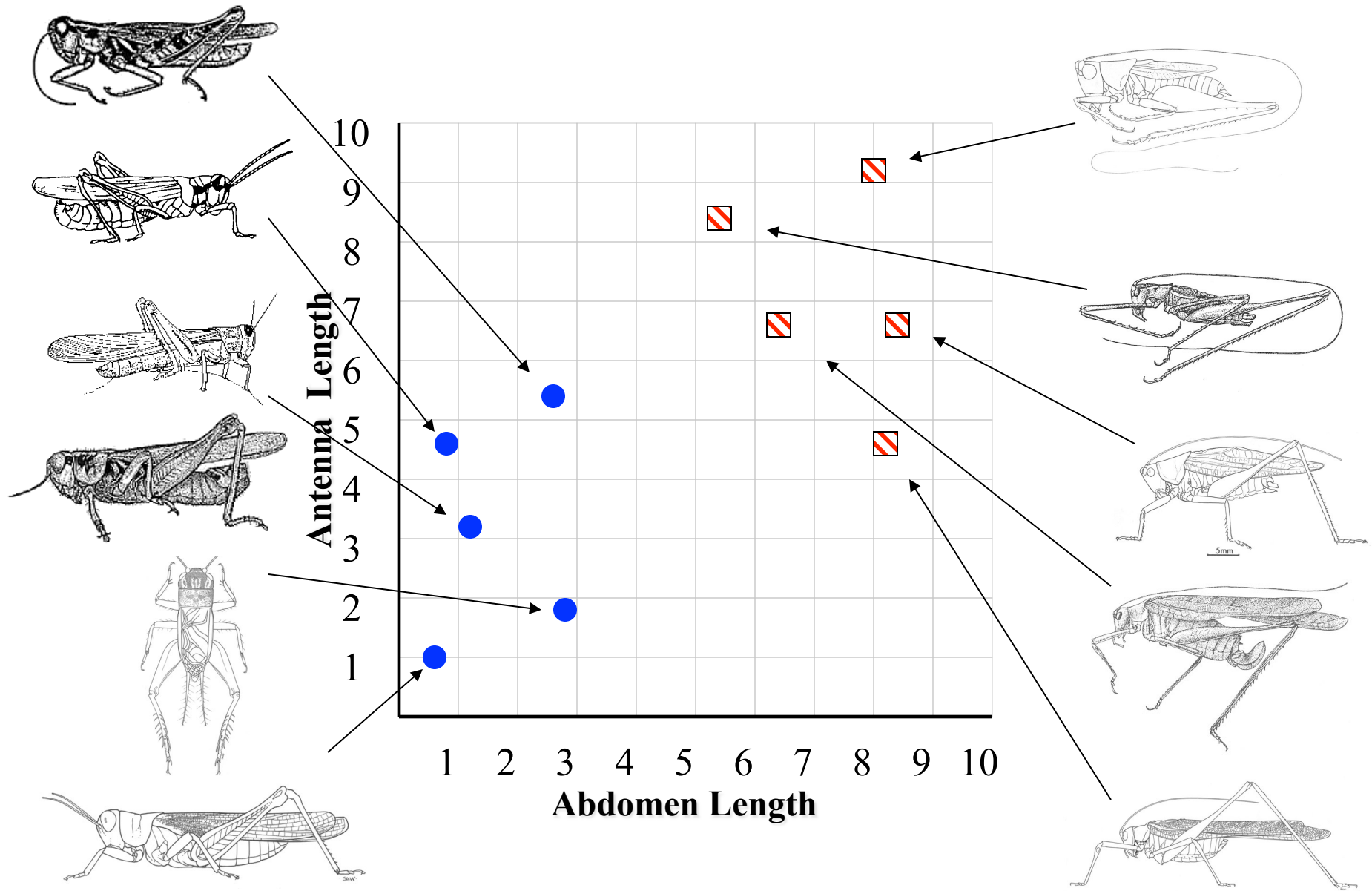


The rule again:

if the square of the sum of the two bars is less than or equal to 100, it is an **A**. Otherwise it is a **B**.

Grasshoppers

Katydid



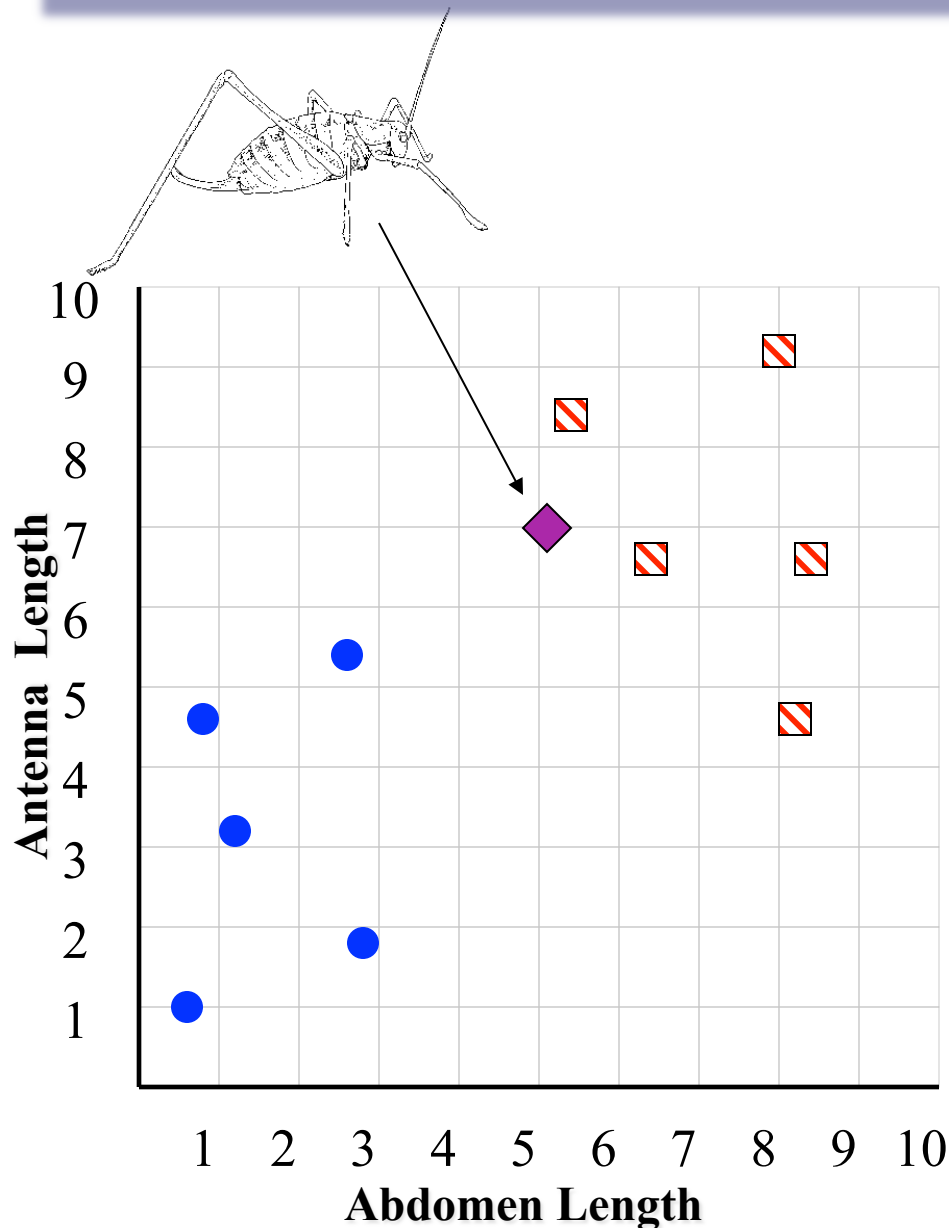
previously unseen instance =

11

5.1

7.0

??????



We can “project” the previously unseen instance into the same space as the database.

We have now abstracted away the details of our particular problem. It will be much easier to talk about points in space.

▣ **Katydid**

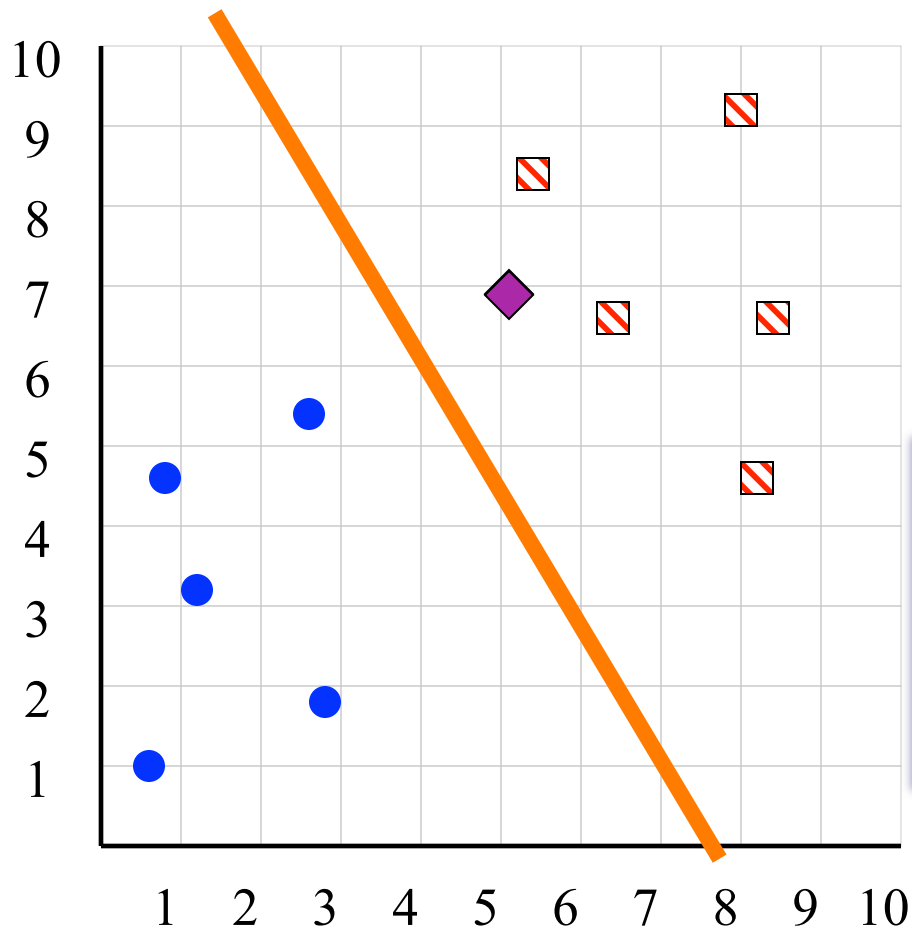
● **Grasshoppers**

Simple Linear Classifier



R.A. Fisher

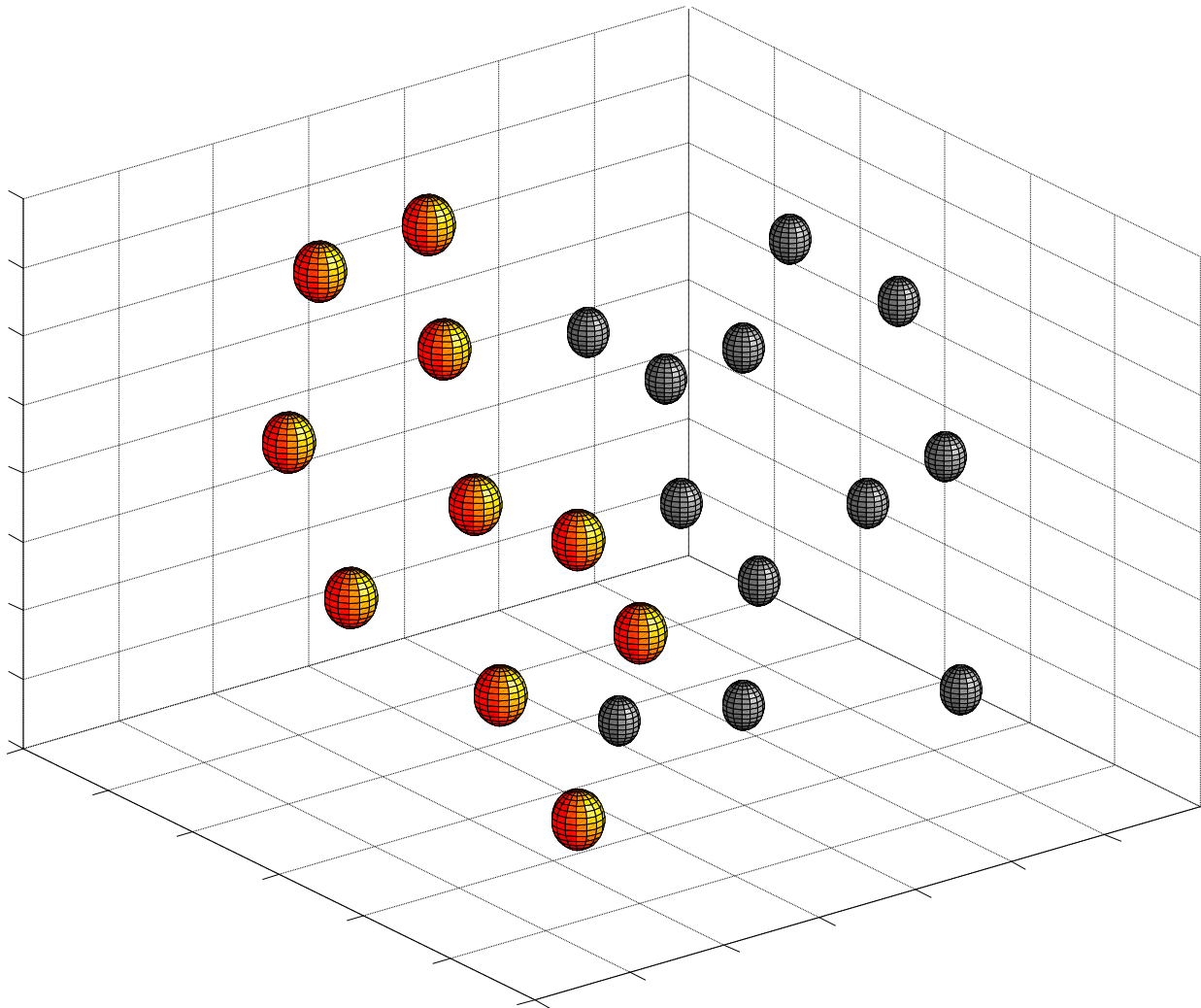
1890-1962



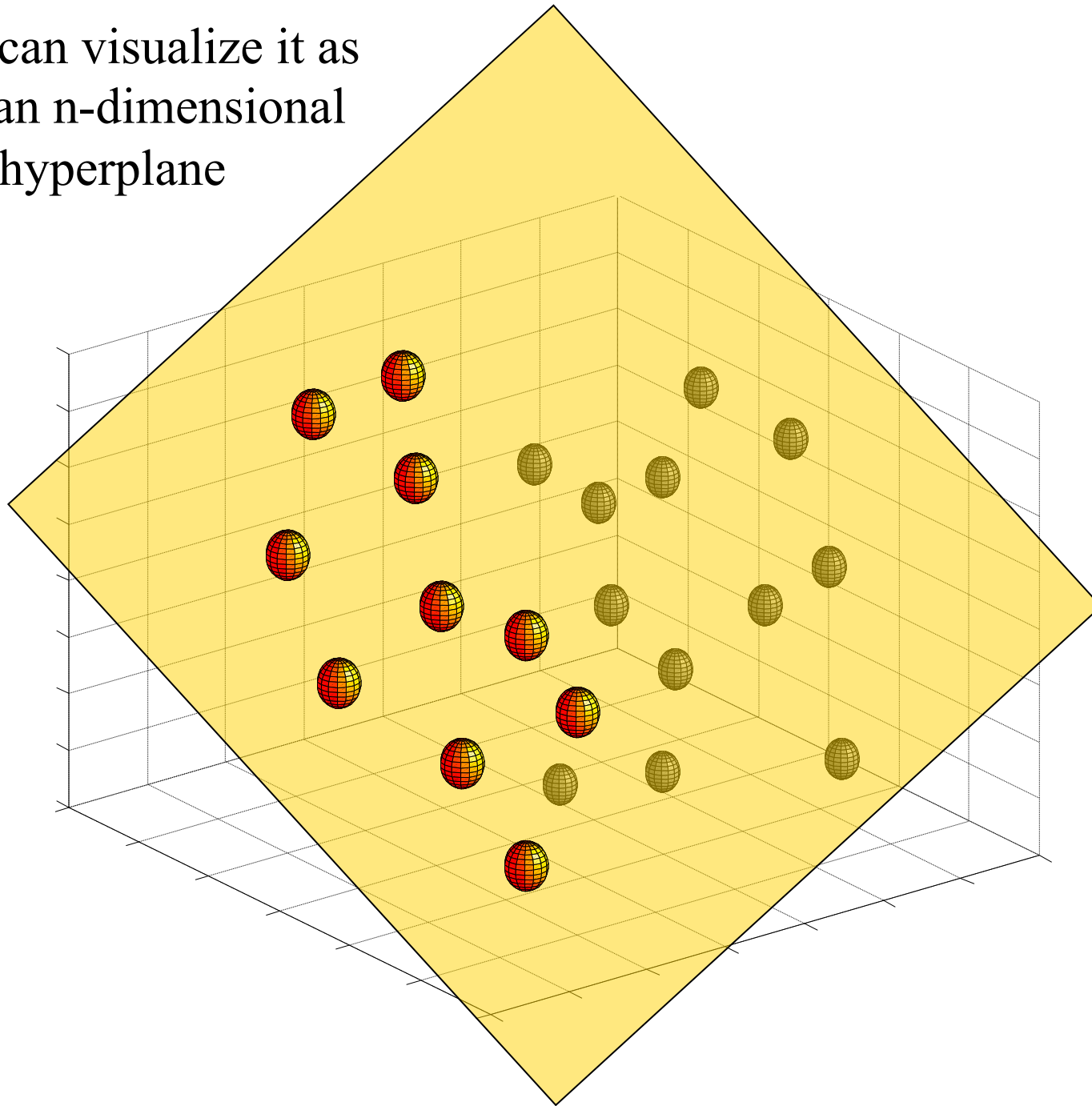
If previously unseen instance above the line
then
class is **Katydid**
else
class is **Grasshopper**

 **Katydids**
 **Grasshoppers**

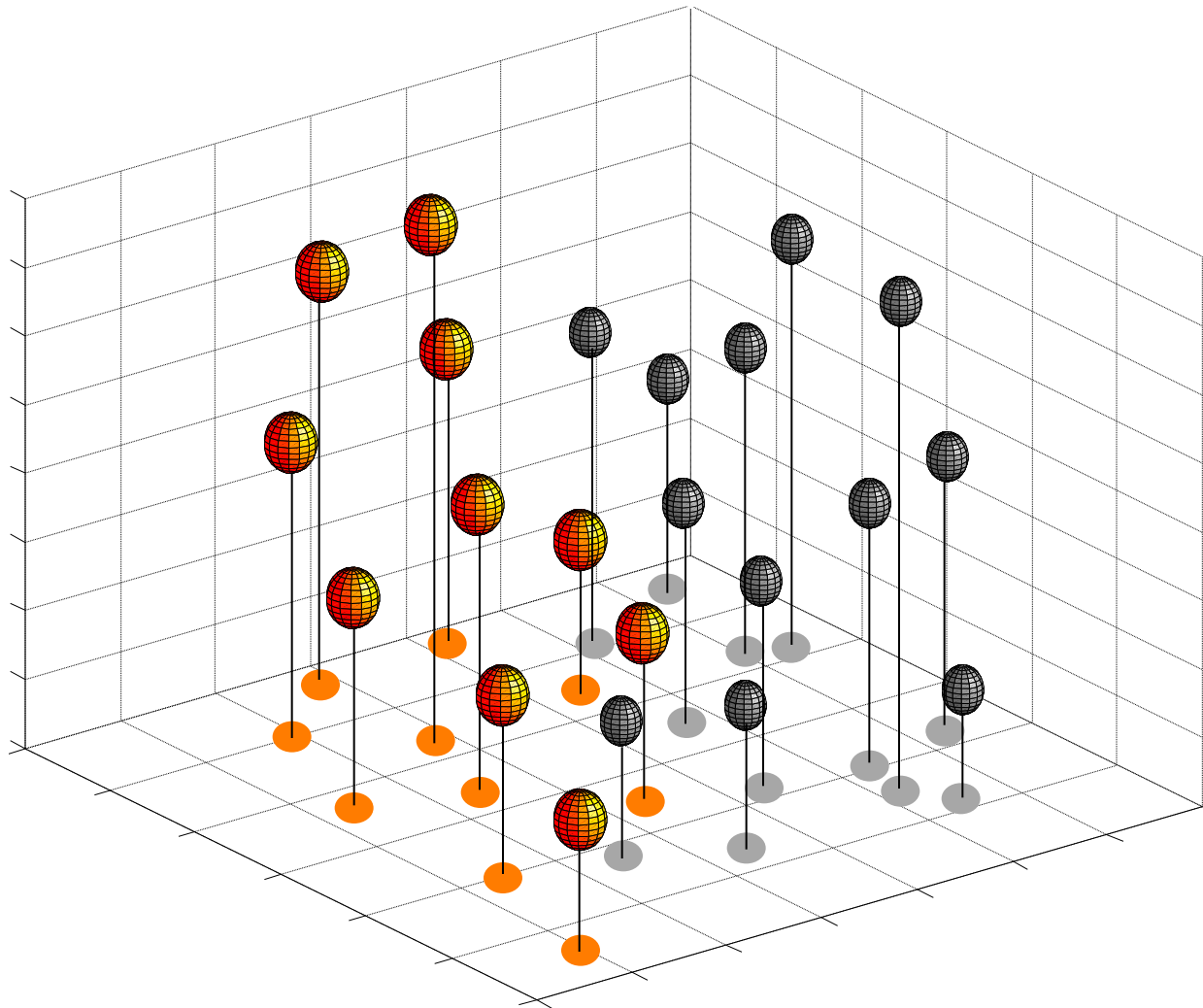
The simple linear classifier is defined for higher dimensional spaces...



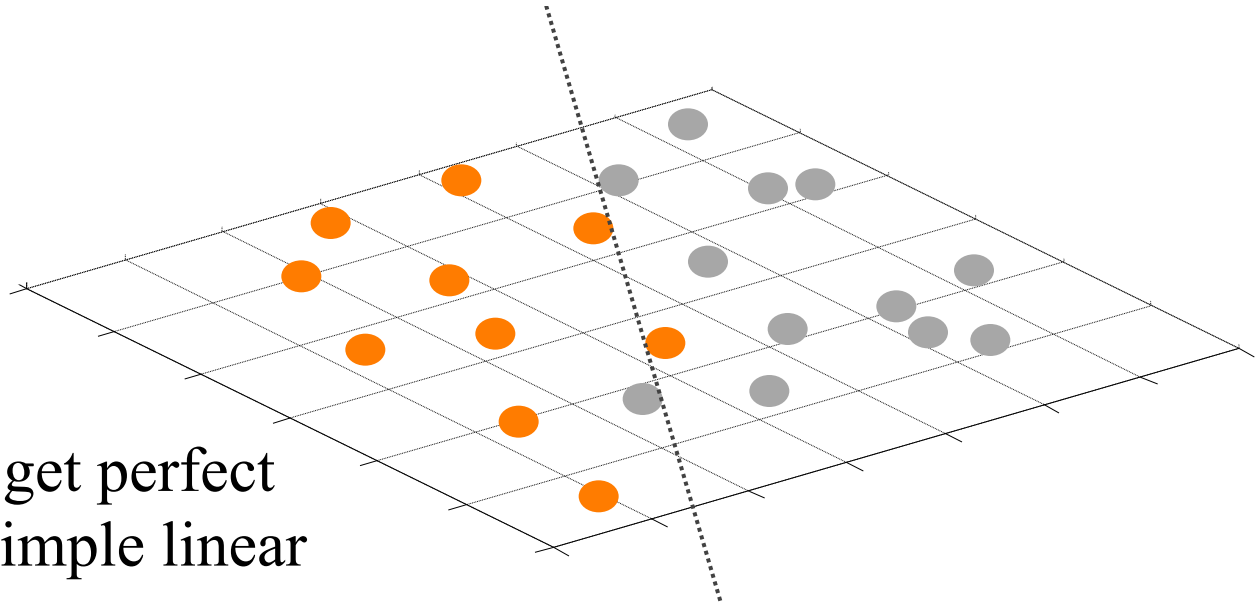
... we can visualize it as
being an n-dimensional
hyperplane



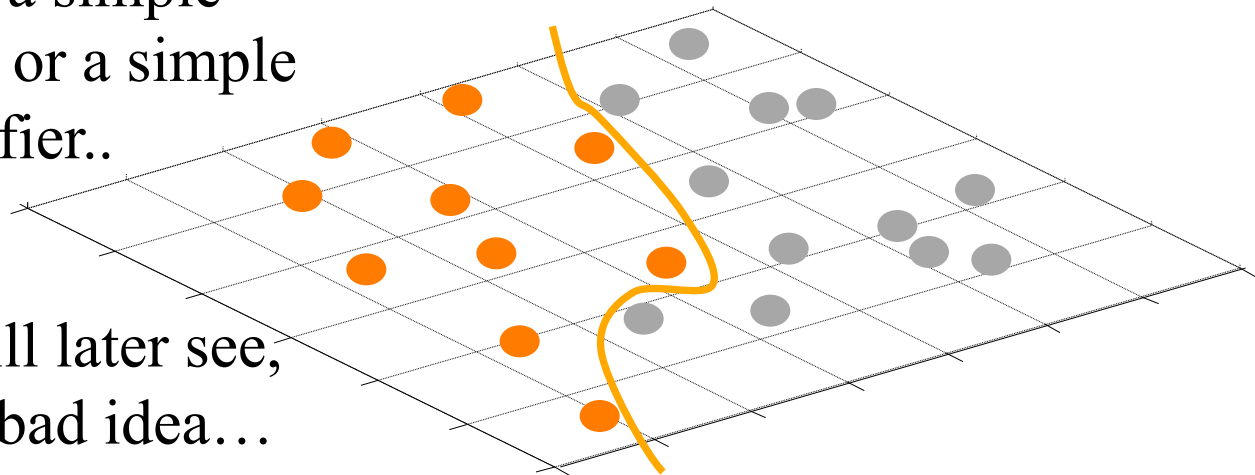
It is interesting to think about what would happen in this example if we did not have the 3rd dimension...



We can no longer get perfect accuracy with the simple linear classifier...



We could try to solve this problem by using a simple *quadratic* classifier or a simple *cubic* classifier..



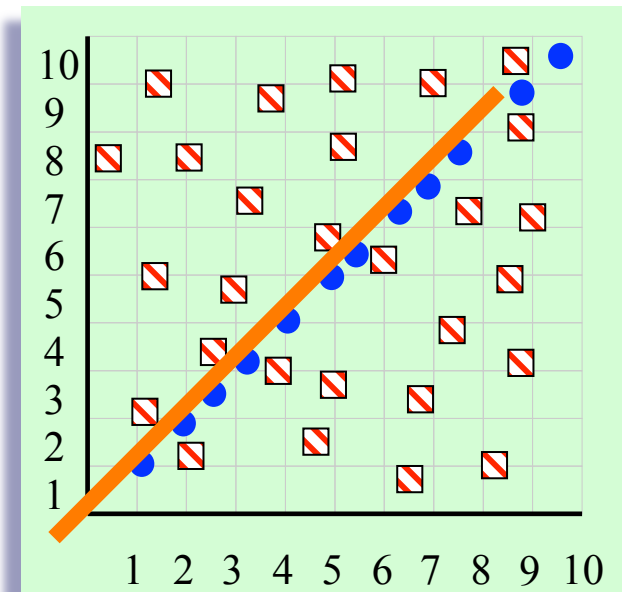
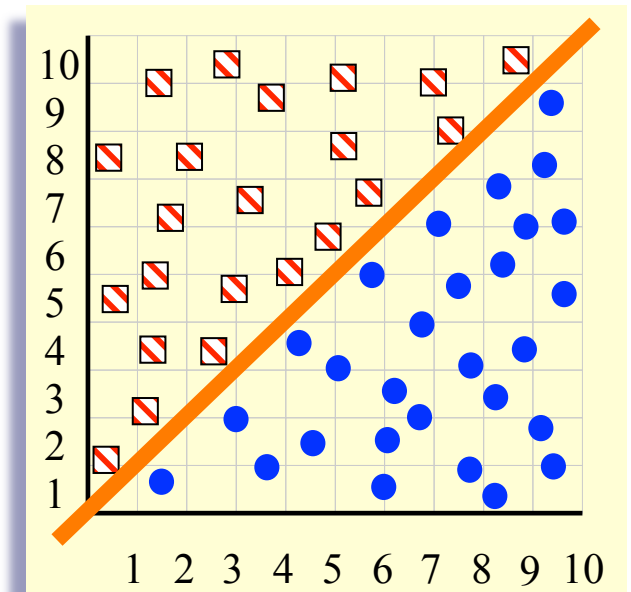
However, as we will later see, this is probably a bad idea...

Which of the “Pigeon Problems” can be solved by the Simple Linear Classifier?

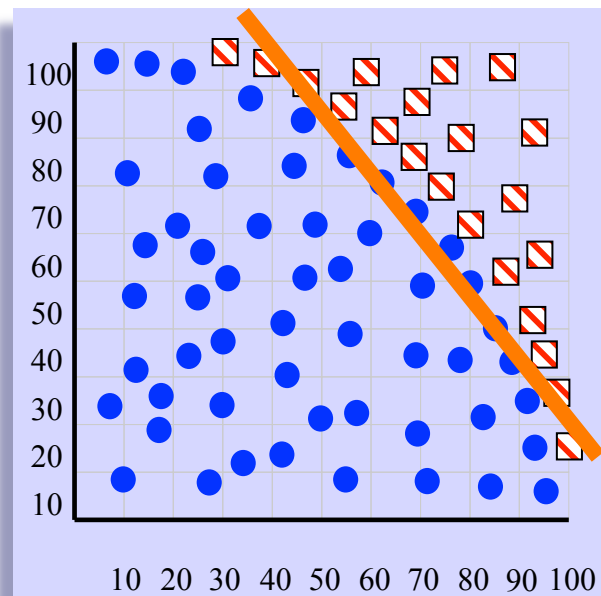
Perfect

Useless

Pretty Good



Problems that can be solved by a linear classifier are called **linearly separable**.



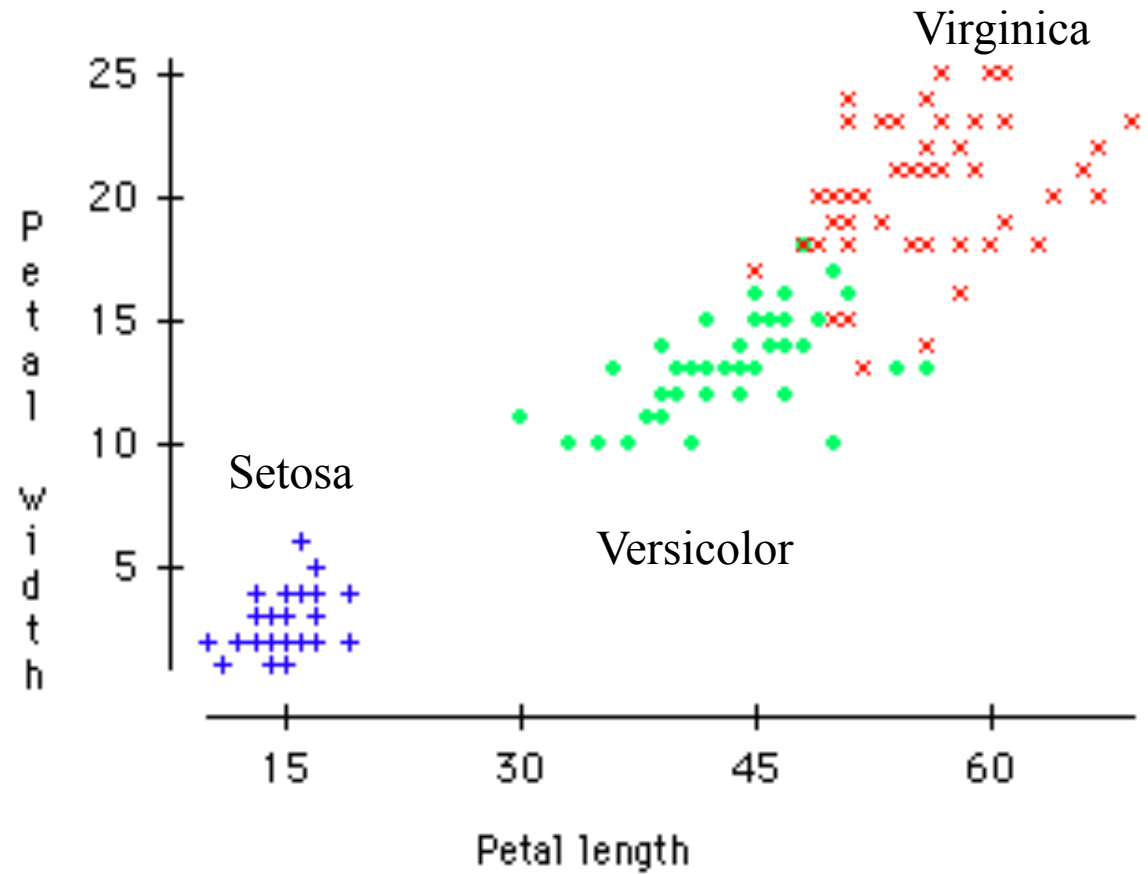
A Famous Problem

R. A. Fisher's Iris Dataset.

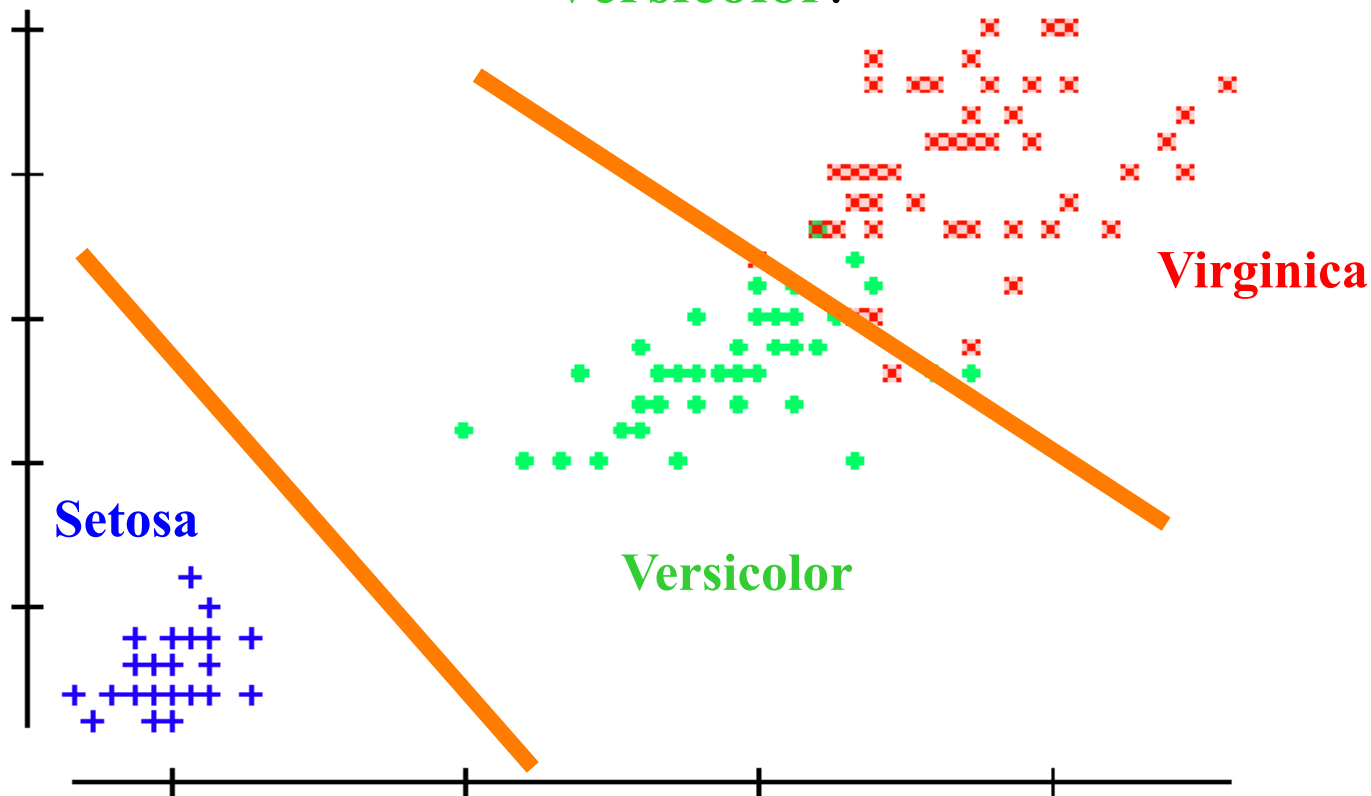
3 classes

50 of each class

The task is to classify Iris plants into one of 3 varieties using the Petal Length and Petal Width.



We can generalize the piecewise linear classifier to N classes, by fitting N-1 lines. In this case we first learned the line to (perfectly) discriminate between **Setosa** and **Virginica/Versicolor**, then we learned to approximately discriminate between **Virginica** and **Versicolor**.



If petal width $> 3.272 - (0.325 * \text{petal length})$ **then** class = **Virginica**
Elseif petal width...

We have now seen one classification algorithm, and we are about to see more.
How should we compare them?

- Predictive accuracy
- Speed and scalability
 - ★ time to construct the model
 - ★ time to use the model
 - ★ efficiency in disk-resident databases
- Robustness
 - ★ handling noise, missing values and irrelevant features, streaming data
- Interpretability:
 - ★ understanding and insight provided by the model

Predictive Accuracy I

How do we *estimate* the **accuracy** of our classifier?

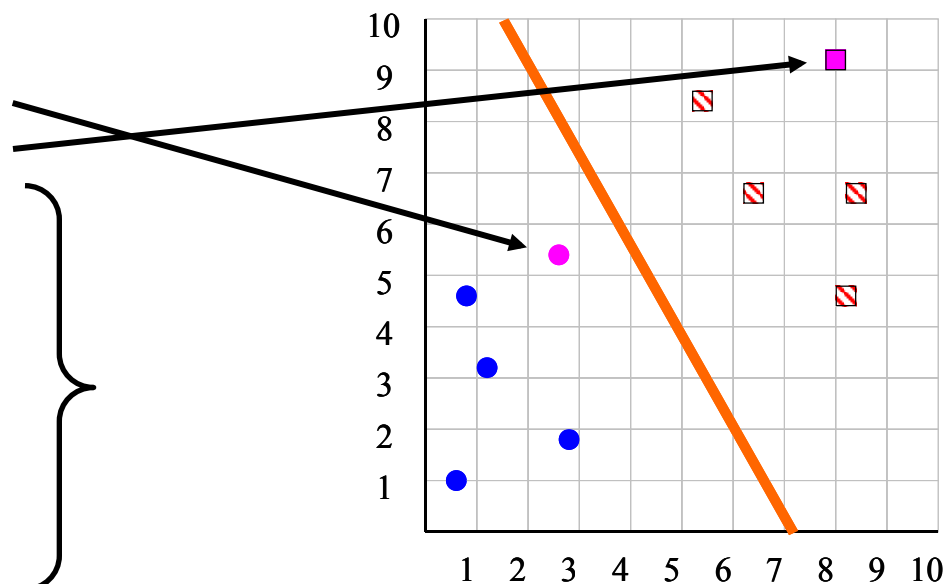
We can use ***K*-fold cross validation**

We divide the dataset into K equal sized sections. The algorithm is tested K times, each time leaving out one of the K section from building the classifier, but using it to *test* the classifier instead

$$\text{Accuracy} = \frac{\text{Number of correct classifications}}{\text{Number of instances in our database}}$$

$K = 5$

Insect ID	Abdomen Length	Antennae Length	Insect Class
1	2.7	5.5	Grasshopper
2	8.0	9.1	Katydid
3	0.9	4.7	Grasshopper
4	1.1	3.1	Grasshopper
5	5.4	8.5	Katydid
6	2.9	1.9	Grasshopper
7	6.1	6.6	Katydid
8	0.5	1.0	Grasshopper
9	8.3	6.6	Katydid
10	8.1	4.7	Katydid

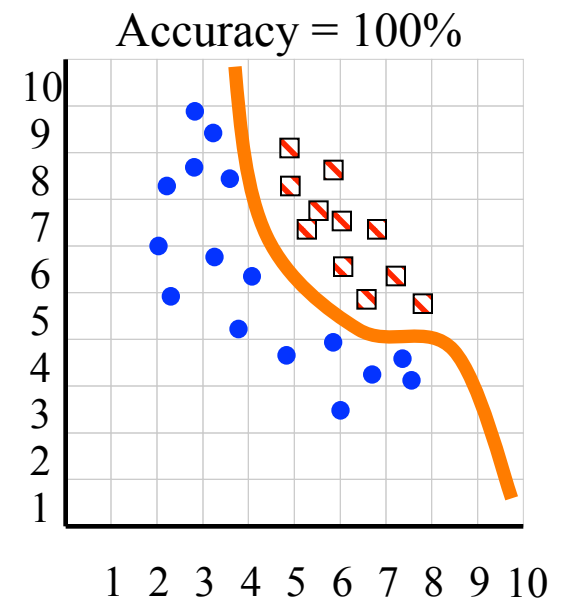
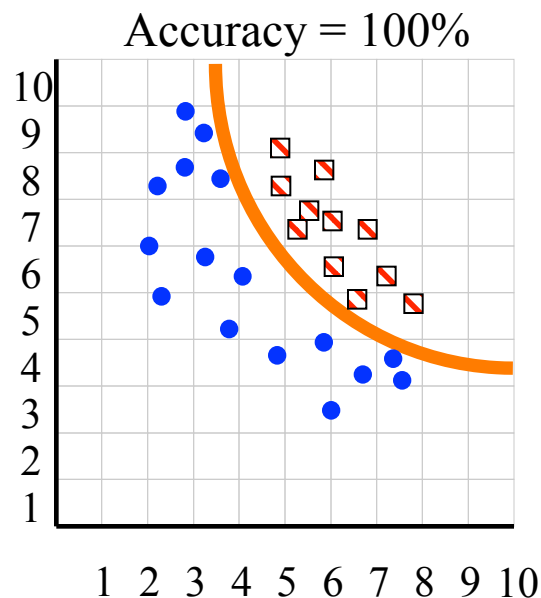
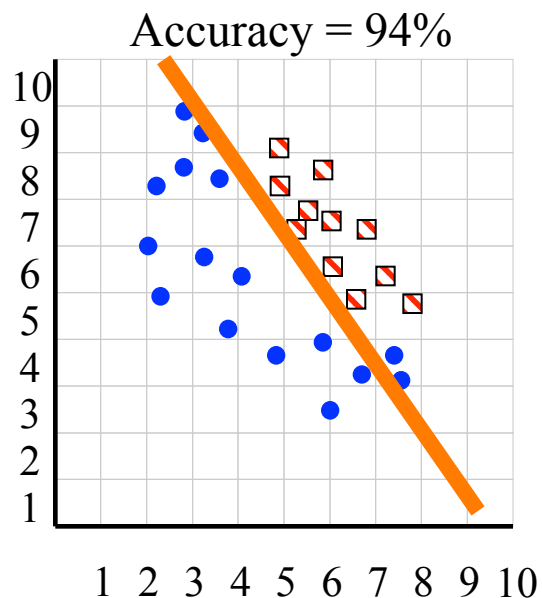


Predictive Accuracy II

Using K-fold cross validation is a good way to set any parameters we may need to adjust in (any) classifier.

We can do K-fold cross validation for each possible setting, and choose the model with the highest accuracy. Where there is a tie, we choose the simpler model.

Actually, we should probably penalize the more complex models, even if they are more accurate, since more complex models are more likely to overfit (discussed later).



Predictive Accuracy III

$$\text{Accuracy} = \frac{\text{Number of correct classifications}}{\text{Number of instances in our database}}$$

Accuracy is a single number, we may be better off looking at a **confusion matrix**. This gives us additional useful information...

True label is...

	Cat	Dog	Pig
Cat	100	0	0
Dog	9	90	1
Pig	45	45	10

Classified as a...

Metrics for Performance Evaluation

- Focus on the predictive capability of a model
 - ★ Rather than how fast it takes to classify or build models, scalability, etc.
- Confusion Matrix:

	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)
b: FN (false negative)
c: FP (false positive)
d: TN (true negative)

Metrics for Performance Evaluation...

ACTUAL CLASS	PREDICTED CLASS	
	Class=Yes	Class=No
Class=Yes	a (TP)	b (FN)
	c (FP)	d (TN)

- Most widely-used metric:

$$\text{Accuracy} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

Cost Matrix

	PREDICTED CLASS		
	$C(i j)$	Class=Yes	Class=No
	Class=Yes	$C(\text{Yes} \text{Yes})$	$C(\text{No} \text{Yes})$
	Class=No	$C(\text{Yes} \text{No})$	$C(\text{No} \text{No})$

$C(i|j)$: Cost of misclassifying class j example as class i

Cost-Sensitive Measures

$$\text{Precision (p)} = \frac{a}{a + c}$$

$$\text{Recall (r)} = \frac{a}{a + b}$$

$$\text{F - measure (F)} = \frac{2rp}{r + p} = \frac{2a}{2a + b + c}$$

- Precision is biased towards $C(\text{Yes}|\text{Yes})$ & $C(\text{Yes}|\text{No})$
- Recall is biased towards $C(\text{Yes}|\text{Yes})$ & $C(\text{No}|\text{Yes})$
- F-measure is biased towards all except $C(\text{No}|\text{No})$

$$\text{Weighted Accuracy} = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$$

Speed and Scalability I

We need to consider the time and space requirements for the two distinct phases of classification:

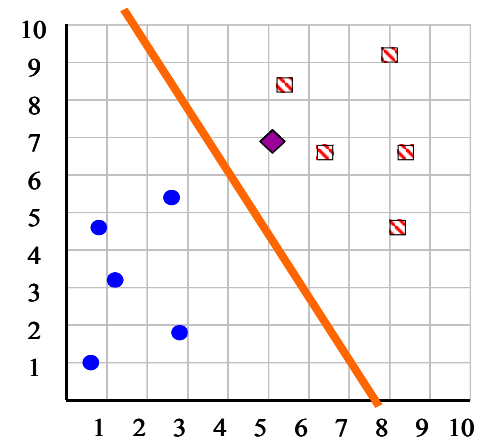
Time to **construct** the classifier

In the case of the simpler linear classifier, the time taken to fit the line, this is linear in the number of instances.

Time to **use** the model

In the case of the simpler linear classifier, the time taken to test which side of the line the unlabeled instance is. This can be done in constant time.

As we shall see, some classification algorithms are very efficient in one aspect, and very poor in the other.



Speed and Scalability II

For learning with small datasets, this
is the whole picture



However, for data mining with
massive datasets, it is not so much the
(main memory) time complexity that
matters, rather it is how many times
we have to scan the database.

This is because for most data mining operations, disk access times
completely dominate the CPU times.

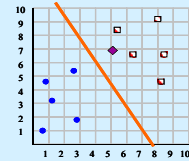
For data mining, researchers often report the number of times you
must scan the database.

Speed and Scalability I

We need to consider the time and space requirements
for the two distinct phases of classification:

- Time to **construct** the classifier
 - In the case of the simpler linear classifier, the time taken to fit the line, this is linear in the number of instances.
- Time to **use** the model
 - In the case of the simpler linear classifier, the time taken to test which side of the line the unlabeled instance is. This can be done in constant time.

As we shall see, some classification
algorithms are very efficient in one aspect,
and very poor in the other.



Robustness I

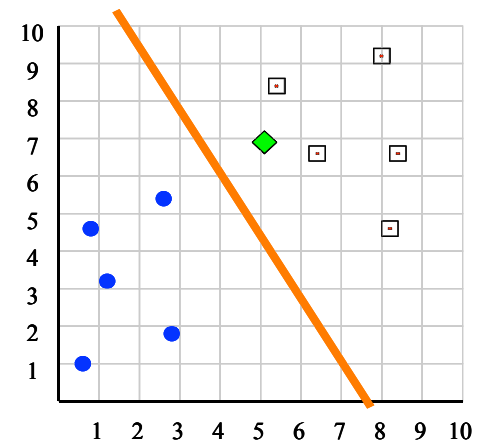
We need to consider what happens when we have:

- Noise

For example, a persons age could have been mistyped as 650 instead of 65, how does this effect our classifier? (This is important only for building the classifier, if the instance to be classified is noisy we can do nothing).

- Missing values

For example suppose we want to classify an insect, but we only know the abdomen length (X-axis), and not the antennae length (Y-axis), can we still classify the instance?



Robustness II

We need to consider what happens when we have:

- Irrelevant features

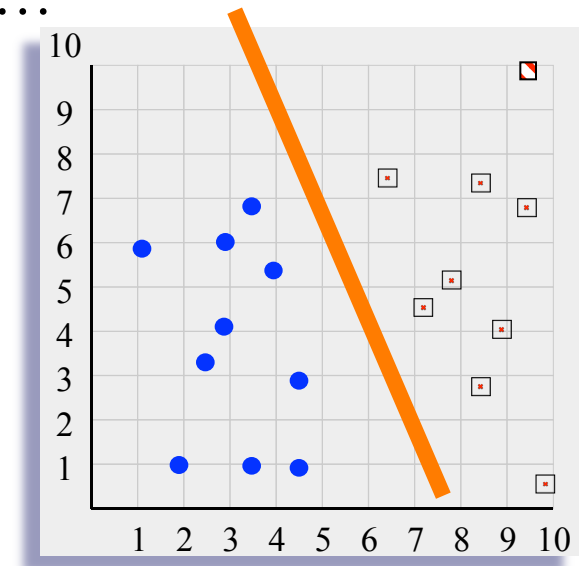
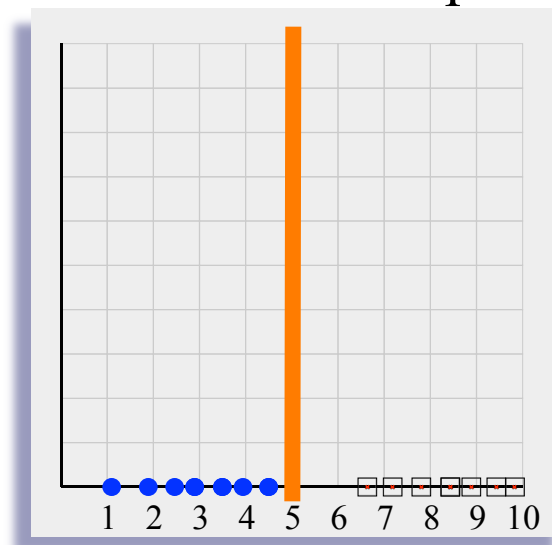
For example, suppose we want to classify people as either

Suitable_Grad_Student

Unsuitable_Grad_Student

And it happens that scoring more than 5 on a particular test is a perfect indicator for this problem...

If we also use
“hair_length” as a
feature, how will this
effect our classifier?



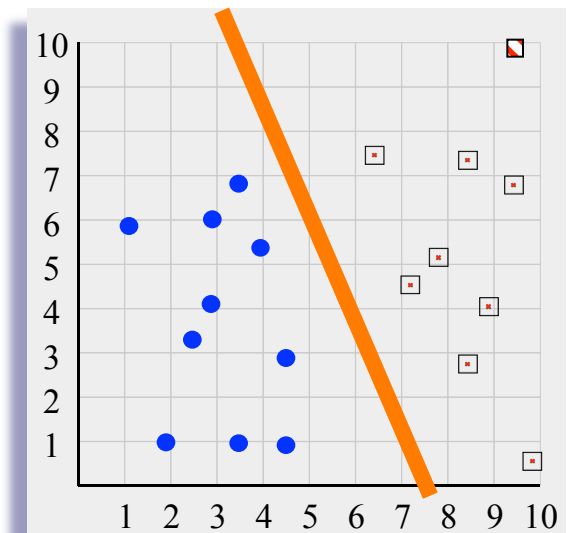
Robustness III

We need to consider what happens when we have:

- Streaming data

For many real world problems, we don't have a single fixed dataset. Instead, the data continuously arrives, potentially forever... (stock market, weather data, sensor data etc)

Can our classifier handle streaming data?

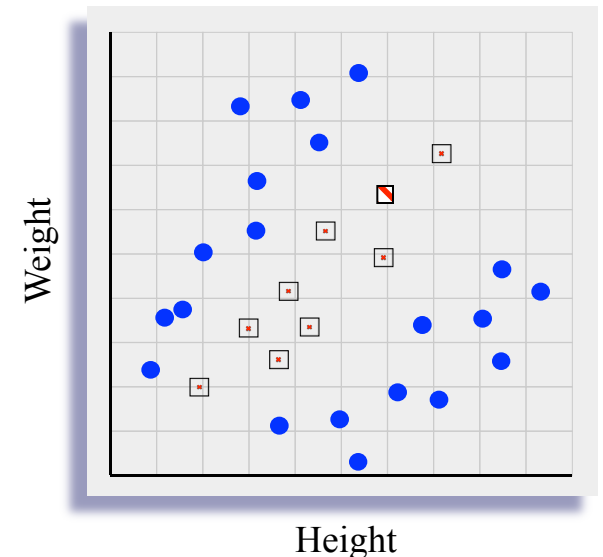


Interpretability

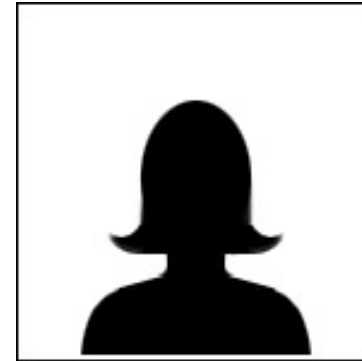
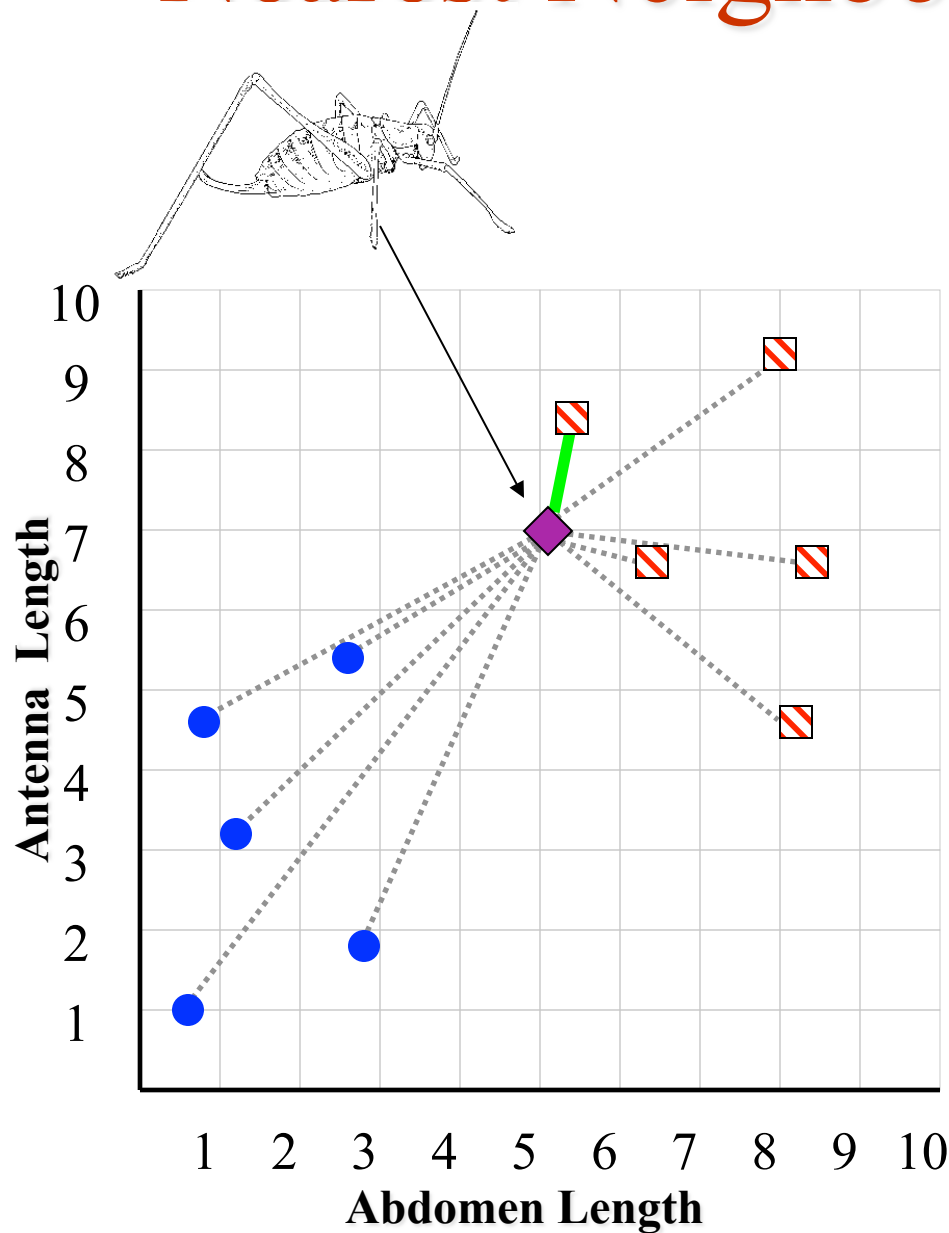
Some classifiers offer a *bonus* feature. The structure of the learned classifier tells use something about the domain.

As a trivial example, if we try to classify peoples health risks based on just their height and weight, we could gain the following insight (Based on the observation that a single linear classifier does not work well, but two linear classifiers do).

There are two ways to be unhealthy, being obese and being too skinny.

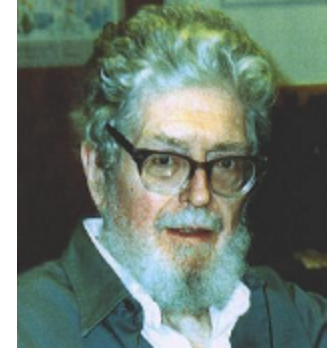


Nearest Neighbor Classifier



Evelyn Fix

1904-1965



Joe Hodges

1922-2000

If the **nearest** instance to the **previously unseen instance** is a **Katydid**

class is **Katydid**

else

class is **Grasshopper**



Katydids



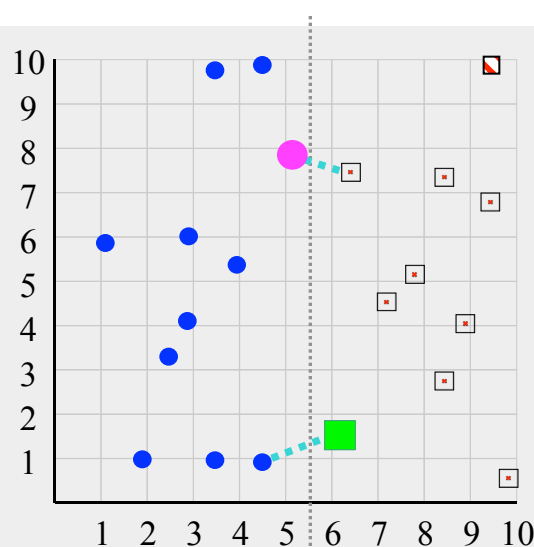
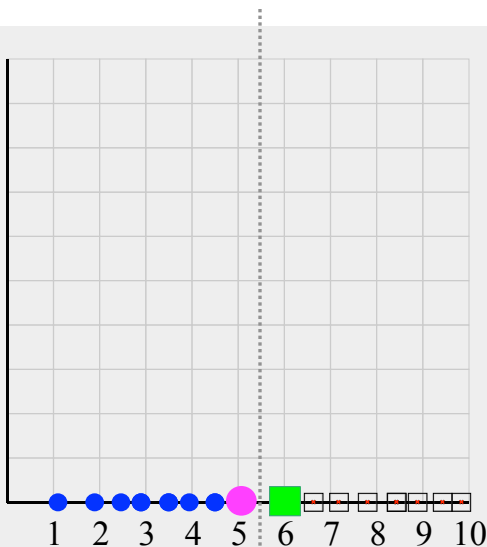
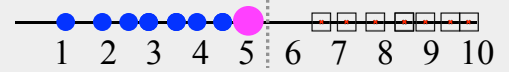
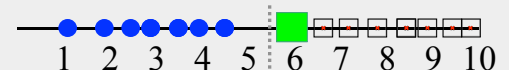
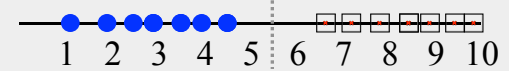
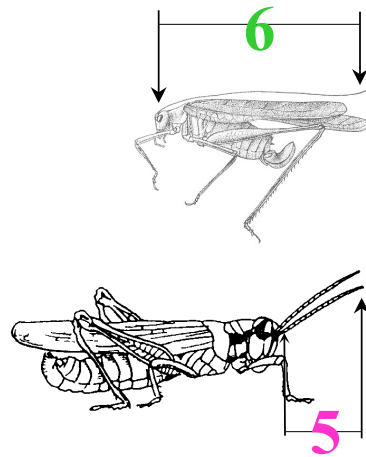
Grasshoppers

The nearest neighbor algorithm is sensitive to irrelevant features...

Suppose the following is true, if an insect's antenna is longer than 5.5 it is a **Katydid**, otherwise it is a **Grasshopper**.

Using just the antenna length we get perfect classification!

Training data



Suppose however, we add in an **irrelevant** feature, for example the insect's mass.

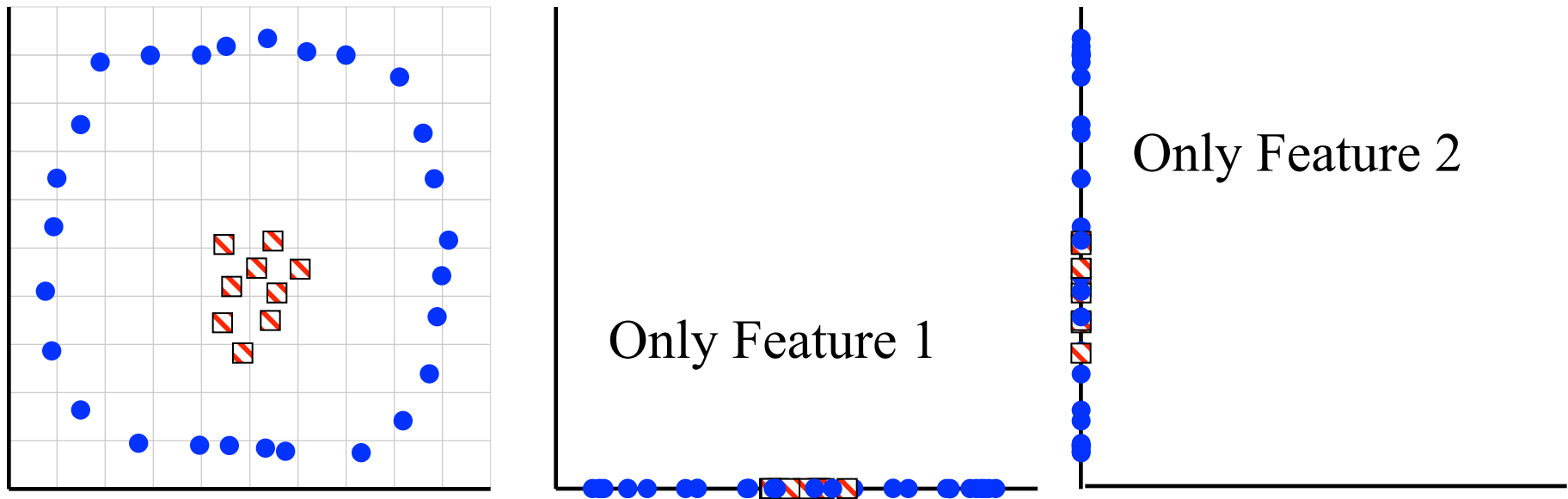
Using both the antenna length and the insect's mass with the 1-NN algorithm we get the wrong classification!

How do we mitigate the nearest neighbor algorithm's sensitivity to irrelevant features?

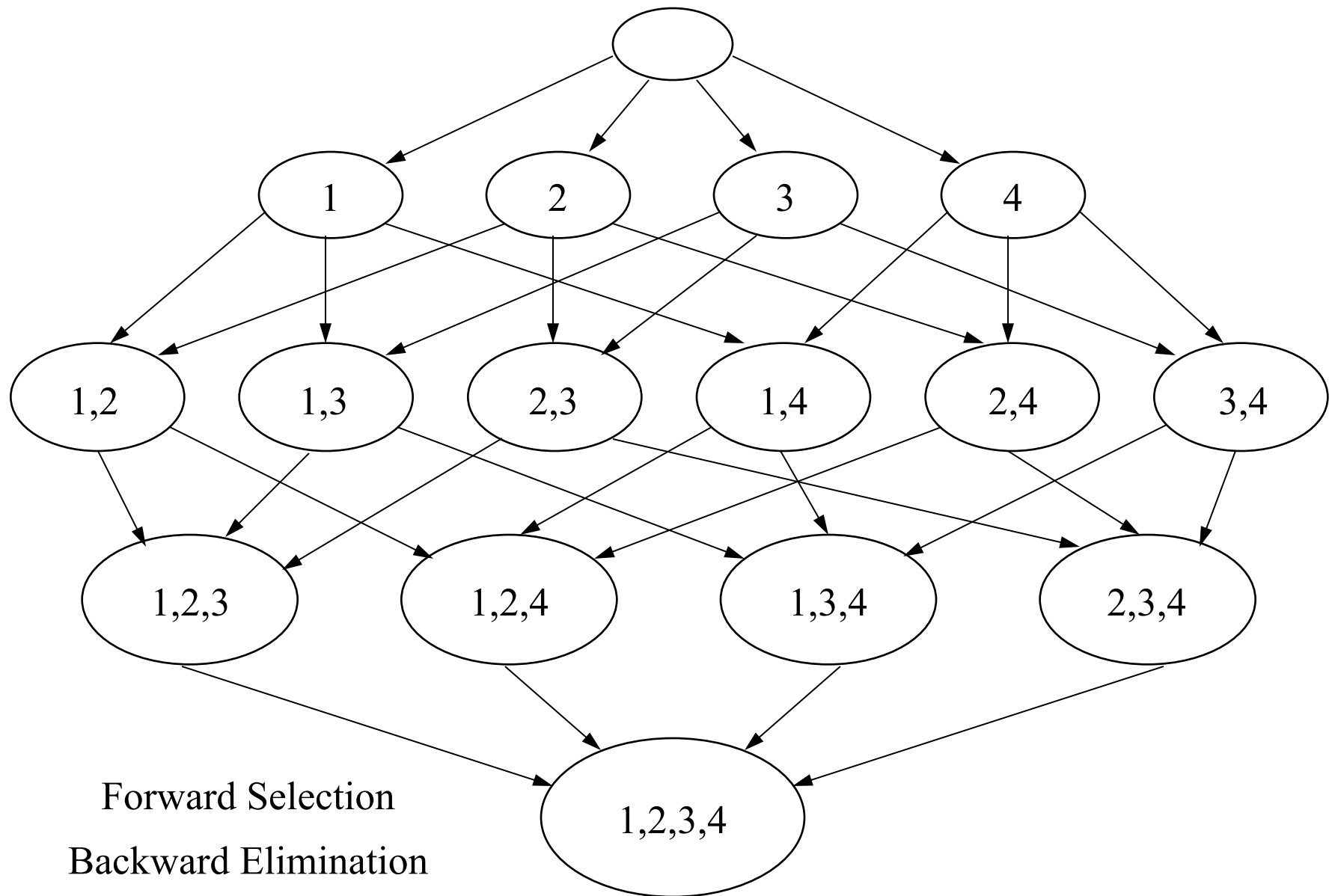
- Use more training instances
- Ask an expert what features are relevant to the task
- Use statistical tests to try to determine which features are useful
- Search over feature subsets (in the next slide we will see why this is hard)

Why searching over feature subsets is hard

Suppose you have the following classification problem, with 100 features, where it happens that Features 1 and 2 (the X and Y below) give perfect classification, but all 98 of the other features are irrelevant...



Using all 100 features will give poor results, but so will using only Feature 1, and so will using Feature 2! Of the $2^{100} - 1$ possible subsets of the features, only one really works.

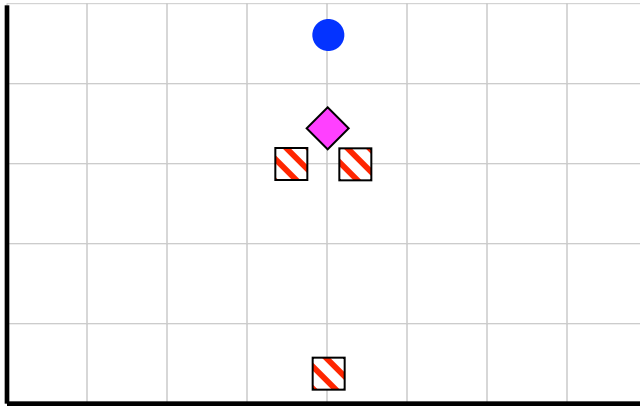


Forward Selection

Backward Elimination

Bi-directional Search

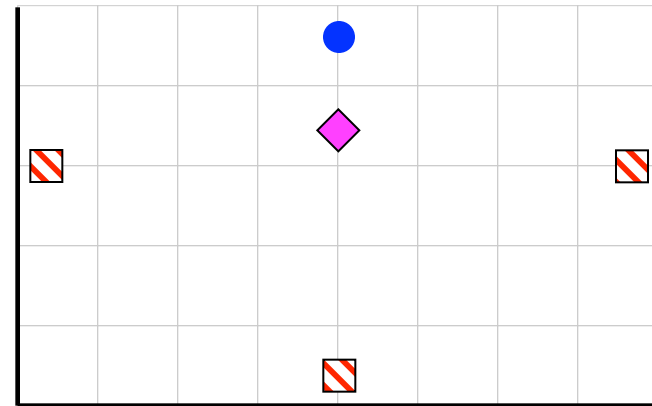
The nearest neighbor algorithm is sensitive to the units of measurement



X axis measured in **centimeters**

Y axis measure in dollars

The nearest neighbor to the **pink** unknown instance is **red**.



X axis measured in **millimeters**

Y axis measure in dollars

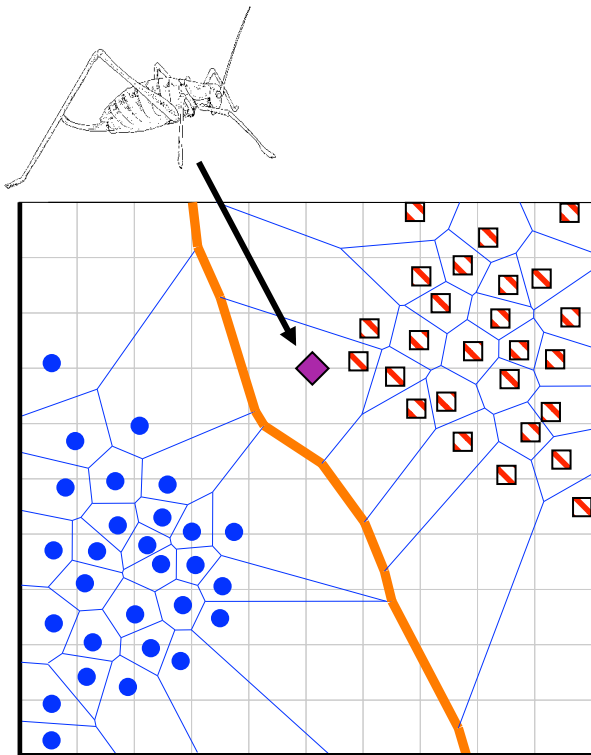
The nearest neighbor to the **pink** unknown instance is **blue**.

One solution is to normalize the units to pure numbers. Typically the features are Z-normalized to have a mean of zero and a standard deviation of one. $X = (X - \text{mean}(X)) / \text{std}(x)$

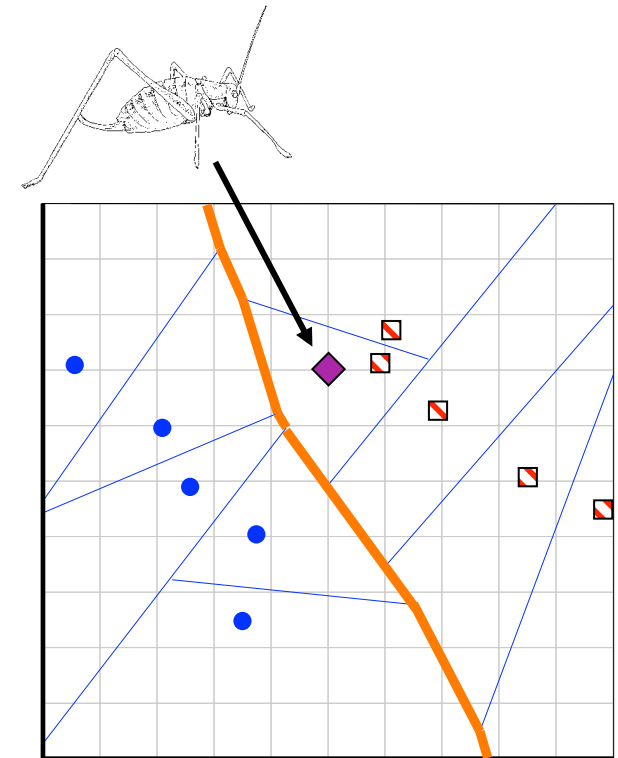
We can speed up nearest neighbor algorithm by “throwing away” some data. This is called data editing.

Note that this can sometimes improve accuracy!

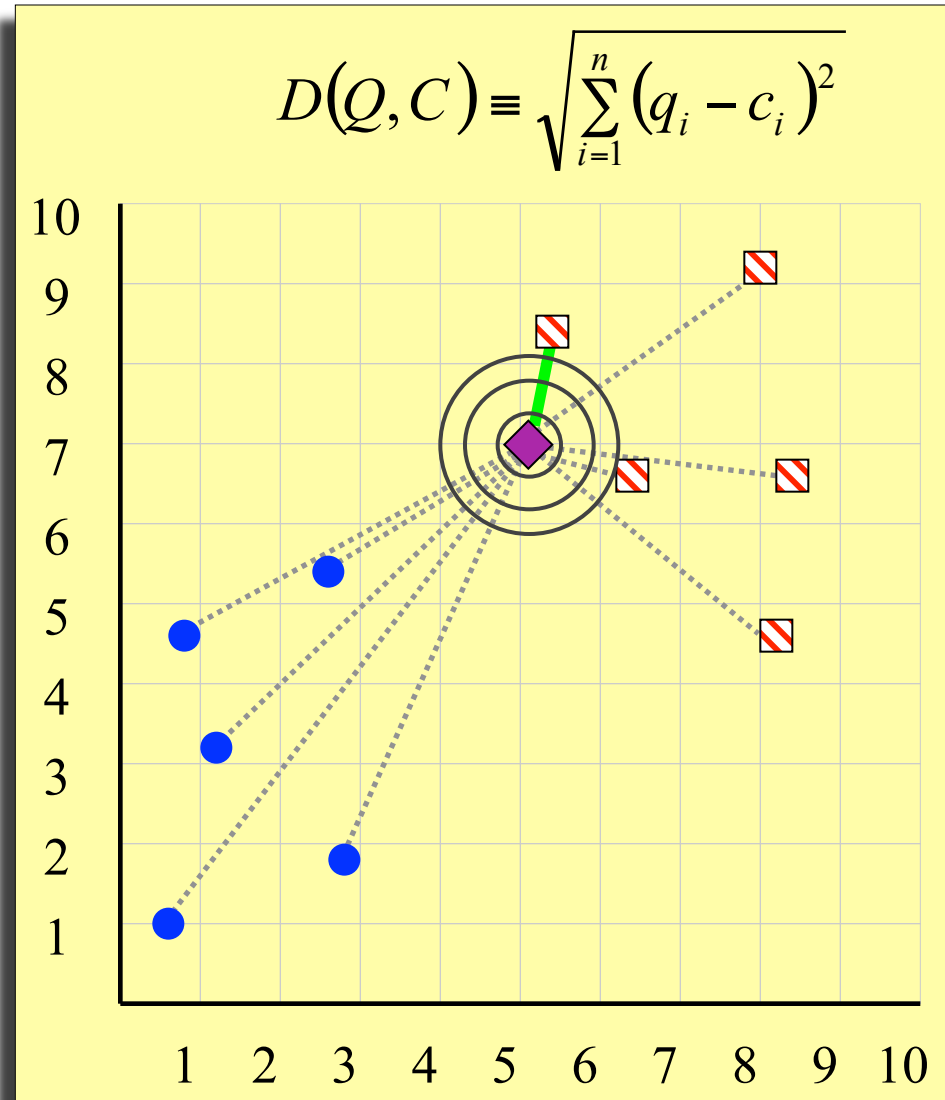
We can also speed up classification with indexing



One possible approach.
Delete all instances that are
surrounded by members of
their own class.



Up to now we have assumed that the nearest neighbor algorithm uses the Euclidean Distance, however this need not be the case...



...In fact, we can use the nearest neighbor algorithm with any distance/similarity function

For example, is “*Faloutsos*” Greek or Irish? We could compare the name “*Faloutsos*” to a database of names using string edit distance...

$$\text{edit_distance}(\textit{Faloutsos}, \textit{Keogh}) = 8$$

$$\text{edit_distance}(\textit{Faloutsos}, \textit{Gunopulos}) = 6$$

Hopefully, the similarity of the name (particularly the suffix) to other Greek names would mean the nearest neighbor is also a Greek name.

ID	Name	Class
1	Gunopulos	Greek
2	Papadopoulos	Greek
3	Kollios	Greek
4	Dardanos	Greek
5	Keogh	Irish
6	Gough	Irish
7	Greenhaugh	Irish
8	Hadleigh	Irish

Specialized distance measures exist for DNA strings, time series, images, graphs, videos, sets, fingerprints etc...

Advantages/Disadvantages of Nearest Neighbor

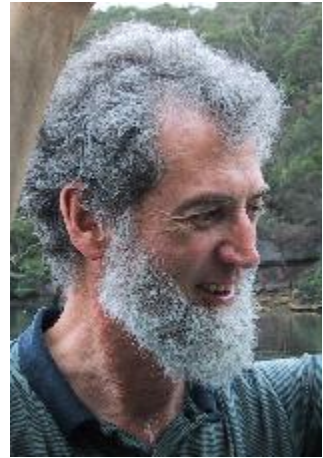
■ Advantages:

- ★ Simple to implement
- ★ Handles correlated features (Arbitrary class shapes)
- ★ Defined for any distance measure
- ★ Handles streaming data trivially

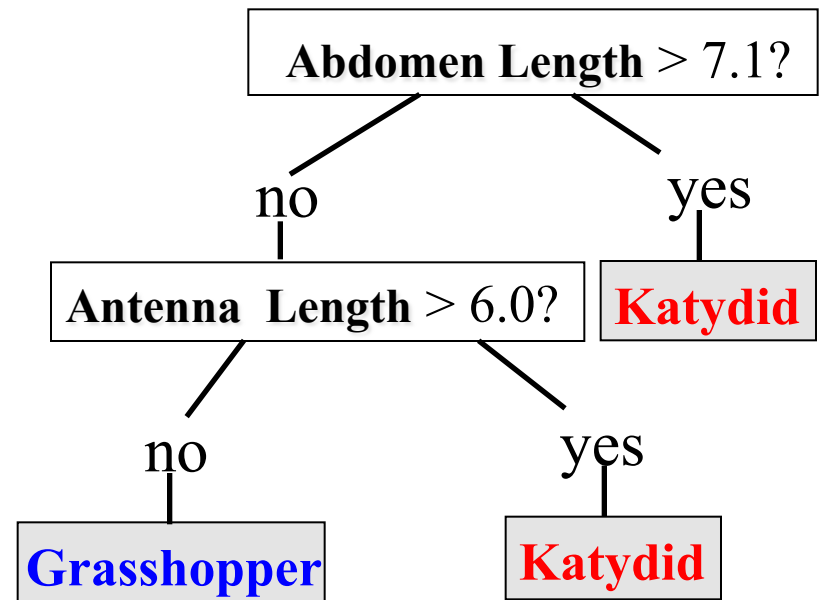
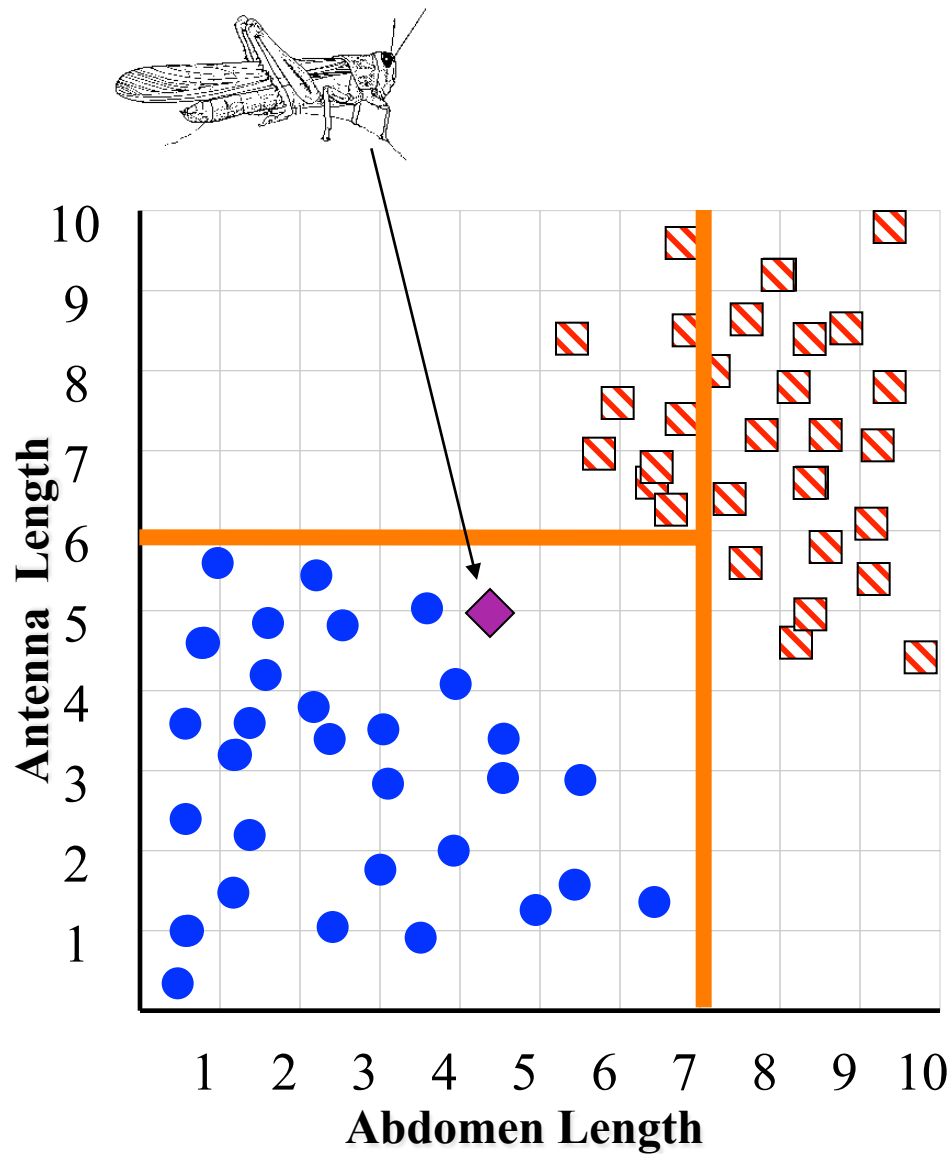
■ Disadvantages:

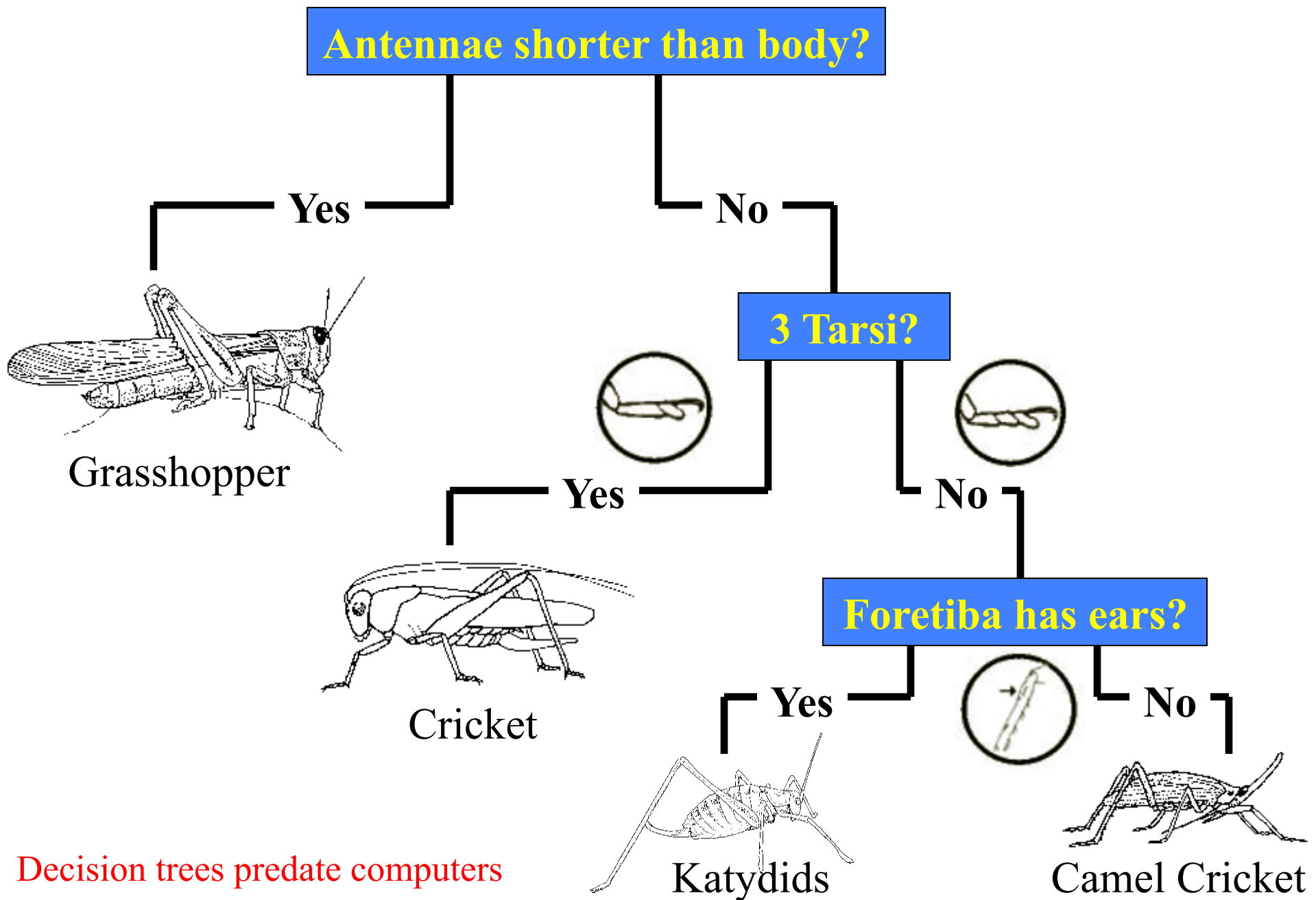
- ★ Very sensitive to irrelevant features.
- ★ Slow classification time for large datasets
- ★ Works best for real valued datasets

Decision Tree Classifier



Ross Quinlan





Decision trees predate computers

Decision Tree Classification

■ Decision tree

- ★ A flow-chart-like tree structure
- ★ Internal node denotes a test on an attribute
- ★ Branch represents an outcome of the test
- ★ Leaf nodes represent class labels or class distribution

■ Decision tree generation consists of two phases

- ★ Tree construction
 - ✓ At start, all the training examples are at the root
 - ✓ Partition examples recursively based on selected attributes
- ★ Tree pruning
 - ✓ Identify and remove branches that reflect noise or outliers

■ Use of decision tree: Classifying an unknown sample

- ★ Test the attribute values of the sample against the decision tree

Information Gain as A Splitting Criteria

- Select the attribute with the highest information gain (information gain is the expected reduction in entropy).
- Assume there are two classes, X and Y
 - ★ Let the set of examples S contain p elements of class X and n elements of class Y
 - ★ The amount of information, needed to decide if an arbitrary example in S belongs to X or Y is defined as










$$E(S) = -\frac{p}{p+n} \log_2 \left(\frac{p}{p+n} \right) - \frac{n}{p+n} \log_2 \left(\frac{n}{p+n} \right)$$

Information Gain in Decision Tree Induction

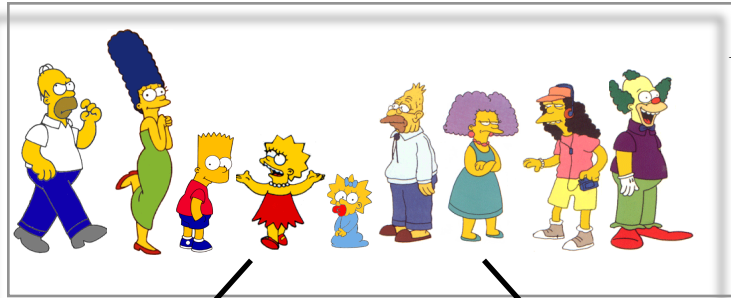
- Assume that using attribute A , a current set will be partitioned into some number of child sets
- The encoding information that would be gained by branching on A

$$Gain(A) = E(Current\ set) - \sum E(all\ child\ sets)$$

Note: entropy is at its minimum if the collection of objects is completely uniform

Person		Hair Length	Weight	Age	Class
	Homer	0"	250	36	M
	Marge	10"	150	34	F
	Bart	2"	90	10	M
	Lisa	6"	78	8	F
	Maggie	4"	20	1	F
	Abe	1"	170	70	M
	Selma	8"	160	41	F
	Otto	10"	180	38	M
	Krusty	6"	200	45	M

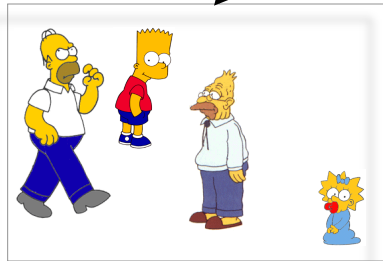
	Comic	8"	290	38	?
---	-------	----	-----	----	---



$$Entropy(S) = -\frac{p}{p+n} \log_2 \left(\frac{p}{p+n} \right) - \frac{n}{p+n} \log_2 \left(\frac{n}{p+n} \right)$$

$$Entropy(4\mathbf{F}, 5\mathbf{M}) = -(4/9) \log_2(4/9) - (5/9) \log_2(5/9) = \mathbf{0.9911}$$

yes
Hair Length <= 5?



$$Entropy(1\mathbf{F}, 3\mathbf{M}) = -(1/4) \log_2(1/4) - (3/4) \log_2(3/4) = \mathbf{0.8113}$$

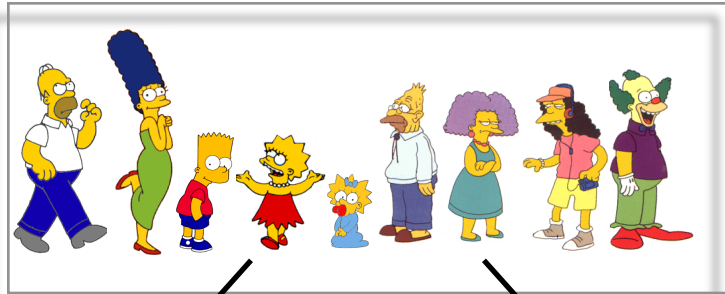


$$Entropy(3\mathbf{F}, 2\mathbf{M}) = -(3/5) \log_2(3/5) - (2/5) \log_2(2/5) = \mathbf{0.9710}$$

Let us try splitting
on *Hair length*

$$Gain(A) = E(Current\ set) - \sum E(all\ child\ sets)$$

$$Gain(Hair\ Length\ \leq\ 5) = \mathbf{0.9911} - (4/9 * \mathbf{0.8113} + 5/9 * \mathbf{0.9710}) = \mathbf{0.0911}$$



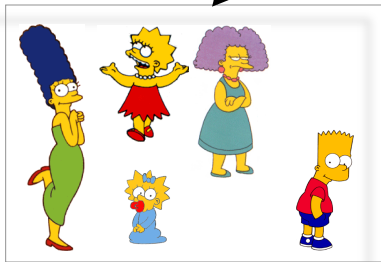
$$Entropy(S) = -\frac{p}{p+n} \log_2 \left(\frac{p}{p+n} \right) - \frac{n}{p+n} \log_2 \left(\frac{n}{p+n} \right)$$

$$Entropy(4\mathbf{F}, 5\mathbf{M}) = -(4/9) \log_2(4/9) - (5/9) \log_2(5/9) = \mathbf{0.9911}$$

yes

no

Weight ≤ 160?



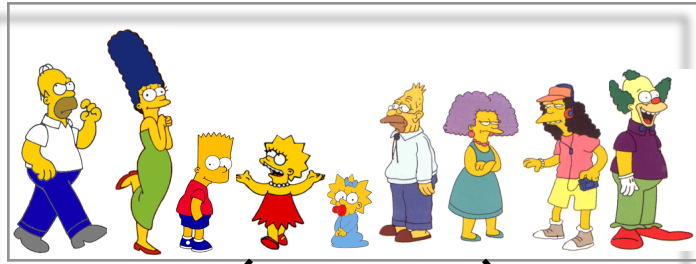
Let us try splitting
on *Weight*

$$Entropy(4\mathbf{F}, 1\mathbf{M}) = -(4/5) \log_2(4/5) - (1/5) \log_2(1/5) = \mathbf{0.7219}$$

$$Entropy(0\mathbf{F}, 4\mathbf{M}) = -(0/4) \log_2(0/4) - (4/4) \log_2(4/4) = \mathbf{0}$$

$$Gain(A) = E(\text{Current set}) - \sum E(\text{all child sets})$$

$$Gain(\text{Weight} \leq 160) = \mathbf{0.9911} - (5/9 * \mathbf{0.7219} + 4/9 * \mathbf{0}) = \mathbf{0.5900}$$



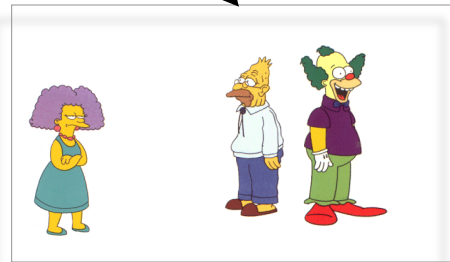
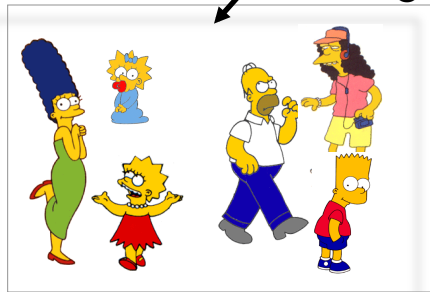
$$Entropy(S) = -\frac{p}{p+n} \log_2 \left(\frac{p}{p+n} \right) - \frac{n}{p+n} \log_2 \left(\frac{n}{p+n} \right)$$

$$Entropy(4\mathbf{F}, 5\mathbf{M}) = -(4/9) \log_2(4/9) - (5/9) \log_2(5/9) = \mathbf{0.9911}$$

yes

age ≤ 40?

no



Let us try splitting
on *Age*

$$Entropy(3\mathbf{F}, 3\mathbf{M}) = -(3/6) \log_2(3/6) - (3/6) \log_2(3/6) = \mathbf{1}$$

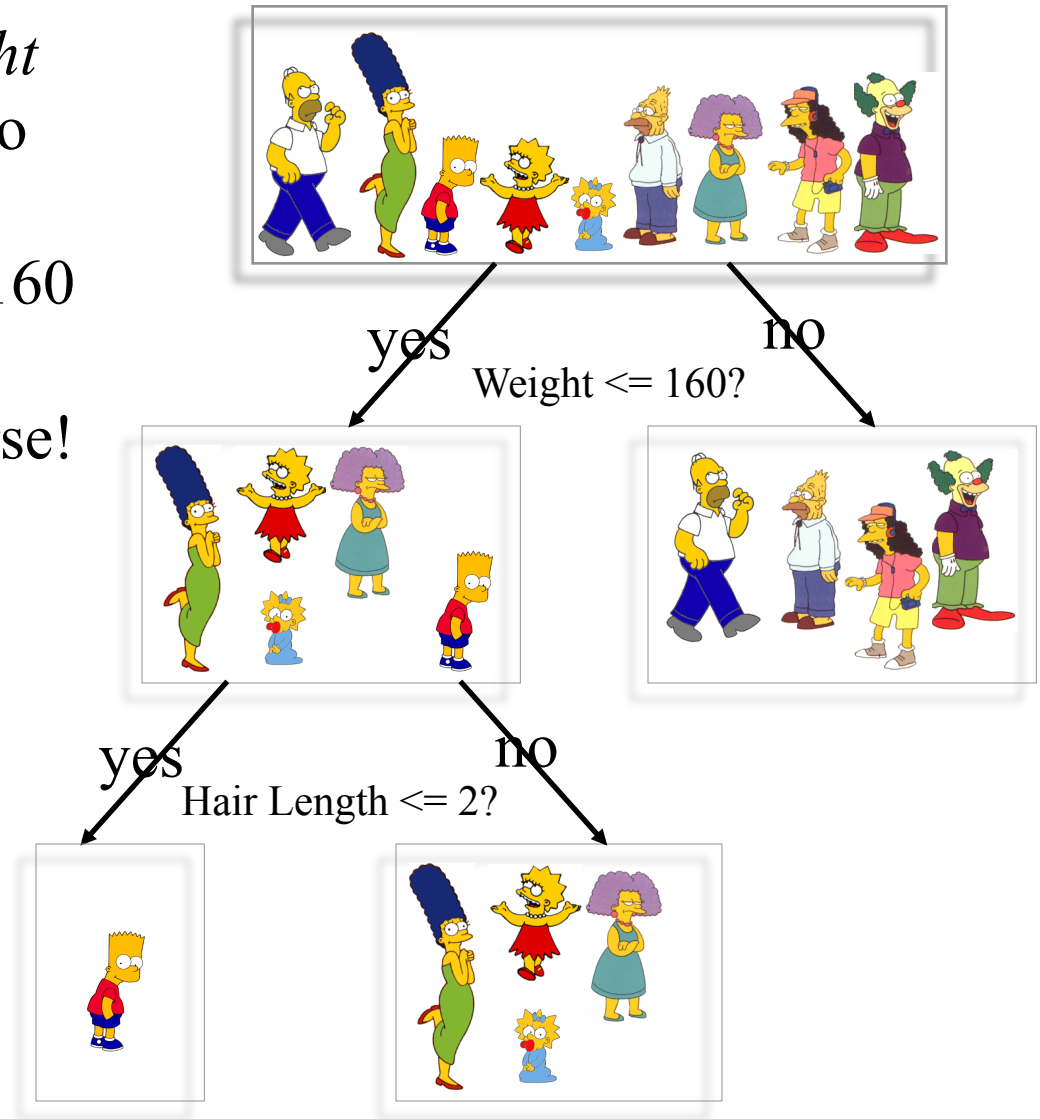
$$Entropy(1\mathbf{F}, 2\mathbf{M}) = -(1/3) \log_2(1/3) - (2/3) \log_2(2/3) = \mathbf{0.9183}$$

$$Gain(A) = E(\text{Current set}) - \sum E(\text{all child sets})$$

$$Gain(\text{Age} \leq 40) = \mathbf{0.9911} - (6/9 * \mathbf{1} + 3/9 * \mathbf{0.9183}) = \mathbf{0.0183}$$

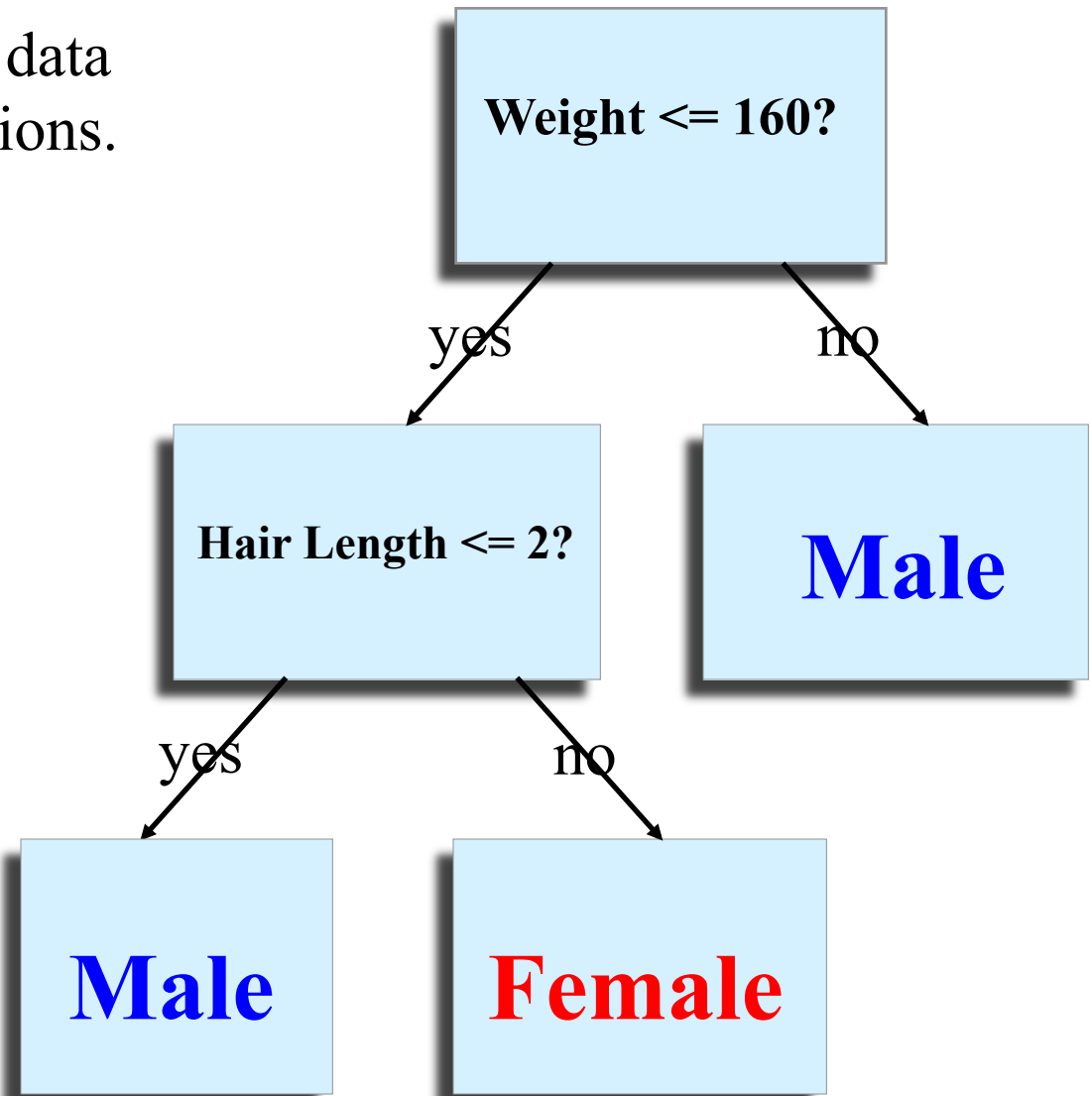
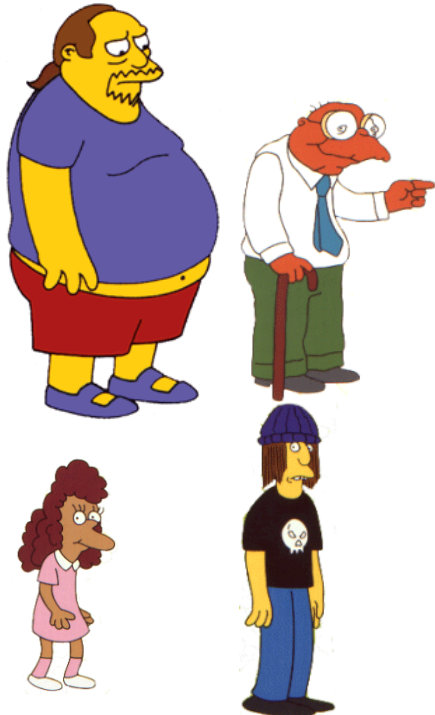
Of the 3 features we had, *Weight* was best. But while people who weigh over 160 are perfectly classified (as males), the under 160 people are not perfectly classified... So we simply recurse!

This time we find that we can split on *Hair length*, and we are done!

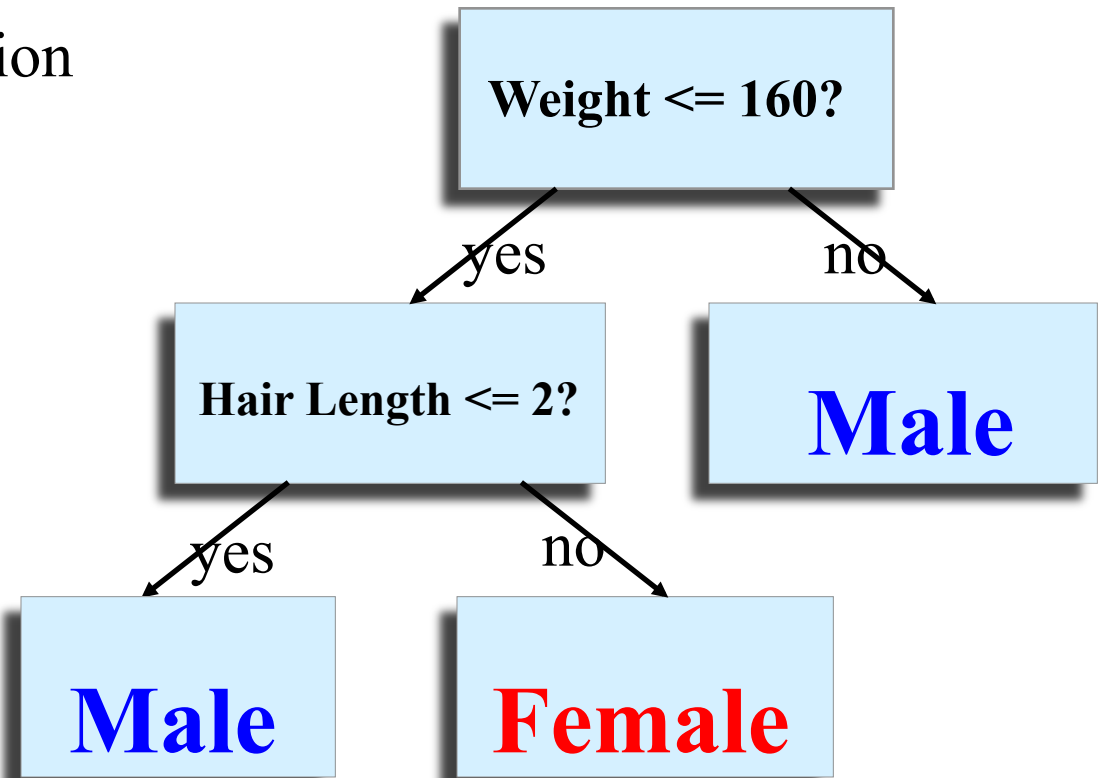


We don't need to keep the data around, just the test conditions.

How would these people be classified?



It is trivial to convert Decision
Trees to rules...



Rules to Classify Males/Females

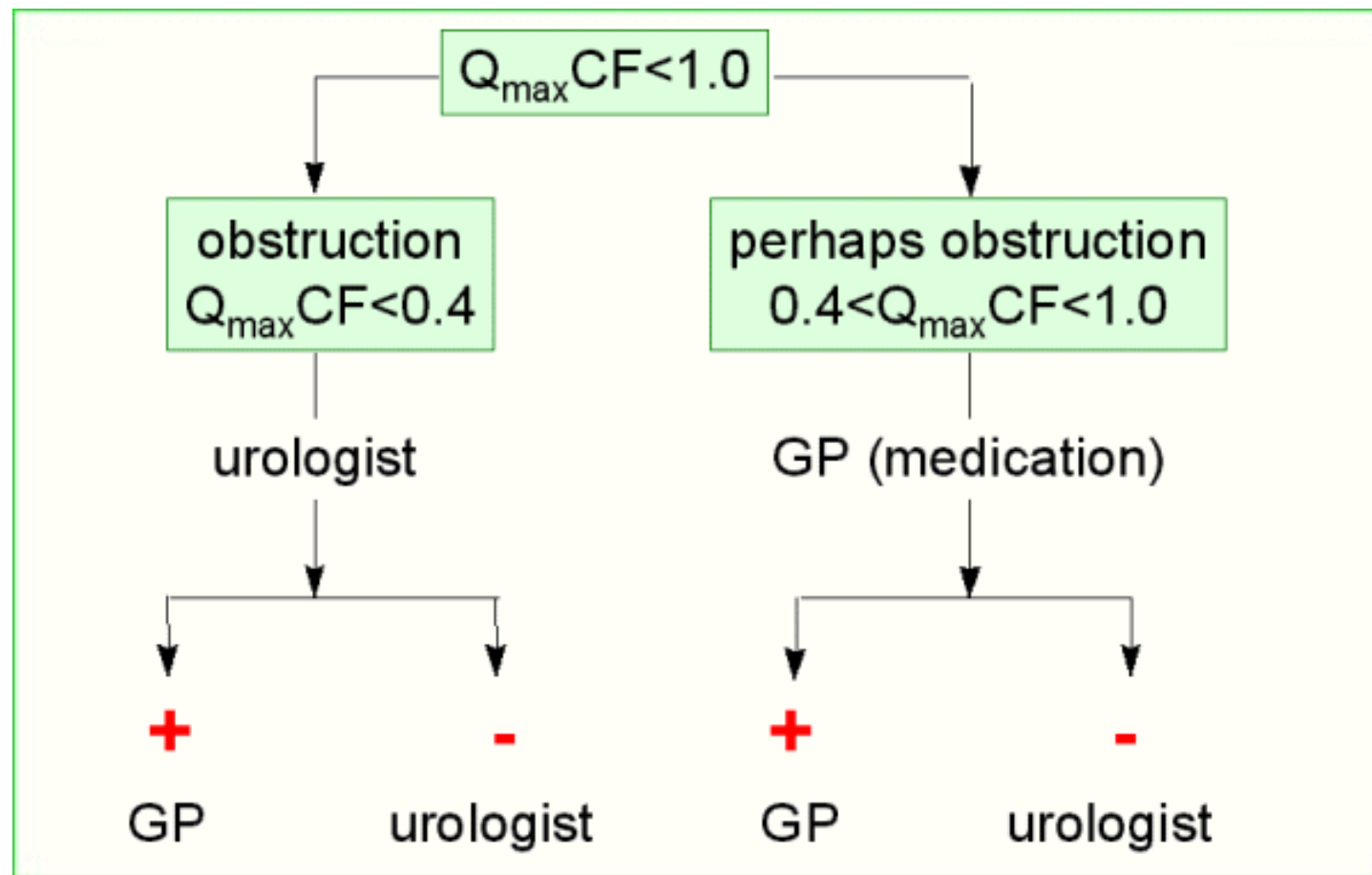
If *Weight* greater than 160, classify as **Male**

Elseif *Hair Length* less than or equal to 2, classify as **Male**

Else classify as **Female**

Once we have learned the decision tree, we don't even need a computer!

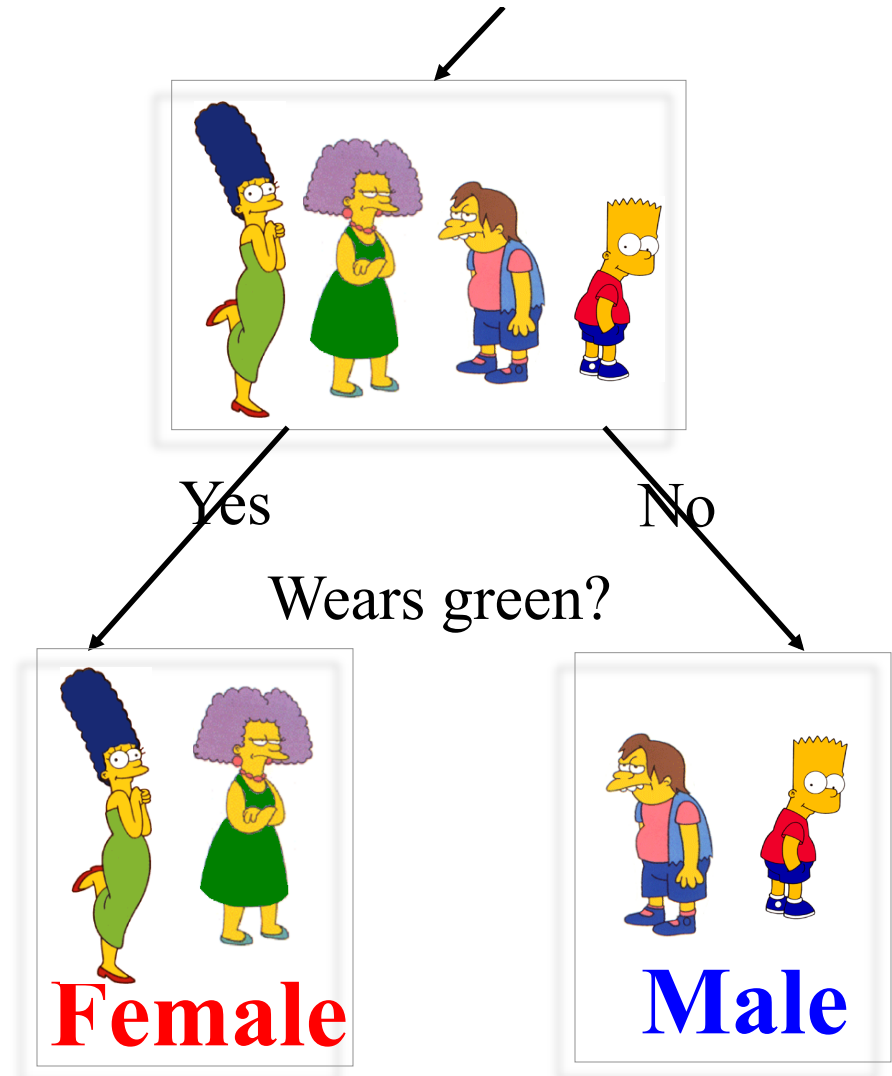
This decision tree is attached to a medical machine, and is designed to help nurses make decisions about what type of doctor to call.



Decision tree for a typical shared-care setting applying the system for the diagnosis of prostatic obstructions.

The worked examples we have seen were performed on small datasets. However with small datasets there is a great danger of overfitting the data...

When you have few datapoints, there are many possible splitting rules that perfectly classify the data, but will not generalize to future datasets.

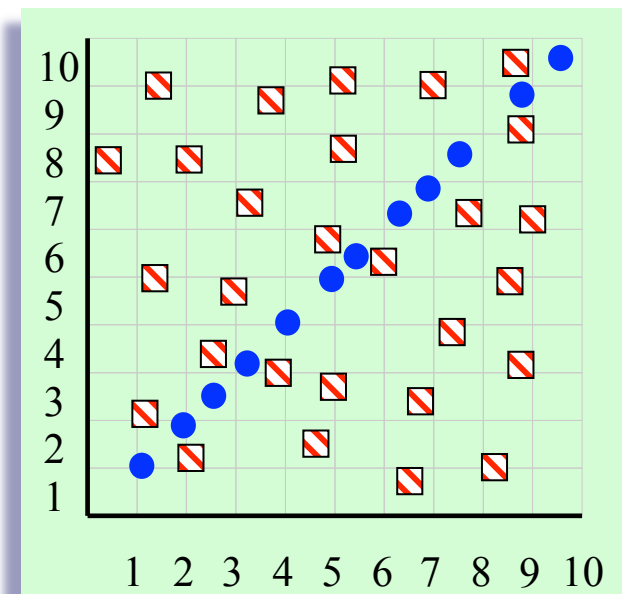
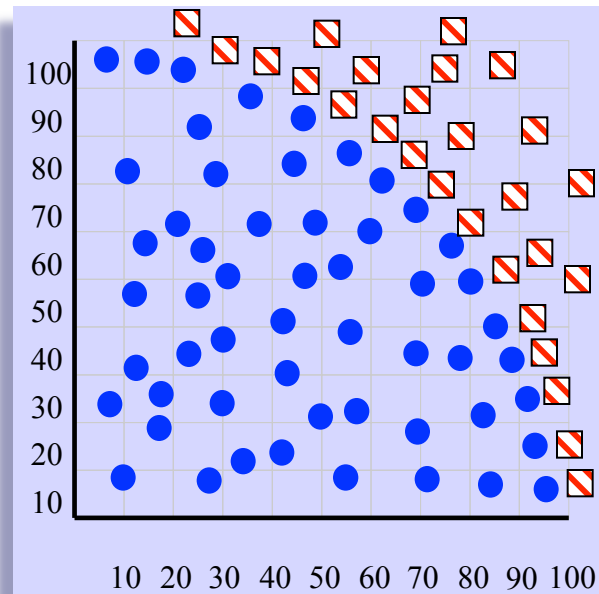
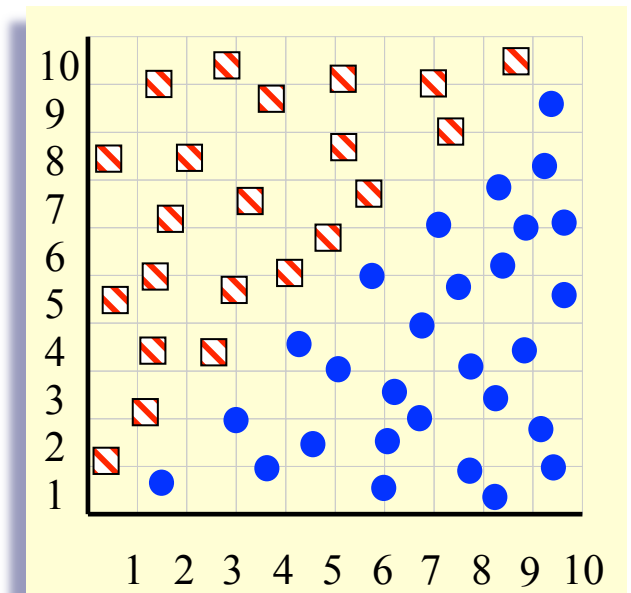


For example, the rule “Wears green?” perfectly classifies the data, so does “Mother’s name is Jacqueline?”, so does “Has blue shoes”...

Avoid Overfitting in Classification

- The generated tree may overfit the training data
 - ★ Too many branches, some may reflect anomalies due to noise or outliers
 - ★ Result is in poor accuracy for unseen samples
- Two approaches to avoid overfitting
 - ★ Prepruning: Halt tree construction early—do not split a node if this would result in the goodness measure falling below a threshold
 - ✓ Difficult to choose an appropriate threshold
 - ★ Postpruning: Remove branches from a “fully grown” tree—get a sequence of progressively pruned trees
 - ✓ Use a set of data different from the training data to decide which is the “best pruned tree”

Which of the “Pigeon Problems” can be solved by a Decision Tree?

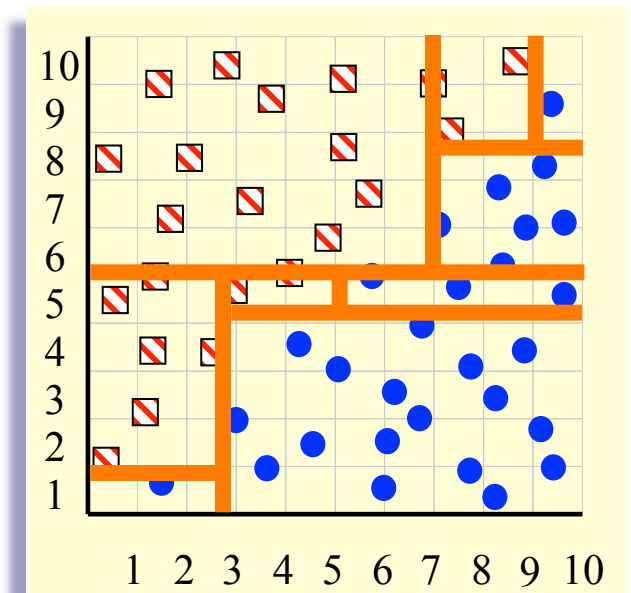


Which of the “Pigeon Problems” can be solved by a Decision Tree?

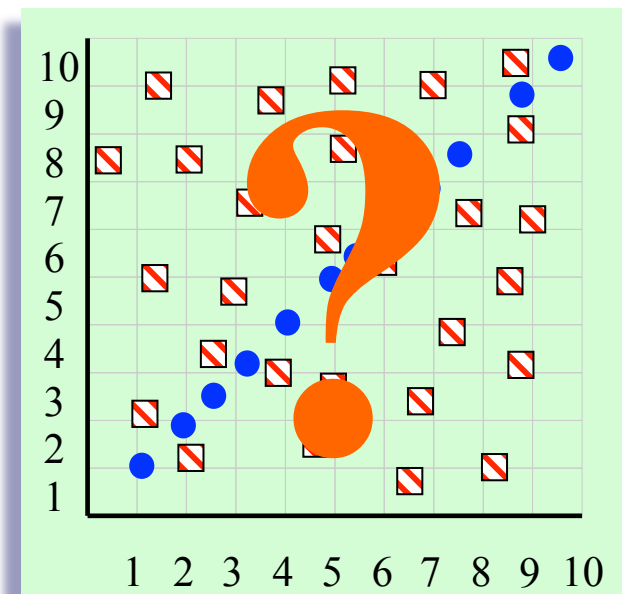
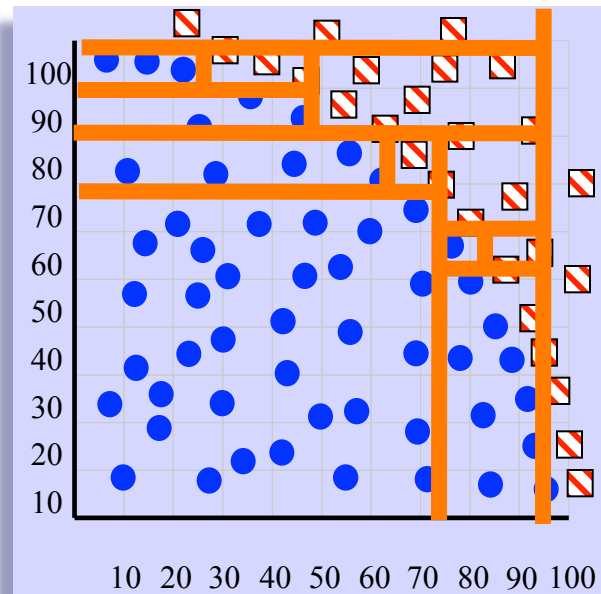
Deep Bushy Tree

Useless

Deep Bushy Tree



The Decision Tree
has a hard time with
correlated attributes



Advantages/Disadvantages of Decision Trees

■ Advantages:

- ★ Easy to understand (Doctors love them!)
- ★ Easy to generate rules

■ Disadvantages:

- ★ May suffer from overfitting.
- ★ Classifies by rectangular partitioning (so does not handle correlated features very well).
- ★ Can be quite large – pruning is necessary.
- ★ Does not handle streaming data easily

Bayesian Methods

- Learning and classification methods based on probability theory.
- Bayes theorem plays a critical role in probabilistic learning and classification.
- Builds a *generative model* that approximates how data is produced
- Uses *prior* probability of each category given no information about an item.
- Categorization produces a *posterior* probability distribution over the possible categories given a description of an item.

Naïve Bayes Classifier



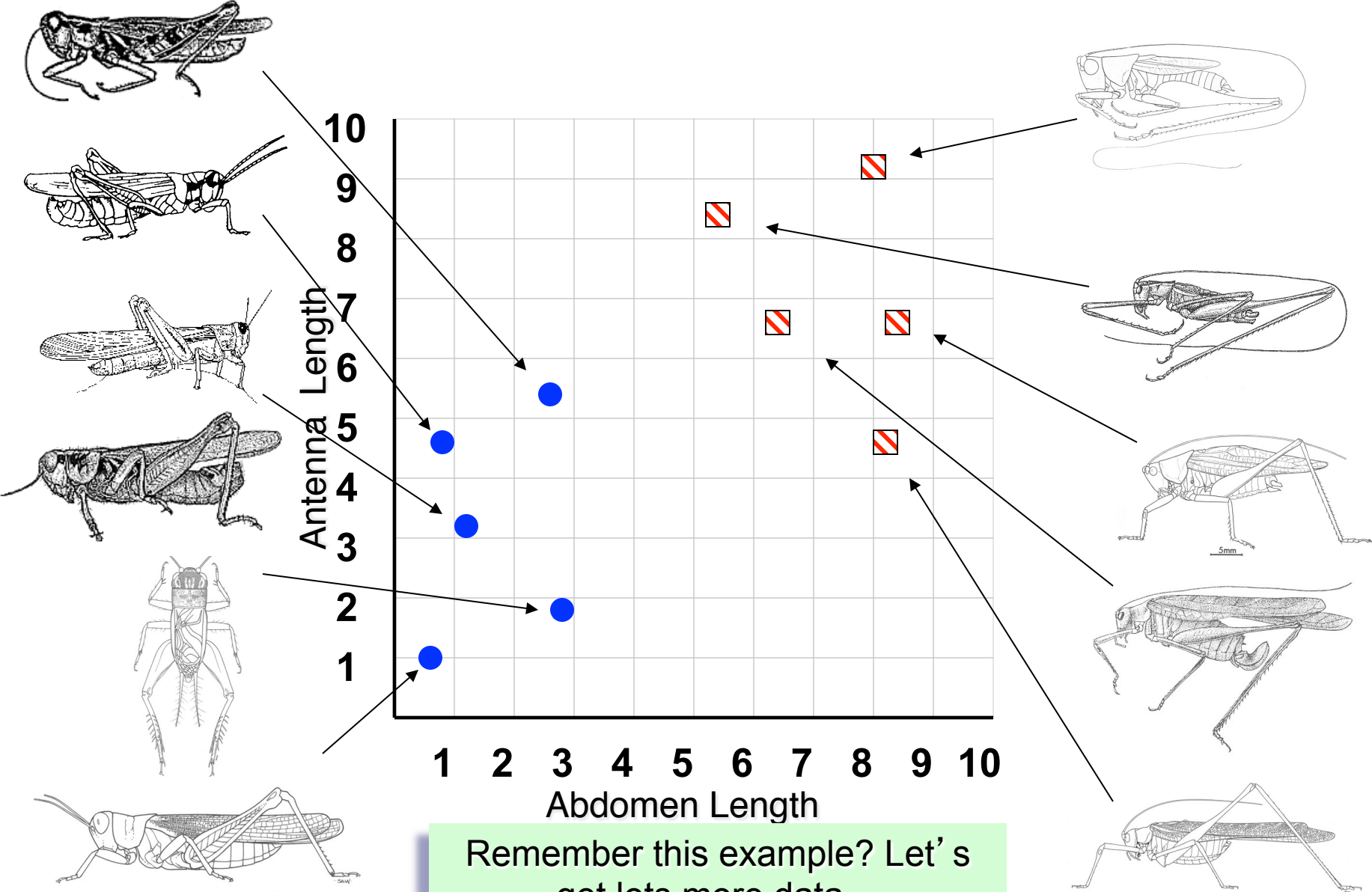
Thomas Bayes

1702 - 1761

We will start off with a visual intuition, before looking at the math...

Grasshoppers

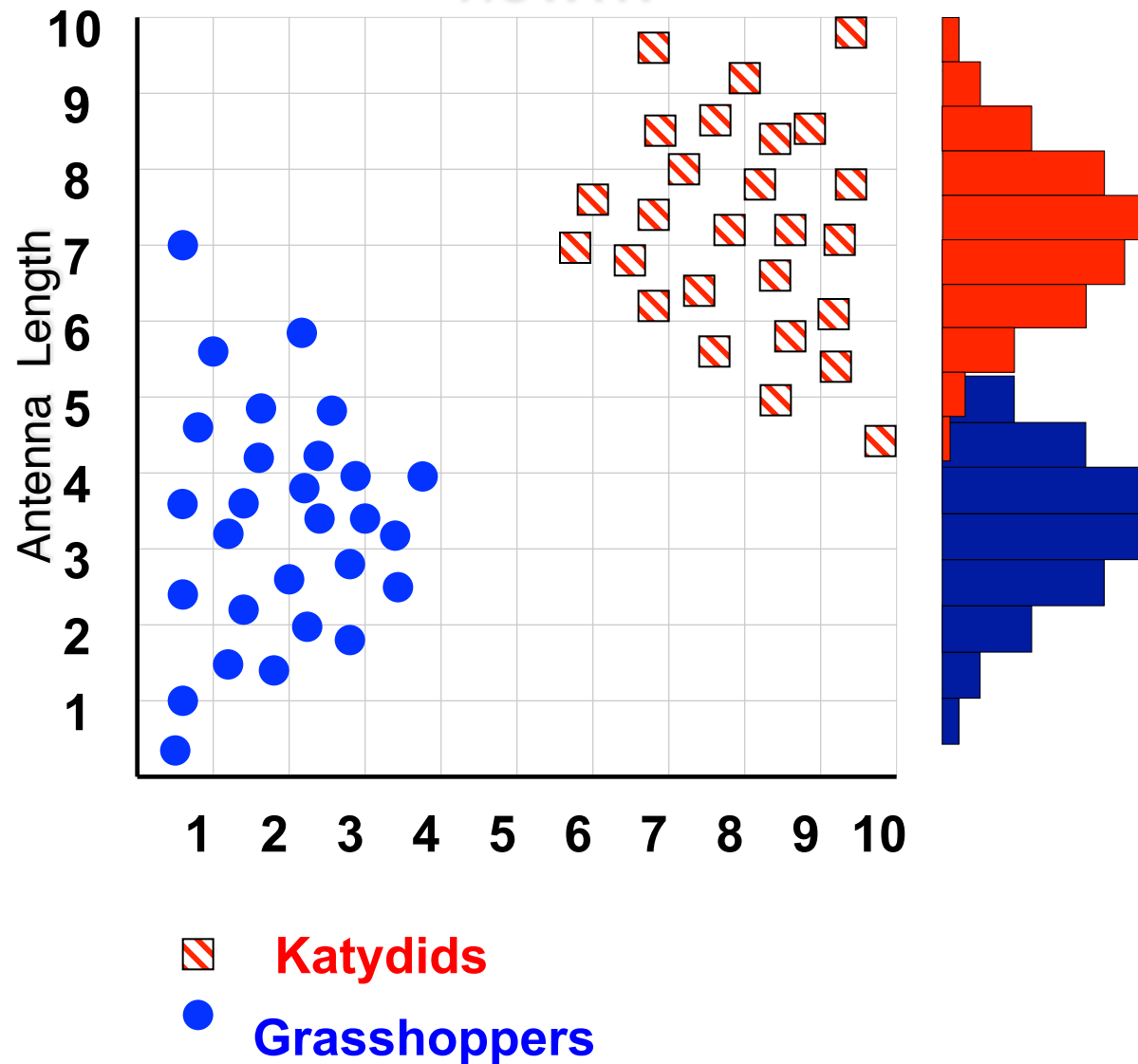
Katydid



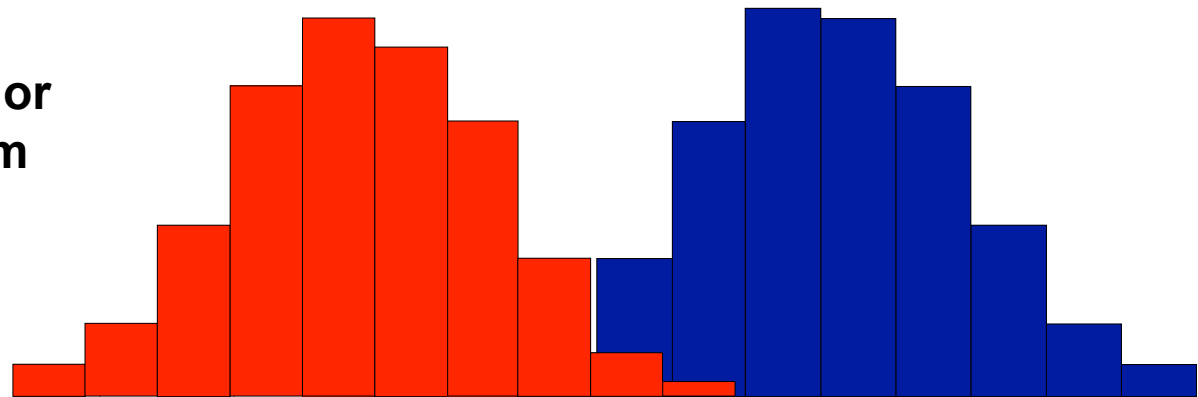
Remember this example? Let's get lots more data...

With a lot of data, we can build a histogram.
Let us just build one for “Antenna Length” for

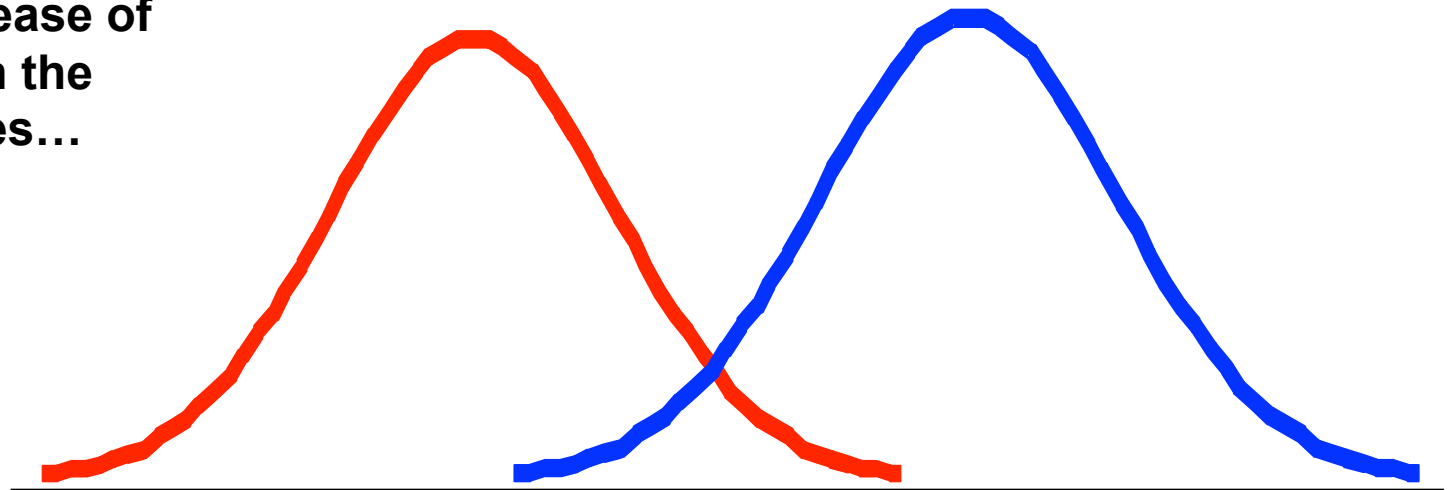
now...



We can leave the histograms as they are, or we can summarize them with two normal distributions.

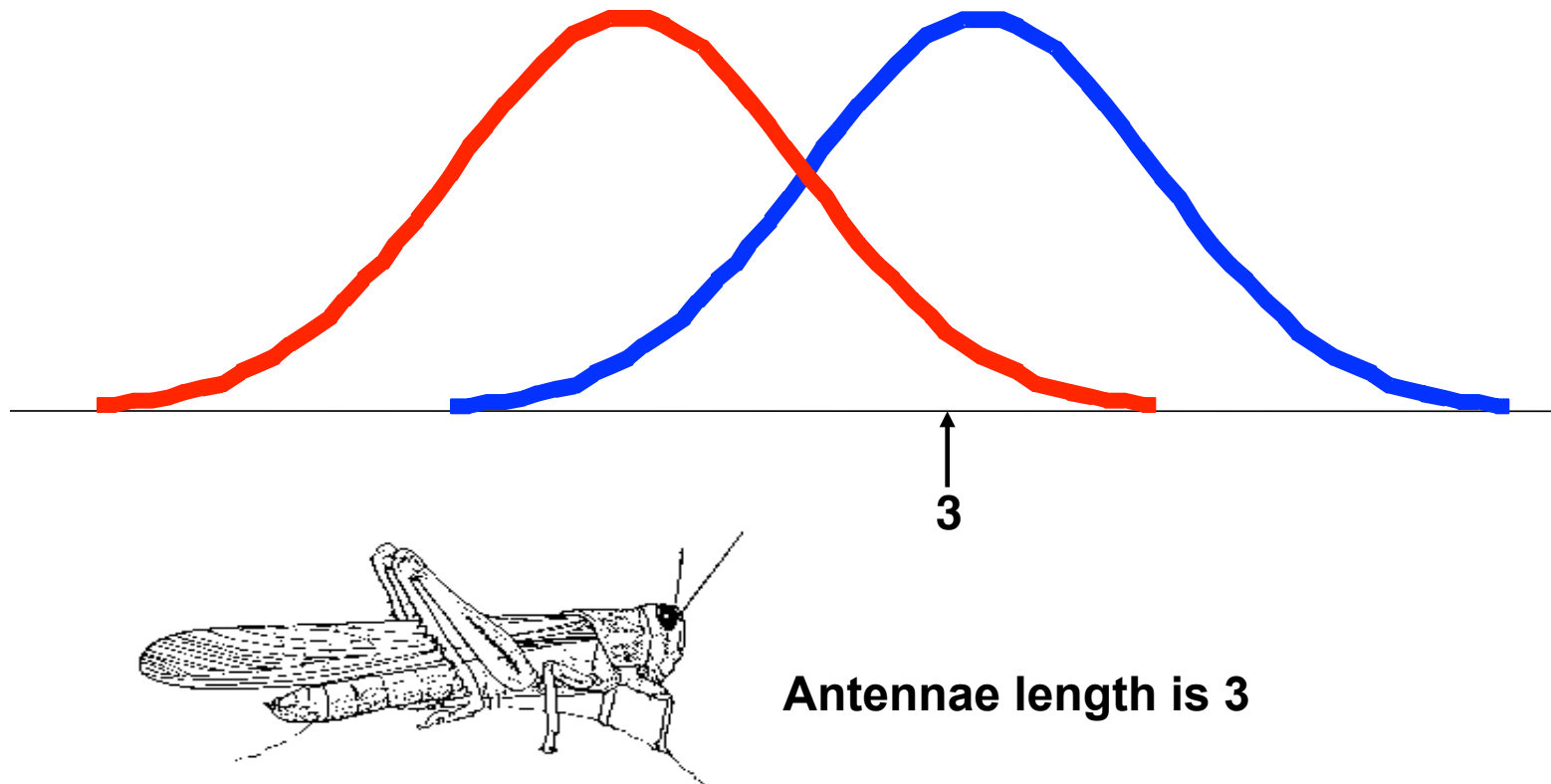


Let us use two normal distributions for ease of visualization in the following slides...



- We want to classify an insect we have found. Its antennae are 3 units long. How can we classify it?
- We can just ask ourselves, give the distributions of antennae lengths we have seen, is it more *probable* that our insect is a **Grasshopper** or a **Katydid**.
- There is a formal way to discuss the most *probable* classification...

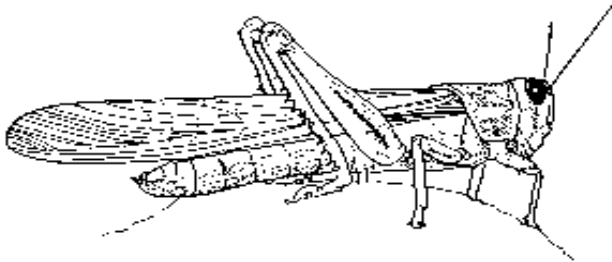
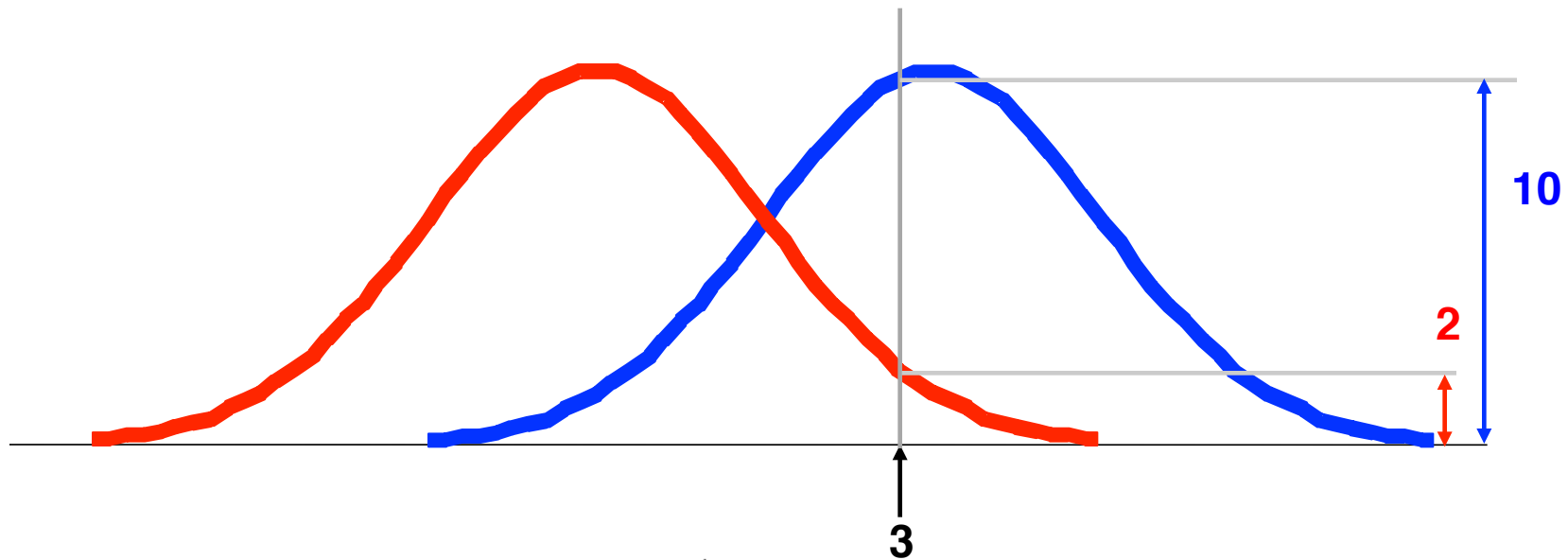
$p(c_j | d)$ = probability of class c_j , given that we have observed d



$p(c_j | d)$ = probability of class c_j , given that we have observed d

P(Grasshopper | 3) = ?

P(Katydid | 3) = ?

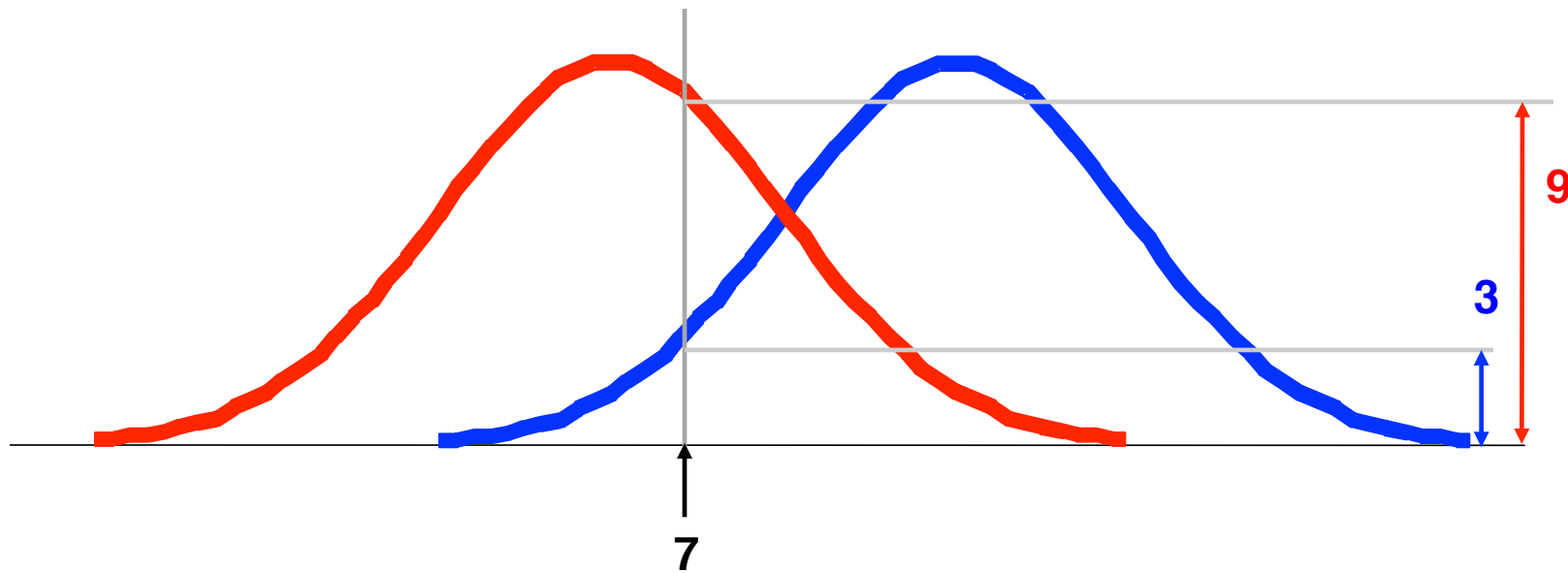


Antennae length is 3

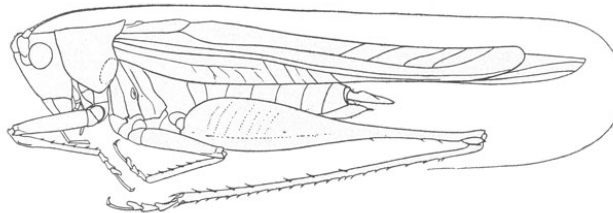
$p(c_j | d)$ = probability of class c_j , given that we have observed d

$P(\text{Grasshopper} | 7) = ?$

$P(\text{Katydid} | 7) = ?$



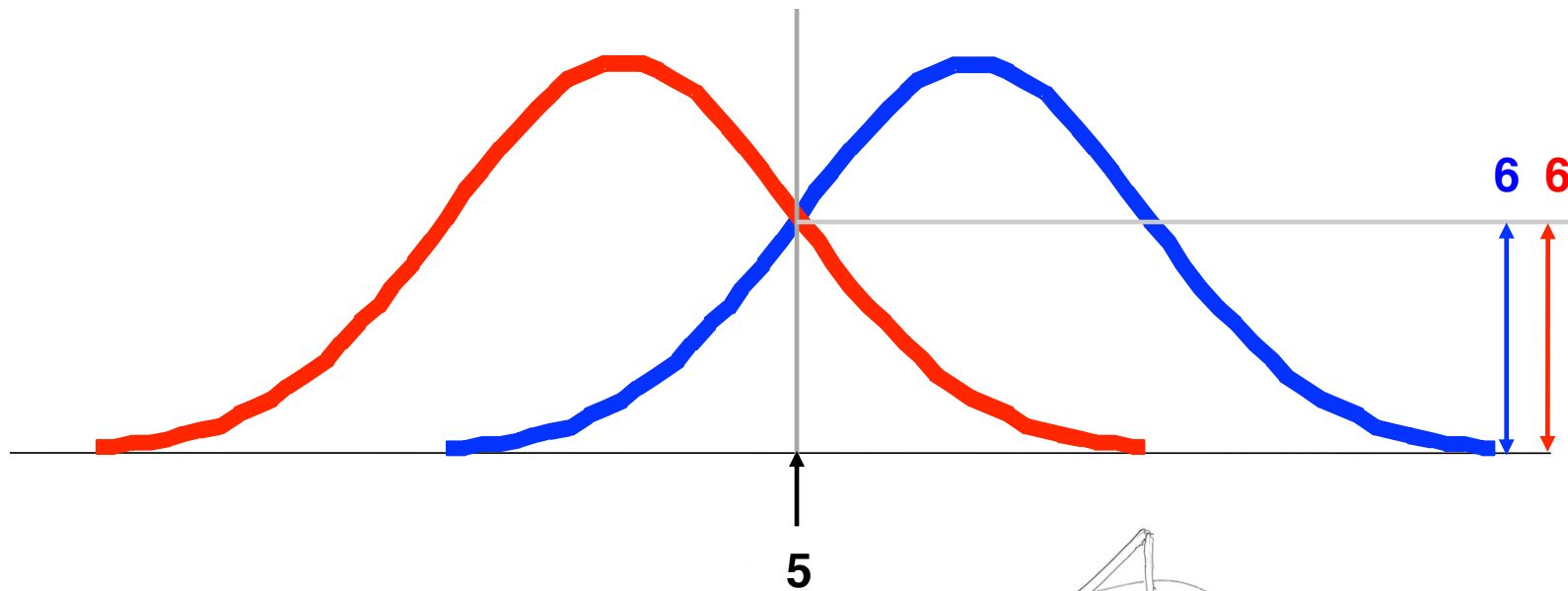
Antennae length is 7



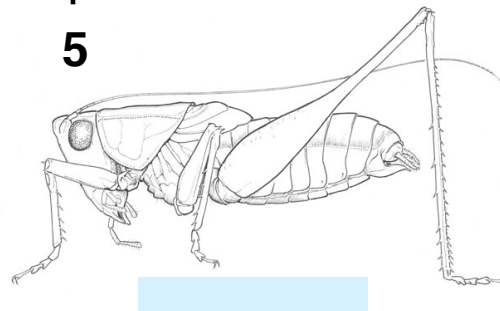
$p(c_j | d)$ = probability of class c_j , given that we have observed d

$P(\text{Grasshopper} | 5) = ?$

$P(\text{Katydid} | 5) = ?$



Antennae length is 5



Bayes Classifiers

- That was a visual intuition for a simple case of the Bayes classifier, also called Naïve Bayes
- We are about to see some of the mathematical formalisms, and more examples, but keep in mind the basic idea.
- *Find out the probability of the **previously unseen instance** belonging to each class, then simply pick the most probable class.*

Bayes Classifier

- A probabilistic framework for solving classification problems

- Conditional Probability:

$$P(C | A) = \frac{P(A, C)}{P(A)}$$

$$P(A | C) = \frac{P(A, C)}{P(C)}$$

- Bayes theorem:

$$P(C | A) = \frac{P(A | C)P(C)}{P(A)}$$

Bayes Classifiers

- Bayesian classifiers use **Bayes theorem**, which says

$$p(c_j | d) = \frac{p(d | c_j) p(c_j)}{p(d)}$$

- $p(c_j | d)$ = probability of instance d being in class c_j ,
This is what we are trying to compute
- $p(d | c_j)$ = probability of generating instance d given class c_j ,
We can imagine that being in class c_j , causes you to have feature d with some probability
- $p(c_j)$ = probability of occurrence of class c_j ,
This is just how frequent the class c_j is in our database
- $p(d)$ = probability of instance d occurring
This can actually be ignored, since it is the same for all classes

Example of Bayes Theorem

■ Given:

- ★ A doctor knows that meningitis causes stiff neck 50% of the time
- ★ Prior probability of any patient having meningitis is $1/50,000$
- ★ Prior probability of any patient having stiff neck is $1/20$

- ## ■ If a patient has stiff neck, what's the probability he/she has meningitis?

Example of Bayes Theorem

■ Given:

- ★ A doctor knows that meningitis causes stiff neck 50% of the time
- ★ Prior probability of any patient having meningitis is 1/50,000
- ★ Prior probability of any patient having stiff neck is 1/20

- ## ■ If a patient has stiff neck, what's the probability he/she has meningitis?

$$P(M | S) = \frac{P(S | M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Bayesian Classifiers

- Consider each attribute and class label as random variables
- Given a record with attributes (A_1, A_2, \dots, A_n)
 - ★ Goal is to predict class C
 - ★ Specifically, we want to find the value of C that maximizes $P(C | A_1, A_2, \dots, A_n)$
- Can we estimate $P(C | A_1, A_2, \dots, A_n)$ directly from data?

Bayesian Classifiers

■ Approach:

- ★ compute the posterior probability $P(C \mid A_1, A_2, \dots, A_n)$ for all values of C using the Bayes theorem

$$P(C \mid A_1 A_2 \dots A_n) = \frac{P(A_1 A_2 \dots A_n \mid C) P(C)}{P(A_1 A_2 \dots A_n)}$$

- ★ Choose value of C that maximizes $P(C \mid A_1, A_2, \dots, A_n)$

- ★ Equivalent to choosing value of C that maximizes $P(A_1, A_2, \dots, A_n \mid C) P(C)$

■ How to estimate $P(A_1, A_2, \dots, A_n \mid C)$?

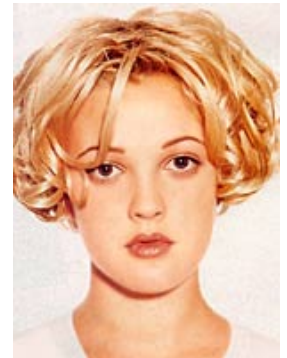
Assume that we have two classes

$C_1 = \text{male}$, and $C_2 = \text{female}$.

We have a person whose sex we do not know, say “*drew*” or *d*.

Classifying *drew* as male or female is equivalent to asking is it more probable that *drew* is **male** or **female**, i.e which is greater $p(\text{male} \mid \text{drew})$ or $p(\text{female} \mid \text{drew})$

(Note: “Drew can be a male or female name”)



Drew Barrymore



Drew Carey

Assume that we have two classes

$C_1 = \text{male}$, and $C_2 = \text{female}$.

We have a person whose sex we do not know, say “*drew*” or d .

Classifying *drew* as male or female is equivalent to asking is it more probable that *drew* is **male** or **female**, i.e which is greater $p(\text{male} \mid \text{drew})$ or $p(\text{female} \mid \text{drew})$

(Note: “Drew can be a male or female name”)



Drew Barrymore



Drew Carey

What is the probability of being called “*drew*” given that you are a **male**?

$$\frac{p(\text{male} \mid \text{drew}) = p(\text{drew} \mid \text{male}) p(\text{male})}{p(\text{drew})}$$

What is the probability of being a **male**?

What is the probability of being named “*drew*”?
(actually irrelevant, since it is the same for all classes)



Officer Drew

This is Officer Drew. Is Officer Drew a
Male or **Female**?

Luckily, we have a small database
with names and gender.

We can use it to apply Bayes
rule...

$$p(c_j | d) = \frac{p(d | c_j) p(c_j)}{p(d)}$$

Name	Sex
Drew	Male
Claudia	Female
Drew	Female
Drew	Female
Alberto	Male
Karin	Female
Nina	Female
Sergio	Male



Officer Drew

$$p(c_j | d) = \frac{p(d | c_j) p(c_j)}{p(d)}$$

Name	Sex
Drew	Male
Claudia	Female
Drew	Female
Drew	Female
Alberto	Male
Karin	Female
Nina	Female
Sergio	Male

$$p(\text{male} | \text{drew}) = \frac{1/3 * 3/8}{3/8} = \frac{0.125}{3/8}$$

$$p(\text{female} | \text{drew}) = \frac{2/5 * 5/8}{3/8} = \frac{0.250}{3/8}$$

Officer Drew is more likely to be a **Female**.

So far we have only considered Bayes Classification when we have one attribute (the “*antennae length*”, or the “*name*”). But we may have many features.

How do we use all the features?

Name	Over 170cm	Eye	Hair length	Sex
Drew	No	Blue	Short	Male
Claudia	Yes	Brown	Long	Female
Drew	No	Blue	Long	Female
Drew	No	Blue	Long	Female
Alberto	Yes	Brown	Short	Male
Karin	No	Blue	Long	Female
Nina	Yes	Brown	Short	Female
Sergio	Yes	Blue	Long	Male

To simplify the task, **naïve Bayesian classifiers** assume attributes have independent distributions, and thereby estimate

$$p(d|c_j) = p(d_1|c_j) * p(d_2|c_j) * \dots * p(d_n|c_j)$$

↑
The probability of class c_j generating instance d , equals....

↑
The probability of class c_j generating the observed value for feature 1, multiplied by..

↑
The probability of class c_j generating the observed value for feature 2, multiplied by..

To simplify the task, **naïve Bayesian classifiers** assume attributes have independent distributions, and thereby estimate

$$p(d|c_j) = p(d_1|c_j) * p(d_2|c_j) * \dots * p(d_n|c_j)$$

$$p(\text{officer drew}|c_j) = p(\text{over_170}_{\text{cm}} = \text{yes}|c_j) * p(\text{eye} = \text{blue}|c_j) * \dots$$



Officer Drew
is blue-
eyed, over
170_{cm} tall,
and has
long hair

$$p(\text{officer drew} | \text{Female}) = 2/5 * 3/5 * \dots$$

$$p(\text{officer drew} | \text{Male}) = 2/3 * 2/3 * \dots$$

Naïve Bayes Classifier

- If one of the conditional probability is zero, then the entire expression becomes zero
- Probability estimation:

$$\text{Original : } P(A_i | C) = \frac{N_{ic}}{N_c}$$

c: number of classes

p: prior probability

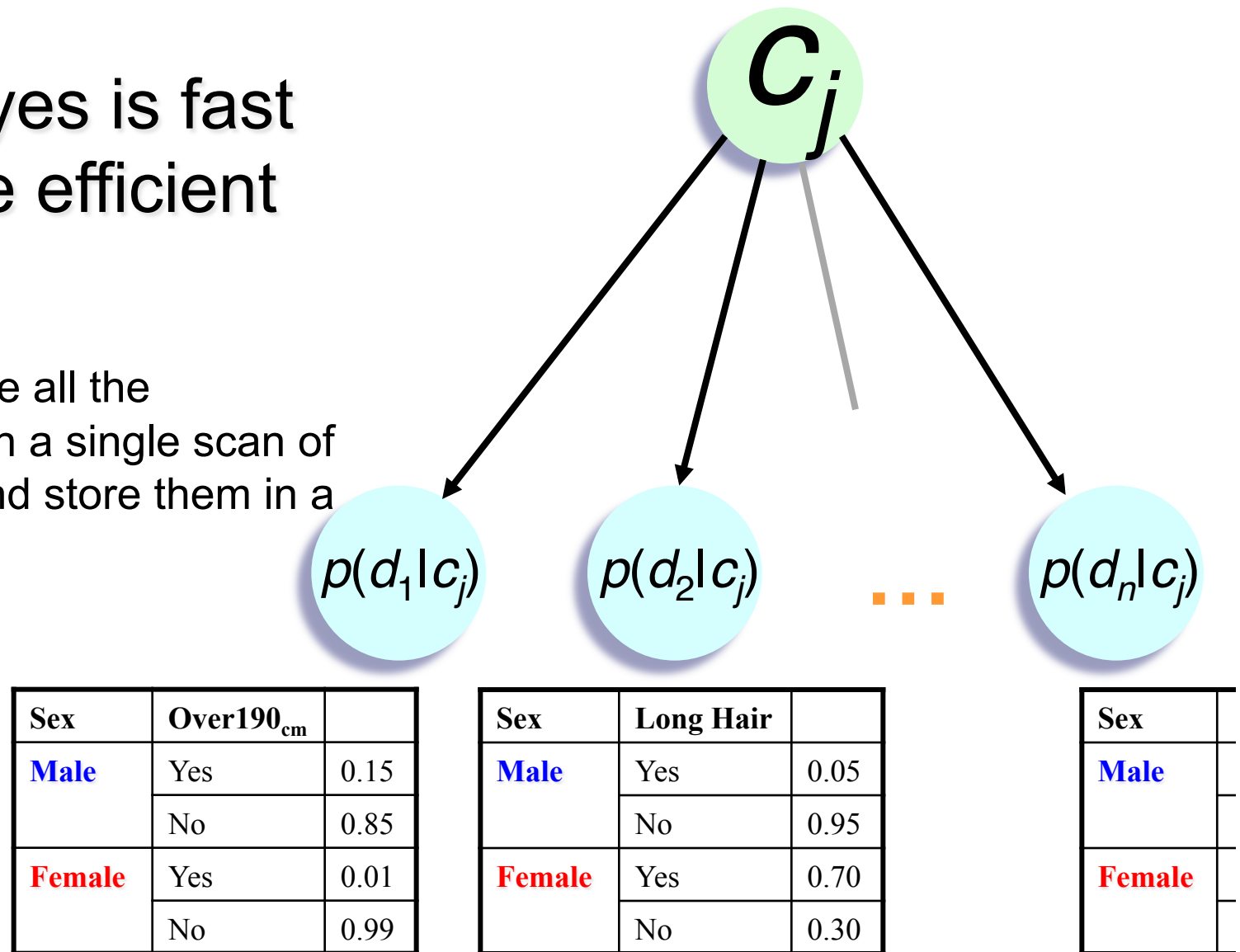
$$\text{Laplace : } P(A_i | C) = \frac{N_{ic} + 1}{N_c + c}$$

m: parameter

$$\text{m - estimate : } P(A_i | C) = \frac{N_{ic} + mp}{N_c + m}$$

Naïve Bayes is fast and space efficient

We can compute all the probabilities with a single scan of the database and store them in a (small) table...



■ Naïve Bayes is NOT sensitive to irrelevant features...

■ Suppose we are trying to classify a person's gender based on several features, including eye color. (Of course, eye color is completely irrelevant to a person's gender)

$$p(\text{Jessica} | c_j) = p(\text{eye} = \text{brown} | c_j) * p(\text{wears_dress} = \text{yes} | c_j) * \dots$$

$$p(\text{Jessica} | \text{Female}) = 9,000/10,000 * 9,975/10,000 * \dots$$

$$p(\text{Jessica} | \text{Male}) = 9,001/10,000 * 2/10,000 * \dots$$

Almost the same!



However, this assumes that we have good enough estimates of the probabilities, so the more data the better.

Advantages/Disadvantages of Naïve Bayes

■ Advantages:

- ★ Fast to train (single scan). Fast to classify
- ★ Not sensitive to irrelevant features
- ★ Not sensitive to isolated noise data
- ★ Handles real and discrete data
- ★ Handles streaming data well
- ★ Handles missing values by ignoring the object during probability estimate computation

■ Disadvantages:

- ★ Assumes independence of features