GENERAL ASSEMBLY

INTRO TO DATA SCIENCE

LECTURE 9: CLUSTERING & DECISION TREE CLASSIFIERS

YUCHEN ZHAO / DAT-14

I. CLUSTER ANALYSIS
II. K-MEANS CLUSTERING
III. INTERPRETING RESULTS

IV. LAB: K-MEANS CLUSTERING (PART 1)

I. CLUSTERING SUMMARY

II. DECISION TREES

III. BUILDING DECISION TREES

LAB:

IV. HOMEWORK 2 REVIEW

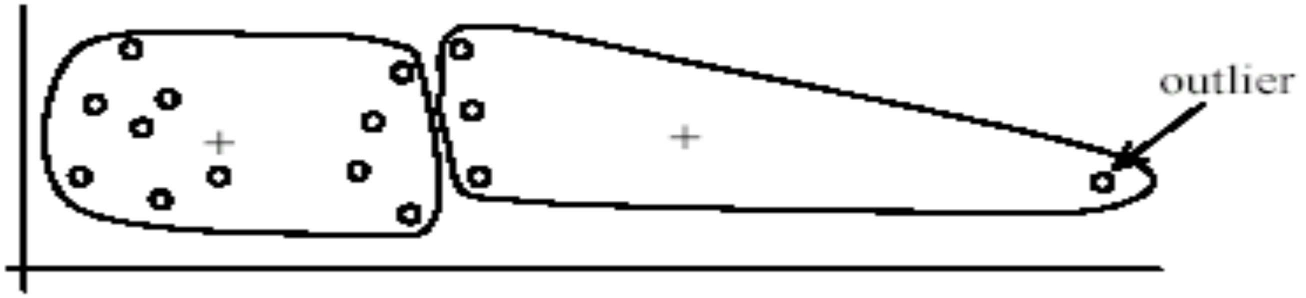
V. K-MEANS CLUSTERING (PART 2)

K-Means Clustering

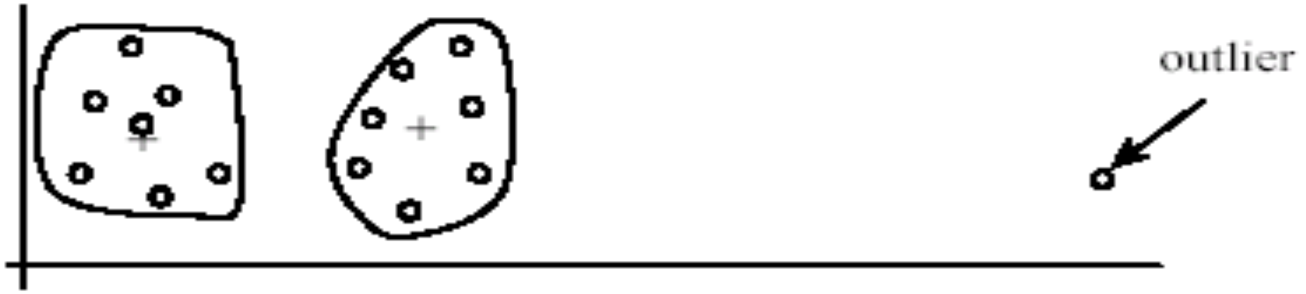
K-Means is the most popular clustering algorithm.

Source: http://www.cs.uic.edu/~liub/teach/cs583-fall-05/CS583-unsupervised-learning.ppt but sensitive to outliers…

K-Means Clustering



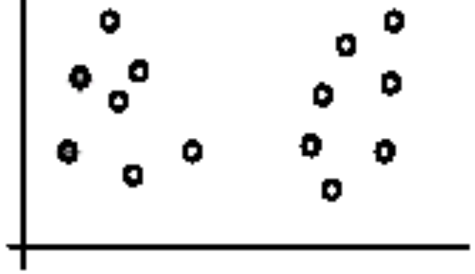
(A): Undesirable clusters

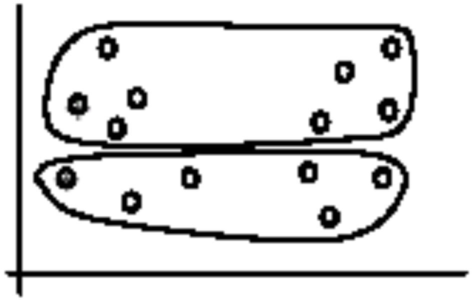


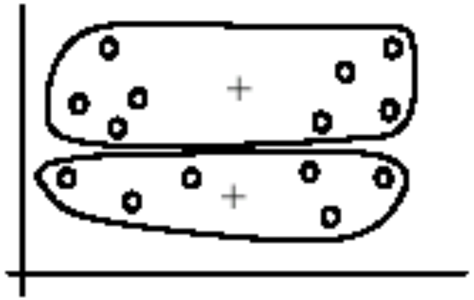
(B): Ideal clusters

also sensitive to initial seeds

K-Means Clustering

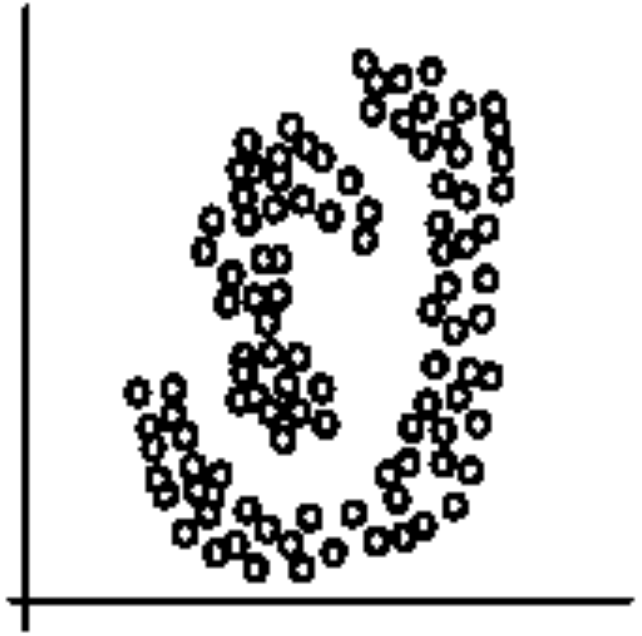

(A). Random selection of seeds (centroids)

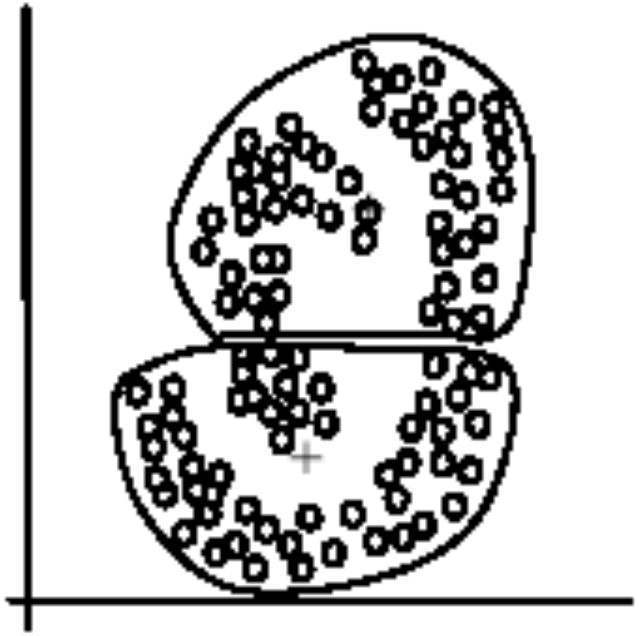

(B). Iteration 1


(C). Iteration 2

and not suitable for discovering
non-hyper-sphere clusters

K-Means Clustering


(A): Two natural clusters


(B): *k*-means clusters

No clear evidence that any other clustering algorithm performs better in general

How to choose a clustering algorithm?

choosing the best algorithm
is a challenge

Every algorithm has limitations and works well with certain data distributions.

In practice, It is very hard, if not impossible, to know what distribution the application data follow.

The common practice is to…

Run several algorithms using different distance functions and parameter settings

then carefully
analyze and compare the results

Clustering is highly **application dependent** and to certain extent **subjective** (personal preferences).

Clustering in Practice:
learning machine data

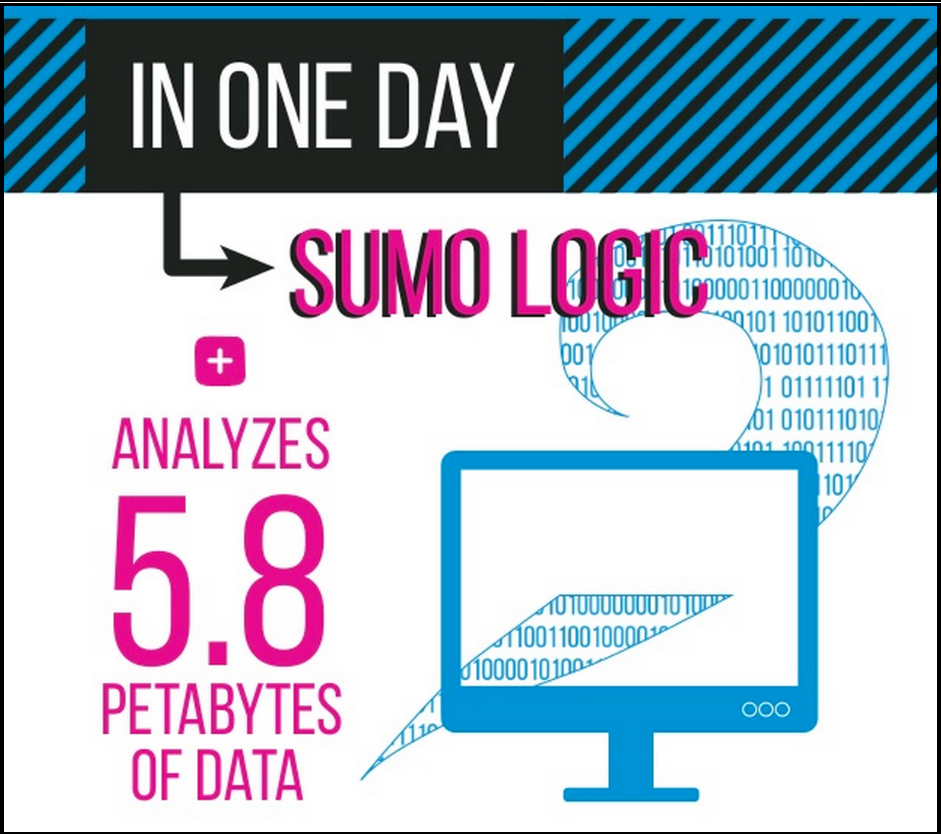
what is machine data?

Clustering Application

http://www.splunk.com/en_us/resources/machine-data.html

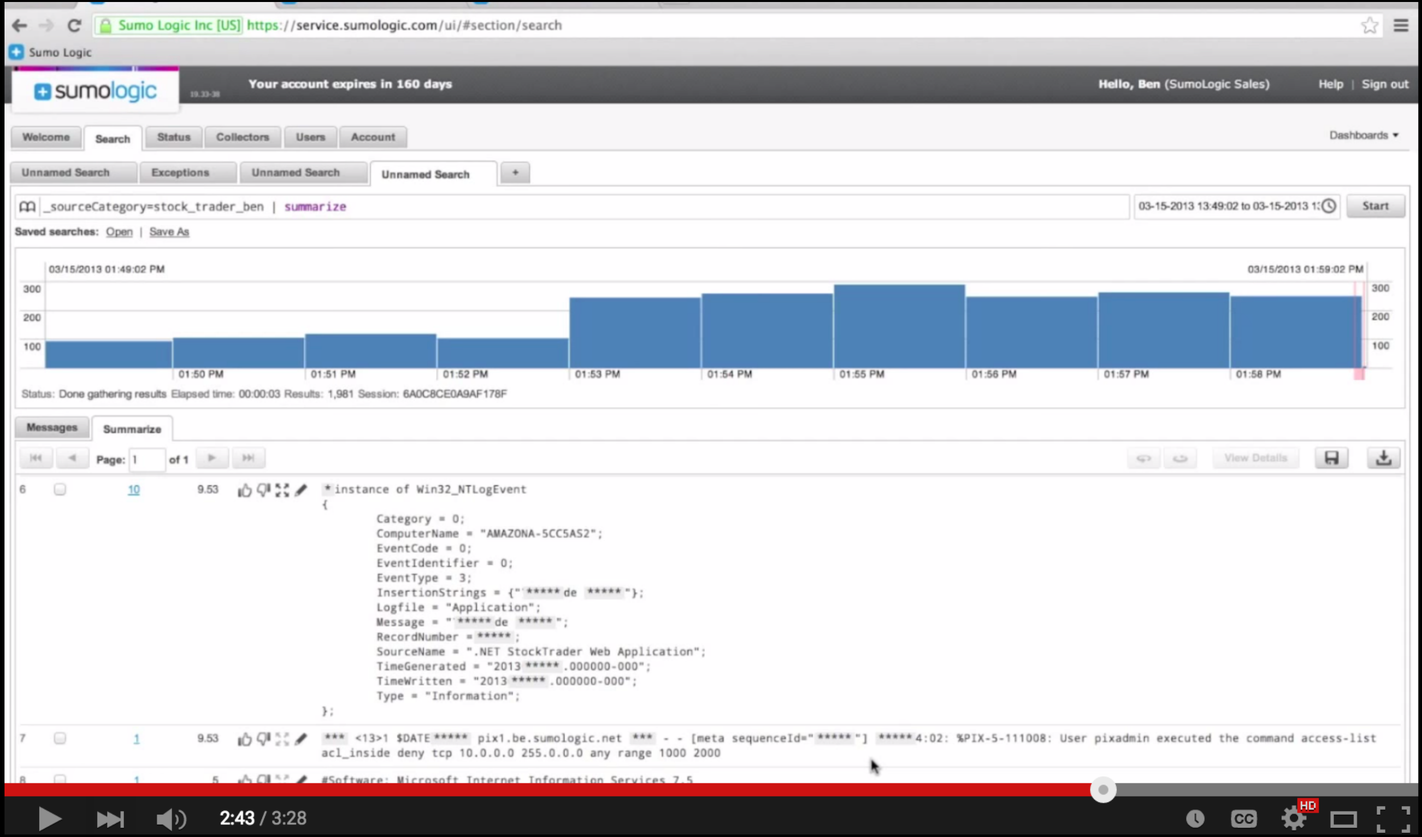
what is the size of
machine data?

Clustering Application



http://www.slideshare.net/Sumo_Logic/sumo-logic-datainoneday

Using Clustering - Log Reduce



```
_sourceCategory=stock_trader_ben | summarize
```

03-15-2013 13:49:02 to 03-15-2013 1

Status: Done gathering results Elapsed time: 00:00:03 Results: 1,981 Session: 6A0C8CE0A9AF178F

```
6   10   9.53   *instance of Win32_NTLogEvent
                {
                        Category = 0;
                        ComputerName = "AMAZONA-5CC5AS2";
                        EventCode = 0;
                        EventIdentifier = 0;
                        EventType = 3;
                        InsertionStrings = {"*****de *****"};
                        Logfile = "Application";
                        Message = "*****de *****";
                        RecordNumber = *****;
                        SourceName = ".NET StockTrader Web Application";
                        TimeGenerated = "2013*****.000000-000";
                        TimeWritten = "2013*****.000000-000";
                        Type = "Information";
                };
7   1    9.53   *** <13>1 $DATE***** pix1.be.sumologic.net *** - - [meta sequenceId="*****"] *****4:02: %PIX-5-111008: User pixadmin executed the command access-list
                acl_inside deny tcp 10.0.0.0 255.0.0.0 any range 1000 2000
8   1    5      #Software: Microsoft Internet Information Services 7.5
```

https://www.youtube.com/watch?v=ZF0zbCu9aps

II. DECISION TREES

Q: What is a decision tree?

Q: What is a decision tree?

A: A non-parametric hierarchical classification technique.

Q: What is a decision tree?

A: A non-parametric hierarchical classification technique.

non-parametric*: no parameters, no distribution assumptions*

Q: What is a decision tree?

A: A non-parametric hierarchical classification technique.

non-parametric*: no parameters, no distribution assumptions*

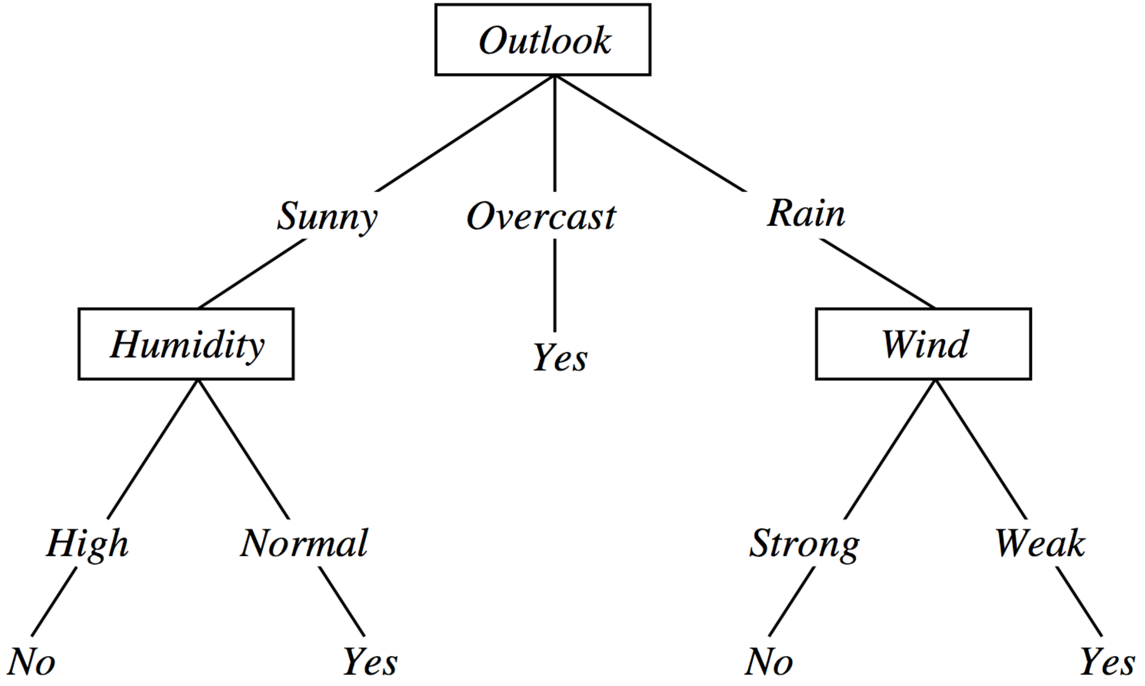
hierarchical*: consists of a sequence of questions which yield a class label when applied to any record*

Q: How is a decision tree represented?

Q: How is a decision tree represented?

A: Using a configuration of **nodes** *and* **edges***.*

DECISION TREE CLASSIFIERS



Classify an instance: <outlook=Sunny, temp = Hot, humidity=High, wind = Strong>

Q: How is a decision tree represented?

A: Using a configuration of **nodes** *and* **edges***.*

More concretely, as a multiway tree, which is a type of (directed acyclic) **graph***.*

Q: How is a decision tree represented?
A: Using a configuration of **nodes** *and* **edges***.*

More concretely, as a multiway tree, which is a type of (directed acyclic) **graph***.*

*In a decision tree, the nodes represent questions (***test conditions***) and the edges are the answers to these questions.*

TYPES OF NODES

The top node of the tree is called the **root node**. *This node has 0 incoming edges, and 2+ outgoing edges.*

TYPES OF NODES

The top node of the tree is called the **root node**. *This node has 0 incoming edges, and 2+ outgoing edges.*

An **internal node** *has 1 incoming edge, and 2+ outgoing edges. Internal nodes represent test conditions.*

TYPES OF NODES

The top node of the tree is called the **root node**. *This node has 0 incoming edges, and 2+ outgoing edges.*

An **internal node** *has 1 incoming edge, and 2+ outgoing edges. Internal nodes represent test conditions.*

A **leaf node** *has 1 incoming edge and, 0 outgoing edges. Leaf nodes correspond to class labels.*

The top node of the tree is called the **root node**. *This node has 0 incoming edges, and 2+ outgoing edges.*

An **internal node** *has 1 incoming edge, and 2+ outgoing edges. Internal nodes represent test conditions.*

A **leaf node** *has 1 incoming edge and, 0 outgoing edges. L correspond to class labels.*

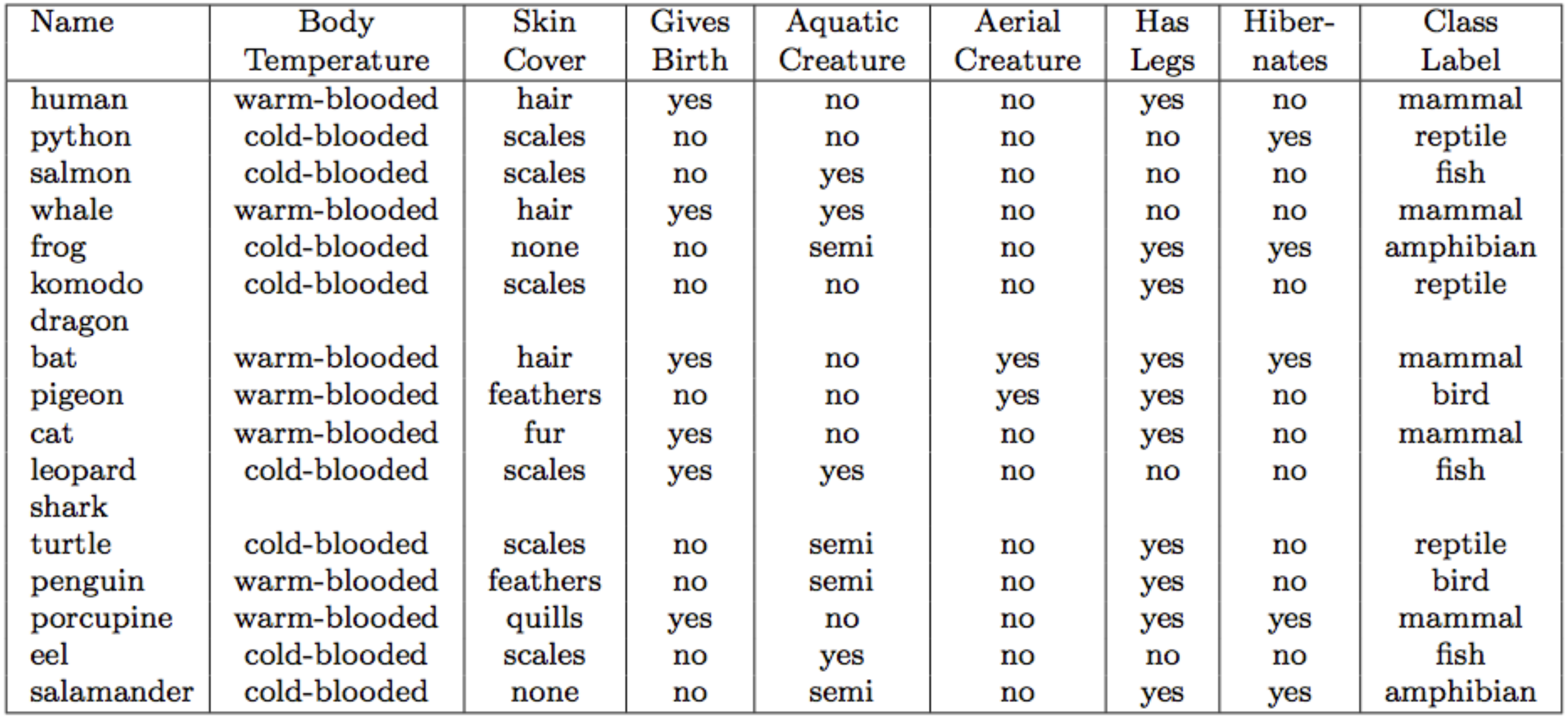
NOTE

The nodes in our tree are connected by directed edges.

These directed edges lead from parent nodes to child nodes.

EXAMPLE – DATA

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

source: http://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf

EXAMPLE – DECISION TREE

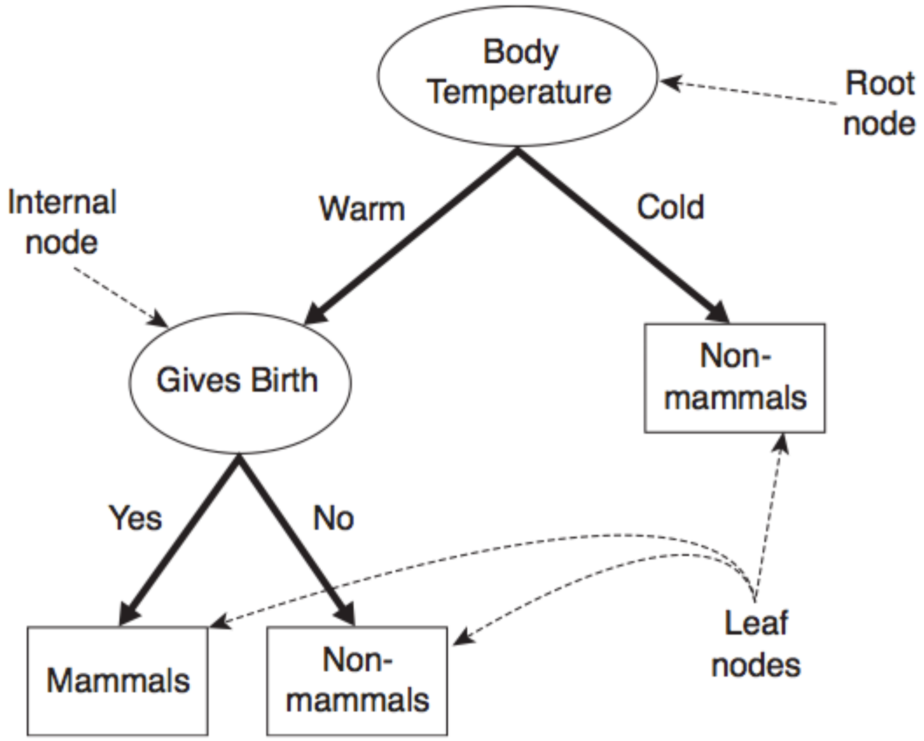


Figure 4.4. A decision tree for the mammal classification problem.

source: http://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf

EXAMPLE – DECISION TREE

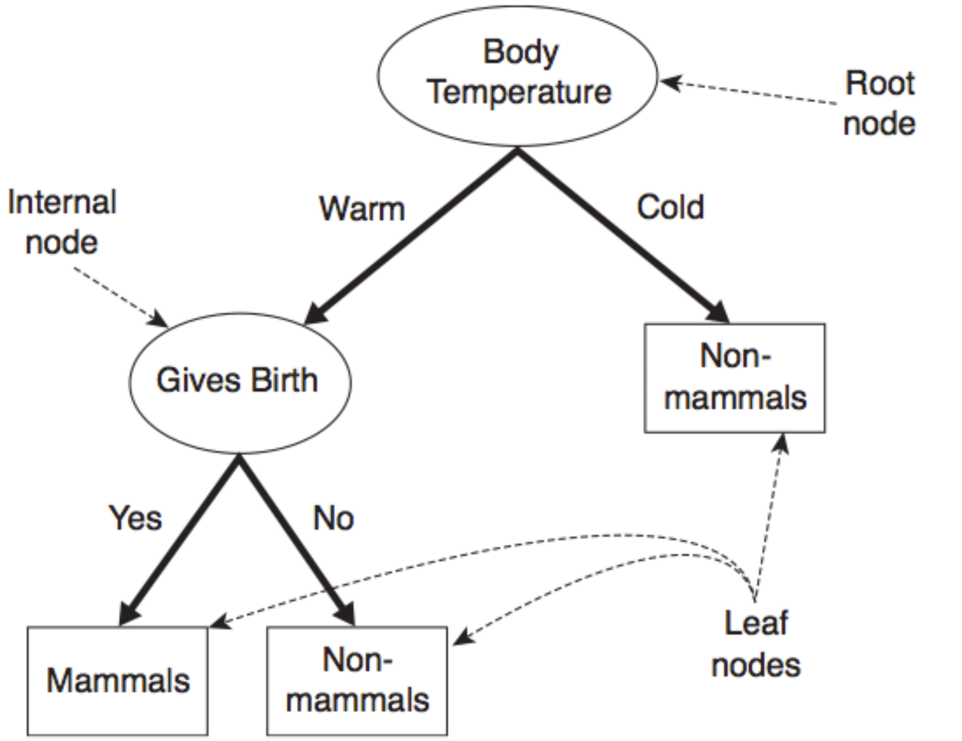


Figure 4.4. A decision tree for the mammal classification problem.

source: http://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf

NOTE

Internal nodes represent test conditions which partition the records at that node.

III. BUILDING DECISION TREES

Q: How do we build a decision tree?

Q: How do we build a decision tree?

A: One possibility would be to evaluate all possible decision trees (eg, all permutations of test conditions) for a given dataset.

Q: How do we build a decision tree?

A: One possibility would be to evaluate all possible decision trees (eg, all permutations of test conditions) for a given dataset.

But this is generally too complex to be practical → $O(2^n)$.

BUILDING A DECISION TREE

Q: How do we build a decision tree?
A: One possibility would be to evaluate all possible decision trees (eg, all permutations of test conditions) for a given dataset.

But this is generally too complex to be practical → $O(2^n)$.

Q: How do we find a practical solution that works?

BUILDING A DECISION TREE

Q: How do we build a decision tree?

A: One possibility would be to evaluate all possible decision trees (eg, all permutations of test conditions) for a given dataset.

But this is generally too complex to be practical → $O(2^n)$.

Q: How do we find a practical solution that works?

A: Use a **heuristic** *algorithm.*

The basic method used to build (or "grow") a decision tree is **Hunt's algorithm**.

The basic method used to build (or "grow") a decision tree is **Hunt's algorithm***.*

This is a **greedy recursive** *algorithm that leads to a* **local optimum***.*

The basic method used to build (or "grow") a decision tree is **Hunt's algorithm***.*

This is a **greedy recursive** *algorithm that leads to a* **local optimum***.*

greedy – *algorithm makes locally optimal decision at each step*
recursive – *splits task into subtasks, solves each the same way*
local optimum – *solution for a given neighborhood of points*

Hunt's algorithm builds a decision tree by recursively partitioning records into smaller & smaller subsets.

BUILDING A DECISION TREE

Hunt's algorithm builds a decision tree by recursively partitioning records into smaller & smaller subsets.

The partitioning decision is made at each node according to a metric called **purity**.

BUILDING A DECISION TREE

Hunt's algorithm builds a decision tree by recursively partitioning records into smaller & smaller subsets.

The partitioning decision is made at each node according to a metric called **purity**.

A partition is 100% pure when all of its records belong to a single class.

Consider a binary classification problem with classes X, Y. Given a set of records D_t at node t, Hunt's algorithm proceeds as follows:

Consider a binary classification problem with classes X, Y. Given a set of records D_t at node t, Hunt's algorithm proceeds as follows:

1) If all records in D_t belong to class X, then t is a leaf node corresponding to class X.

Consider a binary classification problem with classes X, Y. Given a set of records D_t at node t, Hunt's algorithm proceeds as follows:

1) If all records in D_t belong to class X, then t is a leaf node corresponding to class X.

NOTE

This is the base case for the recursive algorithm.

Consider a binary classification problem with classes X, Y. Given a set of records D_t at node t, Hunt's algorithm proceeds as follows:

2) If D_t contains records from both classes, then a test condition is created to partition the records further. In this case, t is an internal node whose outgoing edges correspond to the possible outcomes of this test condition.

Consider a binary classification problem with classes X, Y. Given a set of records D_t at node t, Hunt's algorithm proceeds as follows:

2) If D_t contains records from both classes, then a test condition is created to partition the records further. In this case, t is an internal node whose outgoing edges correspond to the possible outcomes of this test condition.

These outgoing edges terminate in **child nodes**. *A record d in D_t is assigned to one of these child nodes based on the outcome of the test condition applied to d.*

BUILDING A DECISION TREE

Consider a binary classification problem with classes X, Y. Given a set of records D_t at node t, Hunt's algorithm proceeds as follows:

3) These steps are then recursively applied to each child node.

BUILDING A DECISION TREE

Consider a binary classification problem with classes X, Y. Given a set of records D_t at node t, Hunt's algorithm proceeds as follows:

3) These steps are then recursively applied to each child node.

NOTE

Decision trees are easy to interpret, but the algorithms to create them are a bit complicated.

Q: How do we partition the training records?

Q: How do we partition the training records?

A: There are a few ways to do this.

Q: How do we partition the training records?
A: There are a few ways to do this.

Test conditions can create **binary splits***:*

Q: How do we partition the training records?

A: There are a few ways to do this.

Test conditions can create **binary splits***:*

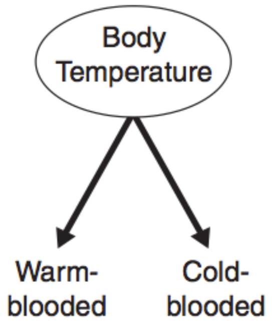
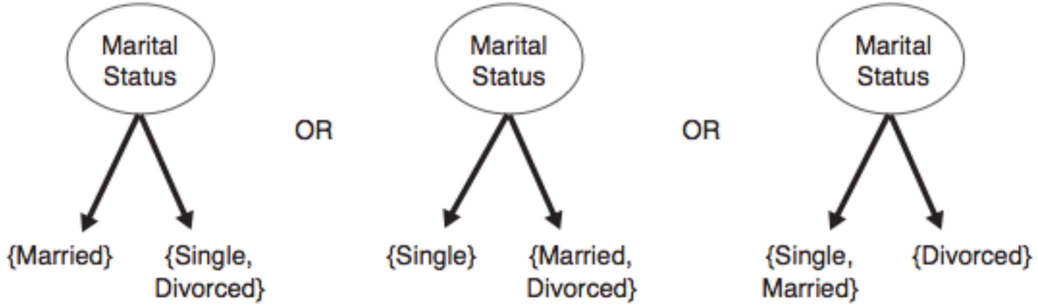


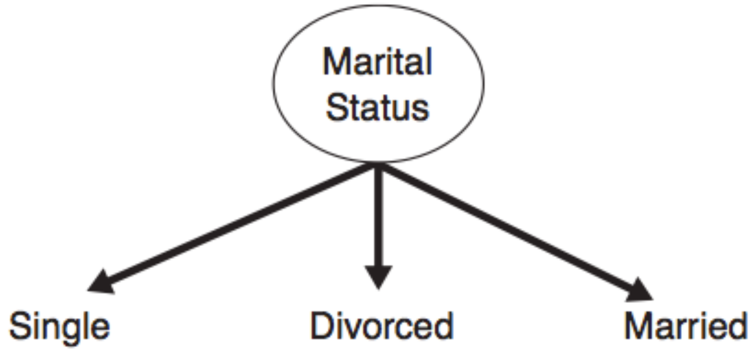
Figure 4.8. Test condition for binary attributes.



(b) Binary split {by grouping attribute values}

Q: How do we partition the training records?
A: There are a few ways to do this.

Alternatively, we can create **multiway splits***:*

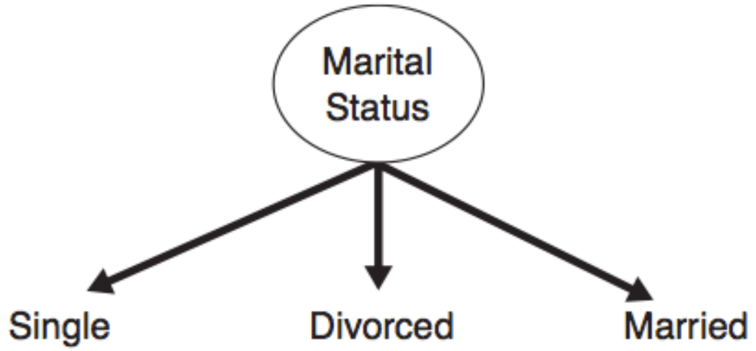


(a) Multiway split

Q: How do we partition the training records?

A: There are a few ways to do this.

Alternatively, we can create **multiway splits***:*



(a) Multiway split

NOTE

Multiway splits can produce purer subsets, but may lead to overfitting!

Q: How do we partition the training records?

A: There are a few ways to do this.

For continuous features, we can use either method:

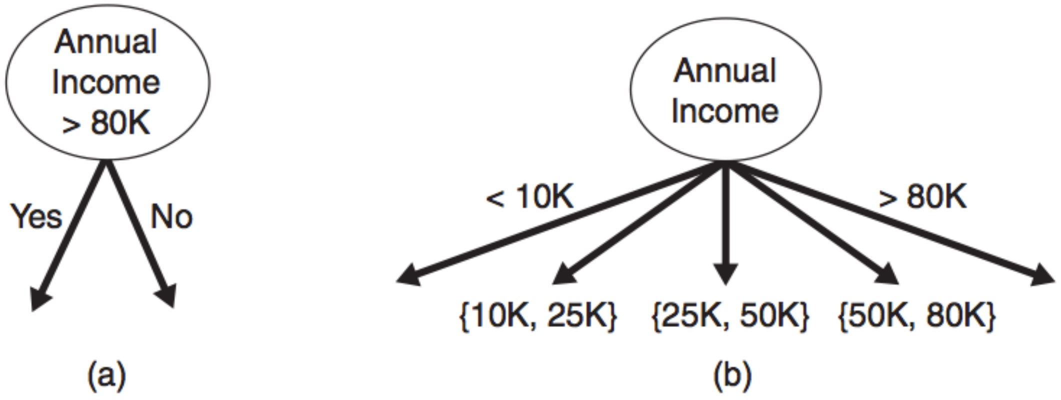


Figure 4.11. Test condition for continuous attributes.

Q: How do we partition the training records?

A: There are a few ways to do this.

For continuous features, we can use either method:

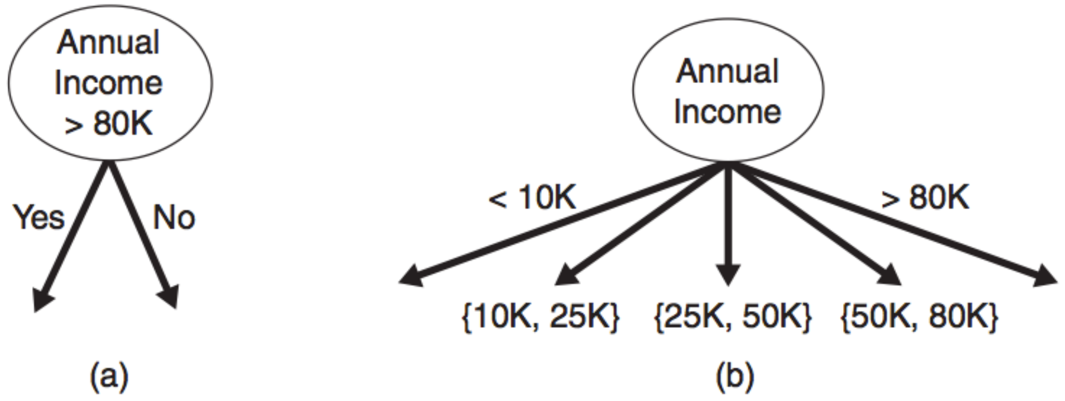


Figure 4.11. Test condition for continuous attributes.

NOTE

There are optimizations that can improve the naïve quadratic complexity of determining the optimum split point for continuous atrributes.

Q: How do we determine the best split?

Q: How do we determine the best split?

A: Recall that no split is necessary (at a given node) when all records belong to the same class.

CREATING PARTITIONS

Q: How do we determine the best split?

A: Recall that no split is necessary (at a given node) when all records belong to the same class.

Therefore we want each step to create the partition with the highest possible purity.

CREATING PARTITIONS

Q: How do we determine the best split?

A: Recall that no split is necessary (at a given node) when all records belong to the same class.

Therefore we want each step to create the partition with the highest possible purity.

We need an objective function to optimize!

We’ll discuss various objective functions in the next lecture

LAB:
HOMEWORK 2 REVIEW
K-MEANS CLUSTERING (PART 2)

DISCUSSION