Data Mining Tutorial

Mark A. Austin

University of Maryland

austin@umd.edu

ENCE 688P, Fall Semester 2021

October 16, 2021

Overview

Quick Review

Artificial Intelligence (AI) and Machine Learning (ML)

Technical Implementation (2020, Google, Siemens, IBM)

- AI and ML will be deeply embedded in new software and algorithms.

Artificial Intelligence:

- Knowledge representation and reasoning with ontologies and rules. Semantic graphs. Executable event-based processing.

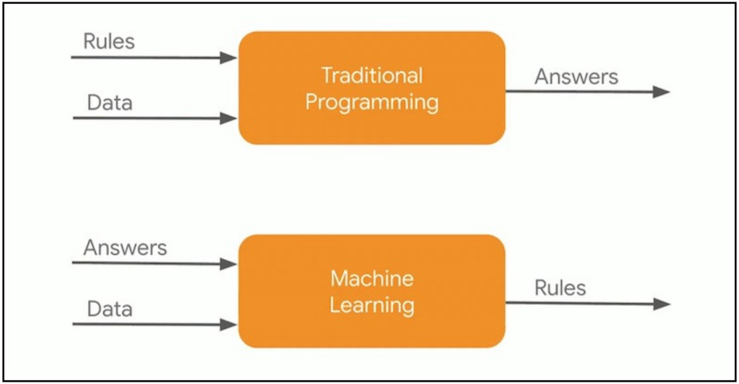
Machine Learning:

- Modern neural networks. Input-to-output prediction.
- Data mining.
- Identify objects, events, and anomalies.
- Learn structure and sequence. Remember stuff.

Man and Machine (AI-ML View)

Man	AI-ML Machine
• Good at formulating solutions to problems. • Can work with incomplete data and information. • Creative. • Reasons logically, but very slow. Forgetful. • Performance is static. • Humans make the rules, then they break them.	• Manipulates Os and 1s. • Can work with incomplete data and information. • Creative. • Fast logical reasoning. • Performance doubles every 18-24 months. • Data mining can discover the rules.

Traditional Programming vs AI-ML Workflow



Introduction to Data Mining

Numerous Definitions

Data Mining

The field of data mining addresses the question of how to best use historical data to discover general regularities and improve future decisions (Mitchell, 1999).

Data Mining

Data mining is the extraction of implicit, previously unknown, and potentially useful information – structural patterns – from data (Witten et al., 2017).

The process of discovering useful patterns from data must be automatic (or at least semi-automatic). Useful patterns allow us to make nontrivial predictions on new data.

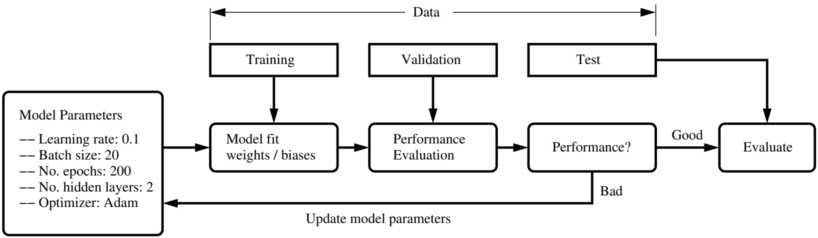
Data Mining Techniques

Working with Initial Dataset

- Data cleaning and curation
- Remove redundant features
- Identify input variables and output variable.

Preprocessed Dataset:

- Data split: 80% for training, 20% for validation and testing.



Data Mining Techniques

Training Dataset

- The sample of data used to fit the model.

Validation Dataset

- The sample of data used to provide an unbiased evaluation of the model fit on the training dataset while training the model parameters.

Testing Dataset

- The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.

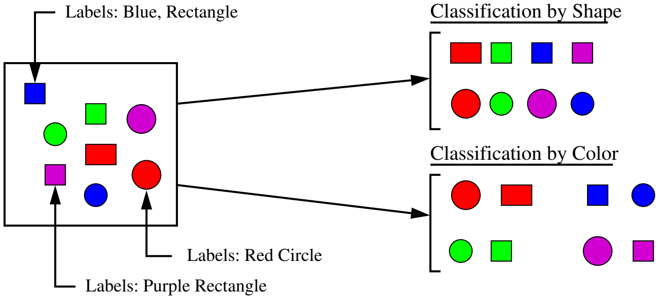
Data Mining Techniques

Data Mining Techniques

Classification Analysis

Classification analysis learns a method for predicting the instance class from pre-labeled (classified) instances.

Classification by Shape/Color (Supervised Learning)



Data Mining Techniques

Classification Problem

- **Given** a set of n attributes (ordinal or categorical), a set of k classes, and a set of labeled training instances,

$$[(i_i, l_i), \cdots, (i_j, l_j)], \tag{1}$$

  where $i = (v_1, v_2, \cdots, v_n)$,
  and $l \in (c_1, c_2, \cdots, c_k)$.
- **Goal** is to determine a classification rule – sequence of tests on the attributes – that predicts the class of any instance from the values of its attributes.

Note

- This is a generalization of the concept learning problem since typically there are more than two (outcome) classes.
- Data will contain scatter; may have missing values.

Data Mining Techniques

Decision Trees.

A structure that includes a root node, branches, and leaf nodes. Each internal node represents a test on an attribute; each branch represents the outcome of a test; and each leaf represents a class label.

Arbitrary Boolean Functions

- Each attribute is binary valued (true or false).
- Example trees: XOR, AND and OR, etc ...

Continuous Domains

- Each attribute is real valued (true or false).
- Tests check if $a_i >$ value.

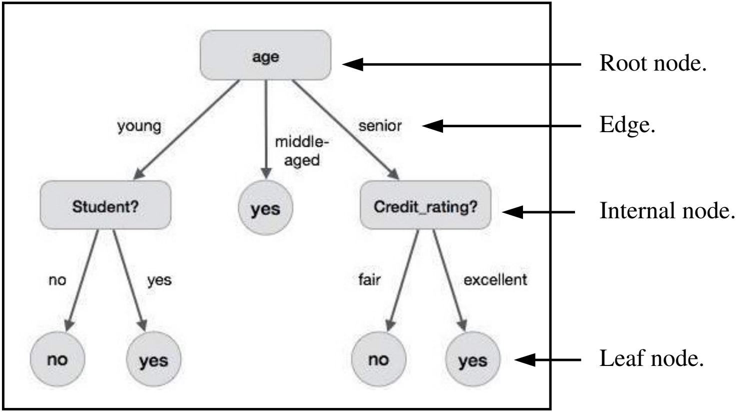
Data Mining Techniques

Sample Dataset. Will customer buy a computer?

ID	Age Group	Income	Student	Credit Rating	Buys Computer
1	young	high	no	fair	no
2	young	high	no	excellent	no
3	middle	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle	low	yes	excellent	yes
8	young	medium	no	fair	no
9	young	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	young	medium	yes	excellent	yes
12	middle	medium	no	excellent	yes
13	middle	high	yes	fair	yes
14	senior	medium	no	excellent	no

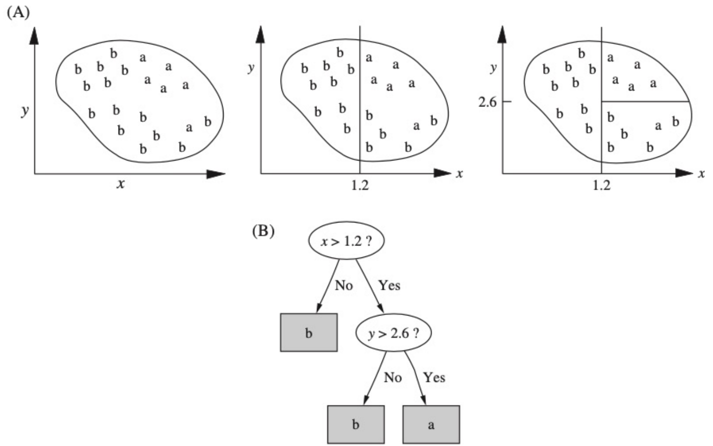
Data Mining Techniques

Sample Decision Tree (Split on Discrete Domain)



Data Mining Techniques

Covering Algorithm and Rule Construction (Split on Continuous Domain)

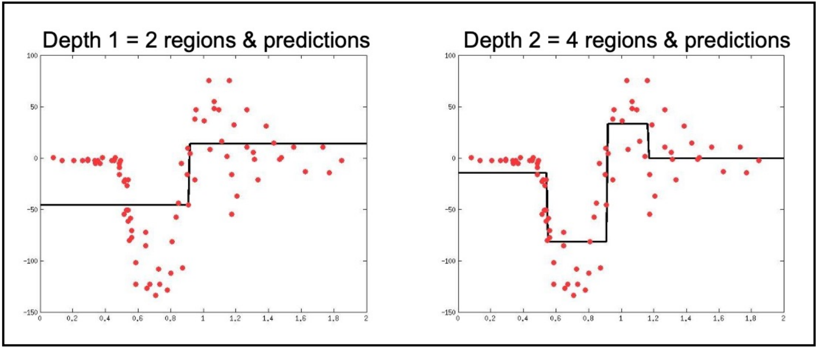


Data Mining Techniques

Decision Trees for Regression (One-Dimensional Regression)

- Goal is to predict real-valued numbers at the leaf nodes.

Prediction of a Single Scalar Feature

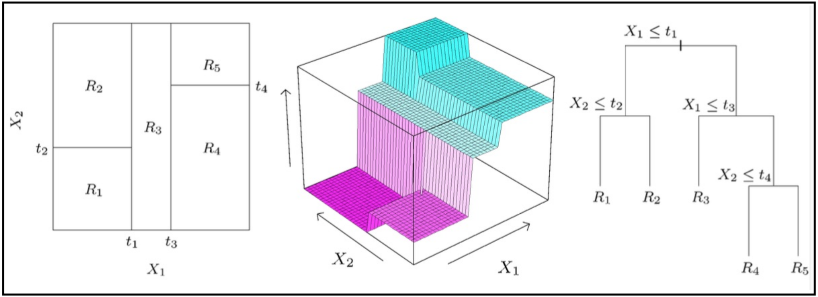


Data Mining Techniques

Decision Trees for Regression (Two-Dimensional Regression)

- Each node splits tree according to a single feature.
- Mean values of training data are predicted at leaf nodes.

Example



Data Mining Techniques

Basic Questions:

- How to choose the attribute (or value) to split on at each level of the tree?
- When should a node be declared a leaf?
- If a leaf is impure, how should it be labeled?
- If the tree is too large, how can it be pruned?

Notes on Strategy:

- When all of the data in a single node comes from the same class, can declare the node to be a leaf and stop splitting.
- When a group of data points have exactly the same attribute values, we cannot split any further. Declare the node to be a leaf, and output the class that is the majority.

Data Mining Techniques

Algorithms

- Perceptron.
- Logistic Regression.
- Decision tree algorithms (C4.5, J48)
- Support Vector Machines (SVM).
- Random Forest.

Applications

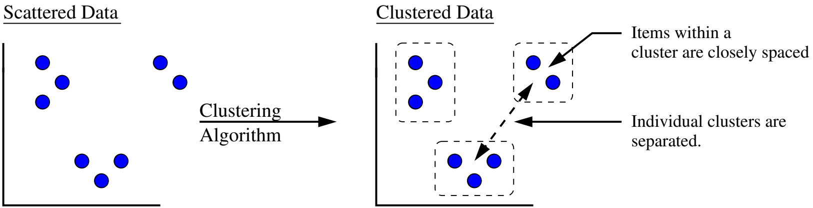
- Anomaly (Fraud) detection.
- Medical diagnosis.
- Industrial applications.

Data Mining Techniques

Clustering Problems

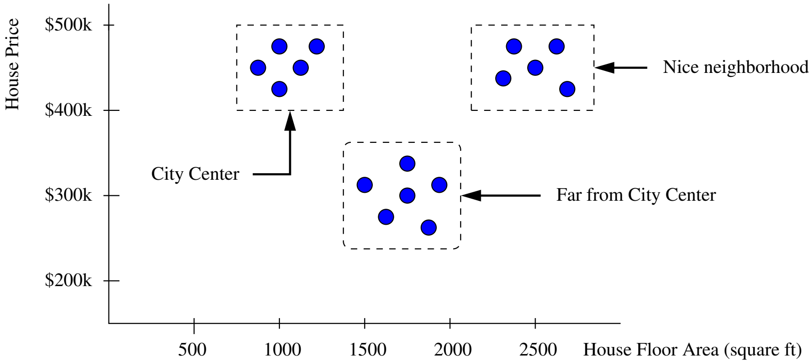
Clustering techniques apply when there is no class to be predicted, but when un-labeled instances need to be divided into common natural groups.

Clustering Process (Unsupervised Learning)



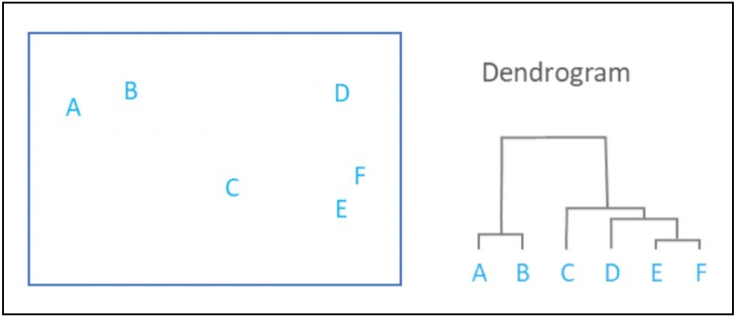
Data Mining Techniques

Example 1. Clustering of House Prices and Floor Areas



Data Mining Techniques

Example 2. Hierarchical Clustering and Dendrograms



Dendrogram

A dendrogram is a branching (tree) diagram that represents relationships of similarity among groups of entities.

Data Mining Techniques

Algorithms

- K-means clustering.
- Hierarchical clustering.

Applications

- Preprocessing step for many scientific applications.
- Natural language processing.
- Market segmentation.
- Netflix/movie recommendations.

Data Mining Techniques

Association

Association is a data mining function that discovers the probability the co-occurrence of items (or patterns) in a collection of data.

Association Rules

- Identify relationships between co-occurring items can be expressed as association rules (e.g., if X, then Y).

Key Challenges

- How to identify useful correlations among all correlations?
- Correlation relationships are not the same as dependency relationships – *if X, then Y* does not *imply if Y, then X* !
- Historical data does not necessarily predict the future.

Data Mining Techniques

Goals of Predictive Analysis

- For a customer who purchases product A, what other products will they purchase?
- Will coupons increase same-store sales?
- Will a reduced price mean higher sales?

Retail Strategies

- Put most frequently purchased item (e.g., milk) at the back of the store.
- Co-locate items that are bought together – can lead to increase in sales for both.

Data Mining Techniques

Example 1. iPhone Color and Personality Traits.

Phone Color	Personality Traits
Green	Fresh, harmonious, healthy, hopeful.
Blue	Confident, dependable, trustworthy.
Yellow	Happy, honorable, intelligent.
Pink	Compassionate, energetic, playful.
White	Balanced, calm, clean.

Customers want to select an iPhone Color that correlates with their personality traits.

Data Mining Techniques

Example 2. Urban Legend from early 1990s: Diapers and Beer

ID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}
...	...

market basket transactions

Examples of Association Rules

- $\{Diapers\} \longrightarrow \{Beer\}$,
- $\{Milk, Bread\} \longrightarrow \{Eggs, Coke\}$,
- $\{Beer, Bread\} \longrightarrow \{Milk\}$.

Data Mining Techniques

Itemset and **k-Itemset**

- A collection of one or more items (e.g., $\{Milk, Bread\}$.
- k-Itemset is an itemset containing k items.

Support Count σ

- Frequency of ocurrence of an itemset.
- Example: $\sigma(\{Milk, Bread, Diaper\}) = 2$.

Support

- Indicates how frequently the if/then relationship appears in the data.

Association Rule

- Expression of the form X $\longrightarrow$ Y, where X and Y are itemsets.

Data Mining Techniques (Rule Evaluation Metrics)

Support (s)

- Fraction of transactions that contain both X and Y.
- Support(s) $= \frac{\sigma\{Milk, Diaper, Beer\}}{T} = 2/5 = 0.4.$

Confidence (c)

- Measures how often items in Y appear in transactions that contain X.
- Confidence(c) $= \frac{\{Milk, Diaper, Beer\}}{\{Milk, Diaper\}} = 2/3 = 0.67.$

Data Mining for Association Rules

Given a set of transactions T, find all rules having:

- Support(s) $\geq$ min support threshhold.
- Confidence(c) $\geq$ min confidence threshold.

Data Mining Techniques (Brute-Force Enumeration)

Brute-Force Enumeration

- Compute support and confidence for all possible association rules.
- Prune rules that do not meet min support/confidence thresholds.

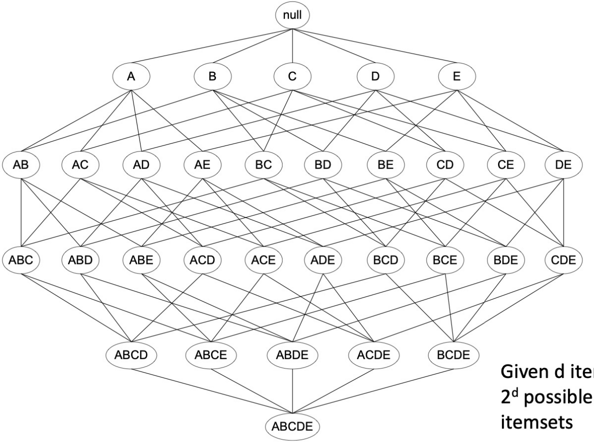
TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Rules:

{Milk,Diaper} → {Beer} (s=0.4, c=0.67)
{Milk,Beer} → {Diaper} (s=0.4, c=1.0)
{Diaper,Beer} → {Milk} (s=0.4, c=0.67)
{Beer} → {Milk,Diaper} (s=0.4, c=0.67)
{Diaper} → {Milk,Beer} (s=0.4, c=0.5)
{Milk} → {Diaper,Beer} (s=0.4, c=0.5)

Data Mining Techniques (Brute-Force Enumeration)

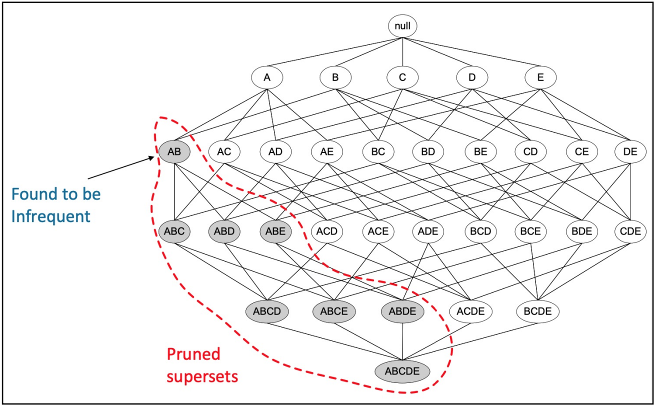
Computational Complexity: Given d items, there are 2^d possible candidate itemsets.



Given d items, there are 2^d possible candidate itemsets

Data Mining Techniques (Brute-Force Enumeration)

Need strategies to reduce computational effort by systematically pruning the low scoring items from candidate space.



Data Mining Techniques

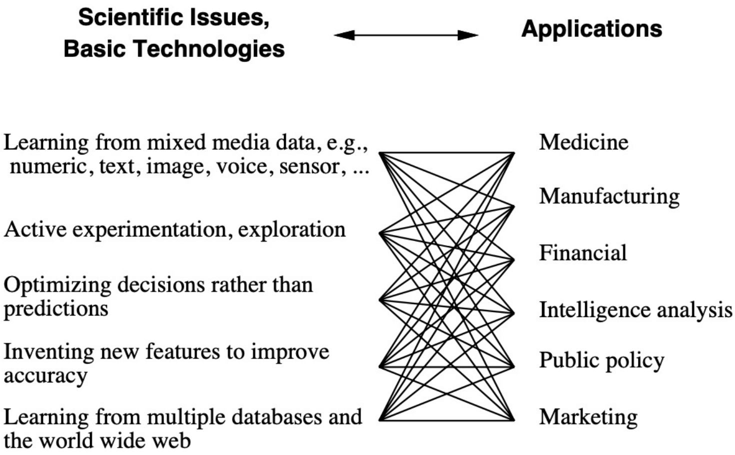
Algorithms (see Chapter 6 of Witten et al.)

- **Apriori**: Follows a generate-and-test methodology for finding frequent item sets, generating successively longer candidate item sets, and then scanning the item sets to see if they meet threshold limits.
- **Frequent Pattern Trees**: Begins by counting the number of times individual items – attribute-value pairs – occur in the dataset. This is a single pass. Then, a (sorted) tree structure is constructed with the goal of identifying large (frequent) item sets.

Applications

- Weather prediction,
- Medical diagnosis,
- Purchasing habits of retail customers.

Scientific Research Enabling Applications



Source: Mitchell, 1999.

Entropy

(Quantitative Measure of Uncertainty)

Definition

Definition of Entropy

As it relates to machine learning, entropy is is a measure of the randomness (disorder or uncertainty) of information being processed.

Simple Example: Tossing a Fair Coin (High Entropy):

- A fair coin has no affinity (or preference) for heads or tails.
- The outcome any number of tosses is difficult to predict because there no relationship between coin flipping and the outcome.

Mathematical Models of Entropy

Principle of Maximum Entropy (Jaynes, 1957)

Given some partial information about a random variate, we should choose the probability distribution that is is consistent with the given information (e.g., boundary constraints), but otherwise has maximum entropy associated with it.

Relationship of Entropy to Uncertainty and Probability

- Every probability distribution has some uncertainty associated with it. Entropy provides a quantitative measure of this uncertainty.
- A principle goal of data mining models and algorithms is to reduce uncertainty.

Measuring Uncertainty of a Probability Distribution:

Definition of a Probability Distribution:

Let the probabilities of n possible outcomes $A_1, A_2, \cdots, A_n$, of an experiment be $p_1, p_2, \cdots, p_n$, respectively. The distribution:

$$P = (p_1, p_2, p_3, \cdots, p_n), \tag{2}$$

satisfies the constraints:

$$\sum_{i=1}^{n} p_i = 1, \tag{3}$$

and

$$p_1 \geq 0, p_2 \geq 0, \cdots, p_n \geq 0. \tag{4}$$

Measuring Uncertainty of a Probability Distribution

Requirements for Measuring Uncertainty (Kapur, 1989):

- It should be a function of $p_1, p_2, \cdots, p_n$, i.e.,

$$H = H_n(P) = H(p_1, p_2, \cdots, p_n). \tag{5}$$

- $H_n(P)$ should be a continuous and symmetric function.
- The maximum value of H_n should increase as n increases.
- It should be minimum (and possibly zero) when there is no uncertainty about the outcome. In other words, it should vanish when one of the outcomes is certain.

$$H_n(P) = 0 \text{ when } p_i = 1 \text{ and } p_j = 0, \ (j \neq i). \tag{6}$$

Measuring Uncertainty of a Probability Distribution

- H_n should be maximum when there is maximum uncertainty, which arises when the outcomes are equally likely, i.e.,

$$p_1 = p_2 = \cdots = p_n = \frac{1}{n}. \tag{7}$$

- For two independent probability distributions P and Q,

$$\sum_{i=1}^{n} p_i = 1, \text{ and } \sum_{j=1}^{m} q_j = 1, \tag{8}$$

the uncertainty of the joint scheme $P \cup Q$ should be:

$$H_{m+n}(P \cup Q) = H_n(P) + H_m(Q). \tag{9}$$

If P and Q have outcomes $A_1, A_2, \cdots, A_n$ and $B_1, B_2, \cdots,$ r_n, then the joint outcomes are $A_i B_j$ with probabilies $p_i q_j$.

Mathematical Models of Entropy

Shanon's Measure of Entropy

Shanon (1949) proposed the following measure:

$$H_n(P) = \sum_{i=1}^{n} p_i ln(\frac{1}{p_i}) = -\sum_{i=1}^{n} p_i ln(p_i). \qquad (10)$$

Intial Observations:

- This function is continuous, symmetric, and convex.
- When one of the probabilities is 1, the others are zero. The entropy is zero and is a minimum value – no surprise.
- All of the commonly used probability distributions – uniform, normal, poisson, logarithmic – can be framed in terms of maximum entropy subject to constraints.

Mathematical Models of Entropy

Maximum Value of Entropy

We can use Lagrange's equations to find a maximum value, i.e.

$$-\sum_{i=1}^{n} p_i ln(p_i) - \lambda \left[\sum_{i=1}^{n} p_i - 1 \right]. \tag{11}$$

This gives (uniform distribution):

$$p_1 = p_2 = \cdots = p_n = \frac{1}{n}. \tag{12}$$

The maximum value of H_n is:

$$H_n = -\sum_{i=1}^{n} \frac{1}{n} ln(\frac{1}{n}) = ln(n) \rightarrow \text{ increases linearly with n.} \tag{13}$$

Mathematical Models of Entropy

Illustrative Example

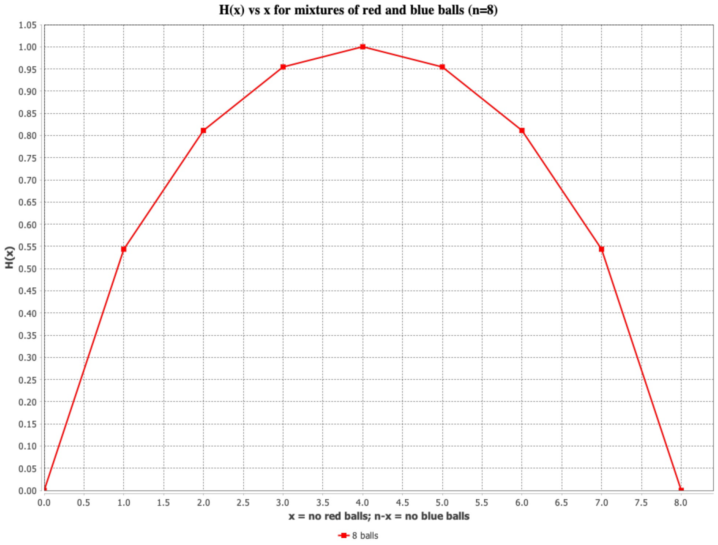
Suppose that an urn contains a mixture of red (n_r) red and blue (n_b) balls (i.e., $n = n_r + n_b$). The entropy is:

$$H_2(P) = -\left[\frac{n_r}{n}\right]\log_2\left[\frac{n_r}{n}\right] - \left[\frac{n_b}{n}\right]\log_2\left[\frac{n_b}{n}\right]. \tag{14}$$

Sample Calculation. Let $n_r = 2$, $n_b = 6$.

$$\begin{aligned} H_2(P) &= -\left[\frac{2}{8}\right]\log_2\left[\frac{2}{8}\right] - \left[\frac{6}{8}\right]\log_2\left[\frac{6}{8}\right] \\ &= \frac{1}{4}\cdot 2.0 + \frac{3}{4}\cdot 0.415 = 0.811 \end{aligned} \tag{15}$$

Mathematical Models of Entropy



Mathematical Models of Entropy

Key Points:

- Minimum values of entropy occur when the urn contains only red balls (i.e., $x = 0$) or only blue balls (i.e., $x = 8$). There is no disorder.
- The maximum value of entropy occurs when the urn system has maximum disorder – that is, four blue balls and four red balls.

$$H_2(P) = -\left[\frac{4}{8}\right]\log_2\left[\frac{4}{8}\right] - \left[\frac{4}{8}\right]\log_2\left[\frac{4}{8}\right] = 1.0 \tag{16}$$

- Even higher levels of entropy (disorder) can be obtained by adding more colors to the urn, e.g., 2 blue balls, 2 green balls, 3 red balls, 1 purple ball. Now, $P = \left(\frac{1}{4}, \frac{1}{4}, \frac{3}{8}, \frac{1}{8}\right)$.

Information Gain in Decision Trees

Mathematical Framework

Information Gain

The amount of information that is gained by knowing the value of an attribute. It equals the entropy of a distribution before a split minus the entropy of a distribution after a split.

$$IG(Y,X) = H(Y) - H(Y|X). \tag{17}$$

Here:

- Information gain $IG(X, Y)$ is the reduction of uncertainty about Y given an additional piece of information X about Y.
- $H(Y)$ is the entropy of Y (before split).
- $H(Y|X)$ is the conditional entropy of Y given the value of attribute X (after split).

Decision Trees

Design of Data Partitions for Classification Tree:

- Use information gain as measure for attribute selection.
- Pick atttribute split that maximizes information gain IG(Y,X), i.e.,

$$IG(D, A) = H(D) - \sum_{i=1}^{v} \frac{D_j}{D} H(D_j) \tag{18}$$

Here:

- D is a prescribed data partition and A is an attribute.
- Split D into v partitions (or subsets) $\{D_1, D_2, \cdots, D_j\}$, where D_j contains those tuples in D that have outcome a_j of A.

Decision Trees

Basic Algorithm (This is a greedy algorithm) ...

- Decision tree is constructed in a top-down recursive divide-and-conquer manner.
- When the construction process begins, all training examples are at the root.
- Attributes are categorical – if continuous-valued they are discretized in advance.
- Need to design a sequence of selected attributes to partition dataset recursively.
- Test attributes are selected on basic of heuristic or statistical measure (e.g., information gain).

Conditions for Stopping Partitioning

- All samples for a given node belong to the same class.

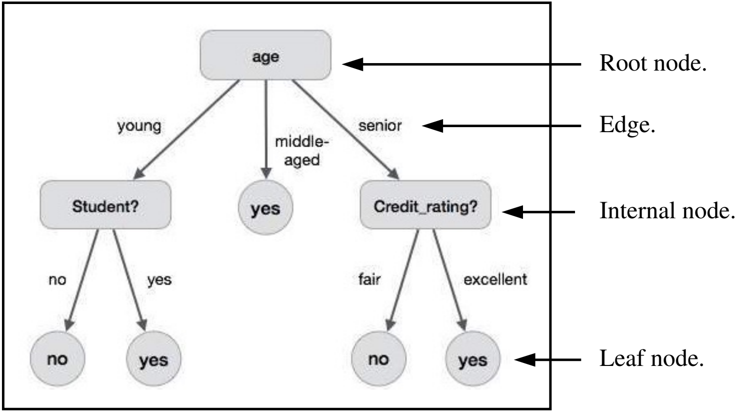
Example 1 (Buy Computer)

Initial Dataset. Will customer buy a computer?

ID	Age Group	Income	Student	Credit Rating	Buys Computer
1	young	high	no	fair	no
2	young	high	no	excellent	no
3	middle	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle	low	yes	excellent	yes
8	young	medium	no	fair	no
9	young	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	young	medium	yes	excellent	yes
12	middle	medium	no	excellent	yes
13	middle	high	yes	fair	yes
14	senior	medium	no	excellent	no

Example 1 (Buy Computer)

Sample Decision Tree: Initially split data based on age group.



Is this a good decision?

Example 1 (Buy Computer)

Entropy of Base Dataset

Purchase outcomes: $\{no = 5, yes = 9\}$.

$$H(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.94 \qquad (19)$$

Partitioned Dataset. Split dataset by age ...

```
 ID   Age Group    Income  Student  Credit Rating   Buys Computer
-----------------------------------------------------------------
  1       young      high       no           fair              no
  2       young      high       no      excellent              no
  8       young    medium       no           fair              no
  9       young       low      yes           fair             yes
 11       young    medium      yes      excellent             yes
-----------------------------------------------------------------
```

Example 1 (Buy Computer)

Partitioned Dataset. Split dataset by age ...

ID	Age Group	Income	Student	Credit Rating	Buys Computer
3	middle	high	no	fair	yes
7	middle	low	yes	excellent	yes
12	middle	medium	no	excellent	yes
13	middle	high	yes	fair	yes

ID	Age Group	Income	Student	Credit Rating	Buys Computer
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
10	senior	medium	yes	fair	yes
14	senior	medium	no	excellent	no

Example 1 (Buy Computer)

Entropy of Partitioned Dataset. Split by age group ...

$$
\begin{aligned}
IG(D, Age) &= H(D) - \sum_{v \in \{you, mid, sen\}} \frac{S_v}{S} H(S_v) \\
&= H(D) - \frac{5}{14} H(S_{you}) - \frac{4}{14} H(S_{mid}) - \frac{5}{14} H(S_{sen}) \\
&= 0.246.
\end{aligned} \tag{20}
$$

Remaining split options:

- IG(D,Income) $= 0.029$,
- IG(D,Student) $= 0.151$,
- IG(D,Credit Rating) $= 0.048$.

Example 1 (Buy Computer)

Conclusion

- Attribute age has the highest information gain and therefore becomes the splitting attribute at the root node.

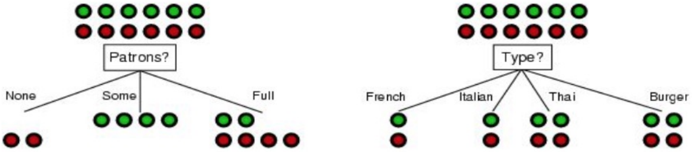
Actions

- Branches are grown for each outcome of age.
- Repeat process on lower-level nodes using split attributes of student and credit rating.

Example 2 (Customer Wait for Table at Restaurant?)

Customer Dataset (Source: Russell and Norvig, 2010)

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T



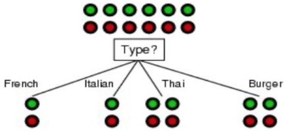
Example 2 (Customer Wait for Table at Restaurant?)

Dataset Attributes

- **Alternate:** Is there a suitable alternate restaurant nearby?
- **Bar:** Does restaurant have comfortable bar area to wait in?
- **Fri/Sat:** True on Fridays and Saturdays.
- **Hungry:** True when customer is hungry.
- **Patrons:** How many people are in the restaurant? (none, some, and full).
- **Price:** The restaurant price range ($, $$ and $$$).
- **Raining:** Is it raining outside?
- **Reservation:** Did customer make a reservation?
- **Type:** Type of restaurant (French, Italian, Thai, or Burger).
- **WaitEstimate:** Wait time estimated by host (0-10 mins, 10-30, 30-60, or > 60).

Example 2 (Customer Wait for Table at Restaurant?)

Split on Restaurant Type Attribute

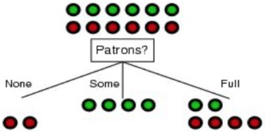


Root entropy: 0.5 * log(2) + 0.5 * log(2) = 1 bit

Leaf entropies: 2/12 * 1 + 2/12 * 1 + … = 1 bit

No reduction!

Split on Patrons Attribute



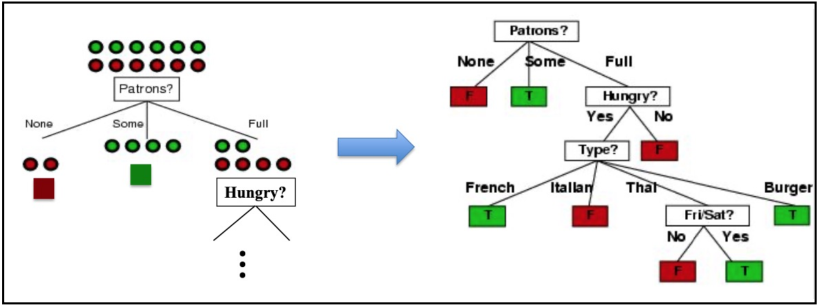
Root entropy: 0.5 * log(2) + 0.5 * log(2) = 1 bit

Leaf entropies: 2/12 * 0 + 4/12 * 0 + 6/12 * 0.9

Lower entropy after split!

Example 2 (Customer Wait for Table at Restaurant?)

Decision Tree Synthesis



Classification with Decision Trees (Summary)

Advantages

- Decision trees are simple to understand and interpret.
- Requires only a small number of observations.
- Best and expected values can be determined for different scenarios.

Disadvantages

- Difficulties in handling data with missing values.
- Information gain criterion is biased in favor of attributes with more levels.
- Calculations become complex if values are uncertain or outcomes are linked.

Ensemble Learning

Ensemble Methods (General Idea)

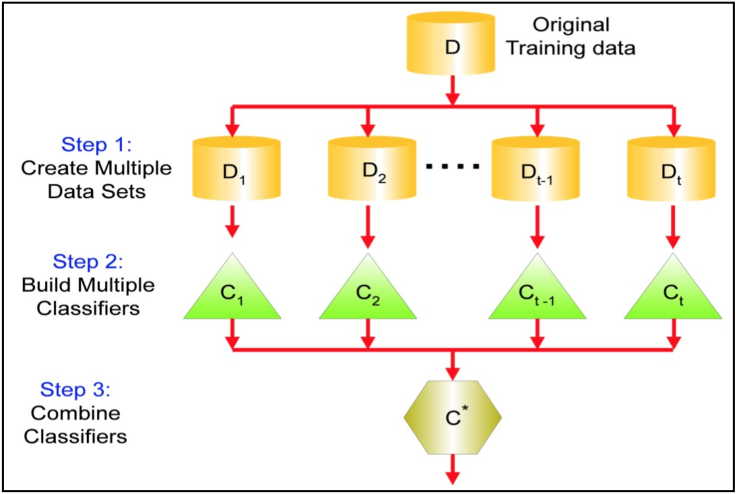
Ensemble Methods

Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any one constituent learning algorithm.

Motivation and Approach

- Supervised learning algorithms search through a hypothesis space to find a hypothesis that will make good predictions.
- Even if the hypothesis space contains hypotheses that are well suited to a particular problem space, find a good hypothesis can still be very difficult.
- Ensembles combine hypotheses in the hope of finding a new one with superior predictive capabilities.

Ensemble Learning (General Idea)



Ensemble Learning (General Idea)

Ensemble Learning

- Combine predictions from multiple learning algorithms $\longrightarrow$ ensemble.
- Often leads to better predictive performance than a single learner.
- Works well then small differences in the training data produce very different classifiers (e.g., decision trees).

Drawbacks

- Increased computational effort.
- Reduced level of interpretability.

Ensemble Learning (Why does it work?)

Why does it work?

- Assume classifiers $C_1, \cdots, C_k$ are independent, i.e.,

$$\text{correlation} \quad \sigma\left(C_1, C_2\right) = 0. \tag{21}$$

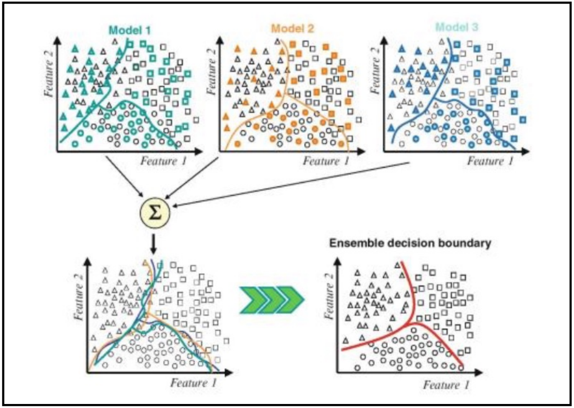
- Assume, for example, that there are 25 classifiers, each having an error rate $\eta = 0.35$.
- Probability that the ensemble classifier makes a wrong prediction:

$$\sum_{i=13}^{25} \binom{25}{i} \eta^i (1-\eta)^{25-i} = 0.06. \tag{22}$$

which is much lower than any individual classifier.

Ensemble Learning (Diversity in Prediction)

Use of ensemble methods can lead to improvements in prediction accuracy through reduction of variability.



Source: Zhang, et al, Ensemble Machine Learning, Springer, 2012.

Ensemble Learning

Constructing Ensembles: Methods for obtaining sets of classifiers

- **Bagging.**
- **Random Forest.**
- **Cross-Validation.** Two key ideas: (1) instead of different classifiers, train same classifier on different data, (2) since training data is expensive, reuse data bu subsampling.

Combining Classifiers: Methods for combining different classifiers

- Stacking
- Bayesian Model Averaging
- Boosting
- AdaBoost

Ensemble Techniques (Bagging)

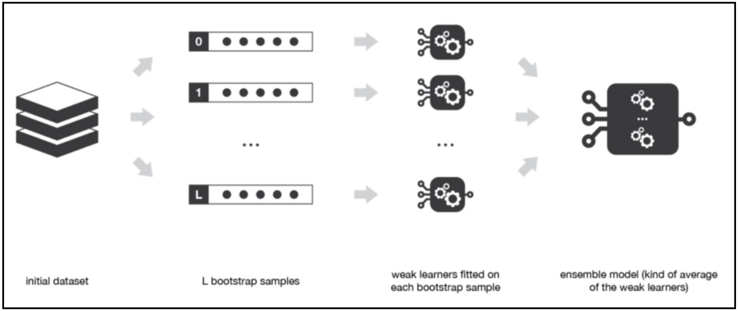
Bagging (Breiman, 1996). Bootstrapping on data.

- Create a data set by sampling data points with replacement.

```
-----------------------------------------------------------
Original Data       :   1   2   3   4   5   6   7   8   9  10
-----------------------------------------------------------
Bagging (Round 1):   7   2   9   7   3   2   1   1   4   5
Bagging (Round 2):   6  10   4   2  10   3   8   9   7   4
Bagging (Round 3):   4   6   8   2   5   1   6   3   1   9
Bagging (Round 4):   .....
Bagging (Round 5):   .....
-----------------------------------------------------------
```

- Create models based on the data sets.
- Generate more data sets and models.
- Make predictions by combining votes – Classification $\rightarrow$ majority vote; prediction $\rightarrow$ average.

Ensemble Techniques (Bagging)



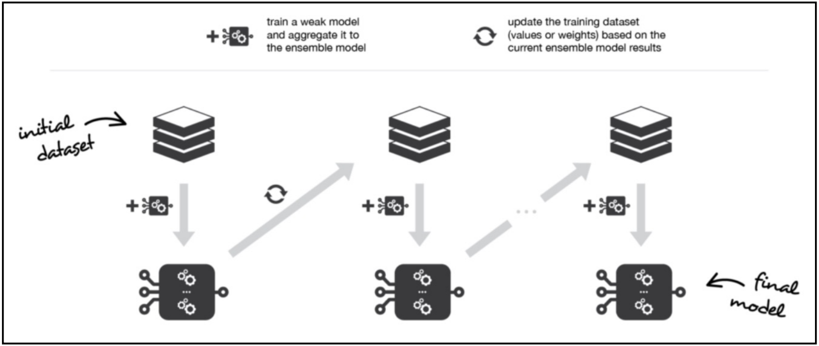
Advantages/Disadvantages:

- Helps when classifier is unstable (has high variance).
- Not helpful when classifier is stable and has large bias.

Ensemble Techniques (Overview)

Boosting (Schapire, 1998). Recursively reweight data.

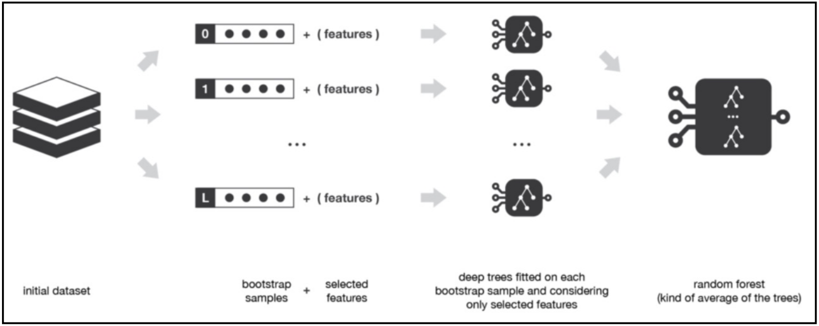
- Records wrongly classified will have their weights increased.
- Records correctly classified will have their weights decreased.



Ensemble Techniques (Random Forest)

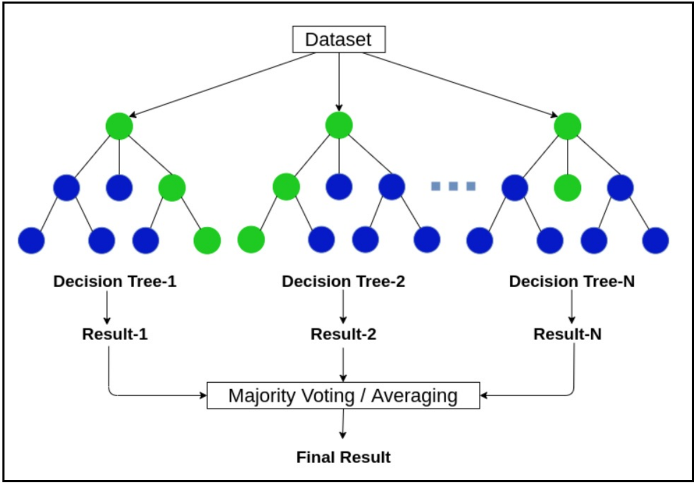
Random Forest (Breiman, 2001).

- Randomly pick features and data to generate diversity of classifiers (decision trees).



Ensemble Techniques (Random Forest)

Random Forest (Breiman, 2001).



Metrics of Evaluation

Metrics of Evaluation

Cross Validation Model

Cross validation is a method for assessing how the results of a data mining (statistical) analysis will generalize to an independent dataset. It is mainly used in predictive model applications.

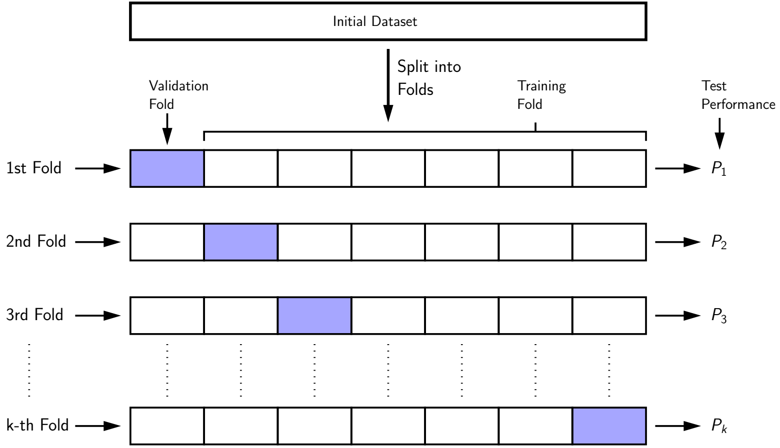
K-Fold Cross Validation Method

- Divide the sample data into k equal parts.
- Use $k-1$ parts for training and one for testing.
- Repeat the procedure k times, rotating the test dataset.
- Compute metrics of performance across the iterations, i.e.,

$$\text{Performance} = \sum_{i=1}^{k} P_i. \tag{23}$$

Metrics of Evaluation

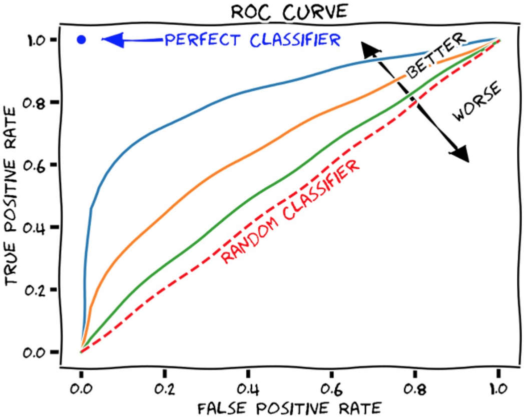
K-Fold Cross Validation



Metrics of Evaluation

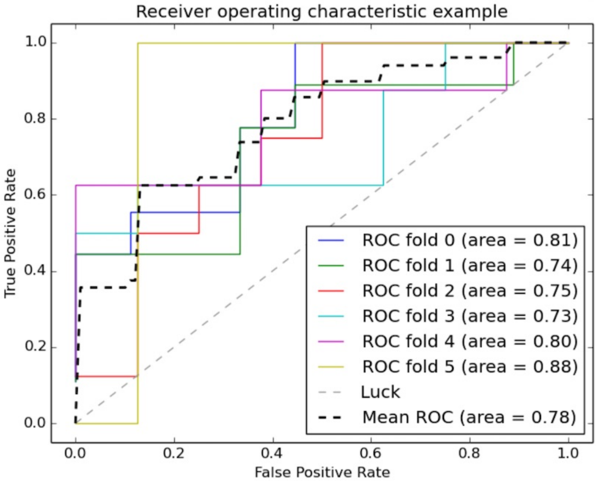
Receiver Operating Curve

A receiver operating curve (ROC) illustrates diagnostic ability of a binary classifier as its discrimination threshold is varied.



Metrics of Evaluation

Typical ROC Curves



Working with Weka

Introduction

WEKA

WEKA (Waikato Environment for Knowledge Acquisition) is a workbench for data mining and machine learning.

Software Download and Installation

- WEKA is written in Java, so it will run on both PCs and Macs.
- Download from: https://www.cs.waikato.ac.nz/weka/

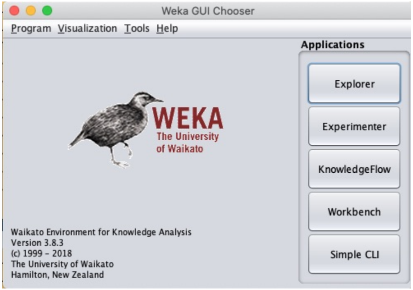
Online Resources

- See class web page for evolving list of links to WEKA resources ...
- Videos learning machine learning with WEKA are available on YouTube.

Getting Started

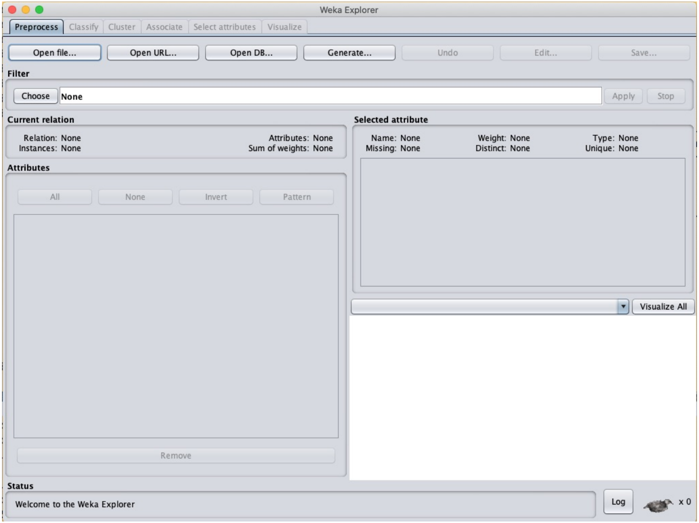
From the Terminal Window

```
prompt >> java -jar weka.jar
```

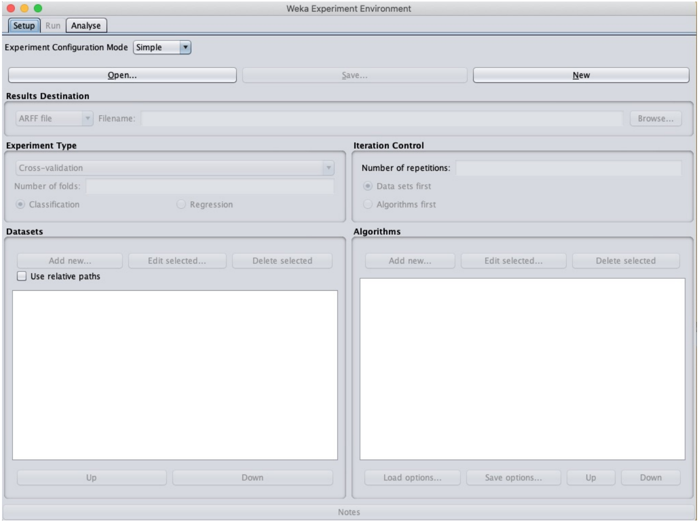


You can also write and run custom applications through the WEKA API.

Weka GUI Explorer



Weka GUI Experimenter



Data Mining

Examples

Example 1. Will Customer Buy Computer?

Input datafile (arff format)

```
% ==================================================================
% ENCE 688P: Classification for Buy Computer?
% ==================================================================

@relation 'computer'
@attribute id real
@attribute age { young, middle, senior}
@attribute income { low, medium, high}
@attribute student {yes, no}
@attribute credit { fair, excellent}
@attribute purchase { no, yes}

@data
1,young,high,no,fair,no
2,young,high,no,excellent,no
3,middle,high,no,fair,yes
4,senior,medium,no,fair,yes
5,senior,low,yes,fair,yes
6,senior,low,yes,excellent,no
7,middle,low,yes,excellent,yes
8,young,medium,no,fair,no
9,young,low,yes,fair,yes
10,senior,medium,yes,fair,yes
11,young,medium,yes,excellent,yes
12,middle,medium,no,excellent,yes
13,middle,high,yes,fair,yes
14,senior,medium,no,excellent,no
```

Example 1. Will Customer Buy Computer?

Java Program Source Code

See: java-code-ml-weka2018/src/ence688p/ClassificationTask.java

Abbreviated Program Output (J48 unpruned tree)

```
age = young
|   student = yes: yes (2.0)
|   student = no: no (3.0)
age = middle: yes (4.0)
age = senior
|   credit = fair: yes (3.0)
|   credit = excellent: no (2.0)

Number of Leaves  :  5
Size of the tree :  8
```

Example 1. Will Customer Buy Computer?

Classification Accuracy wrt Training Dataset

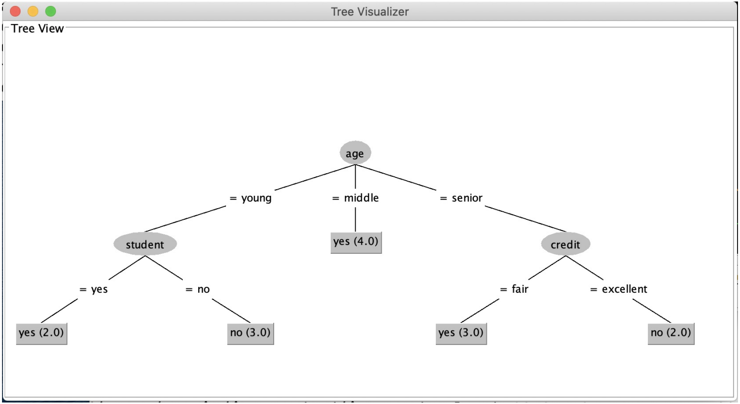
```
Correctly Classified Instances          14      100 %
Incorrectly Classified Instances         0        0 %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0 %
Root relative squared error              0 %
Total Number of Instances               14

=== Confusion Matrix ===

    a b   <-- classified as
    5 0 | a = no
    0 9 | b = yes
```

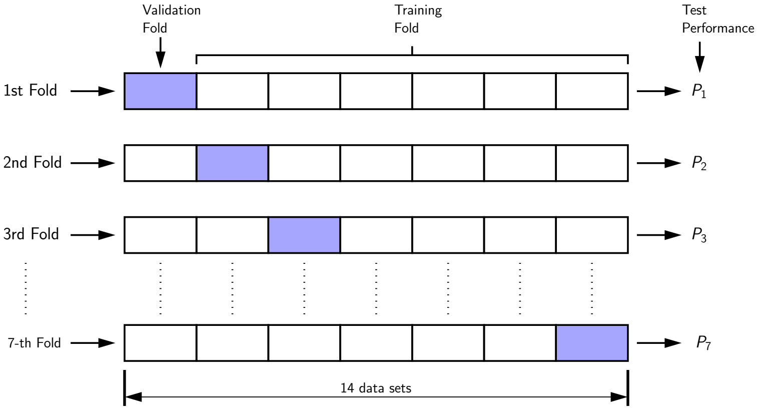
Example 1. Will Customer Buy Computer?

Classification Accuracy wrt Training Dataset



Example 1. Will Customer Buy Computer?

Cross Validation Model (nofolds = 7)



Example 1. Will Customer Buy Computer?

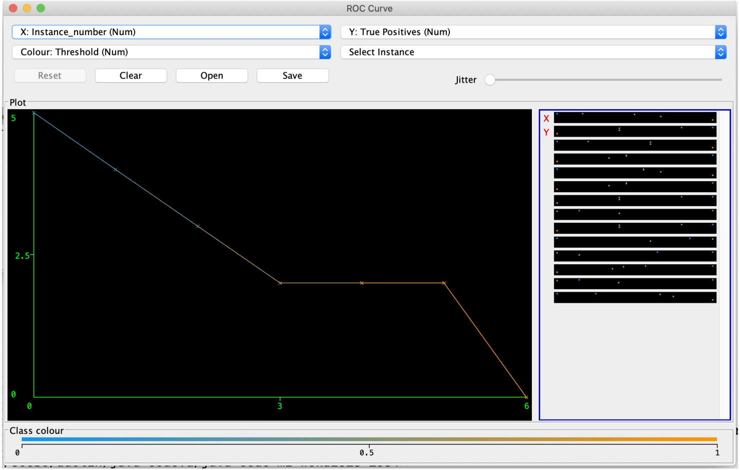
Cross Validation Model (after classification) (nofolds = 7)

```
Correctly Classified Instances          10               71.4286 %
Incorrectly Classified Instances         4               28.5714 %
Kappa statistic                          0.3778
Mean absolute error                      0.2798
Root mean squared error                  0.4393
Relative absolute error                 58.3333 %
Root relative squared error             88.6322 %
Total Number of Instances               14

=== Confusion Matrix ===

    a b   <-- classified as
    3 2 | a = no
    2 7 | b = yes
```

Example 1. Will Customer Buy Computer?



Example 2. Milk, Diapers and Beer

Input datafile (arff format)

```
% ================================================================
% ENCE 688P: Customer purchases at supermarket ..
%
% Mark Austin                                          March, 2021
% ================================================================

@relation 'supermarket'
@attribute id real
@attribute beer {t}
@attribute bread {t}
@attribute coke {t}
@attribute diapers {t}
@attribute eggs {t}
@attribute milk {t}

@data
1,?,t,?,?,?,t
2,t,t,?,t,t,?
3,t,?,t,t,?,t
4,t,t,?,t,?,t
5,?,t,t,t,?,t
```

Example 2. Milk, Diapers and Beer

Java Program Source Code (Weka Code)

See: java-code-ml-weka2018/src/ence688p/Supermarket.java

Abbreviated Program Output (Print modified input file)

```
@relation supermarket-weka.filters.unsupervised.attribute.Remove-R1

@attribute beer {t}
... attributes removed ...
@attribute milk {t}

@data
?,t,?,?,?,t
t,t,?,t,t,?
t,?,t,t,?,t
t,t,?,t,?,t
?,t,t,t,?,t
```

Example 2. Milk, Diapers and Beer

Abbreviated Program Output (Apriori Model)

```
Size of set of large itemsets L(1): 6
Size of set of large itemsets L(2): 13
Size of set of large itemsets L(3): 12
Size of set of large itemsets L(4): 4

Best rules found:

 1. beer=t 3 ==> diapers=t 3 <conf:(1)> lift:(1.25) lev:(0.12) [0] conv
 2. coke=t 2 ==> diapers=t 2 <conf:(1)> lift:(1.25) lev:(0.08) [0] conv
 3. coke=t 2 ==> milk=t 2 <conf:(1)> lift:(1.25) lev:(0.08) [0] conv:(0
 4. beer=t bread=t 2 ==> diapers=t 2 <conf:(1)> lift:(1.25) lev:(0.08)
 5. beer=t milk=t 2 ==> diapers=t 2 <conf:(1)> lift:(1.25) lev:(0.08) [
 6. coke=t milk=t 2 ==> diapers=t 2 <conf:(1)> lift:(1.25) lev:(0.08) [
 7. coke=t diapers=t 2 ==> milk=t 2 <conf:(1)> lift:(1.25) lev:(0.08) [
 8. coke=t 2 ==> diapers=t milk=t 2 <conf:(1)> lift:(1.67) lev:(0.16) [
 9. eggs=t 1 ==> beer=t 1 <conf:(1)> lift:(1.67) lev:(0.08) [0] conv:(0
10. eggs=t 1 ==> bread=t 1 <conf:(1)> lift:(1.25) lev:(0.04) [0] conv:(
```

Example 2. Milk, Diapers and Beer

Abbreviated Program Output (FPGrowth Model)

```
FPGrowth found 38 rules (displaying top 10)

 1. [coke=t]: 2 ==> [milk=t]: 2 <conf:(1)> lift:(1.25) lev:(0.08) conv:
 2. [beer=t]: 3 ==> [diapers=t]: 3 <conf:(1)> lift:(1.25) lev:(0.12) co
 3. [coke=t]: 2 ==> [diapers=t]: 2 <conf:(1)> lift:(1.25) lev:(0.08) co
 4. [eggs=t]: 1 ==> [diapers=t]: 1 <conf:(1)> lift:(1.25) lev:(0.04) co
 5. [eggs=t]: 1 ==> [bread=t]: 1 <conf:(1)> lift:(1.25) lev:(0.04) conv
 6. [eggs=t]: 1 ==> [beer=t]: 1 <conf:(1)> lift:(1.67) lev:(0.08) conv:
 7. [milk=t, beer=t]: 2 ==> [diapers=t]: 2 <conf:(1)> lift:(1.25) lev:(
 8. [coke=t]: 2 ==> [milk=t, diapers=t]: 2 <conf:(1)> lift:(1.67) lev:(
 9. [milk=t, coke=t]: 2 ==> [diapers=t]: 2 <conf:(1)> lift:(1.25) lev:(
10. [diapers=t, coke=t]: 2 ==> [milk=t]: 2 <conf:(1)> lift:(1.25) lev:(

--- ===================================== ...
--- Finished !! ...
```

References

- Jaynes E.T., Information Theory and Statistical Mechanics. II, Phys. Rev. 108, 171, October 1957.
- Kapur J.N., Maximum-Entropy Models in Science and Engineering, John Wiley and Sons, 1989.
- Mitchell T.M., Machine Learning and Data Mining, Communications of the ACM, Vol. 42., No. 11, November 1999.
- Russell S., and Norvig P., Artificial Intelligence: A Modern Approach (Third Edition), Prentice-Hall, 2010.
- Shanon C.E., and Weaver W., The Mathematical Theory of Communication, University of Illinois, Urbana, Chicago, 1949.
- Witten I.H., Frank E., Hall M.A., and Pal C.J., Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2017.