CHAPTER 1

Machine Learning Semantics

Seed, Cleanse and group dataset
Raw Data

Training Dataset
Build and refine the model
Machine Learning Algorithm (Learner)

Testing Dataset
Validate the model

Prediction Rule

New or Evaluation Dataset

Predicted behavior
Distribution of load on both the machines

Profits
Learning Sub-fields

- Supervised learning
- Unsupervised learning
- Deep learning
- Reinforcement learning
- Semi-supervised learning
The image contains a graph with a logarithmic scale on the x-axis labeled as "Hypothesis" and the y-axis labeled as "Minimum samples needed." The graph shows a trend line that indicates an increase in minimum samples needed as the hypothesis value increases.

The equation for Mean Squared Error (MSE) is given as:

\[ MSE = \frac{\sum_{i=1}^{n} (P_i - A_i)^2}{n} \]

The equation for Mean Absolute Error (MAE) is given as:

\[ MAE = \frac{\sum_{i=1}^{n} |P_i - A_i|}{n} \]

The equation for Normalized Mean Squared Error (NMSE) is given as:

\[ NMSE = \frac{MSE \text{ of developed model}}{MSE \text{ of naive model}} \]

A graph below the equations shows the relationship between error and model complexity, with two curves labeled "Variance" and "Bias^2" that tend to decrease as model complexity increases, reaching a minimum at the "Optimum Model Complexity," and then increasing again. The total error also decreases as model complexity increases up to the optimum point and then increases again.
Machine learning has many learning models.

Categories:
- Association rules based
- Decision tree based
- Ensemble method based
- Deep learning based
- Clustering methods based
- Regression Analysis based
- Bayesian method based
- Instance based
- Kernel method based
- Dimensionality reduction based
Parallel Processor

SIMD (Single Instruction Multiple data stream)

- Shared memory (tightly coupled)
  - Master-Slave architecture

- SMP (Symmetric multiprocessors)

MIMD (Multiple Instruction Multiple data stream)

- Distributed memory (loosely coupled)
  - Clusters

Diagram:
- Shared Memory
  - Network connectivity for data share
  - Parallel processing: Multiple CPUs within a shared memory
  - Distributed processing: Multiple machines with their own memory connected via a network
<table>
<thead>
<tr>
<th>ID#</th>
<th>Model</th>
<th>Year</th>
<th>Color</th>
<th>Dealer</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>4523</td>
<td>Civic</td>
<td>2002</td>
<td>Blue</td>
<td>MN</td>
<td>$18,000</td>
</tr>
<tr>
<td>3476</td>
<td>Corolla</td>
<td>1999</td>
<td>White</td>
<td>IL</td>
<td>$15,000</td>
</tr>
<tr>
<td>7623</td>
<td>Camry</td>
<td>2001</td>
<td>Green</td>
<td>NY</td>
<td>$21,000</td>
</tr>
<tr>
<td>9834</td>
<td>Prius</td>
<td>2001</td>
<td>Green</td>
<td>CA</td>
<td>$18,000</td>
</tr>
<tr>
<td>6734</td>
<td>Civic</td>
<td>2001</td>
<td>White</td>
<td>OR</td>
<td>$17,000</td>
</tr>
<tr>
<td>5342</td>
<td>Altima</td>
<td>2001</td>
<td>Green</td>
<td>FL</td>
<td>$19,000</td>
</tr>
<tr>
<td>3845</td>
<td>Maxima</td>
<td>2001</td>
<td>Blue</td>
<td>NY</td>
<td>$22,000</td>
</tr>
<tr>
<td>8354</td>
<td>Accord</td>
<td>2000</td>
<td>Green</td>
<td>VT</td>
<td>$18,000</td>
</tr>
<tr>
<td>4395</td>
<td>Civic</td>
<td>2001</td>
<td>Red</td>
<td>CA</td>
<td>$17,000</td>
</tr>
<tr>
<td>7352</td>
<td>Civic</td>
<td>2002</td>
<td>Red</td>
<td>WA</td>
<td>$18,000</td>
</tr>
</tbody>
</table>

Diagram:
- **Map**
- **Shuffle**
- **Reduce**

Process 0: Data

May I Send?

Process 1: Yes

Data

Time
<table>
<thead>
<tr>
<th>Year</th>
<th>Evolution / Progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002-2003</td>
<td>Work on Nutch was started by Doug Cutting and Mike Cafarella</td>
</tr>
<tr>
<td>2003 - 2004</td>
<td>Google published work on GFS and MapReduce</td>
</tr>
<tr>
<td>2004</td>
<td>Doug Cutting added support for GFS and MapReduce to Nutch</td>
</tr>
<tr>
<td>2006</td>
<td>Hadoop spins out of Nutch when Yahoo hired Doug Cutting</td>
</tr>
<tr>
<td>2007</td>
<td>NY Times converts 4TB of image archives over 100 EC2s</td>
</tr>
<tr>
<td></td>
<td>Facebook launched Hive, an SQL support for Hadoop</td>
</tr>
<tr>
<td>2008</td>
<td>Fastest sort over 910 nodes taking 3.5 mins</td>
</tr>
<tr>
<td></td>
<td>Cloudera founded</td>
</tr>
<tr>
<td>2009</td>
<td>First Hadoop Summit with 750 attendees</td>
</tr>
<tr>
<td></td>
<td>Doug Cutting joined Cloudera</td>
</tr>
</tbody>
</table>
A Large File of size 1000 MB

101011101010111110010110101010101011011101010110110101011010000111100
110101011010001010101011011101010111111001010101010101101011111011
1010101111010110001111001010101001010101101010111101011111101010110
101010101010101010101010101011010101010110101111010101011111101101
1010101010101010101010101010101011101010101010101011110101010101010
1010101010101010101010101010101010101111010101010111111011010101010
11101010111101010100011111001010101010010101010101011110101010101010
11101010101010101010101010101010111101010101010101111110101010101010
10101010101010111010101010101010101010111111010101010101111110101010
10101010101010101010101010101010101010111101010101011111101010101010
10101010101010101010101010101010101010101111010101010111111010101010
10101010101010101010101010101010101010101011101010101111110101010101
01010101010111010101010101010101010101010101010101010101010101010101
01010101010101010101010101010101010101010101010101010101010101010101
01010101010101010101010101010101010101010101010101010101010101010101
01010101010101010101010101010101010101010101010101010101010101010101
01010101010101010101010101010101010101010101010101010101010101010101

Block 1  Block 2  Block 3  ...  Block 15  Block 16
64MB    64MB    64MB    64MB    40MB

Data Node 1
Block 1
Block 2
(Copy 1)
Block 3
(Copy 2)

Data Node 2
Block 2
(Copy 1)
Block 3
(Copy 1)

Data Node 3
Block 3
(Copy 1)
Block 1
(Copy 3)

Data Node 4
Block 1
(Copy 1)
Block 2
(Copy 3)
Data is chopped and stored on the HDFS (Hadoop Distributed File System).

Data in HDFS is scattered across multiple nodes for Fault Tolerance.

Metadata is stored in the name node and the data blocks are stored in data nodes.

The master slave architecture has Name Node (the master node) and the data nodes (slave nodes).

Both Name node and Data nodes are stored on commodity servers and each node offers local storage and computation.

A Large File

10101110101011110101
1010110101010101010101
10111111010101111011
01000111110011010101

Split

Blocks of 64MB files

B1  B2  B3
B4  B5  B6
B7  ...  Bn

HDFS Client

Request Meta-data for Writing data into HDFS

Details of what block to be stored where are passed

Name Node

Metadata

Large file, n=3, \{B1, B2, B3, ..., Bn\}
B1 = \{ Data Node 1, Data Node 3, Data Node 4 \}
B2 = \{ Data Node 2, Data Node 3, Data Node 4 \}
B3 = \{ Data Node 1, Data Node 2, Data Node 3 \}
...  Bn = ...

Name Node

Heart-beat signal
Block report
Replication

Rack 1

Data Node 1  Data Node 2

Rack 2

Data Node 3  Data Node 4

Rack 1

Data Node 1  Data Node 2

Rack 2

Data Node 3  Data Node 4
Hadoop Command Line

Usage: hadoop fs [generic options]
[-appendToFile <localsrc> ... <dst>]
[-cat [-ignoreGrc] <src> ...]
[-checksum <src> ...]
[-chgrp [-R] GROUP PATH...]
[-chmod [-R] <MODE_MODE> ... ! OCTALMODE> PATH...]
[-chown [-R] [OWNER1:GROUP]] PATH...]
[-copyFromLocal [-f] [-p] <localsrc> ... <dst>]
[-copyToLocal [-p] [-ignoreGrc] [-crc] <src> ... <localdst>]
[-count [-q] <path> ...]
[-cp [-f] [-p] <src> ... <dst>]
[-createSnapshot <snapshotDir> [<snapshotName>]]
[-deleteSnapshot <snapshotDir> <snapshotName>]
[-df [-h] [<path> ...]]
[-du [-s] [-h] <path> ...]
[-expunge]
[-get [-p] [-ignoreGrc] [-crc] <src> ... <localdst>]
[-getfacl [-R] <path>]
[-getmerge [-nl] <src> <localdst>]
[-help [cmd ...]]
[-ls [-d] [-h] [-R] <path> ...]
[-mkdir [-p] <path> ...]
[-moveFromLocal <localsrc> ... <dst>]
[-moveToLocal <src> <localdst>]
[-mv <src> ... <dst>]
[-put [-f] [-p] <localsrc> ... <dst>]
[-renameSnapshot <snapshotDir> <oldName> <newName>]
[-rmrdir [{-ignore-fail-on-non-empty} <dir> ...]
[-setfacl [-R] [[-h] k] [-ni-x <acl_spec> <path>]]{-set <acl_spec> <path>]]
[-set [format] <path> ...]
[-tail [-f] <file>]
[-test [-defsz] <path>]
[-text [-ignoreGrc] <src> ...]
[-touch <path> ...]
[-usage [cmd ...]]

Generic options supported are
-\-conf <configuration file> specify an application configuration file
-\-D <property=value> use value for given property
-\-fs <local|namenode:port> specify a namenode
-\-jt <local|jobtracker:port> specify a job tracker
-\-files <comma separated list of files> specify comma separated files to be copied to the map reduce cluster
-\-libjars <comma separated list of jars> specify comma separated jar files to include in the classpath.
-\-archives <comma separated list of archives> specify comma separated archives to be unarchived on the compute machines.

The general command line syntax is
bin/hadoop command [genericOptions] [commandOptions]
Input data

InputFormat

(K1, V1)

Map

(K2, V2)

Shuffle and Sort

(K2, (V2, V2, V2...))

Reduce

(K3, V3)
ARCHITECTURE COMPARISON
Hadoop 1.0 vs. Hadoop 2.0.

Single Use System
Batch Apps

HADOOP 1.0

MapReduce
(cluster resource management & data processing)

HDFS
(redundant, reliable storage)

HADOOP 2.0

MapReduce
(batch)
Tez
(interactive)
Others
(varied)

YARN
(operating system: cluster resource management)

HDFS2
(redundant, reliable storage)

Multi Use Data Platform
Batch, Interactive, Online, Streaming, …

SOURCE: HORTONWORKS
master@Hadoopupgrade:$ sudo addgroup hadoop
Adding group 'hadoop' (GID 1001) ...
Done.
master@Hadoopupgrade:$ sudo adduser --ingroup hadoop hduser
Adding user 'hduser' ...
Adding new user 'hduser' (1001) with group 'hadoop' ...
Creating home directory '/home/hduser' ...
Copying files from '/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
password: password updated successfully
Changing the user information for hduser
Enter the new value, or press ENTER for the default
   Full Name []:
   Room Number []:
   Work Phone []:
   Home Phone []:
   Other []:
Is the information correct? [Y/n] Y
master@Hadoopupgrade:$

<configuration>

<!-- Site specific YARN configuration properties -->

<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>

<property>
  <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>

</configuration>

<configuration>

<property>
  <name>fs.default.name</name>
  <value>hdfs://localhost:9000</value>
</property>

</configuration>

<configuration>

<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>

<property>
  <name>dfs.namenode.name.dir</name>
  <value>/file:/usr/local/hadoop/yarn_data/hdfs/namenode</value>
</property>

<property>
  <name>dfs.datanode.data.dir</name>
  <value>/file:/usr/local/hadoop/yarn_data/hdfs/datanode</value>
</property>

</configuration>
<configuration>
<property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
</property>
</configuration>

# Set Hadoop-related environment variables
export HADOOP_PREFIX="/usr/local/hadoop"
export HADOOP_HOME="/usr/local/hadoop"
export HADOOP_MAPRED_HOME=${HADOOP_HOME}
export HADOOP_COMMON_HOME=${HADOOP_HOME}
export HADOOP_HDFS_HOME=${HADOOP_HOME}
export YARN_HOME=${HADOOP_HOME}
export HADOOP_CONF_DIR=${HADOOP_HOME}/etc/hadoop

# Native Path
export HADOOP_COMMON_LIB_NATIVE_DIR=${HADOOP_PREFIX}/lib/native
export HADOOP_OPTS="-Djava.library.path=${HADOOP_PREFIX}/lib"

# Java path
export JAVA_HOME="/usr/local/Java/jdk1.7.0_45"
export HADOOP_HOME/bin directory to PATH
export PATH=$PATH:$HADOOP_HOME/bin:$JAVA_HOME/bin:$HADOOP_HOME/sbin

hduser@Hadoopupgrade:~$ hadoop-daemon.sh start namenode
starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-namenode-Hadoopupgrade.out
hduser@Hadoopupgrade:~$ jps
1244 NameNode
1280 Jps
dhuser@Hadoopupgrade:~$ hadoop-daemon.sh start datanode
starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser-datanode-Hadoopupgrade.out
hduser@Hadoopupgrade:~$ jps
1400 Jps
1244 NameNode
1332 DataNode
hduser@Hadoopupgrade:~$ yarn-daemon.sh start resourcemanager
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resourcemanager-Hadoopupgrade.out
hduser@Hadoopupgrade:~$ jps
1474 Jps
1244 NameNode
1433 ResourceManager
1332 DataNode
Vectors Implementation in Mahout

- Dense Vectors
- Sparse Vectors

  - Random Access Sparse Vectors
  - Sequential Access Sparse Vectors

---

R Console

```r
> library(zoo)
> library(rproxy)

Warning messages:
1: package 'zoo' was built under R version 2.15.2
2: package 'rproxy' was built under R version 2.15.2
[Previously saved workspace restored]
```

---

```r
> help.start()
```

---

```
R is free software and comes with ABSOLUTELY NO WARRANTY. You are welcome to redistribute it under certain conditions. Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and 'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or 'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Loading required package: zoo
Loading required package: rproxy

Warning messages:
1: package 'zoo' was built under R version 2.15.2
2: package 'rproxy' was built under R version 2.15.2
[Previously saved workspace restored]
```
R version 2.15.1 (2012-06-22) -- "Roasted Marshmallows"
Copyright (C) 2012 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: i386-pc-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Loading required package: room
Loading required package: rscproxy
Warning messages:
1: package 'room' was built under R version 2.15.2
2: package 'rscproxy' was built under R version 2.15.2
[Previously saved workspace restored]

> chooseCRANmirror()
Warning messages:
1: package 'room' was built under R version 2.15.2
2: package 'rscproxy' was built under R version 2.15.2
[Previously saved workspace restored]

```r
> chooseCRANmirror()
> 1+2
> [1] 3
> log(10)
> [1] 2.302585
> 1+2
> [1] 3
> a = 2
> a
> [1] 2
> a <-2
> a
> [1] 2
> a <-10
> a
> [1] 10
> b <-5
> b
> [1] 5
```

```
f <- function(x1,y1) (1-x1)^2 + 100*(y1 - x1)^2
optim( c(0,0), f )

f <- function(x) (1-x[1])^2 + 100*(x[2]-x[1])^2
optim( c(0,0), f )

x <- seq(-1.5,1.5,by=.2)
y <- seq(-1.5,1.5,by=.2)
z <- outer(x,y,f)
persp(x,y,z,phi=45,theta=45,col="yellow",shad=1)
```
```julia
function mandel(z)
    c = z
    maxiter = 80
    for n = 1:maxiter
        abs(z) ≥ 2 && return n-1
        z = z^2 + c
    end
    return max
end

mandel(0) 80 maxabs!
maximum maxintfloat
maximum!(x, y) max_text_extents
```
Spark assembly has been built with Hive, including Datanucleus jars on classpath.


Welcome to

Using Scala version 2.10.4 (OpenJDK 64-Bit Server VM, Java 1.7.0_75)

Type in expressions to have them evaluated.
Type :help for more information.

INFO SecurityManager: Changing view acs to: ubuntu
INFO SecurityManager: Changing modify acs to: ubuntu
INFO SecurityManager: SecurityManager: authentication disabled; ui acs disabled; users with view permissions: Set(ubuntu); users with modify permissions: Set(ubuntu)
INFO HttpServer: Starting HTTP Server on port 59689. Successfully started service 'HTTP class server' on port 59689.
INFO HttpServer: Starting HTTP Server
INFO HttpServer: Starting HTTP file server' on port 60397.
INFO HttpServer: Successfully started service 'SparkUI' on port 4040.
INFO Executor: Using REPL class URI: http://172.31.21.139:59689
INFO AkkaUtils: Connecting to HeartbeatReceiver: akka.tcp://sparkDriver@ip-172-31-21-139.us-west-2.compute.internal:33030/user/HeartbeatReceiver
INFO NettBlockTransferService: Server created on 44185
INFO BlockManagerMaster: Trying to register BlockManager
INFO BlockManagerMasterActor: Registering block manager localhost:44185 with 267.3 MB RAM, BlockManagerId<driver>, localhost, 44185
INFO BlockManagerMaster: Registered BlockManager
INFO SparkUI: Created spark context...

Spark context available as sc.
runBatchLayer() {
    while(true) recomputeFunctions()
}

<table>
<thead>
<tr>
<th>ID</th>
<th>Age</th>
<th>Experience</th>
<th>Income</th>
<th>Family</th>
<th>CCAvg</th>
<th>Personal Loan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25</td>
<td>1</td>
<td>49</td>
<td>4</td>
<td>1.60</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>45</td>
<td>19</td>
<td>34</td>
<td>3</td>
<td>1.50</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>39</td>
<td>15</td>
<td>11</td>
<td>1</td>
<td>1.00</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>35</td>
<td>9</td>
<td>100</td>
<td>1</td>
<td>2.70</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>35</td>
<td>8</td>
<td>45</td>
<td>4</td>
<td>1.00</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>37</td>
<td>13</td>
<td>29</td>
<td>4</td>
<td>0.40</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>34</td>
<td>9</td>
<td>180</td>
<td>1</td>
<td>8.90</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>38</td>
<td>14</td>
<td>130</td>
<td>4</td>
<td>4.70</td>
<td>1</td>
</tr>
<tr>
<td>19</td>
<td>46</td>
<td>21</td>
<td>193</td>
<td>2</td>
<td>8.10</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>38</td>
<td>13</td>
<td>119</td>
<td>1</td>
<td>3.30</td>
<td>1</td>
</tr>
<tr>
<td>39</td>
<td>42</td>
<td>18</td>
<td>141</td>
<td>3</td>
<td>5.00</td>
<td>1</td>
</tr>
<tr>
<td>43</td>
<td>32</td>
<td>7</td>
<td>132</td>
<td>4</td>
<td>1.10</td>
<td>1</td>
</tr>
<tr>
<td>48</td>
<td>37</td>
<td>12</td>
<td>194</td>
<td>4</td>
<td>0.20</td>
<td>1</td>
</tr>
</tbody>
</table>

13 Records (6-No & 7-Yes)

CCAvg

Low
8 Records (6-No & 2-Yes)

Medium
3 Records (3-Yes)

High
2 Records (2-Yes)

Income

Low
5 Records (5-No)

Medium
2 Records (1-No & 1-Yes)

High
1 Record (1-Yes)
\[ E = - \sum_{i=1}^{m} p_i \log_2(p_i) \]

\[ E_A = \sum_{i=1}^{v} \frac{D_i}{D} E(D_i) \]

<table>
<thead>
<tr>
<th>ID</th>
<th>Age</th>
<th>Income</th>
<th>Family</th>
<th>CCAvg</th>
<th>Personal Loan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Young</td>
<td>Low</td>
<td>4</td>
<td>Low</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Old</td>
<td>Low</td>
<td>3</td>
<td>Low</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Middle</td>
<td>Low</td>
<td>1</td>
<td>Low</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>Middle</td>
<td>Medium</td>
<td>1</td>
<td>Low</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>Middle</td>
<td>Low</td>
<td>4</td>
<td>Low</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>Middle</td>
<td>Low</td>
<td>4</td>
<td>Low</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>Middle</td>
<td>High</td>
<td>1</td>
<td>High</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>Middle</td>
<td>Medium</td>
<td>4</td>
<td>Medium</td>
<td>1</td>
</tr>
<tr>
<td>19</td>
<td>Old</td>
<td>High</td>
<td>2</td>
<td>High</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>Middle</td>
<td>Medium</td>
<td>1</td>
<td>Medium</td>
<td>1</td>
</tr>
<tr>
<td>39</td>
<td>Old</td>
<td>Medium</td>
<td>3</td>
<td>Medium</td>
<td>1</td>
</tr>
<tr>
<td>43</td>
<td>Young</td>
<td>Medium</td>
<td>4</td>
<td>Low</td>
<td>1</td>
</tr>
<tr>
<td>48</td>
<td>Middle</td>
<td>High</td>
<td>4</td>
<td>Low</td>
<td>1</td>
</tr>
</tbody>
</table>

\[-\frac{6}{13} \log_2\left(\frac{6}{13}\right) - \frac{7}{13} \log_2\left(\frac{7}{13}\right) = 0.995727\]

<table>
<thead>
<tr>
<th></th>
<th>Fraction</th>
<th>CCAvg</th>
<th>Loan=No</th>
<th>Loan=Yes</th>
<th>Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>0.615385</td>
<td>0.25</td>
<td>0.75</td>
<td>0.81</td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td>0.230769</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>0.153846</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>Entropy</td>
<td>0.499248</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
\[ \text{Entropy}_{\text{CCAvg}} = \frac{8}{13} E(6,2) + \frac{3}{13} E(0,3) + \frac{2}{13} E(0,2) = 0.499248 \]

\[ I_{\text{CCAvg}} = 0.995727 - 0.499248 = 0.496479 \]

\[ I_{\text{Family}} = 0.995727 - 0.923077 = 0.07265 \]

\[ \left( \frac{6}{13} \right)^2 + \left( \frac{7}{13} \right)^2 \]

<table>
<thead>
<tr>
<th>CCAvg</th>
<th>Fraction</th>
<th>Loan=No</th>
<th>Loan=Yes</th>
<th>Gini</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>0.615385</td>
<td>0.250000</td>
<td>0.750000</td>
<td>0.625</td>
</tr>
<tr>
<td>Medium</td>
<td>0.230769</td>
<td>0.000000</td>
<td>1.000000</td>
<td>1.000</td>
</tr>
<tr>
<td>High</td>
<td>0.153846</td>
<td>0.000000</td>
<td>1.000000</td>
<td>1.000</td>
</tr>
<tr>
<td>Gini Index</td>
<td></td>
<td></td>
<td></td>
<td>0.769231</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Family</th>
<th>Fraction</th>
<th>Loan=No</th>
<th>Loan=Yes</th>
<th>Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.307692</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>0.076923</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0.153846</td>
<td>0.50</td>
<td>0.50</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>0.461538</td>
<td>0.50</td>
<td>0.50</td>
<td>0.5</td>
</tr>
<tr>
<td>Gini Index</td>
<td></td>
<td></td>
<td></td>
<td>0.538462</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Men</th>
<th>Number</th>
<th>Age</th>
<th>Married</th>
</tr>
</thead>
<tbody>
<tr>
<td>Man 1</td>
<td>1</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Man 2</td>
<td>2</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Man 40</td>
<td>40</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Female</th>
<th>Number</th>
<th>Age</th>
<th>Married</th>
</tr>
</thead>
<tbody>
<tr>
<td>Woman 1</td>
<td>41</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Woman 2</td>
<td>42</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Woman 59</td>
<td>99</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Woman 60</td>
<td>99</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>
Partition $D$ into $D_{train}$ (training / "growing"), $D_{validation}$ (validation / "pruning")

Build complete tree $T$ on $D_{train}$

UNTIL accuracy on $D_{validation}$ decreases DO

FOR each non-leaf node candidate in $T$

$\text{Temp}[\text{candidate}] \leftarrow \text{Prune} (T, \text{candidate})$

$\text{Accuracy}[\text{candidate}] \leftarrow \text{Test} (\text{Temp}[\text{candidate}], D_{validation})$

$T \leftarrow T' \in \text{Temp}$ with best value of $\text{Accuracy}$ (best increase; greedy)

RETURN (pruned)
<table>
<thead>
<tr>
<th>S No</th>
<th>Variable</th>
<th>Variable Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X1</td>
<td>X1, X2, X3</td>
</tr>
<tr>
<td>2</td>
<td>X2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>X3</td>
<td>X3, X4, X5</td>
</tr>
<tr>
<td>4</td>
<td>X4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>X5</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>Xn</td>
<td>Xa, Xb, Xn</td>
</tr>
</tbody>
</table>

\[
\text{MSE} \left[ \hat{f}_m(p) \right] = O \left( \frac{1}{m^{4(D+4)}} \right)
\]

<table>
<thead>
<tr>
<th>Dimensionality</th>
<th>Required Sample Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>786</td>
</tr>
<tr>
<td>7</td>
<td>10,700</td>
</tr>
<tr>
<td>10</td>
<td>842,000</td>
</tr>
</tbody>
</table>

\[
d_H \left( P(Y_+), P(Y_-) \right) = \sqrt{\sum_{i=1}^D \left( \sqrt{P(Y_+ | X_i)} - \sqrt{P(Y_- | X_i)} \right)^2}
\]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>Low</td>
<td>0</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>1</td>
</tr>
</tbody>
</table>
\[ D(x, x') = \sqrt{\sum |x_d - x'_d|^2} \]
\[ D(x, x') = \sum_{d} 1_{x_d = x'_d} \]
\[ D(x, x') = \sqrt[\text{p}]{{\sum |x_d - x'_d|^\text{p}}} \]
\[ a^* = \sum_{i=0}^{N} \lambda_i y_i \vec{x}_i \]

\[ \sum_{i=0}^{N} \lambda_i y_i = 0 \]
\begin{align*}
\omega_0 &= 1 - 8a \\
3a + 1 - 8a &= -1 \\
5a &= 2 \\
a &= \frac{2}{5}
\end{align*}

\begin{align*}
\omega_0 &= 1 - 8 \frac{2}{5} = \frac{5 - 16}{5} \\
\omega_0 &= -\frac{11}{5}
\end{align*}
\[ \hat{\omega} = \left( \frac{2}{5}, \frac{4}{5} \right) \]

\[ g(\bar{x}) = \frac{2}{5} x_1 + \frac{4}{5} x_2 - \frac{11}{5} \]

\[ g(\bar{x}) = x_1 + 2x_2 - 5.5 \]
Minimize : $\Phi(w) = \frac{1}{2}w^Tw$
Subject to : $d_i(w^Tx_i + b) \geq 1 \quad \forall i$

Introduce slack variables $\xi_i \geq 0$

Minimize : $\Phi(w) = \frac{1}{2}w^Tw$
Subject to : $d_i(w^Tx_i + b) \geq 1 - \xi_i \quad \forall i$

Polynomial: $K_p(X,Y) = (1 + X \cdot Y)^p$

Radial Basis Function (RBF) or Gaussian: $K_r(X,Y) = e^{-\frac{1}{2\sigma^2}\|X-Y\|^2_2}$

Hyperbolic Tangent: $K_s(X,Y) = \tanh(\beta_0 X \cdot Y + \beta_1)$
Support = \frac{\text{freq}(X,Y)}{N}

Confidence = \frac{\text{freq}(X,Y)}{\text{freq}(X)}

Lift = \frac{\text{Support}}{\text{Supp}(X) \times \text{Supp}(Y)}
Support = $\frac{freq(X,Y)}{N}$

Overall Support over the support for individual attributes

Lift = \frac{Support}{Supp(X) \times Supp(Y)}

The frequency of different attributes contributing to the purpose of a single attribute in isolation

Confidence = \frac{freq(X,Y)}{freq(X)}
\[ \sum_{k=1}^{d-1} \binom{d}{k} \times \sum_{j=1}^{d-k} \binom{d-k}{j} \]

\[ \sum_{k=1}^{d-1} \binom{d}{k} \times \sum_{j=1}^{d-k} \binom{d-k}{j} = 3^d - 2^{d+1} + 1 \]
$C_k$: Candidate itemset of size $k$
$L_k$: frequent itemset of size $k$
$L_1 = \{\text{frequent items}\}$;
for ($k = 1; L_k \neq \varnothing; k++$) do begin
  $C_{k+1}$ = candidates generated from $L_k$;
  for each transaction $t$ in database do
    increment the count of all candidates in $C_{k+1}$
    that are contained in $t$
  $L_{k+1}$ = candidates in $C_{k+1}$ with min_support
  end
return $\bigcup_k L_k$;
Clustering

Hierarchical Clustering
- agglomerative
- divisive

Partitional Clustering

Raw Data

Clustering Algorithm

Data grouped into Clusters
- Cluster A
- Cluster B
- Cluster C

Similarity Scale

$X_1$, $X_2$, $X_3$, $X_4$, $X_5$
CHAPTER 9

Machine learning has many Learning models categories:
- Association rule based
- Decision tree based
- Ensemble method based
- Deep learning based (ANN)
- Clustering method based
- Regression analysis based
- Dimensionality reduction based
- Kernel based
- Instance based

\[ \sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2} \]

\[ \sum_{x} \sum_{y} P(X = x \text{ and } Y = y) = 1 \]

\[ P(X = x) = \sum_{y} P(X = x, Y = y) = \sum_{y} P(X = x \mid Y = y)P(Y = y) \]
<table>
<thead>
<tr>
<th>(X1)</th>
<th>(X2)</th>
<th>(Y (X1+X2))</th>
<th>(Z (X1-X2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
<td>-2</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>5</td>
<td>-3</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>6</td>
<td>-4</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>7</td>
<td>-5</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>5</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>6</td>
<td>-2</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>7</td>
<td>-3</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>8</td>
<td>-4</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>7</td>
<td>-1</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>8</td>
<td>-2</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>9</td>
<td>-3</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>9</td>
<td>-1</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>10</td>
<td>-2</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>11</td>
<td>-1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>Z</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-------</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>-5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-4</td>
<td>1/36</td>
<td>1/36</td>
<td></td>
</tr>
<tr>
<td>-3</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
</tr>
<tr>
<td>-2</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
</tr>
<tr>
<td>-1</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
</tr>
<tr>
<td>0</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
</tr>
<tr>
<td>1</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
</tr>
<tr>
<td>2</td>
<td>1/36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1/36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1/36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1/36</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1/36</td>
<td>1/18</td>
<td>1/12</td>
</tr>
</tbody>
</table>

**Discrete**
- Bernoulli
- Binomial
- Negative binomial
- Geometric
- Poisson

**Continuous**
- Normal
- T distribution
- Gamma
- Chi Square
- Exponential
- Weibull
- F Distribution

<table>
<thead>
<tr>
<th>Possible demand X</th>
<th>Number of days</th>
<th>Probability [P(X)]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
<td>0.06</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>0.14</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>0.24</td>
</tr>
<tr>
<td>6</td>
<td>14</td>
<td>0.28</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>0.2</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>0.08</td>
</tr>
</tbody>
</table>
### Table 1: Possible demand X, Probability [P(X)], Weighted Value [XP(X)]

<table>
<thead>
<tr>
<th>Possible demand X</th>
<th>Probability [P(X)]</th>
<th>Weighted Value [XP(X)]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.06</td>
<td>0.18</td>
</tr>
<tr>
<td>4</td>
<td>0.14</td>
<td>0.56</td>
</tr>
<tr>
<td>5</td>
<td>0.24</td>
<td>1.2</td>
</tr>
<tr>
<td>6</td>
<td>0.28</td>
<td>1.68</td>
</tr>
<tr>
<td>7</td>
<td>0.2</td>
<td>1.4</td>
</tr>
<tr>
<td>8</td>
<td>0.08</td>
<td>0.64</td>
</tr>
<tr>
<td><strong>1</strong></td>
<td></td>
<td><strong>E(X) = 5.66</strong></td>
</tr>
</tbody>
</table>

### Table 2: Possible demand X, Probability [P(X)], Weighted Value [XP(X)], Squared demand (X^2), Weighted Square [X^2P(X)]

<table>
<thead>
<tr>
<th>Possible demand X</th>
<th>Probability [P(X)]</th>
<th>Weighted Value [XP(X)]</th>
<th>Squared demand (X^2)</th>
<th>Weighted Square [X^2P(X)]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.06</td>
<td>0.18</td>
<td>9</td>
<td>0.54</td>
</tr>
<tr>
<td>4</td>
<td>0.14</td>
<td>0.56</td>
<td>16</td>
<td>2.24</td>
</tr>
<tr>
<td>5</td>
<td>0.24</td>
<td>1.2</td>
<td>25</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>0.28</td>
<td>1.68</td>
<td>36</td>
<td>10.08</td>
</tr>
<tr>
<td>7</td>
<td>0.2</td>
<td>1.4</td>
<td>49</td>
<td>9.8</td>
</tr>
<tr>
<td>8</td>
<td>0.08</td>
<td>0.64</td>
<td>64</td>
<td>5.12</td>
</tr>
<tr>
<td><strong>1</strong></td>
<td></td>
<td><strong>E(X) = 5.66</strong></td>
<td><strong>E(X^2) = 33.78</strong></td>
<td></td>
</tr>
</tbody>
</table>

\[
\lim_{n \to \infty} \frac{\lambda^r}{r!} \left( \frac{1}{n} \right)^r \left( \frac{n}{\lambda} \right)^{n-r} = 0
\]

\[
\lim_{n \to \infty} \frac{n!}{r!(n-r)!} \frac{\lambda^r}{n^r} \left( \frac{1}{\lambda} \right)^{n-r} = 0
\]

\[
P(X = r) = \frac{\lambda^r e^{-\lambda}}{r!}
\]

\[
P(X = 0) = \frac{e^{-\lambda} \lambda^0}{0!} = e^{-\lambda}
\]
The function shown is a normal distribution:

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

- This is called Bernoulli trial
- Simple coin toss
- Flip the coin a number of times
- Binomial distribution
- Increase the number of times the coin is flipped, you get a continuous similar to normal. Increase the flips but simultaneously reduce the probability of success, you get Poisson
- Normal
- Poisson
- Instead of looking at the number of successes, if you ask how much time, space it takes to look at the first success, you get a continuous function
- Exponential
\[ p(A | B) = \frac{p(A)p(B | A)}{p(B)} \]

\[ p(H | D) = \frac{p(H)p(D | H)}{p(D)} \]

**TrainMultinomialNB(C, D)**

1. \(V \leftarrow \text{ExtractVocabulary}(D)\)
2. \(N \leftarrow \text{CountDocs}(D)\)
3. for each \(c \in C\) do
   4. \(N_c \leftarrow \text{CountDocsInClass}(D, c)\)
   5. \(\text{prior}[c] \leftarrow N_c / N\)
   6. \(\text{text}_c \leftarrow \text{ConcatenateTextOfAllDocsInClass}(D, c)\)
   7. for each \(t \in V\) do
      8. \(T_{ct} \leftarrow \text{CountTokensOfTerm}(\text{text}_c, t)\)
   9. for each \(t \in V\) do
      10. \(\text{condprob}[t][c] \leftarrow \frac{T_{ct} + 1}{\sum_t (T_{ct} + 1)}\)
4. return \(V, \text{prior}, \text{condprob}\)

**ApplyMultinomialNB(C, V, prior, condprob, d)**

1. \(W \leftarrow \text{ExtractTokensFromDoc}(V, d)\)
2. for each \(c \in C\) do
   3. \(\text{score}[c] \leftarrow \log \text{prior}[c]\)
   4. for each \(t \in W\) do
      5. \(\text{score}[c] \leftarrow \text{score}[c] + \log \text{condprob}[t][c]\)
4. return \(\arg \max_{c \in C} \text{score}[c]\)


\textbf{TrainBernoulliNB(}C, D\textbf{)}

1. \( V \leftarrow \text{ExtractVocabulary}(D) \)
2. \( N \leftarrow \text{CountDocs}(D) \)
3. for each \( c \in C \)
   4. do \( N_c \leftarrow \text{CountDocsInClass}(D, c) \)
   5. \( \text{prior}[c] \leftarrow N_c / N \)
   6. for each \( t \in V \)
   7. do \( N_{ct} \leftarrow \text{CountDocsInClassContainingTerm}(D, c, t) \)
   8. \( \text{condprob}[t][c] \leftarrow (N_{ct} + 1) / (N_c + 2) \)
9. return \( V, \text{prior}, \text{condprob} \)

\textbf{ApplyBernoulliNB(}C, V, prior, condprob, d\textbf{)}

1. \( V_d \leftarrow \text{ExtractTermsFromDoc}(V, d) \)
2. for each \( c \in C \)
   3. do \( \text{score}[c] \leftarrow \log \text{prior}[c] \)
   4. for each \( t \in V \)
   5. do if \( t \in V_d \)
   6. then \( \text{score}[c] += \log \text{condprob}[t][c] \)
   7. else \( \text{score}[c] += \log(1 - \text{condprob}[t][c]) \)
8. return \( \arg \max_{c \in C} \text{score}[c] \)

\( d = \{t_1, \ldots, t_k, \ldots, t_n\}, t_k \in V \)

\( d = \{e_1, \ldots, e_i, \ldots, e_M\}, \)

\( e_i \in \{0,1\} \)

\( \hat{P}(X = t | c) \)

\( \hat{P}(U_i = e | c) \)

\( \hat{P}(c) \prod_{k \leq k \leq n} \hat{P}(X = t_k | c) \)

\( \hat{P}(c) \prod_{i \in x} \hat{P}(U_i = e_i | c) \)

\( \hat{P}(X = \text{the} | c) \approx 0.05 \)

\( \hat{P}(U_{\text{the}} = 1 | c) \approx 1.0 \)
\[ \rho_{xy} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{n} \]

\[ \text{Cor}(x, y) = \frac{\rho_{xy}}{\sigma_x \sigma_y} \]
Mean = \mu = \int_{a}^{b} xP(x)dx

Variance = \sigma^2 = \int_{a}^{b} x^2 P(x)dx
<table>
<thead>
<tr>
<th>Company A</th>
<th>Company B</th>
<th>Company C</th>
<th>A (returns)</th>
<th>B (returns)</th>
<th>C (returns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>498.3</td>
<td>243.7</td>
<td>250.15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>515.25</td>
<td>245.75</td>
<td>250.25</td>
<td>0.0340157</td>
<td>-0.008412</td>
<td>0.0004</td>
</tr>
<tr>
<td>506.4</td>
<td>242.7</td>
<td>250.25</td>
<td>-0.017176</td>
<td>-0.012411</td>
<td>0</td>
</tr>
<tr>
<td>504.8</td>
<td>244.65</td>
<td>253.55</td>
<td>-0.00316</td>
<td>0.0080346</td>
<td>0.013187</td>
</tr>
<tr>
<td>536.95</td>
<td>250.95</td>
<td>236.8</td>
<td>0.0636886</td>
<td>0.0257511</td>
<td>-0.06606</td>
</tr>
<tr>
<td>512.55</td>
<td>227.4</td>
<td>219.1</td>
<td>-0.045442</td>
<td>-0.0938434</td>
<td>-0.07475</td>
</tr>
<tr>
<td>525.65</td>
<td>240.3</td>
<td>206.5</td>
<td>0.0255585</td>
<td>0.0567282</td>
<td>-0.05751</td>
</tr>
<tr>
<td>538.95</td>
<td>243.8</td>
<td>216.45</td>
<td>0.025302</td>
<td>0.0145651</td>
<td>0.048184</td>
</tr>
<tr>
<td>510.45</td>
<td>235.25</td>
<td>217.9</td>
<td>-0.052881</td>
<td>-0.0350697</td>
<td>0.006699</td>
</tr>
<tr>
<td>503</td>
<td>238.35</td>
<td>215.3</td>
<td>-0.014595</td>
<td>0.0131775</td>
<td>-0.01193</td>
</tr>
<tr>
<td>500.75</td>
<td>231.4</td>
<td>218.15</td>
<td>-0.004473</td>
<td>-0.0291588</td>
<td>0.013237</td>
</tr>
<tr>
<td>496.45</td>
<td>228.15</td>
<td>217.55</td>
<td>-0.008587</td>
<td>-0.0140449</td>
<td>-0.00275</td>
</tr>
<tr>
<td>492.3</td>
<td>219.7</td>
<td>215.15</td>
<td>-0.008359</td>
<td>-0.037037</td>
<td>-0.01103</td>
</tr>
<tr>
<td>496.6</td>
<td>218</td>
<td>205.4</td>
<td>0.00873845</td>
<td>-0.0077378</td>
<td>-0.04532</td>
</tr>
<tr>
<td></td>
<td>A (returns)</td>
<td>B (returns)</td>
<td>C (returns)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0340157</td>
<td>0.008412</td>
<td>0.0004</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.017176</td>
<td>-0.012411</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.00316</td>
<td>0.0080346</td>
<td>0.013187</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0636886</td>
<td>0.0257511</td>
<td>-0.06606</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.045442</td>
<td>-0.0938434</td>
<td>-0.07475</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0255585</td>
<td>0.0567282</td>
<td>-0.05751</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.025302</td>
<td>0.0145651</td>
<td>0.048184</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.052881</td>
<td>-0.0350697</td>
<td>0.006699</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.014595</td>
<td>0.0131775</td>
<td>-0.01193</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.004473</td>
<td>-0.0291588</td>
<td>0.013237</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.008587</td>
<td>-0.0140449</td>
<td>-0.00275</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.008359</td>
<td>-0.037037</td>
<td>-0.01103</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0087345</td>
<td>-0.0077378</td>
<td>-0.04532</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sum</strong></td>
<td><strong>0.0026265</strong></td>
<td><strong>-0.1026342</strong></td>
<td><strong>-0.18764</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Mean</strong></td>
<td><strong>0.000202</strong></td>
<td><strong>-0.0078949</strong></td>
<td><strong>-0.01443</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>A (returns)</th>
<th>B (returns)</th>
<th>C (returns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.05288</td>
<td>-0.09384</td>
<td>-0.07475</td>
<td></td>
</tr>
<tr>
<td>-0.045442</td>
<td>-0.03704</td>
<td>-0.06606</td>
<td></td>
</tr>
<tr>
<td>-0.01718</td>
<td>-0.03507</td>
<td>-0.05751</td>
<td></td>
</tr>
<tr>
<td>-0.01459</td>
<td>-0.02916</td>
<td>-0.04532</td>
<td></td>
</tr>
<tr>
<td>-0.00859</td>
<td>-0.01404</td>
<td>-0.01193</td>
<td></td>
</tr>
<tr>
<td>-0.00836</td>
<td>-0.01241</td>
<td>-0.01103</td>
<td></td>
</tr>
<tr>
<td><strong>Median</strong></td>
<td><strong>-0.00447</strong></td>
<td><strong>-0.00774</strong></td>
<td><strong>-0.00275</strong></td>
</tr>
<tr>
<td>-0.00316</td>
<td>0.008035</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0.008735</td>
<td>0.008412</td>
<td>0.0004</td>
<td></td>
</tr>
<tr>
<td>0.025302</td>
<td>0.013177</td>
<td>0.006699</td>
<td></td>
</tr>
<tr>
<td>0.025558</td>
<td>0.014565</td>
<td>0.013187</td>
<td></td>
</tr>
<tr>
<td>0.034016</td>
<td>0.025751</td>
<td>0.013237</td>
<td></td>
</tr>
<tr>
<td>0.063689</td>
<td>0.056728</td>
<td>0.048184</td>
<td></td>
</tr>
</tbody>
</table>
\[
E\left(\sum_{i=1}^{n} a_i x_i\right) = \sum_{i=1}^{n} a_i E(x_i)
\]

\[
v \sum a_i x_i = \sum a_i^2 v(x_i) + 2 \sum \sum a_i a_j \text{cov}(x_i, x_j)
\]

<table>
<thead>
<tr>
<th></th>
<th>Gold</th>
<th>IT</th>
<th>Bank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns</td>
<td>15</td>
<td>25</td>
<td>17</td>
</tr>
<tr>
<td>SD</td>
<td>5</td>
<td>15</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Gold</th>
<th>IT</th>
<th>Bank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gold</td>
<td>1</td>
<td>-1</td>
<td>-0.5</td>
</tr>
<tr>
<td>IT</td>
<td>-1</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>Bank</td>
<td>-0.5</td>
<td>0.5</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Portfolio</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portfolio2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>24.2</td>
</tr>
<tr>
<td>Portfolio3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>23.4</td>
</tr>
<tr>
<td>Portfolio4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>22.6</td>
</tr>
<tr>
<td>Portfolio5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>21.8</td>
</tr>
</tbody>
</table>
Null hypothesis:

$$H_0: \mu_1 = \mu_2 = \mu_3$$

Direct path way

We want to measure if smoking has any direct effect on coronary heart diseases

Data brought out another variable age that can also play a major role in coronary heart diseases

Back door path way

And that Non-smokers are younger
Regression Analysis

Key Assumptions:
- Number of cases
- Multicollinearity and singularity
- Accuracy of data
- Missing data
- Outliers
- Homoscedasticity
- Linearity
- Normality

Histogram:
- Age (years)
- Std. Dev = 13.33
- Mean = 56.4
- N = 1207.00
<table>
<thead>
<tr>
<th>Trns #</th>
<th>Commision ($)</th>
<th>error</th>
<th>error $^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5000</td>
<td>-5000</td>
<td>25000000</td>
</tr>
<tr>
<td>2</td>
<td>17000</td>
<td>7000</td>
<td>490000000</td>
</tr>
<tr>
<td>3</td>
<td>11000</td>
<td>1000</td>
<td>10000000</td>
</tr>
<tr>
<td>4</td>
<td>8000</td>
<td>-2000</td>
<td>40000000</td>
</tr>
<tr>
<td>5</td>
<td>14000</td>
<td>4000</td>
<td>160000000</td>
</tr>
<tr>
<td>6</td>
<td>5000</td>
<td>-5000</td>
<td>250000000</td>
</tr>
</tbody>
</table>

**SSE** = 120000000
\[ \hat{y}_i = b_0 + b_1 x_i \]

\[ b_1 = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} \]

\( \bar{x} = \text{mean of the independent variable} \quad x_i = \text{value of independent variable} \)

\( \bar{y} = \text{mean of the dependent variable} \quad y_i = \text{value of dependent variable} \)

<table>
<thead>
<tr>
<th>Transaction ($)</th>
<th>Commission ($)</th>
<th>Txn Deviation</th>
<th>Comm Deviation</th>
<th>Dev Product</th>
<th>Square Txn Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>34000</td>
<td>5000</td>
<td>-40000</td>
<td>-5000</td>
<td>200000000</td>
<td>1600000000</td>
</tr>
<tr>
<td>108000</td>
<td>17000</td>
<td>34000</td>
<td>7000</td>
<td>238000000</td>
<td>1156000000</td>
</tr>
<tr>
<td>64000</td>
<td>11000</td>
<td>-10000</td>
<td>1000</td>
<td>-10000000</td>
<td>1000000000</td>
</tr>
<tr>
<td>88000</td>
<td>8000</td>
<td>14000</td>
<td>-2000</td>
<td>-28000000</td>
<td>1960000000</td>
</tr>
<tr>
<td>99000</td>
<td>14000</td>
<td>25000</td>
<td>4000</td>
<td>100000000</td>
<td>6250000000</td>
</tr>
<tr>
<td>51000</td>
<td>5000</td>
<td>-23000</td>
<td>-5000</td>
<td>115000000</td>
<td>5290000000</td>
</tr>
<tr>
<td>Mean</td>
<td>74000</td>
<td>10000</td>
<td>Sum</td>
<td>615000000</td>
<td>4205000000</td>
</tr>
</tbody>
</table>

\[ \hat{y}_i = b_0 + b_1 x_i \quad b_0 = -0.8188 \quad b_1 = 0.1462 \]

intercept \quad slope

\[ \hat{y}_i = -0.8188 + 0.1462x \]

OR

\[ \hat{y}_i = 0.1462x - 0.8188 \]
\[ y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon \]

linear parameters

error

\[ E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p \]

error term assumed to be zero

\[ \hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p \]

\( b_0, b_1, b_2, \ldots, b_p \) are the estimates of \( \beta_0, \beta_1, \beta_2, \ldots, \beta_p \)

\( \hat{y} \) = predicted value of the dependent variable
\[ p_i = \frac{1}{1 + e^{-(a + bx_i)}} \]  
(This is called the logistic response function)

\[ \log \frac{p_i}{1-p_i} = a + bx_i \]

\[ \frac{p_i}{1-p_i} = e^{\hat{a} + \hat{b}x_i} \]

\[ \hat{p}_i = \frac{e^{\hat{a} + \hat{b}x_i}}{1 + e^{\hat{a} + \hat{b}x_i}} \]
Neural Networks work on the concept of a Human brain, which is the basic unit of a Neuron. Neurons have different types, including:
- Binary threshold neurons
- Linear neurons
- Idealized neurons
- Rectified Linear Neurons
- Sigmoid neurons
- Stochastic binary neurons
\[ y = b + \sum_{i=1}^{n} (w_i x_i) \]

\[ z = b + \sum_{i} x_i w_i \]

\[ y = \begin{cases} 
  z & \text{if } z > 0 \\
  0 & \text{otherwise} 
\end{cases} \]

\[ z = \sum_{i=1}^{n} (w_i x_i) \]

\[ z = b + \sum_{i=1}^{n} (w_i x_i) \]
$z = b + \sum_{i=1}^{n} (w_i x_i)$

$y = \frac{1}{1 + e^{-z}}$
\[ z = b + \sum_{i=1}^{n} (w_i x_i) \]

\[ p(s=1) = \frac{1}{1 + e^{-z}} \]
3 hidden neurons

6 hidden neurons

20 hidden neurons

Image/Video Pixels → Layer 1 → Layer 2 → Layer 3 → Simple Classifier

Feature representation

Input data

Lee et al., ICML 2009; CACM 2011

3rd layer “Objects”

2nd layer “Object parts”

1st layer “Edges”

Pixels
Neural Networks
- Elman Networks
- Hopfield Networks
- Jordan Networks
- Multilayer Fully Connected Feedforward Networks (MLP)
- Radial Basis Function Network (RBF)
- Dynamic Learning Vector Quantization (DLVQ) Networks

\[
C = \frac{1}{2n} \sum_x \| y(x) - a^L(x) \|^2
\]

\[
C = \frac{1}{n} \sum_x C_x
\]

\[
C_x = \frac{1}{2} \| y - a^L \|^2
\]
\[ C = \frac{1}{2} \| y - a^L \|^2 = \frac{1}{2} \sum_j (y_j - a^L_j)^2 \]

\[ \partial C / \partial w^l_{jk} \text{ and } \partial C / \partial b^l_j \]

\[ \delta^l_j \equiv \frac{\partial C}{\partial z^l_j} \]

Measures how the activation function changes at the current position in the network

\[ \delta^L_j = \left( \frac{\partial C}{\partial a^L_j} \right) \sigma'(z^L_j) \]

Measures how the cost function changes based on the jth activation output
\[ \delta^l = ((w^{l+1})^T \delta^{l+1}) \circ \sigma'(z^l) \]

- **Transpose of the weight matrix at \( l+1 \), this moves the error backwards through the network**
- **Hadamard product, this moves the error backward through the activation layer \( l \)**

\[
\begin{bmatrix}
1 \\
2
\end{bmatrix} \circ \begin{bmatrix}
3 \\
4
\end{bmatrix} = \begin{bmatrix}
1 \times 3 \\
2 \times 4
\end{bmatrix} = \begin{bmatrix}
3 \\
8
\end{bmatrix}
\]

\[
\frac{\partial C}{\partial b^l_j} = \delta^l_j
\]

\[
\frac{\partial C}{\partial b} = \delta
\]

\[
\frac{\partial C}{\partial w^l_{jk}} = a^{l-1}_k \delta^l_j
\]

\[ z^l = w^l a^{l-1} + b^l \text{ and } a^l = \sigma(z^l) \]
A diagram illustrating a neuron's structure and function. The neuron receives inputs from dendrites, which synapse with the cell body, producing an output axon. The mathematical representation of the neuron's operation is shown as:

\[ f \left( \sum_{i} w_i x_i + b \right) \]

where \( x_i \) are the inputs, \( w_i \) are the weights, \( b \) is the bias, and \( f \) is the activation function.
CHAPTER 12

Reinforcement learning

Decision making

Functional Approximation

Markov Decision Process (MDP)

Temporal Difference (TD)

Learning Subfields
- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- Semi-supervised Learning
- Deep Learning
Agent and environment interact at discrete time steps: $t = 0, 1, 2, \ldots$

Agent observes state at step $t$: $s_t \in S$ produces action at step $t$: $a_t \in A(s_t)$

gets resulting reward: $r_{t+1} \in \mathbb{R}$ and resulting next state: $s_{t+1}$

$$\pi^* = \arg\max_\pi \left( E \left[ \sum_{t=0}^{\infty} r^t R(s_t)/\pi \right] \right)$$

$$R(s)\neq U^\pi(s) = E \left[ \sum_{t=0}^{\infty} r^t R(s_t) / \pi, s_0 = s \right]$$

$$\pi^* = \arg\max_\pi E \left[ \sum_{s_1} T(s, a, s_1) U(s_1) \right]$$

$$U(s) = R(s) + \gamma \max_\pi E \left[ \sum_{s_1} T(s, a, s_1) U(s_1) \right]$$
\[ V^*(s) = \max_{a \in \mathcal{A}(s)} Q^{\pi^*}(s, a) \]
\[ = \max_a E_{\pi^*} \left\{ R_t \mid s_t = s, a_t = a \right\} \]
\[ = \max_a E_{\pi^*} \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a \right\} \]
\[ = \max_a E_{\pi^*} \left\{ r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_t = s, a_t = a \right\} \]
\[ = \max_a E \left\{ r_{t+1} + \gamma V^*(s_{t+1}) \mid s_t = s, a_t = a \right\} \]
\[ = \max_{a \in \mathcal{A}(s)} \sum_{s'} P_{ss'}^a \left[ R_{ss'}^a + \gamma V^*(s') \right] . \]

1. Initialization
   \[ V(s) \in \mathbb{R} \text{ and } \pi(s) \in \mathcal{A}(s) \text{ arbitrarily for all } s \in \mathcal{S} \]

2. Policy Evaluation
   Repeat
   \[ \Delta \leftarrow 0 \]
   For each \( s \in \mathcal{S} \):
   \[ v \leftarrow V(s) \]
   \[ V(s) \leftarrow \sum_{s'} P_{ss'}^{\pi(s)} \left[ R_{ss'}^{\pi(s)} + \gamma V(s') \right] \]
   \[ \Delta \leftarrow \max(\Delta, |v - V(s)|) \]
   until \( \Delta < \theta \) (a small positive number)

3. Policy Improvement
   \[ \text{policy-stable} \leftarrow \text{true} \]
   For each \( s \in \mathcal{S} \):
   \[ b \leftarrow \pi(s) \]
   \[ \pi(s) \leftarrow \arg \max_a \sum_{s'} P_{ss'}^a \left[ R_{ss'}^a + \gamma V(s') \right] \]
   If \( b \neq \pi(s) \), then \( \text{policy-stable} \leftarrow \text{false} \)
   If \( \text{policy-stable} \), then stop; else go to 2
Initialize $V$ arbitrarily, e.g., $V(s) = 0$, for all $s \in S^+$

Repeat

$\Delta \leftarrow 0$

For each $s \in S$:

$v \leftarrow V(s)$

$V(s) \leftarrow \max_a \sum_{s'} \mathcal{P}_{ss'}^a \left[ \mathcal{R}_{ss'}^a + \gamma V(s') \right]$

$\Delta \leftarrow \max(\Delta, |v - V(s)|)$

until $\Delta < \theta$ (a small positive number)

Output a deterministic policy, $\pi$, such that

$$
\pi(s) = \arg \max_a \sum_{s'} \mathcal{P}_{ss'}^a \left[ \mathcal{R}_{ss'}^a + \gamma V(s') \right]
$$

$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[ r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right]$

Initialize $Q(s, a)$ arbitrarily

Repeat (for each episode):

Initialize $s$

Choose $a$ from $s$ using policy derived from $Q$ (e.g., $\varepsilon$-greedy)

Repeat (for each step of episode):

Take action $a$, observe $r$, $s'$

Choose $a'$ from $s'$ using policy derived from $Q$ (e.g., $\varepsilon$-greedy)

$$
Q(s, a) \leftarrow Q(s, a) + \alpha \left[ r + \gamma Q(s', a') - Q(s, a) \right]
$$

$s \leftarrow s'$; $a \leftarrow a'$;

until $s$ is terminal

$$
Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[ r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right].
$$
Initialize $Q(s, a)$ arbitrarily
Repeat (for each episode):
    Initialize $s$
    Repeat (for each step of episode):
        Choose $a$ from $s$ using policy derived from $Q$ (e.g., $\varepsilon$-greedy)
        Take action $a$, observe $r, s'$
        $Q(s, a) \leftarrow Q(s, a) + \alpha \left[ r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]$
        $s \leftarrow s'$
    until $s$ is terminal

Initialize $\rho$ and $Q(s, a)$, for all $s, a$, arbitrarily
Repeat forever:
    $s \leftarrow$ current state
    Choose action $a$ in $s$ using behavior policy (e.g., $\varepsilon$-greedy)
    Take action $a$, observe $r, s'$
    $Q(s, a) \leftarrow Q(s, a) + \alpha \left[ r - \rho + \max_{a'} Q(s', a') - Q(s, a) \right]$
    If $Q(s, a) = \max_a Q(s, a)$, then:
    $\rho \leftarrow \rho + \beta \left[ r - \rho + \max_{a'} Q(s', a') - \max_a Q(s, a) \right]$
Truth label: -1

error rate: 35%, acc: 65%

\[
\sum_{i=2}^{3} \binom{3}{i} \varepsilon^i (1 - \varepsilon)^{3-i} = 3 \times 0.35^2 \times 0.65 + 1 \times 0.35^3 \times 1 = 0.2817
\]

\[
\sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1 - \varepsilon)^{25-i} = 0.06
\]
The base classifiers are identical (perfectly correlated)

The base classifiers are independent

---

Congratulations! Movies we think You will ❤️

Add movies to your Queue, or Rate ones you've seen for even better suggestions.
Ensemble method for Supervised learning
Combining the “learning” technique

Labelled Data

Derived Subset dataset D1
Classifier C1

Derived Subset dataset D2
Classifier C2

Derived Subset dataset Dn
Classifier Cn

Combines the learning from the labelled data

Ensemble Model

Unlabelled data
Prediction

Training dataset

Testing dataset
Ensemble method for Supervised learning
Combining the "consensus" technique

Labelled Data

Derived Subset dataset D1 → Classifier C1
Derived Subset dataset D2 → Classifier C2
Derived Subset dataset Dn → Classifier Cn

Ensemble Model

Unlabeled data

Combines the predictions based on majority voting → Final Prediction

Training dataset

Testing dataset

Training Sample → h₁
Weighted Sample → h₂
...
Weighted Sample → hₜ

H
\[ H(x) = \text{sign}\left( \sum_{t=1}^{T} \alpha_t h_t(x) \right) \]

**Given** \((x_1, y_1), \ldots, (x_m, y_m)\) where \(x \in \mathcal{X}, y \in \{-1, +1\}\)

**Initialise** weights \(D_1(i) = 1/m\)

**Iterate** \(t=1, \ldots, T:\)
- Train weak learner using distribution \(D_t\)
- Get weak classifier: \(h_t : \mathcal{X} \rightarrow \mathbb{R}\)
- Choose \(\alpha_t \in \mathbb{R}\)
- Update: \(D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_t h_t(x_i))}{Z_t}\)

where \(Z_t\) is a normalization factor (chosen so that \(D_{t+1}\) will be a distribution), and \(\alpha_t:\)

\[
\alpha_t = \frac{1}{2} \ln \left( \frac{1 - \varepsilon_t}{\varepsilon_t} \right) > 0
\]

**Output** – the final classifier

\[ H(x) = \text{sign}\left( \sum_{t=1}^{T} \alpha_t h_t(x) \right) \]

---

**Data points class label**
- \(y_t = +1\)
- \(y_t = -1\)

**Data points weight value**
- \(w_t = 1\)
Data points class label

- $y_t = +1$
- $y_t = -1$

Data points weight value

- $w_t = 1$

$H = p(\text{error}) = 0.5$

Data points class label

- $y_t = +1$
- $y_t = -1$

Weight value updated

- $w_t = w_t \exp(-y_t, H)$
Data points class label

- $y_t = +1$
- $y_t = -1$

Weight value updated

$w_t = w_t \exp\{-y_t, H\}$
BAGGING

Training phase
1. Initialize the parameters
   • $D = \emptyset$, the ensemble.
   • $L$, the number of classifiers to train.
2. For $k = 1, \ldots, L$
   • Take a bootstrap sample $S_k$ from $Z$.
   • Build a classifier $D_k$ using $S_k$ as the training set.
   • Add the classifier to the current ensemble, $D = D \cup D_k$.
3. Return $D$.

Classification phase
4. Run $D_1, \ldots, D_L$ on the input $x$.
5. The class with the maximum number of votes is chosen as the label for $x$. 
Expected Error = Variance + Bias

\[ E[(\hat{f}(X) - E[f(X)])^2] + E[(\hat{f}(X) - E[\hat{f}(X)])^2] \]

- \( E[(\hat{f}(X) - E[f(X)])^2] \): the expected discrepancy between the estimated and true function
- \( E[(\hat{f}(X) - E[\hat{f}(X)])^2] \): is squared discrepancy between averaged estimated and true function
- \( E[f(X)] \): expected divergence of the estimated function vs. its average value

Original Data: 1 2 3 4 5 6 7 8 9 10

Round 1: 7 8 10 8 2 5 10 10 5 9

Round 2: 1 4 9 1 2 3 2 7 3 2

Round 3: 1 8 5 10 5 5 9 6 3 7
**Require:** $I$ (an inducer), $T$ (the number of iterations), $S$ (the training set), $d$ (weighting distribution).

**Ensure:** $M_t; t = 1, \ldots, T$

1: $t \leftarrow 1$
2: repeat
3: $S_t \leftarrow S$ with random weights drawn from $d$.
4: Build classifier $M_t$ using $I$ on $S_t$
5: $t \leftarrow t + 1$
6: until $t > T$

At each node:

- choose some small subset of variables at random
- find a variable (and a value for that variable) which optimizes the split
Algorithm 1 Friedman’s Gradient Boost algorithm

Inputs:
- input data \((x, y)_{i=1}^{N}\)
- number of iterations \(M\)
- choice of the loss-function \(\Psi(y, f)\)
- choice of the base-learner model \(h(x, \theta)\)

Algorithm:
1: initialize \(\hat{f}_0\) with a constant
2: for \(t = 1\) to \(M\) do
3:    compute the negative gradient \(g_t(x)\)
4:    fit a new base-learner function \(h(x, \theta_t)\)
5:    find the best gradient descent step-size \(\rho_t\):

\[\rho_t = \arg\min_{\rho} \sum_{i=1}^{N} \Psi[y_i, \hat{f}_{t-1}(x_i) + \rho h(x_i, \theta_t)]\]
6:    update the function estimate:

\[\hat{f}_t \leftarrow \hat{f}_{t-1} + \rho_t h(x, \theta_t)\]
7: end for
Ensemble method for Unsupervised learning
Combining the “consensus” technique

- Clustering Algorithm 1
- Clustering Algorithm 2
- Clustering Algorithm n

Unlabeled data

Combines the partitioning based on consensus

Final Clustering

base clustering models

<table>
<thead>
<tr>
<th></th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$v_2$</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$v_3$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$v_4$</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$v_5$</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>$v_6$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

objects

they may not represent the same cluster!

The goal: get the consensus clustering
Data Architectures for modern Machine learning
Semantified Common Data Repository

Data Ingestion
Data Management
Query Management

- Low Cost, High Performance Storage
- Flexible, Easy-to-Use Data Organization
- Performance-Optimized Analytics
- Automation of most manual Development and Query Activities
- Self-Service End-User Features
- Intelligent Processing

delivering