MODULE 1
HDFS BASICS, RUNNING EXAMPLE PROGRAMS AND BENCHMARKS, HADOOP MAPREDUCE FRAMEWORK, MAPREDUCE PROGRAMMING

Hadoop Distributed File System Basics

- HADOOP DISTRIBUTED FILE SYSTEM DESIGN FEATURES
- HDFS COMPONENTS:
  - ✓ HDFS Block Replication
  - ✓ HDFS Safe Mode
  - ✓ Rack Awareness
  - ✓ NameNode High Availability
  - ✓ HDFS NameNode Federation
  - ✓ HDFS Checkpoints and Backups
  - ✓ HDFS Snapshots
  - ✓ HDFS NFS Gateway
- HDFS USER COMMANDS

- Google File System (GFS)
- Hadoop Distributed File System (HDFS)
- HDFS block size is typically 64MB or 128MB

Important aspects of HDFS:

- ✓ The write-once/read-many design is intended to facilitate streaming reads.
- ✓ Files may be appended, but random seeks are not permitted. There is no caching of data.
- ✓ Converged data storage and processing happen on the same server nodes.
- ✓ "*Moving computation is cheaper than moving data*."
- ✓ A reliable file system maintains multiple copies of data across the cluster. Consequently, failure of a single node (or even a rack in a large cluster) will not bring down the file system.
- ✓ A specialized file system is used, which is not designed for general use.

HDFS COMPONENTS

- ✓ The design of HDFS is based on two types of nodes: a NameNode and multiple DataNodes
- ✓ NameNode manages all the metadata needed to store and retrieve the actual data from the DataNodes.
- ✓ No data is actually stored on the NameNode.

- ✓ The design is a master/slave architecture in which the master (NameNode) manages the file system namespace and regulates access to files by clients.
- ✓ File system namespace operations such as opening, closing, and renaming files and directories are all managed by the NameNode
- ✓ The NameNode also determines the mapping of blocks to DataNodes and handles DataNode failures
- ✓ The NameNode manages block creation, deletion, and replication
- ✓ The slaves (DataNodes) are responsible for serving read and write
- ✓ requests from the file system to the clients

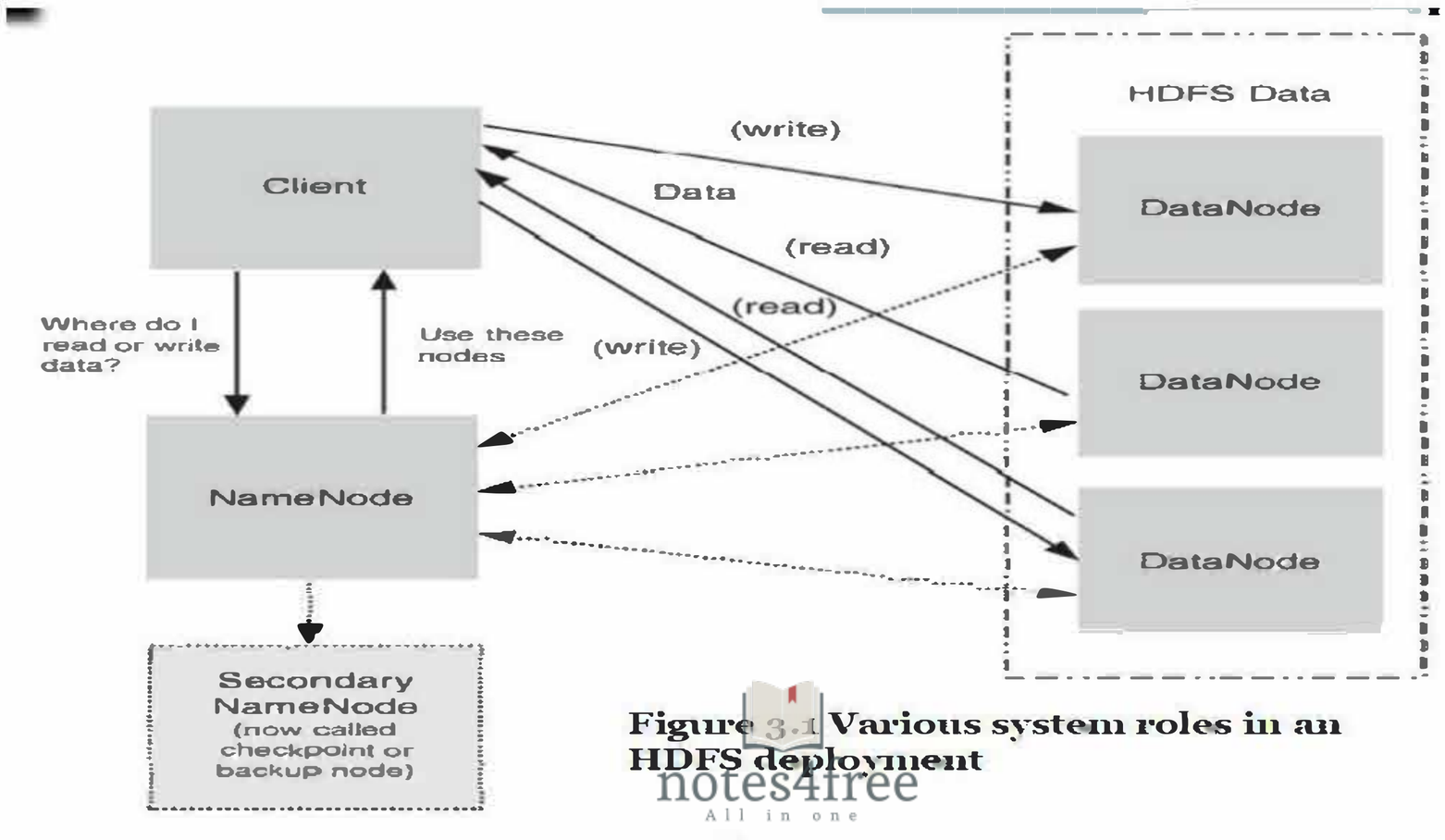


Figure 3.1 Various system roles in an HDFS deployment

HDFS Safe Mode

- When the NameNode starts, it enters a read-only *safe mode* where blocks cannot be replicated or deleted. Safe Mode enables the NameNode to perform two important processes:

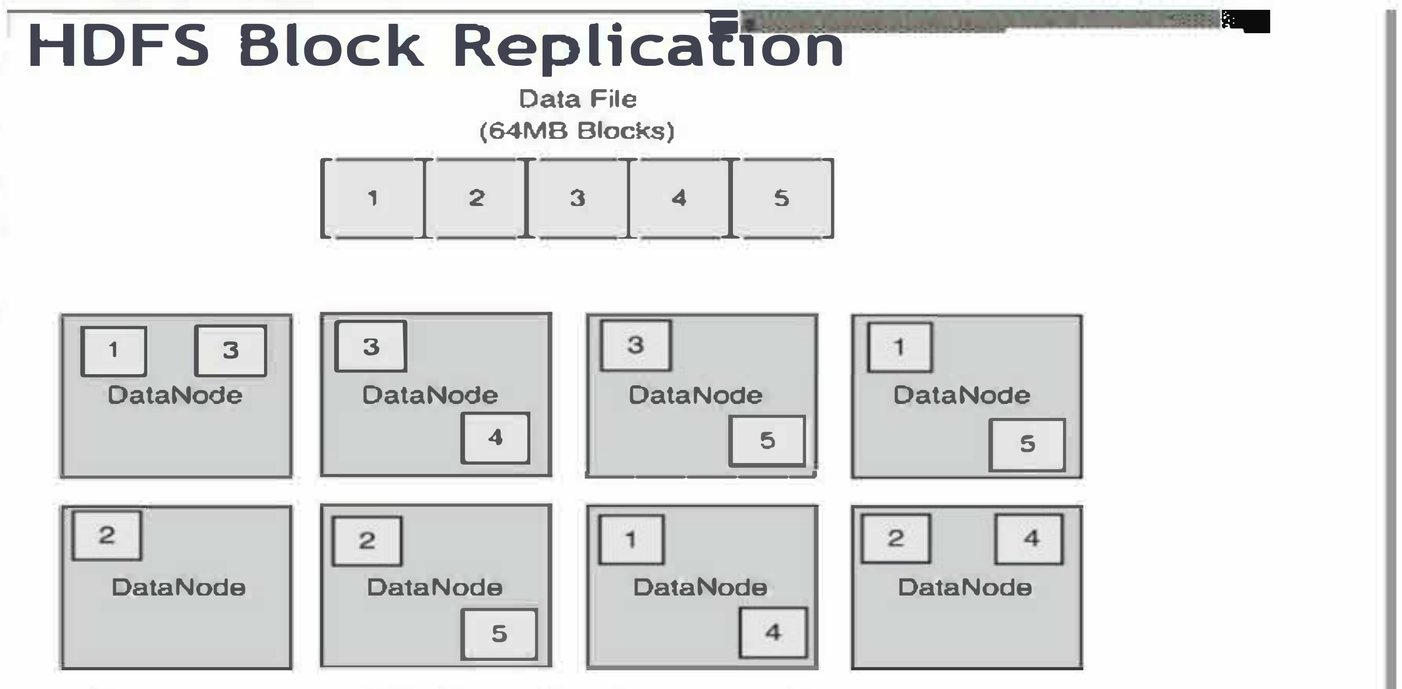


Figure 3.2 HDFS block replication example

1. The previous file system state is reconstructed by loading the fsimage file into memory and replaying the edit log.
2. The mapping between blocks and data nodes is created by waiting for enough of the

DataNodes to register so that at least one copy of the data is available. Not all DataNodes are required to register before HDFS exits from Safe Mode.The registration process may continue for some time.

- HDFS may also enter Safe Mode for maintenance using the hdfs dfsadmin-safemode command or when there is a file system issue that must be addressed by the administrator.

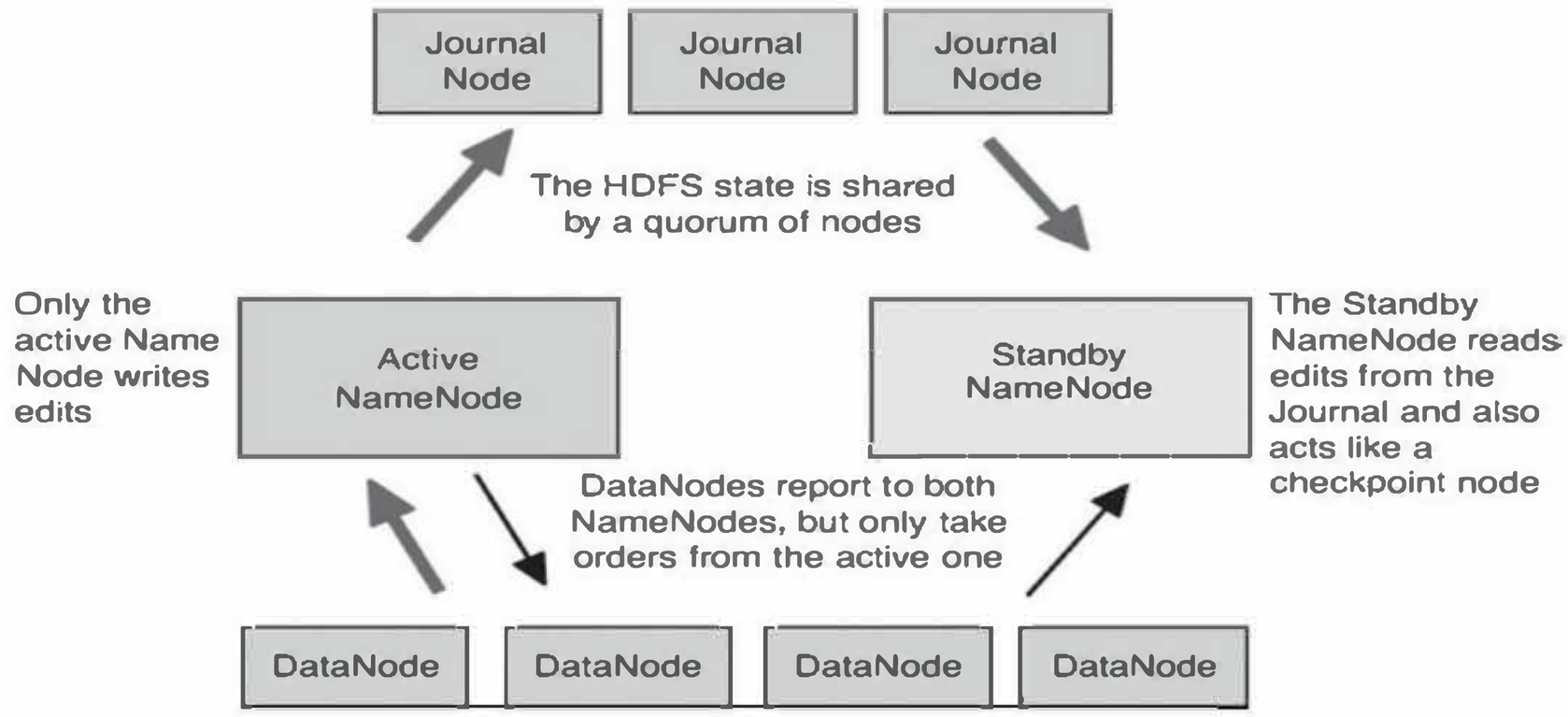
Rack Awareness

- Rack awareness deals with data locality.
- The main design goals of Hadoop MapReduce is to move the computation to the data. Assuming that most data center networks do not offer full bisection bandwidth, a typical Hadoop cluster will exhibit three levels of data locality:
  1. Data resides on the local machine (best).
  2. Data resides in the same rack (better).
  3. Data resides in a different rack (good).
- When the YARN scheduler is assigning MapReduce containers to work as mappers, it will try to place the container first on the local machine, then on the same rack, and finally on another rack.
- In addition, the NameNode tries to place replicated data blocks on multiple racks for improved fault tolerance. In such a case, an entire rack failure will not cause data loss or stop HDFS from working. Performance may be degraded, however.
- HDFS can be made rack-aware by using a user-derived script that enables the master node to map the network topology of the cluster.
- A default Hadoop installation assumes all the nodes belong to the same (large) rack. In that case, there is no option 3.

NameNode High Availability

- The NameNode was a single point of failure that could bring down the entire Hadoop cluster.
- NameNode hardware often employed redundant power supplies and storage to guard against such problems, but it was still susceptible to other failures.
- The solution was to implement NameNode High Availability (HA) as a means to provide true failover service.
- As shown in Figure 3.3, an HA Hadoop cluster has two (or more) separatNameNode machines.
- Each machine is configured with exactly the same software.
- One of the NameNode machines is in the Active state, and the other is in the Standby state.
- Like a single NameNode cluster,the Active NameNode is responsible for all client HDFS operations in the cluster.
- The Standby NameNode maintains enough state to provide a fast failover (if required).

Figure 3.3 HDFS High Availability design



HDFS Checkpoints and Backups

- *The NameNode stores the metadata of the HDFS file system in a file called fsimage.*
- File systems modifications are written to an edits log file, and at startup the NameNode merges the edits into a new fsimage.
- The Secondary NameNode or CheckpointNode periodically fetches edits from the NameNode, merges them, and returns an updated fsimage to the NameNode.
- An HDFS BackupNode is similar, but also maintains an up-to-date copy of the file system namespace both in memory and on disk.
- Unlike a CheckpointNode, the BackupNode does not need to download the fsimage and edits files from the active NameNode because it already has an up-to-date namespace state in memory.
- A NameNode supports one BackupNode at a time.
- No CheckpointNodes may be registered if a Backup node is in use.

HDFS Snapshots

- HDFS snapshots are similar to backups, but are created by administrators using the hdfs dfs snapshot command.
- HDFS snapshots are read-only point-in-time copies of the file system.

They offer the following features:

1. Snapshots can be taken of a sub-tree of the file system or the entire file system.
2. Snapshots can be used for data backup, protection against user errors, and disaster recovery.
3. Snapshot creation is instantaneous.
4. Blocks on the DataNodes are not copied, because the snapshot files record the block list and the file size. There is no data copying, although it appears to the user that there are duplicate files.
5. Snapshots do not adversely affect regular HDFS operations.

HDFS NFS Gateway

- The HDFS NFS Gateway supports NFSv3 and enables HDFS to be mounted as part of the client's local file system.
- Users can browse the HDFS file system through their local file systems that provide an NFSv3 client compatible operating system. This feature offers users the following capabilities:
    - Users can easily download/upload files from/to the HDFS file system to/from their local file system.
    - Users can stream data directly to HDFS through the mount point. Appending to a file is supported, but random write capability is not supported.

HDFS commands

- Syntax
- **hdfs [--config confdir] COMMAND**
- where COMMAND is one of:

1. **dfs** :run a file system command on the file systems supported in Hadoop.
2. **namenode –format:** format the DFS file system
3. **secondarynamenode** : run the DFS secondary namenode
4. **namenode:** run the DFS namenode
5. **journalnode:** run the DFS journalnode
6. **zkfc**: run the ZK Failover Controller daemon
7. **datanode** : run a DFS datanode
8. **dfsadmin:** run a DFS admin client
9. **haadmin** : run a DFS HA admin client
10. **fsck:** run a DFS file system checking utility
11. **balancer** : run a cluster balancing utility
12. **jmxget :** get JMX exported values from NameNode or DataNode.
13. **mover:** run a utility to move block replicas across storage types
14. **oiv:** apply the *offline fsimage viewer* to an fsimage
15. **oiv_legacy :** apply the *offline fsimage viewer* to an legacy fsimage
16. **oev** : apply the *offline edits viewer* to an edits file
17. **fetchdt:** fetch a delegation token from the NameNode
18. **getconf** : get config values from configuration
19. **groups** : get the groups which users belong to
20. **snapshotDiff** : diff two snapshots of a directory or diff the current directory contents with a snapshot
21. **lsSnapshottableDir** : list all snapshottable dirs owned by the current user Use -help to see options
22. **portmap :** run a portmap service
23. **nfs3 :** run an NFS version 3 gateway
24. **cacheadmin** : configure the HDFS cache
25. **crypto** : configure HDFS encryption zones

26. **storagepolicies :** get all the existing block storage policies
27. **version :** print the version

General HDFS Commands

- **hdfs version**
- **hdfs dfs**

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|resourcemanager:port> specify a ResourceManager
- -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.

List Files in HDFS

To list the files in the root HDFS directory

$ hdfs dfs -ls /

To list files in your home directory

$ hdfs dfs -ls

The same result can be obtained by issuing the following command

$ hdfs dfs -ls /user/hdfs

Make a Directory in HDFS

To make a directory in HDFS

$ hdfs dfs -mkdir stuff

Copy Files to HDFS

To copy a file from your current local directory into HDFS

$ hdfs dfs -put test stuff

Copy Files from HDFS

Files can be copied back to your local file system

$ hdfs dfs -get stuff/test test-local

Copy Files within HDFS

The following command will copy a file in HDFS

$ hdfs dfs -cp stuff/test test.hdfs

Delete a File within HDFS

To delete the HDFS file test.dhfs that was created previously

$ hdfs dfs -rm test.hdfs

Moved:'hdfs://limulus:8020/user/hdfs/stuff/test'totrashat:hdfs://limulus:8020/user/hdfs/.Trash/Current

• Note that when the fs.trash.interval option is set to a non-zero value in core-site.xml, all deleted files are moved to the user's .Trash directory. This can be avoided by including the - skipTrash option.

$ hdfs dfs -rm -skipTrash stuff/test

Deleted stuff/test

Delete a Directory in HDFS

To delete the HDFS directory stuff and all its contents

$ hdfs dfs -rm -r -skipTrash stuff

Deleted stuff

Get an HDFS Status Report

users can get an abbreviated HDFS status report using the following command

$ hdfs dfsadmin -report

Hadoop MapReduce Framework

THE MAPREDUCE MODEL

- The MapReduce computation model provides a very powerful tool for many applications and is more common than most users realize.
- There are two stages: a mapping stage and a reducing stage.
- The MapReduce model is inspired by the map and reduce functions commonly used in many functional programming languages.
- The functional nature of MapReduce has some important properties:
  1. Data flow is in one direction (map to reduce). It is possible to use the output of a reduce step as the input to another MapReduce process.
  2. As with functional programing, the input data are not changed. By applying the mapping and reduction functions to the input data, new data are produced. In effect, the original state of the Hadoop data lake is always preserved.
- Distributed (parallel) implementations of MapReduce enable large amounts of data to be analyzed quickly.
- The mapper process is fully scalable and can be applied to any subset of the input data. Results from multiple parallel mapping functions are then combined in the reducer phase.
- Hadoop accomplishes parallelism by using a distributed file system (HDFS) to slice and spread data over multiple servers.
- MapReduce will try to move the mapping tasks to the server that contains the data slice. Results from each data slice are then combined in the reducer step.

MAPREDUCE PARALLEL DATA FLOW

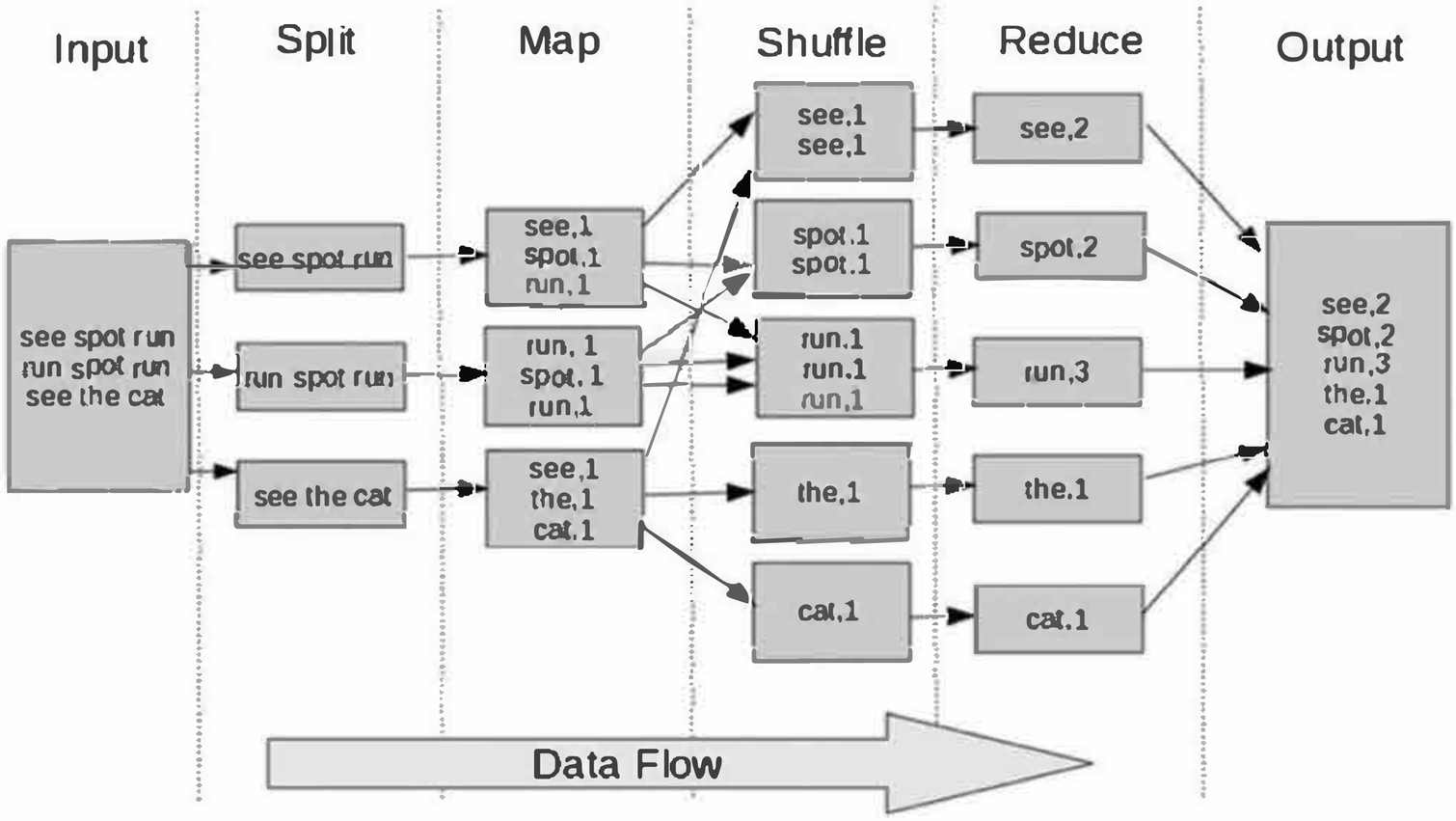
Parallel execution of MapReduce requires other steps in addition to the mapper and reducer processes. The basic steps are as follows:

1. **Input Splits.**
   - The default data chunk or block size is 64MB.
   - Thus, a 500MB file would be broken into 8 blocks and written to different machines in the cluster.
   - The data are also replicated on multiple machines (typically three machines).
   - The input splits used by MapReduce are logical boundaries based on the input data.
2. **Map Step.**
   - The mapping process is where the parallel nature of Hadoop comes into play.
   - For large amounts of data, many mappers can be operating at the same time.
   - The user provides the specific mapping process.
   - MapReduce will try to execute the mapper on the machines where the block resides.
   - Because the file is replicated in HDFS, the least busy node with the data will be chosen.
   - If all nodes holding the data are too busy, MapReduce will try to pick a node that is closest to the node that hosts the data block (a characteristic called rack-awareness).
   - The last choice is any node in the cluster that has access to HDFS.
3. **Combiner Step.**
   - It is possible to provide an optimization or prereduction as part of the map stage where **key–value pairs** are combined prior to the next stage. The combiner stage is optional.
4. **Shuffle Step.**
   - Before the parallel reduction stage can complete, all similar keys must be combined and counted by the same reducer process.
   - Therefore, results of the map stage must be collected by **key–value** pairs and shuffled to the same reducer process.
   - If only a single reducer process is used, the shuffle stage is not needed.
5. **Reduce Step.**
   - The final step is the actual reduction.
   - In this stage, the data reduction is performed as per the programmer's design. The reduce step is also optional.
   - The results are written to HDFS.
   - Each reducer will write an output file.

For example, a MapReduce job running four reducers will create files called part-0000, part-0001, part-0002, and part-0003.

Figure 5.1 is an example of a simple Hadoop MapReduce data flow for a word count program. The map process counts the words in the split, and the reduce process calculates the total for each word. As mentioned earlier, the actual computation of the map and reduce stages are up to the programmer. The MapReduce data flow shown in Figure 5.1 is the same regardless of the specific map and reduce tasks.

Figure 5.1 Apache Hadoop parallel MapReduce data flow



The input to the MapReduce application is the following file in HDFS with three lines of text.

- The goal is to count the number of times each word is used.

  see spot run
  run spot run
  see the cat

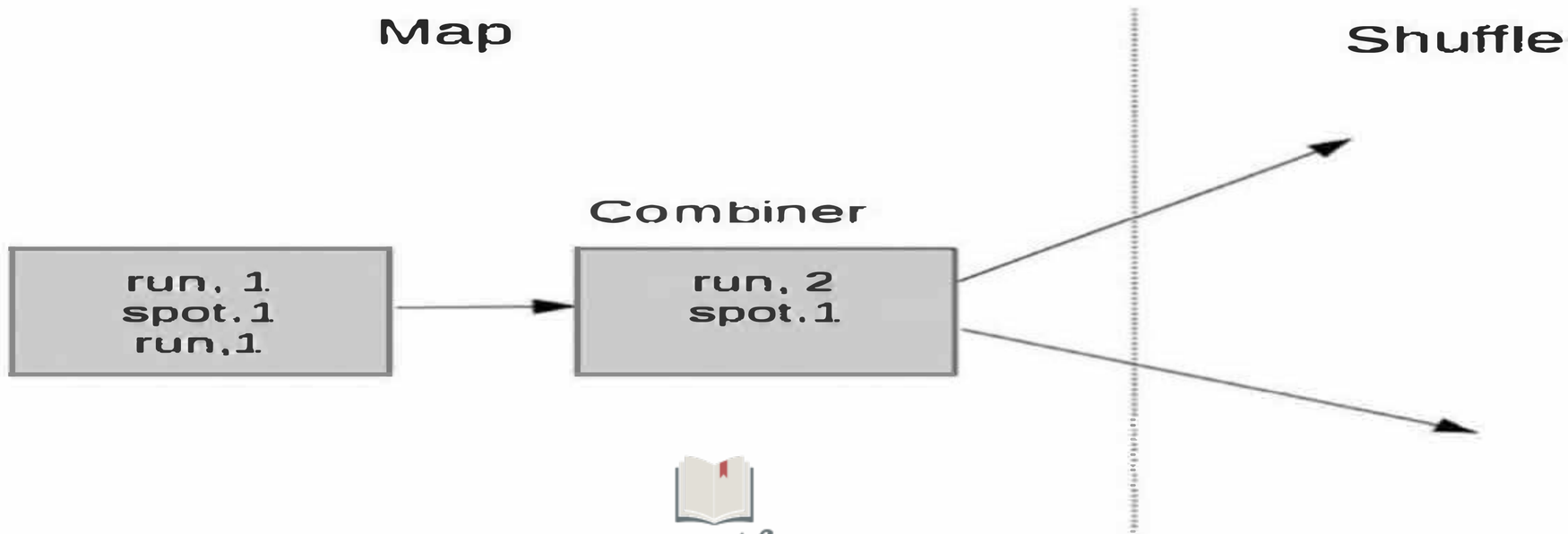
- The **first** thing MapReduce will do is create the data splits.
- For simplicity, each line will be one split.
- Since each split will require a map task, there are three ***mapper processes*** that count the number of words in the split. On a cluster, the results of each map task are written to local disk and not to HDFS.
- Next, similar keys need to be collected and sent to a reducer process.
- The shuffle step requires data movement and can be expensive in terms of processing time.
- Depending on the nature of the application, the amount of data that must be shuffled throughout the cluster can vary from small to large.
- Once the data have been collected and sorted by key, the reduction step can begin (even if only partial results are available).
- It is not necessary—and not normally recommended to have a reducer for each ***key–value***.
- In some cases, a single reducer will provide adequate performance; in other cases, multiple reducers may be required to speed up the reduce phase.

- The number of reducers is a tunable option for many applications.
- The final step is to write the output to HDFS.
- A combiner step enables some pre-reduction of the map output data. For instance, in the previous example, one map produced the following counts:

(run,1)
(spot,1)
(run,1)

As shown in Figure 5.2, the count for run can be combined into -(run,2) before the shuffle. This optimization can help minimize the amount of data transfer needed for the shuffle phase.

Figure 5.2 Adding a combiner process to the map step in MapReduce

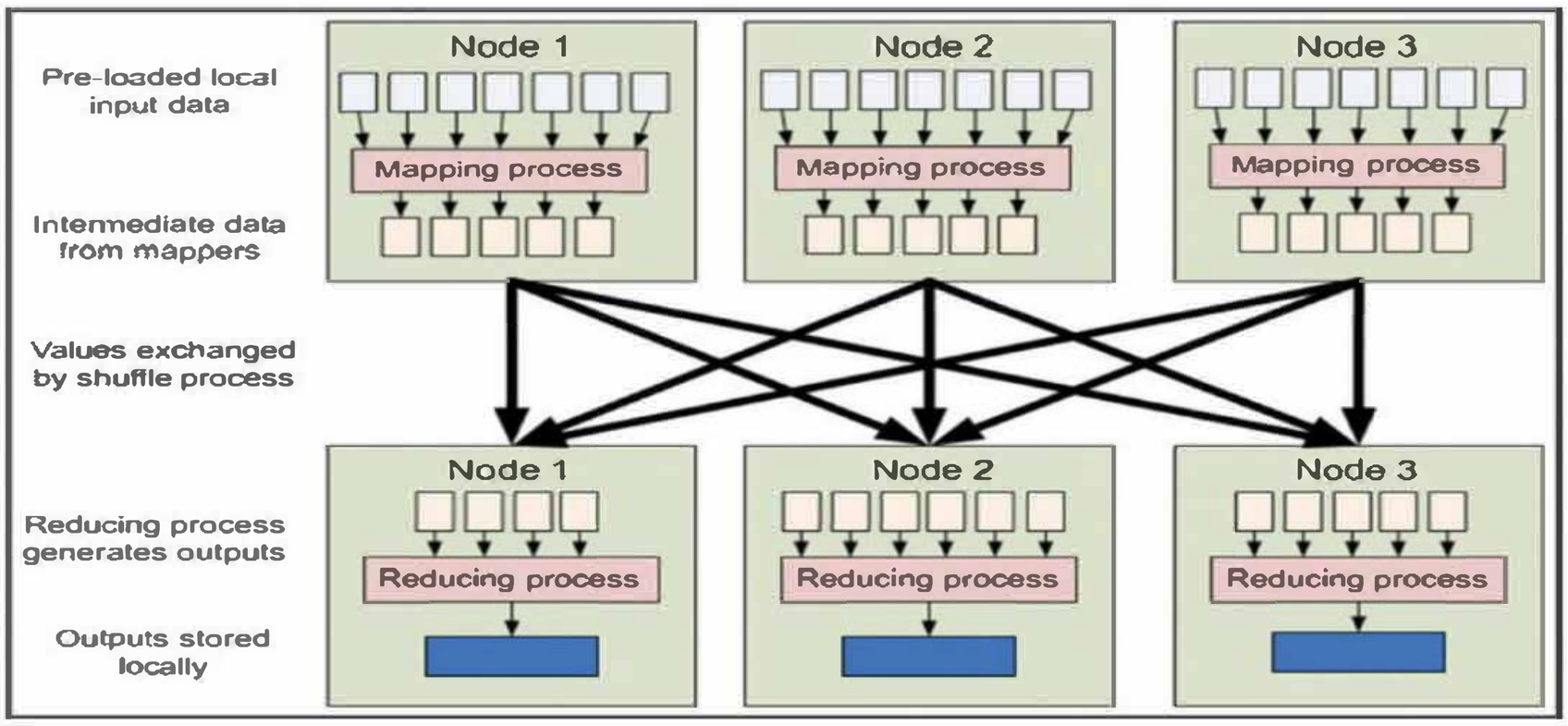


- The Hadoop YARN resource manager and the MapReduce framework determine the actual placement of mappers and reducers.
- The MapReduce framework will try to place the map task as close to the data as possible.
- It will request the placement from the YARN scheduler but may not get the best placement due to the load on the cluster.
- In general, nodes can run both mapper and reducer tasks.
- Indeed, the dynamic nature of YARN enables the work containers used by completed map tasks to be returned to the pool of available resources.

Figure 5.3 shows a simple three-node MapReduce process.

- Once the mapping is complete, the same nodes begin the reduce process.
- The shuffle stage makes sure the necessary data are sent to each mapper.
- Also note that there is no requirement that all the mappers complete at the same time or that the mapper on a specific node be complete before a reducer is started.
- Reducers can be set to start shuffling based on a threshold of percentage of mappers that have finished.

Figure 5.3 Process placement during MapReduce (Adapted from Yahoo Hadoop Documentation)



FAULT TOLERANCE AND SPECULATIVE EXECUTION

- One of the most interesting aspects of parallel MapReduce operation is the strict control of data flow throughout the execution of the program.

  For example, mapper processes do not exchange data with other mapper processes, and data can only go from mappers to reducers—not the other direction.
- The confined data flow enables MapReduce to operate in a fault-tolerant fashion.
- The design of MapReduce makes it possible to easily recover from the failure of one or many map processes.

  For example, should a server fail, the map tasks that were running on that machine could easily be restarted on another working server because there is no dependence on any other map task.
- In a similar fashion, failed reducers can be restarted.
- If reduce tasks remain to be completed on a down node, the MapReduce ApplicationMaster will need to restart the reducer tasks.
- If the mapper output is not available for the newly restarted reducer, then these map tasks will need to be restarted.
- This process is totally transparent to the user and provides a fault-tolerant system to run applications.

Speculative Execution

- One of the challenges with many large clusters is the inability to predict or manage unexpected system bottlenecks or failures. this problem represents a difficult challenge for large systems.
- Thus, it is possible that a congested network, slow disk controller, failing disk, high processor load, or some other similar problem might lead to slow performance without anyone noticing.

- When one part of a MapReduce process runs slowly, it ultimately slows down everything else because the application cannot complete until all processes are finished.
- The nature of the parallel MapReduce model provides an interesting solution to this problem. By starting a copy of a running map process without disturbing any other running mapper processes.

For example, suppose that as most of the map tasks are coming to a close, the ApplicationMaster notices that some are still running and schedules redundant copies of the remaining jobs on less busy or free servers. *Should the secondary processes finish first, the other first processes are then terminated (or vice versa). This process is known as* ***speculative execution.***

The same approach can be applied to reducer processes that seem to be taking a long time. Speculative execution can reduce cluster efficiency because redundant resources are assigned to applications that seem to have a slow spot.

Hadoop MapReduce Hardware

- The capability of Hadoop MapReduce and HDFS to tolerate server or even whole rack—failures can influence hardware designs.
- The use of commodity (typically x86_64) servers for Hadoop clusters has made low-cost, high-availability implementations of Hadoop possible for many data centers.
- Indeed, the Apache Hadoop philosophy seems to assume servers will always fail and takes steps to keep failure from stopping application progress on a cluster.
- The use of server nodes for both storage (HDFS) and processing (mappers, reducers) is somewhat different from the traditional separation of these two tasks in the data center.
- It is possible to build Hadoop systems and separate the roles (discrete storage and processing nodes).
- However, a majority of Hadoop systems use the general approach where servers enact both roles.
- Another interesting feature of dynamic MapReduce execution is the capability to tolerate dissimilar servers.
- That is, old and new hardware can be used together. Of course, large disparities in performance will limit the faster systems, but the dynamic nature of MapReduce execution will still work effectively on such systems.

Running MapReduce Examples

All Hadoop releases come with MapReduce example applications. Running the existing MapReduce examples is a simple process once the example files are located, that is.

For example, if you installed Hadoop version 2.6.0 from the Apache sources under /opt, the examples will be in the following directory:

```
/opt/hadoop-2.6.0/share/hadoop/mapreduce/
```

In other versions, the examples may be in

```
/usr/lib/hadoopmapreduce/
```

or some other location. The exact location of the example jar file can be found using the find command:

```
$ find / -name "hadoop-mapreduce-examples*.jar" -print
```

The environment variable called HADOOP_EXAMPLES can be defined as follows:

```
$ export HADOOP_EXAMPLES=/usr/hdp/2.2.4.2-2/hadoop-mapreduce
```

Listing Available Examples

A list of the available examples can be found by running the following command. In some cases, the version number may be part of the jar file.

```
$ yarn jar $HADOOP_EXAMPLES/hadoop-mapreduce-examples.jar
```

Running the Pi Example

The pi example calculates the digits of π using a quasi-Monte Carlo method. If you have not added users to HDFS, run these tests as user hdfs. To run the pi example with 16 maps and 1,000,000 samples per map, enter the following command:

```
$ yarn jar $HADOOP_EXAMPLES/hadoop-mapreduce-examples.jar pi 16
1000000
```


Running Basic Hadoop Benchmarks

Many Hadoop benchmarks can provide insight into cluster performance. The best benchmarks are always those that reflect real application performance.

Running the Terasort Test

The terasort benchmark sorts a specified amount of randomly generated data. This benchmark provides combined testing of the HDFS and MapReduce layers of a Hadoop cluster. A full terasort benchmark run consists of the following three steps:

1. Generating the input data via teragen program.
2. Running the actual terasort benchmark on the input data.
3. Validating the sorted output data via the teravalidate program.

1. Run teragen to generate rows of random data to sort.

```
$ yarn jar $HADOOP_EXAMPLES/hadoop-mapreduce-examples.jar teragen 500000000
/user/hdfs/TeraGen-50GB
```

2. Run terasort to sort the database.

```
$ yarn jar $HADOOP_EXAMPLES/hadoop-mapreduce-examples.jar terasort
/user/hdfs/TeraGen-50GB /user/hdfs/TeraSort-50GB
```

3. Run teravalidate to validate the sort.

```
$ yarn jar $HADOOP_EXAMPLES/hadoop-mapreduce-examples.jar teravalidate
/user/hdfs/TeraSort-50GB /user/hdfs/TeraValid-50GB
```

For example, the following command will instruct terasort to use four reducer tasks:

```
$ yarn jar $HADOOP_EXAMPLES/hadoop-mapreduce-examples.jar terasort -
Dmapred.reduce.tasks=4 /user/hdfs/TeraGen-50GB /user/hdfs/TeraSort-50GB
```

Also, do not forget to clean up the terasort data between runs (and after testing is finished). The following command will perform the cleanup for the previous example:

```
$ hdfs dfs -rm -r -skipTrash Tera*
```

Running the TestDFSIO Benchmark

The steps to run TestDFSIO are as follows:

1. Run TestDFSIO in write mode and create data.

```
$ yarn jar $HADOOP_EXAMPLES/hadoop-mapreduce-client-jobclienttests.
jar TestDFSIO -write -nrFiles 16 -fileSize 1000
```

Example results are as follows (date and time prefix removed).

```
fs.TestDFSIO: ----- TestDFSIO ----- : write
fs.TestDFSIO: Date & time: Thu May 14 10:39:33 EDT 2015
fs.TestDFSIO: Number of files: 16
fs.TestDFSIO: Total MBytes processed: 16000.0
fs.TestDFSIO: Throughput mb/sec: 14.890106361891005
fs.TestDFSIO: Average IO rate mb/sec: 15.690713882446289
fs.TestDFSIO: IO rate std deviation: 4.0227035201665595
fs.TestDFSIO: Test exec time sec: 105.631
```

2. Run TestDFSIO in read mode.

```
$ yarn jar $HADOOP_EXAMPLES/hadoop-mapreduce-client-jobclienttests.
jar TestDFSIO -read -nrFiles 16 -fileSize 1000
```

Example results are as follows (date and time prefix removed). The large standard deviation is due to the placement of tasks in the cluster on a small four-node cluster.

```
fs.TestDFSIO: ----- TestDFSIO ----- : read
fs.TestDFSIO: Date & time: Thu May 14 10:44:09 EDT 2015
fs.TestDFSIO: Number of files: 16
fs.TestDFSIO: Total MBytes processed: 16000.0
fs.TestDFSIO: Throughput mb/sec: 32.38643494172466
fs.TestDFSIO: Average IO rate mb/sec: 58.72880554199219
fs.TestDFSIO: IO rate std deviation: 64.60017624360337
```

```
fs.TestDFSIO: Test exec time sec: 62.798
```

3. Clean up the TestDFSIO data.

```
$ yarn jar $HADOOP_EXAMPLES/hadoop-mapreduce-client-jobclienttests.
jar TestDFSIO -clean
```

Managing Hadoop MapReduce Jobs

Hadoop MapReduce jobs can be managed using the mapred job command. The most important options for this command in terms of the examples and benchmarks are -list, -kill, and -status. In particular, if you need to kill one of the examples or benchmarks, you can use the mapred job -list command to find the job-id and then use mapred job -kill <jobid> to kill the job across the cluster. MapReduce jobs can also be controlled at the application level with the yarn application command. The possible options for mapred job are as follows:

```
$ mapred job
Usage: CLI <command> <args>
[-submit <job-file>]
[-status <job-id>]
[-counter <job-id> <group-name> <counter-name>]
[-kill <job-id>]
[-set-priority <job-id> <priority>]. Valid values for
priorities
are: VERY_HIGH HIGH NORMAL LOW VERY_LOW
[-events <job-id> <from-event-#> <#-of-events>]
[-history <jobHistoryFile>]
[-list [all]]
[-list-active-trackers]
[-list-blacklisted-trackers]
[-list-attempt-ids <job-id> <task-type> <task-state>]. Valid
values
for <task-type> are REDUCE MAP. Valid values for <task-state>
are
running, completed
[-kill-task <task-attempt-id>]
[-fail-task <task-attempt-id>]
[-logs <job-id> <task-attempt-id>]
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|resourcemanager:port> specify a ResourceManager
-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]
```

MapReduce Programming

The classic Java WordCount program for Hadoop is compiled and run.

Compiling and Running the Hadoop WordCount Example

The Apache Hadoop WordCount.java program for Hadoop version 2, is the equivalent of the C programing language helloworld.c example. It should be noted that two versions of this program can be found on the Internet. The Hadoop version1 example uses the older org.apache.hadoop.mapred API, while the Hadoop version2 example, shown here in Listing 6.1, uses the newer org.apache.hadoop.mapreduce API. If you experience errors compiling WordCount.java, double-check the source code and Hadoop versions.

WordCount.java

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public class WordCount {
      public static class TokenizerMapper
      extends Mapper<Object, Text, Text, IntWritable>{
      private final static IntWritable one = new IntWritable(1);
      private Text word = new Text();
```

```
public void map(Object key, Text value, Context context) throws
IOException, InterruptedException {
StringTokenizer itr = new StringTokenizer(value.toString());
while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
        }
}
//REDUCER
public static class IntSumReducer
extends Reducer<Text,IntWritable,Text,IntWritable> {
        private IntWritable result = new IntWritable();
        public void reduce(Text key, Iterable<IntWritable> values,Context
        context) throws IOException, InterruptedException {
                int sum = 0;
                for (IntWritable val : values) {
                        sum += val.get();
                }
                result.set(sum);
                context.write(key, result);
        }
}
public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "word count");
        job.setJarByClass(WordCount.class);
        job.setMapperClass(TokenizerMapper.class);
        job.setCombinerClass(IntSumReducer.class);
        job.setReducerClass(IntSumReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
        }
}
```


WordCount is a simple application that counts the number of occurrences of each word in a given input set. The MapReduce job proceeds as follows:

(input) <k1, v1> -> **map** -> <k2, v2> -> **combine** -> <k2, v2> -> **reduce** -> <k3, v3>

(output)

The mapper implementation, via the map method, processes one line at a time as provided by the specified TextInputFormat class. It then splits the line into tokens separated by whitespaces using the StringTokenizer and emits a key–value pair of <word, 1>. The relevant code section is as follows:

```
public void map(Object key, Text value, Context context) throws IOException,
InterruptedException {
        StringTokenizer itr = new StringTokenizer(value.toString());
        while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
        }
}
```

Given two input files with contents Hello World Bye World and Hello Hadoop Goodbye Hadoop, the WordCount mapper will produce two maps:

< Hello, 1>
< World, 1>
< Bye, 1>
< World, 1>
< Hello, 1>
< Hadoop, 1>
< Goodbye, 1>
< Hadoop, 1>

WordCount sets a mapper

```
job.setMapperClass(TokenizerMapper.class);
```

A combiner

```
job.setCombinerClass(IntSumReducer.class);
```

A reducer

```
job.setReducerClass(IntSumReducer.class);
```

Hence, the output of each map is passed through the local combiner (which sums the values in the same way as the reducer) for local aggregation and then sends the data on to the final reducer. Thus, each map above the combiner performs the following pre-reductions:

< Bye, 1>
< Hello, 1>

```
< World, 2>
< Goodbye, 1>
< Hadoop, 2>
< Hello, 1>
```

The reducer implementation, via the reduce method, simply sums the values, which are the occurrence counts for each key. The relevant code section is as follows:

```
public void reduce(Text key, Iterable<IntWritable> values, Context context)
throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
                sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
}
```

The final output of the reducer is the following:

```
< Bye, 1>
< Goodbye, 1>
< Hadoop, 2>
< Hello, 2>
< World, 2>`
```


To compile and run the program from the command line, perform the following steps:

1. Make a local wordcount_classes directory.

```
$ mkdir wordcount_classes
```

2. Compile the WordCount.java program using the 'hadoop classpath' command to include all the available Hadoop class paths.

```
$ javac -cp `hadoop classpath` -d wordcount_classes WordCount.java
```

3. The jar file can be created using the following command:

```
$ jar -cvf wordcount.jar -C wordcount_classes/
```

4. To run the example, create an input directory in HDFS and place a text file in the new directory. For this example, we will use the war-andpeacetxt:

```
$ hdfs dfs -mkdir war-and-peace-input
$ hdfs dfs -put war-and-peace.txt war-and-peace-input
```

5. Run the WordCount application using the following command:

```
$ hadoop jar wordcount.jar WordCount war-and-peace-input war-andpeace-
output
```

References:

Douglas Eadline,"**Hadoop 2 Quick-Start Guide: Learn the Essentials of Big Data Computing in the Apache Hadoop 2 Ecosystem",** 1stEdition, Pearson Education, 2016. ISBN-13: 978-9332570351

MODULE 2
ESSENTIAL HADOOP TOOLS, HADOOP YARN APPLICATIONS, MANAGING HADOOP WITH APACHE AMBARI, BASIC HADOOP ADMINISTRATION PROCEDURES

Essential Hadoop Tools

- **USING APACHE PIG**
- **USING APACHE HIVE**
- **USING APACHE SQOOP TO ACQUIRE RELATIONAL DATA**
- **USING APACHE FLUME TO ACQUIRE DATA STREAMS**
- **MANAGE HADOOP WORKFLOWS WITH APACHE OOZIE**
- **USING APACHE HBASE**

USING APACHE PIG

- Apache Pig is a high-level language that enables programmers to write complex MapReduce transformations using a simple scripting language.
- Pig Latin (the actual language) defines a set of transformations on a data set such as aggregate, join, and sort.
- Pig is often used to extract, transform, and load (ETL) data pipelines, quick research on raw data, and iterative data processing.

Apache Pig usage modes:

- The first is a local mode in which all processing is done on the local machine.
- The non-local (cluster) modes are MapReduce and Tez.
- These modes execute the job on the cluster using either the MapReduce engine or the optimized Tez engine. (Tez, which is Hindi for "speed," optimizes multistep Hadoop jobs such as those found in many Pig queries.)
- There are also interactive and batch modes available; they enable Pig applications to be developed locally in interactive modes, using small amounts of data, and then run at scale on the cluster in a production mode. The modes are summarized in Table 7.1.

Table 7.1 Apache Pig Usage Modes

	Local Mode	Tez Local Mode	MapReduce Mode	Tez Mode
Interactive Mode	Yes	Experimental	Yes	Yes
Batch Mode	Yes	Experimental	Yes	Yes

USING APACHE HIVE

- Apache Hive is a data warehouse infrastructure built on top of Hadoop for providing data summarization, ad hoc queries, and the analysis of large data sets using a SQL-like language called HiveQL.
- Hive is considered the de facto standard for interactive SQL queries over petabytes of data using Hadoop and offers the following features:
    1. Tools to enable easy data extraction, transformation, and loading (ETL)
    2. A mechanism to impose structure on a variety of data formats
    3. Access to files stored either directly in HDFS or in other data storage systems such as HBase
    4. Query execution via MapReduce and Tez (optimized MapReduce)
- Hive provides users who are already familiar with SQL the capability to query the data on Hadoop clusters.
- At the same time, Hive makes it possible for programmers who are familiar with the MapReduce framework to add their custom mappers and reducers to Hive queries.
- Hive queries can also be dramatically accelerated using the Apache Tez framework under YARN in Hadoop version 2.
- Sqoop is a tool designed to transfer data between Hadoop and relational databases.
- You can use Sqoop to import data from a relational database management system (RDBMS) into the Hadoop Distributed File System (HDFS), transform the data in Hadoop, and then export the data back into an RDBMS.
- Sqoop can be used with any Java Database Connectivity (JDBC)– compliant database and has been tested on Microsoft SQL Server, PostgresSQL, MySQL, and Oracle.
- In version 1 of Sqoop, data were accessed using connectors written for specific databases.
- Version 2 (in beta) does not support connectors or version 1 data transfer from a RDBMS directly to Hive or HBase, or data transfer from Hive or HBase to your RDBMS.
- Instead, version 2 offers more generalized ways to accomplish these tasks.

USING APACHE SQOOP TO ACQUIRE RELATIONAL DATA

Apache Sqoop Import and Export Methods

Figure 7.1 describes the Sqoop data import (to HDFS) process.

The data import is done in two steps.

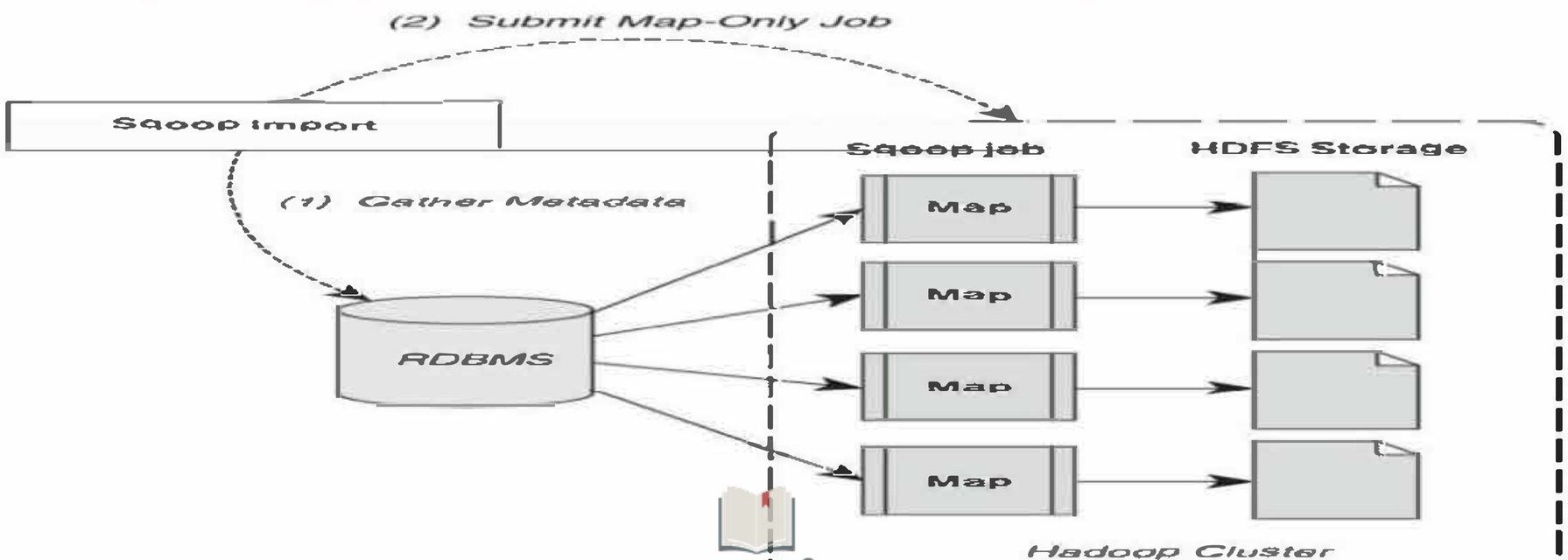
- In the **first step**, shown in the figure, Sqoop examines the database to gather the necessary metadata for the data to be imported.
- The **second step** is a map-only (no reduce step) Hadoop job that Sqoop submits to the cluster.
- This job does the actual data transfer using the metadata captured in the previous step.

- Note that each node doing the import must have access to the database.

Import method

- The imported data are saved in an HDFS directory.
- Sqoop will use the database name for the directory, or the user can specify any alternative directory where the files should be populated.
- By default, these files contain comma-delimited fields, with new lines separating different records.
- You can easily override the format in which data are copied over by explicitly specifying the field separator and record terminator characters.
- Once placed in HDFS, the data are ready for processing.

Figure 7.1 Two-step Apache Sqoop data import method

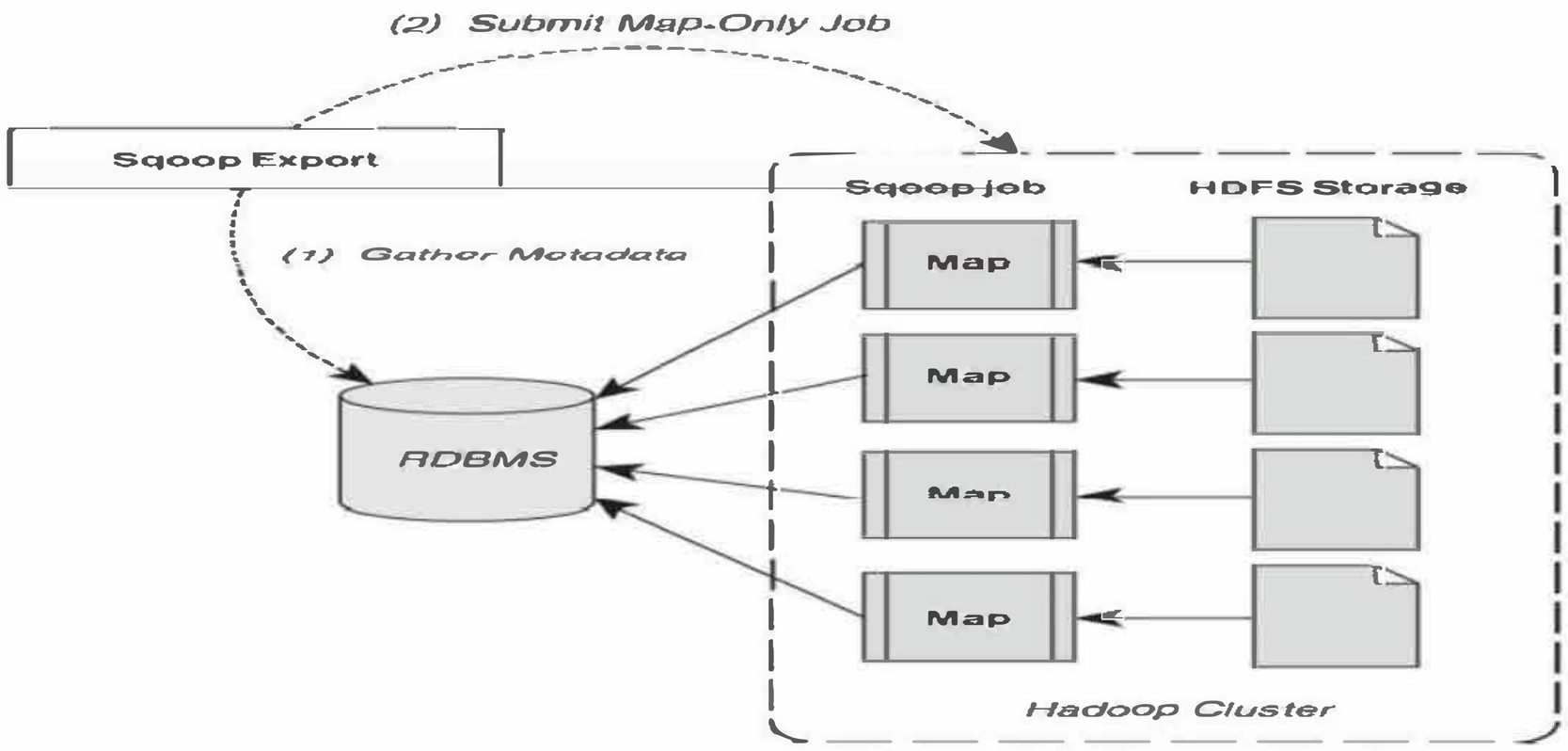


Export method

Data export from the cluster works in a similar fashion.

- The export is done in two steps, as shown in Figure 7.2. As in the import process, the **first step** is to examine the database for metadata.

Figure 7.2 Two-step Sqoop data export method



- The export step again uses a map-only Hadoop job to write the data to the database.

- Sqoop divides the input data set into splits, then uses individual map tasks to push the splits to the database. Again, this process assumes the map tasks have access to the database.

Apache Sqoop Version Changes

- Sqoop **Version 1** uses specialized connectors to access external systems.
- These connectors are often optimized for various RDBMSs or for systems that do not support JDBC.
- Connectors are plug-in components based on Sqoop's extension framework and can be added to any existing Sqoop installation.
- Once a connector is installed, Sqoop can use it to efficiently transfer data between Hadoop and the external store supported by the connector.
- By default, Sqoop version 1 includes connectors for popular databases such as MySQL, PostgreSQL, Oracle, SQL Server, and DB2.
- It also supports direct transfer to and from the RDBMS to HBase or Hive.
- In contrast, to streamline the Sqoop input methods, Sqoop **version 2** no longer supports specialized connectors or direct import into HBase or Hive.
- All imports and exports are done through the JDBC interface.
- Table 7.2 summarizes the changes from version 1 to version 2.
- Due to these changes, any new development should be done with Sqoop version 2.

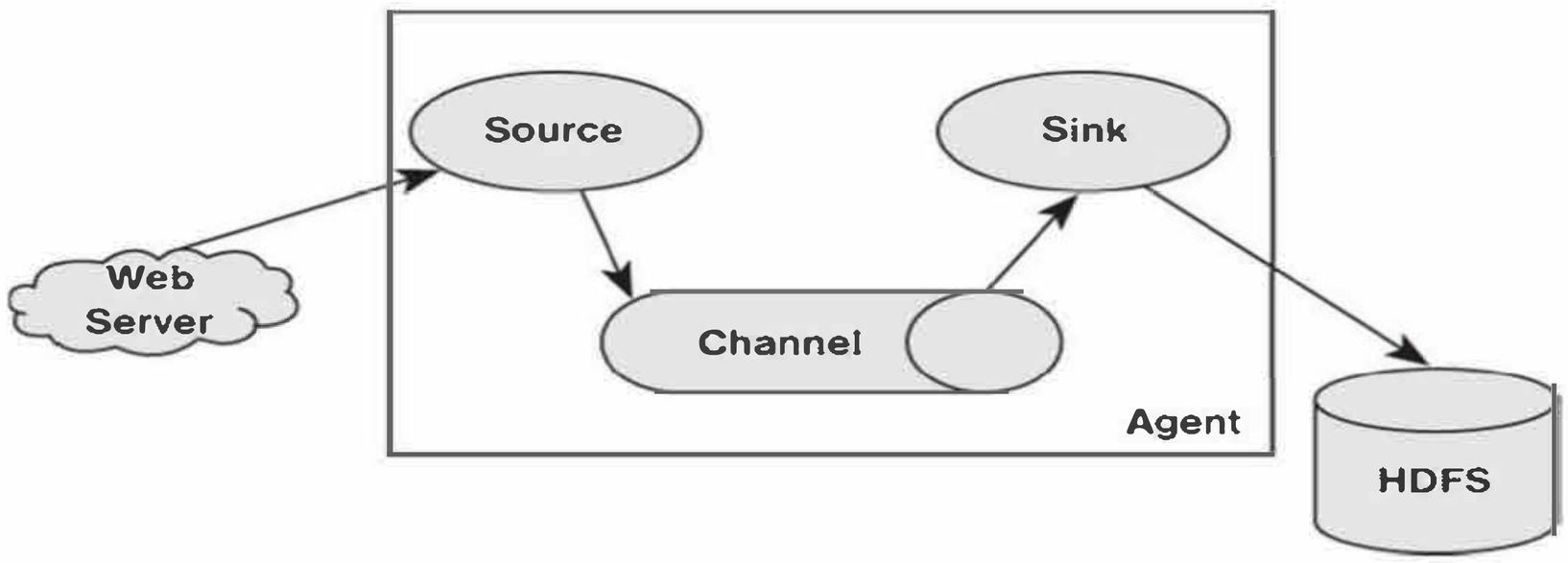
TABLE 7.2 APACHE SQOOP VERSION COMPARISON

Feature	Sqoop Version 1	Sqoop Version 2
Connectors for all major RDBMSs	Supported.	Not supported. Use the generic JDBC connector.
Kerberos security integration	Supported.	Not supported.
Data transfer from RDBMS to Hive or HBase	Supported.	Not supported. First import data from RDBMS into HDFS, then load data into Hive or HBase manually.
Data transfer from Hive or HBase to RDBMS	Not supported. First export data from Hive or HBase into HDFS, and then use Sqoop for export.	Not supported. First export data from Hive or HBase into HDFS, then use Sqoop for export.

USING APACHE FLUME TO ACQUIRE DATA STREAMS

- Apache Flume is an independent agent designed to collect, transport, and store data into HDFS.
- Often data transport involves a number of Flume agents that may traverse a series of machines and locations.
- Flume is often used for log files, social media-generated data, email messages, and just about any continuous data source. As shown in Figure 7.3, a Flume agent is composed of three components.

Figure 7.3 Flume agent with source, channel, and sink



Source. The source component receives data and sends it to a channel. It can send the data to more than one channel. The input data can be from a realtime source (e.g., weblog) or another Flume agent.

Channel. A channel is a data queue that forwards the source data to the sink destination. It can be thought of as a buffer that manages input (source) and output (sink) flow rates.

Sink. The sink delivers data to destination such as HDFS, a local file, or another Flume agent.

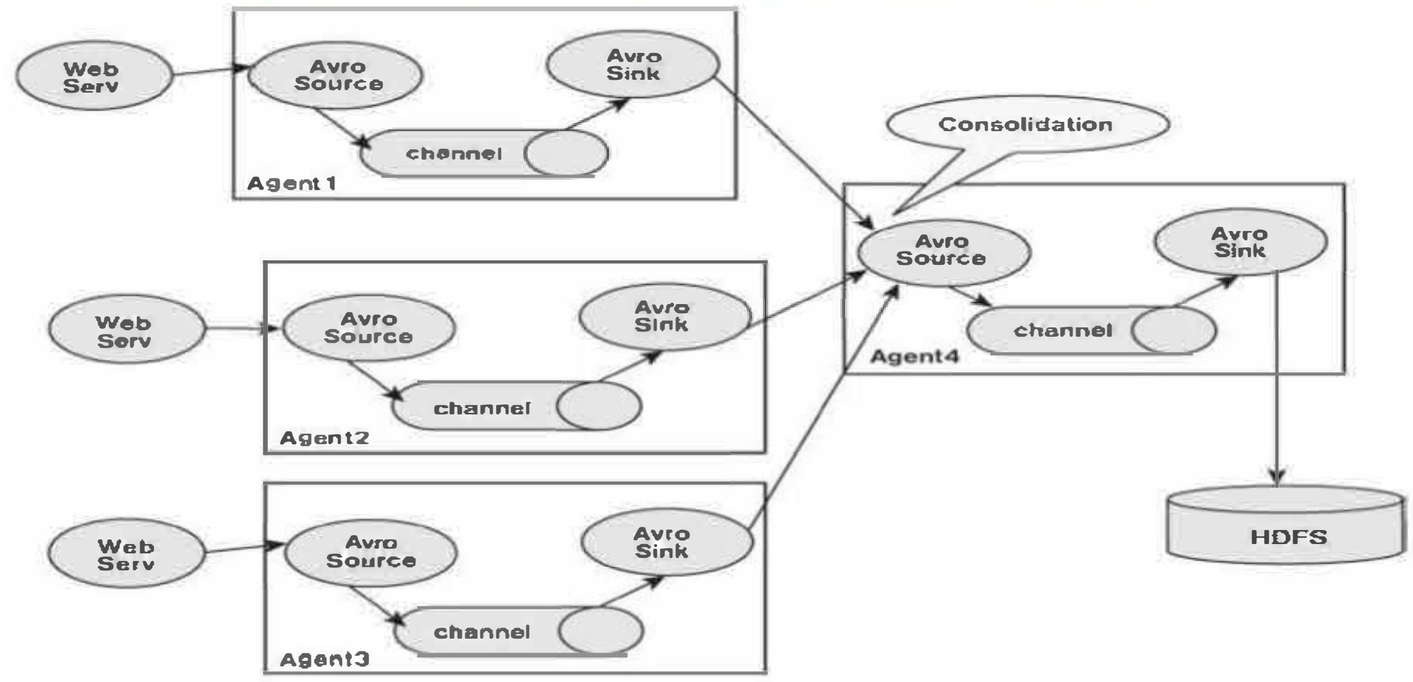
- A Flume agent must have all three of these components defined.
- A Flume agent can have several sources, channels, and sinks.
- Sources can write to multiple channels, but a sink can take data from only a single channel.
- Data written to a channel remain in the channel until a sink removes the data. By default, the data in a channel are kept in memory but may be optionally stored on disk to prevent data loss in the event of a network failure.
- As shown in Figure 7.4, Sqoop agents may be placed in a pipeline, possibly to traverse several machines or domains.
- This configuration is normally used when data are collected on one machine (e.g., a web server) and sent to another machine that has access to HDFS
- In a Flume pipeline, the sink from one agent is connected to the source of another.
- The data transfer format normally used by Flume, which is called Apache Avro, provides several useful features.
- First, Avro is a data serialization/deserialization system that uses a compact binary format.
- The schema is sent as part of the data exchange and is defined using JSON (JavaScript Object Notation).
- Avro also uses remote procedure calls (RPCs) to send data.
- That is, an Avro sink will contact an Avro source to send data.

Sqoop agents placed as pipeline Other configurations

- Another useful Flume configuration is shown in Figure 7.5.

- In this configuration, Flume is used to consolidate several data sources before committing them to HDFS.
- There are many possible ways to construct Flume transport networks.
- In addition, other Flume features not described in depth here include plug-ins and interceptors that can enhance Flume pipelines.

Figure 7.5 A Flume consolidation network



MANAGE HADOOP WORKFLOWS WITH APACHE OOZIE

- Oozie is a workflow director system designed to run and manage multiple related Apache Hadoop jobs.
- For instance, complete data input and analysis may require several discrete Hadoop jobs to be run as a workflow in which the output of one job serves as the input for a successive job.
- Oozie is designed to construct and manage these workflows.
- Oozie is not a substitute for the YARN scheduler.
- That is, YARN manages resources for individual Hadoop jobs, and Oozie provides a way to connect and control Hadoop jobs on the cluster.
- Oozie workflow jobs are represented as directed acyclic graphs (DAGs) of actions. (DAGs are basically graphs that cannot have directed loops.)
- Three types of Oozie jobs are permitted:
    1. **Workflow**—a specified sequence of Hadoop jobs with outcome-based decision points and control dependency. Progress from one action to another cannot happen until the first action is complete.
    2. **Coordinator**—a scheduled workflow job that can run at various time intervals or when data become available.
    3. **Bundle**—a higher-level Oozie abstraction that will batch a set of coordinator jobs.

- Oozie is integrated with the rest of the Hadoop stack, supporting several types of Hadoop jobs out of the box (e.g., Java MapReduce, Streaming MapReduce, Pig,

Hive, and Sqoop) as well as systemspecific jobs (e.g., Java programs and shell scripts). Oozie also provides a CLI and a web UI for monitoring jobs.

Figure 7.6 A simple Oozie DAG workflow

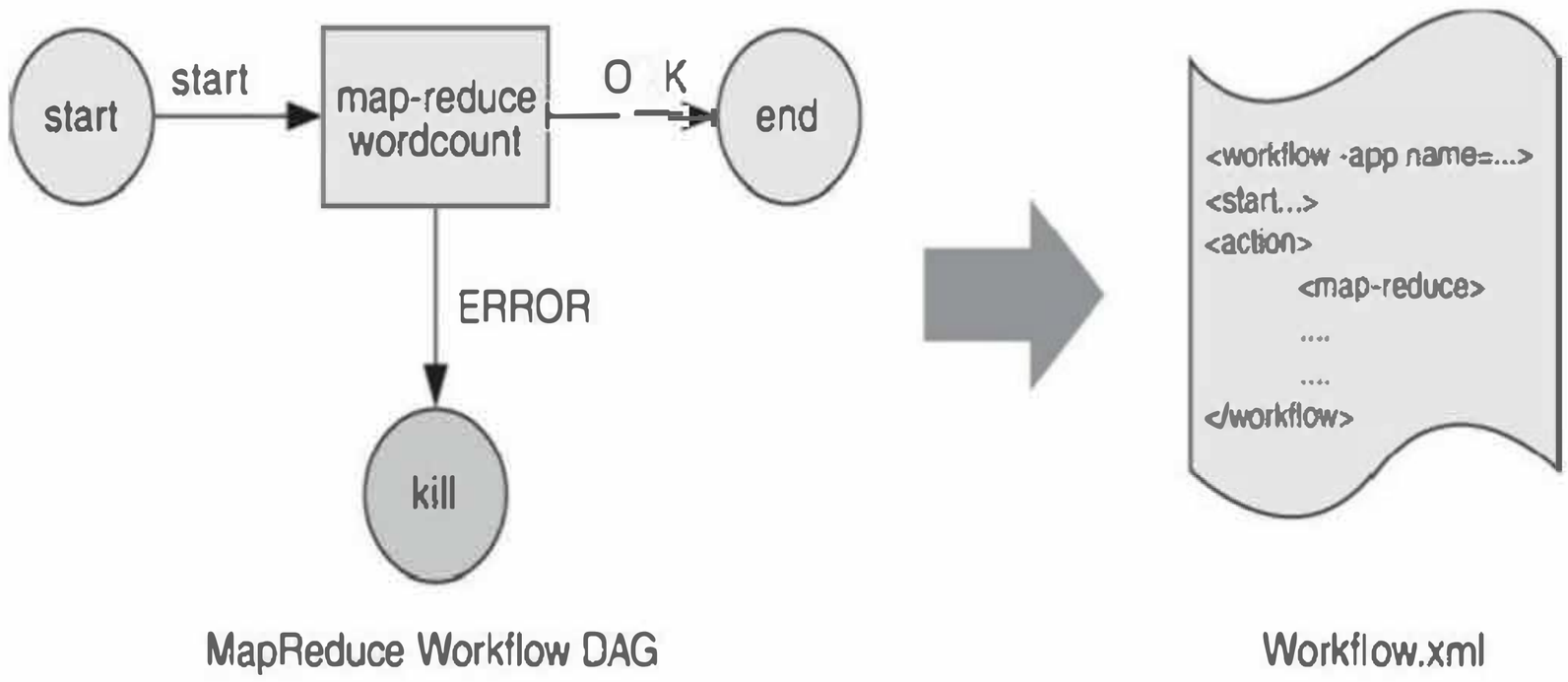
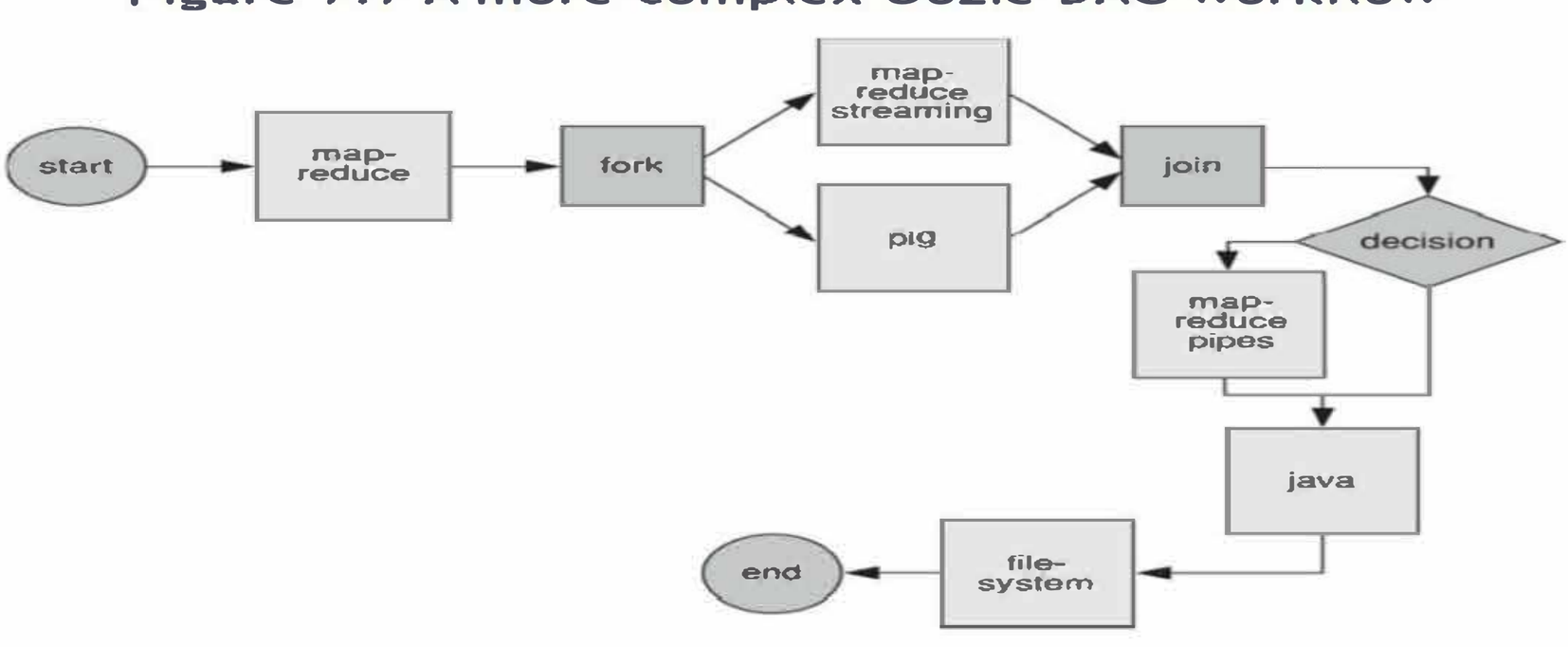


Figure 7.6 depicts a simple Oozie workflow. In this case, Oozie runs a basic MapReduce operation. If the application was successful, the job ends; if an error occurred, the job is killed. Oozie workflow definitions are written in hPDL (an XML Process Definition Language). Such workflows contain several types of nodes:

1. **Control flow** nodes define the beginning and the end of a workflow. They include start, end, and optional fail nodes.
2. **Action nodes** are where the actual processing tasks are defined. When an action node finishes, the remote systems notify Oozie and the next node in the workflow is executed. Action nodes can also include HDFS commands.
3. **Fork/join nodes** enable parallel execution of tasks in the workflow. The fork node enables two or more tasks to run at the same time. A join node represents a rendezvous point that must wait until all forked tasks complete.
4. **Control flow nodes** enable decisions to be made about the previous task. Control decisions are based on the results of the previous action (e.g., file size or file existence). Decision nodes are essentially switch-case statements that use JSP EL (Java Server Pages—Expression Language) that evaluate to either true or false.

Figure 7.7 A more complex Oozie DAG workflow



USING APACHE HBASE

- Apache HBase is an open source, distributed, versioned, **nonrelational** database modeled after **Google's Bigtable.**
- Like Bigtable, HBase leverages the **distributed data storage** provided by the underlying distributed file systems spread across **commodity servers**.
- Apache HBase provides Bigtable-like capabilities on top of Hadoop and HDFS.

Some of the more important features include the following capabilities:

- **Linear** and **modular** scalability
- Strictly **consistent reads and writes**
- Automatic and **configurable sharding of tables**
- Automatic **failover** support between **RegionServers**
- Convenient **base classes** for **backing Hadoop MapReduce jobs** with Apache HBase tables
- **Easy-to-use Java API** for client access

HBase Data Model Overview

- A table in HBase is **similar** to other databases, having **rows and columns**.
- Columns in HBase are grouped into **column families,** all with the **same prefix**.
- For example, consider a table of daily stock prices. There may be a column family called "**price**" that has four *members— price:open, price:close, price:low, and price:high.*
- A column does not need to be a family. For instance, the stock table may have a column named "**volume**" indicating how many shares were traded.
- All **column family** members are **stored** together in the **physical file system**.
- Specific **HBase cell values** are identified by *a row key, column (column family and column), and version (timestamp).*
- It is possible to have many versions of data within an **HBase cell.**
- A version is specified as a **timestamp** and is created *each time data are written to a cell.*
- Almost anything can serve as a *row key*, from strings to binary representations of longs to *serialized data structures.*
- Rows are ***lexicographically*** sorted with the **lowest order appearing first in a table**.
- The **empty byte array denotes** both the start and the end of a ***table's namespace.***
- All ***table accesses*** are via the ***table row key***, which is considered its ***primary key***

YARN DISTRIBUTED-SHELL

- The Hadoop YARN project includes the Distributed-Shell application, which is an example of a ***Hadoop non-MapReduce application built on top of YARN***.
- Distributed-Shell is a ***simple mechanism for running shell commands and scripts in containers on multiple nodes*** in a Hadoop cluster.

- This application is not meant to be a **production administration tool**, but rather a demonstration of the non-MapReduce capability that can be implemented on top of YARN.
- There are multiple **mature implementations** of a distributed shell **that administrators** typically use to manage a cluster of machines.
- In addition, Distributed-Shell can be **used as a starting point** for **exploring and building** Hadoop YARN applications.

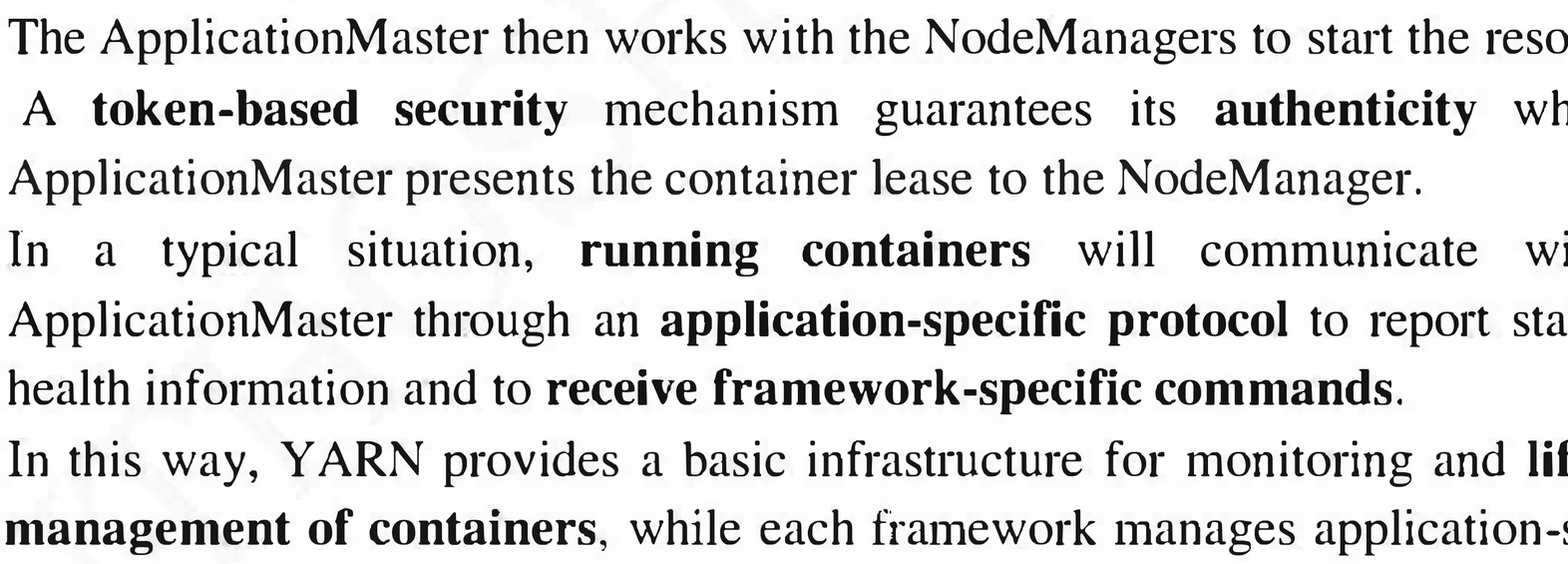
STRUCTURE OF YARN APPLICATIONS

- YARN ResourceManager runs as a scheduling ***daemon on a dedicated machine*** and acts as the central authority for allocating resources to the various competing applications in the cluster.
- The ResourceManager has a ***central and global view of all cluster resources*** and, therefore, can ***ensure fairness, capacity, and locality*** are shared across all users.
- Depending on the application ***demand, scheduling priorities, and resource availability***, the ResourceManager ***dynamically allocates*** resource containers to applications to run on particular nodes.
- A ***container*** is a logical **bundle of resources** (e.g., memory, cores) bound to a particular cluster node.
- To enforce and **track** such assignments, the ResourceManager interacts with a special **system daemon running on each node** called the **NodeManager.**
- Communications between the ResourceManager and NodeManagers are ***heartbeat based for scalability***.
- NodeManagers are **responsible** for ***local monitoring of resource availability, fault reporting, and container life-cycle management*** (e.g., starting and killing jobs).
- The ResourceManager depends on the NodeManagers for its "***global view"*** of the cluster.
- User applications are submitted to the ResourceManager via a **public protocol** and go through an admission control phase during which **security credentials** are validated and various **operational and administrative checks** are performed.
- Those applications that are accepted pass to the ***scheduler and are allowed to run***.
- Once the scheduler has enough ***resources to satisfy the request***, the application is ***moved from an accepted state to a running state.***
- Aside from internal bookkeeping, this process involves ***allocating a container for the single ApplicationMaster*** and spawning it on a node in the cluster. Often called container0, the ApplicationMaster does not have any additional ***resources at this point, but rather must request additional resources from the ResourceManager.***
- The ApplicationMaster is the "master" user job that ***manages all application life-cycle aspects***, including ***dynamically increasing and decreasing resource consumption*** (i.e., containers), ***managing the flow of execution*** (e.g., in case of

MapReduce jobs, running reducers against the output of maps), ***handling faults and computation skew, and performing other local optimizations***.

- The ApplicationMaster is designed to ***run arbitrary user code*** that can be written in any programming language, as all ***communication with the ResourceManager and NodeManager*** is encoded using extensible network protocols.
- YARN makes **few assumptions about the ApplicationMaster**, although in practice it expects most jobs will use a ***higher-level programming framework***.
- By ***delegating all these functions*** to ApplicationMasters, YARN's architecture gains a ***great deal of scalability, programming model flexibility, and improved user agility***.
- For example, upgrading and testing a new MapReduce framework can be done independently of other running MapReduce frameworks.
- Typically, an ApplicationMaster will need to harness the processing power of multiple servers to complete a job.
- To achieve this, the ApplicationMaster ***issues resource requests*** to the ResourceManager.
- The form of ***these requests includes*** specification of ***locality preferences*** (e.g., to accommodate HDFS use) ***and properties of the containers.***
- The ResourceManager will ***attempt to satisfy*** the resource requests coming from each application **according to availability and scheduling policies**.
- When a ***resource is scheduled on behalf*** of an ApplicationMaster, the ResourceManager generates a ***lease for the resource***, which is acquired by a ***subsequent ApplicationMaster heartbeat***.
- The ApplicationMaster then works with the NodeManagers to start the resource.
- A **token-based security** mechanism guarantees its **authenticity** when the ApplicationMaster presents the container lease to the NodeManager.
- In a typical situation, **running containers** will communicate with the ApplicationMaster through an **application-specific protocol** to report status and health information and to **receive framework-specific commands**.
- In this way, YARN provides a basic infrastructure for monitoring and **life-cycle management of containers**, while each framework manages application-specific semantics independently.

Figure 8.1 YARN architecture with two clients (MapReduce and MPI). The darker client (MPI AM2) is running an MPI application, and the lighter client (MR AM1) is running a MapReduce application.

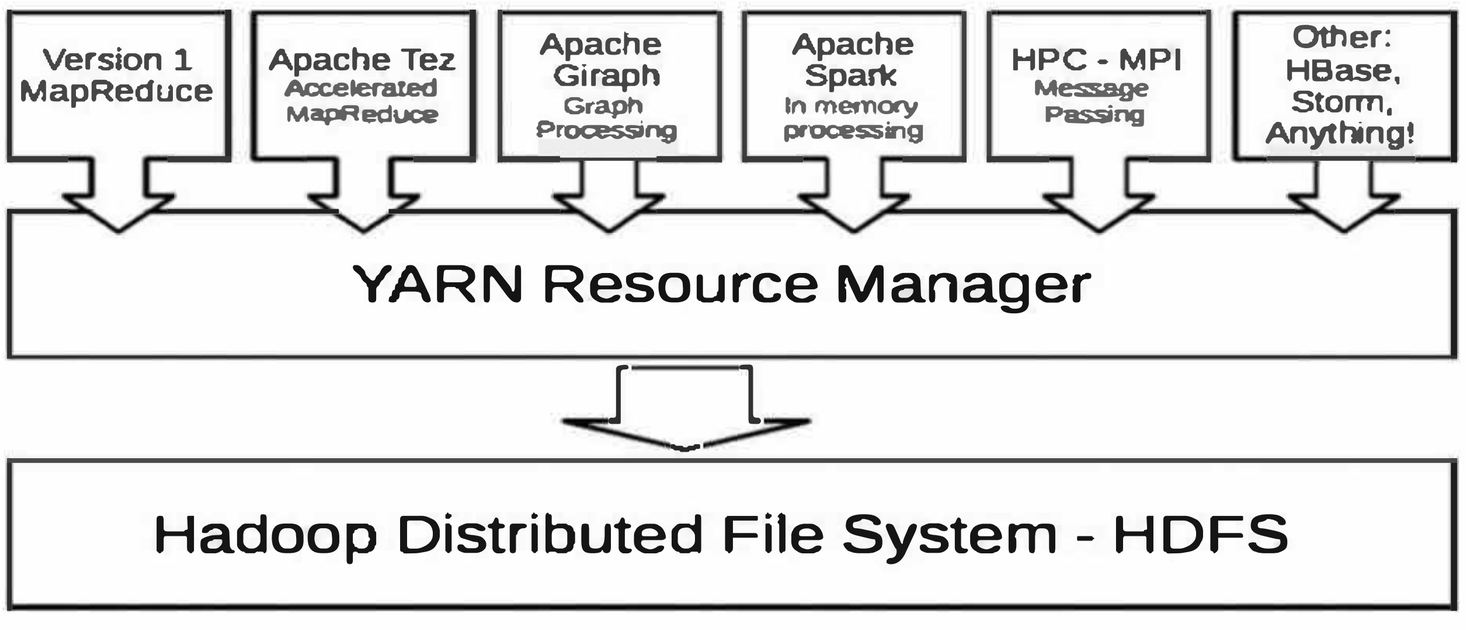


- This design stands in sharp contrast to the original Hadoop version 1 design, in which scheduling was designed and integrated around managing only MapReduce tasks.
- Figure 8.1 illustrates the relationship between the application and YARN components.
- The YARN components appear as the large outer boxes (ResourceManager and NodeManagers), and the two applications appear as smaller boxes (containers), one dark and one light.
- Each application uses a different ApplicationMaster; the darker client is running a Message Passing Interface (MPI) application and the lighter client is running a traditional MapReduce application.

YARN APPLICATION FRAMEWORKS

- One of the most exciting aspects of Hadoop version 2 is the capability to run all types of applications on a Hadoop cluster.
- In Hadoop version 1, the only processing model available to users is MapReduce.
- In Hadoop version 2, MapReduce is separated from the resource management layer of Hadoop and placed into its own application framework.
- Indeed, the growing number of YARN applications offers a high level and multifaceted interface to the Hadoop data lake
- YARN presents a resource management platform, which provides services such as scheduling, fault monitoring, data locality, and more to MapReduce and other frameworks.
- Figure 8.2 illustrates some of the various frameworks that will run under YARN

Figure 8.2 Example of the Hadoop version 2 ecosystem. Hadoop version 1 supports batch MapReduce applications only



Distributed-Shell

- Distributed-Shell is an example application included with the Hadoop core components that demonstrates how to write applications on top of YARN.
- It provides a simple method for running shell commands and scripts in containers in parallel on a Hadoop YARN cluster.

Hadoop MapReduce

- MapReduce was the first YARN framework and drove many of YARN's requirements.
- It is integrated tightly with the rest of the Hadoop ecosystem projects, such as Apache Pig, Apache Hive, and Apache Oozie.

Apache Tez

- Many Hadoop jobs involve the execution of a complex directed acyclic graph (DAG) of tasks using separate MapReduce stages.
- Apache Tez generalizes this process and enables these tasks to be spread across stages so that they can be run as a single, all-encompassing job.
- Tez can be used as a MapReduce replacement for projects such as Apache Hive and Apache Pig.

Apache Giraph

- Apache Giraph is an iterative graph processing system built for high scalability.
- Facebook, Twitter, and LinkedIn use it to create social graphs of users.
- Giraph was originally written to run on standard Hadoop V1 using the MapReduce framework, but that approach proved inefficient and totally unnatural for various reasons.
- The native Giraph implementation under YARN provides the user with an iterative processing model that is not directly available with MapReduce. Support for YARN has been present in Giraph since its own version 1.0 release.
- In addition, using the flexibility of YARN, the Giraph developers plan on implementing their own web interface to monitor job progress.

Hoya: HBase on YARN

- The Hoya project creates dynamic and elastic Apache HBase clusters on top of YARN.
- A client application creates the persistent configuration files, sets up the HBase cluster XML files, and then asks YARN to create an ApplicationMaster.
- YARN copies all files listed in the client's application-launch request from HDFS into the local file system of the chosen server, and then executes the command to start the Hoya ApplicationMaster.
- Hoya also asks YARN for the number of containers matching the number of HBase region servers it needs

Dryad on YARN

- Similar to Apache Tez, Microsoft's Dryad provides a DAG as the abstraction of execution flow.
- This framework is ported to run natively on YARN and is fully compatible with its non-YARN version.
- The code is written completely in native C++ and C# for worker nodes and uses a thin layer of Java within the application.

Apache Spark

- Spark was initially developed for applications in which keeping data in memory improves performance, such as iterative algorithms, which are common in machine learning, and interactive data mining.
- Spark differs from classic MapReduce in two important ways.
- First, Spark holds intermediate results in memory, rather than writing them to disk.
- Second, Spark supports more than just MapReduce functions; that is, it greatly expands the set of possible analyses that can be executed over HDFS data stores. It also provides APIs in Scala, Java, and Python.
- Since 2013, Spark has been running on production YARN clusters at Yahoo!. The advantage of porting and running Spark on top of YARN is the common resource management and a single underlying file system.

Apache Storm

- Traditional MapReduce jobs are expected to eventually finish, but Apache Storm continuously processes messages until it is stopped.
- This framework is designed to process unbounded streams of data in real time. It can be used in any programming language.
- The basic Storm use-cases include real-time analytics, online machine learning, continuous computation, distributed RPC (remote procedure calls), ETL (extract, transform, and load), and more.
- Storm provides fast performance, is scalable, is fault tolerant, and provides processing guarantees.
- It works directly under YARN and takes advantage of the common data and resource management substrate.

Apache REEF: Retainable Evaluator Execution Framework

- YARN's flexibility sometimes requires significant effort on the part of application implementers.
- The steps involved in writing a custom application on YARN include building your own ApplicationMaster, performing client and container management, and handling aspects of fault tolerance, execution flow, coordination, and other concerns.
- The REEF project by Microsoft recognizes this challenge and factors out several components that are common to many applications, such as storage management, data caching, fault detection, and checkpoints.
- Framework designers can build their applications on top of REEF more easily than they can build those same applications directly on YARN, and can reuse these common services/libraries.
- REEF's design makes it suitable for both MapReduce and DAG-like executions as well as iterative and interactive computations

Hamster: Hadoop and MPI on the Same Cluster

- The Message Passing Interface (MPI) is widely used in high-performance computing (HPC).
- MPI is primarily a set of optimized message-passing library calls for C, C++, and Fortran that operate over popular server interconnects such as Ethernet and InfiniBand.
- Because users have full control over their YARN containers, there is no reason why MPI applications cannot run within a Hadoop cluster.
- The Hamster effort is a work-in-progress that provides a good discussion of the issues involved in mapping MPI to a YARN cluster.
- Currently, an alpha version of MPICH2 is available for YARN that can be used to run MPI applications.

Apache Flink: Scalable Batch and Stream Data Processing

- Apache Flink is a platform for efficient, distributed, generalpurpose data processing.
- It features powerful programming abstractions in Java and Scala, a high-performance run time, and automatic program optimization.
- It also offers native support for iterations, incremental iterations, and programs consisting of large DAGs of operations.
- Flink is primarily a stream-processing framework that can look like a batch-processing environment.
- The immediate benefit from this approach is the ability to use the same algorithms for both streaming and batch modes
- However, Flink can provide low-latency similar to that found in Apache Storm, but which is not available in Apache Spark.

Apache Slider: Dynamic Application Management

- Apache Slider (incubating) is a YARN application to deploy existing distributed applications on YARN, monitor them, and make them larger or smaller as desired in real time.
- Applications can be stopped and then started; the distribution of the deployed application across the YARN cluster is persistent and allows for best-effort placement close to the previous locations.
- Applications that remember the previous placement of data (such as HBase) can exhibit fast startup times by capitalizing on this feature.
- YARN monitors the health of "YARN containers" that are hosting parts of the deployed applications.
- If a container fails, the Slider manager is notified. Slider then requests a new replacement container from the YARN ResourceManager.

- Some of Slider's other features include user creation of on-demand applications, the ability to stop and restart applications as needed (preemption), and the ability to expand or reduce the number of application containers as needed.
- The Slider tool is a Java command-line application

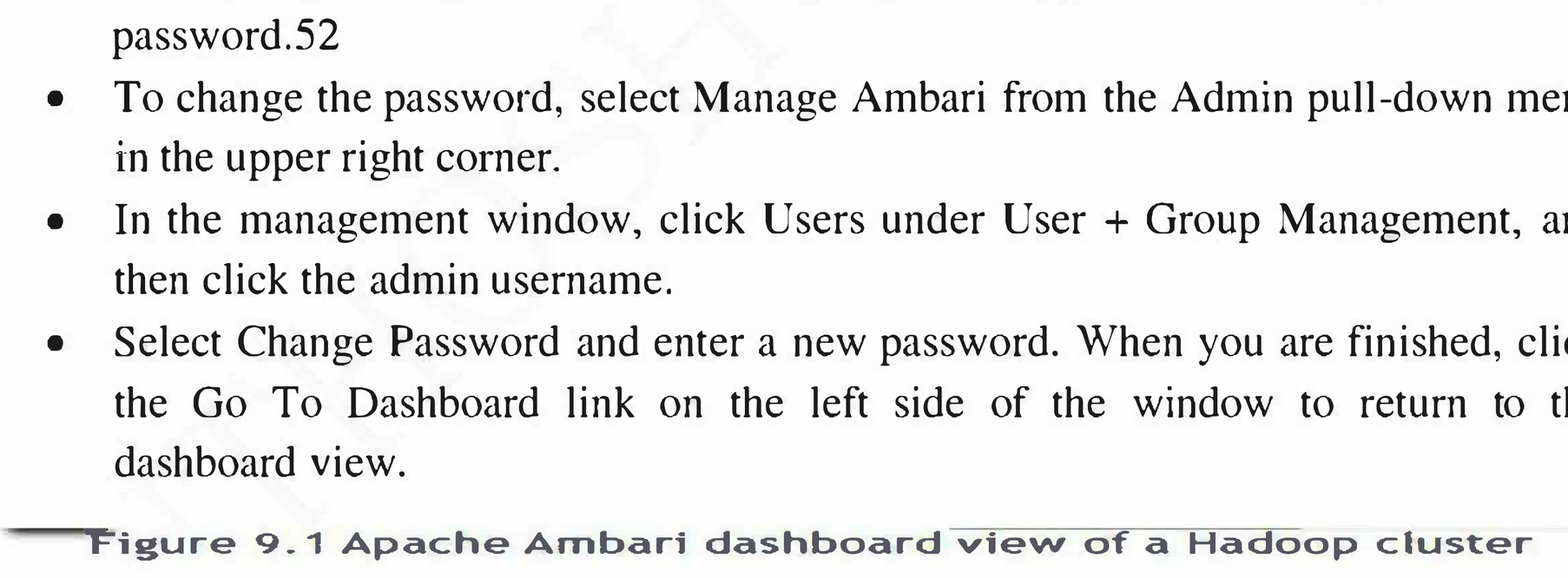
Managing Hadoop with Apache Ambari

- QUICK TOUR OF APACHE AMBARI
- Dashboard View
- Services View
- Hosts View
- Admin View
- Views View
- MANAGING HADOOP SERVICES
- CHANGING HADOOP PROPERTIES

QUICK TOUR OF APACHE AMBARI

- After completing the initial installation and logging into Ambari (as explained in Chapter 2), a dashboard similar to that shown in Figure 9.1 is presented. The same four-node cluster as created.
- If you need to reopen the Ambari dashboard interface,simply enter the following command **$ firefox localhost:8080** The default login and password are admin and admin, respectively. Before continuing any further, you should change the default password.52
- To change the password, select Manage Ambari from the Admin pull-down menu in the upper right corner.
- In the management window, click Users under User + Group Management, and then click the admin username.
- Select Change Password and enter a new password. When you are finished, click the Go To Dashboard link on the left side of the window to return to the dashboard view.

Figure 9.1 Apache Ambari dashboard view of a Hadoop cluster

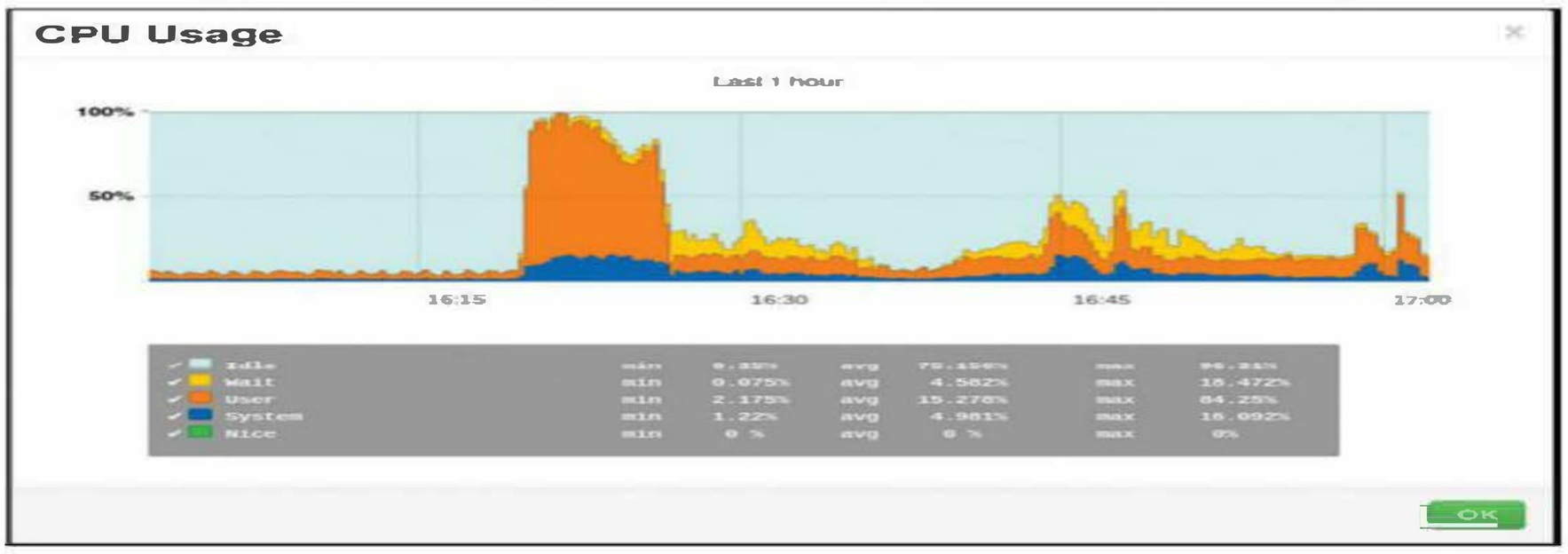


- To leave the Ambari interface, select the Admin pulldown menu at the left side of the main menu bar and click Sign out.
- The dashboard view provides a number of high-level metrics for many of the installed services. A glance at the dashboard should allow you to get a sense of how the cluster is performing.

Dashboard View

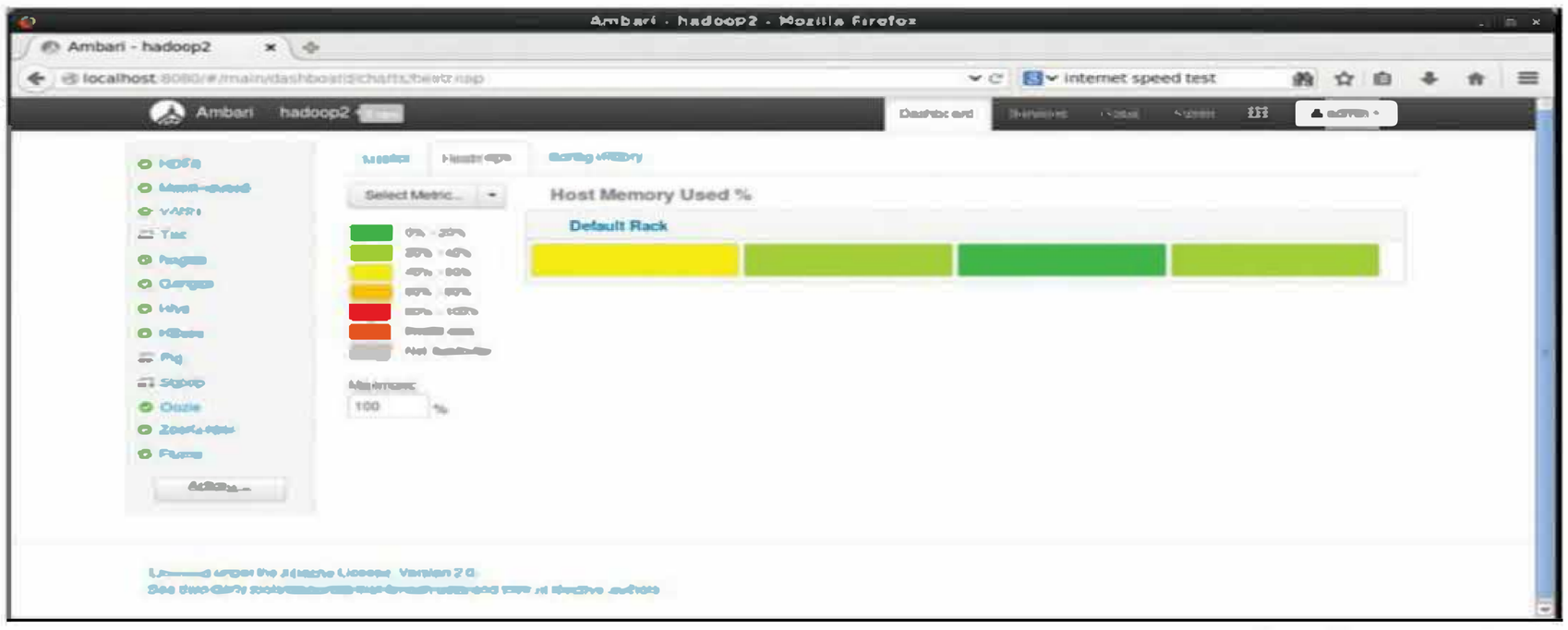
- The Dashboard view provides small status widgets for many of the services running on the cluster.
- The actual services are listed on the left-side vertical menu. You can move, edit, remove, or add these widgets as follows:
  - **Moving:** Click and hold a widget while it is moved about the grid.
  - **Edit:** Place the mouse on the widget and click the gray edit symbol in the upper-right corner of the widget. You can change several different aspects (including thresholds) of the widget.
  - **Remove:** Place the mouse on the widget and click the X in the upper-left corner.
  - **Add:** Click the small triangle next to the Metrics tab and select Add. The available widgets will be displayed. Select the widgets you want to add and click Apply.
- Some widgets provide additional information when you move the mouse over them. For instance, the DataNodes widget displays the number of live, dead, and decommissioning hosts.

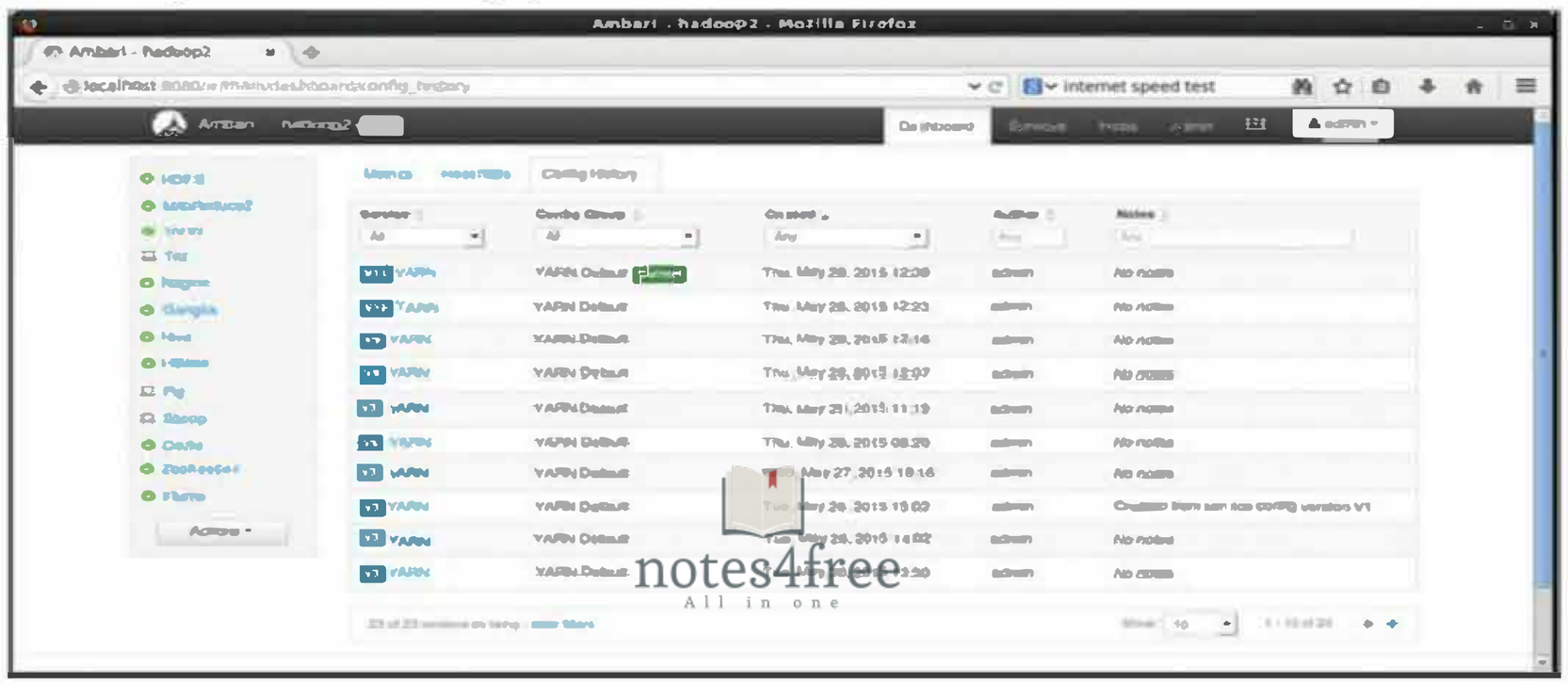
Figure 9.2 Enlarged view of Ambari CPU Usage widget



- The Dashboard view also includes a heatmap view of the cluster. Cluster heatmaps physically map selected metrics across the cluster.
- When you click the Heatmaps tab, a heatmap for the cluster will be displayed.
- To select the metric used for the heatmap, choose the desired option from the Select Metric pull-down menu.
- Note that the scale and color ranges are different for each metric.
- The heatmap for percentage host memory used is displayed in Figure 9.3.

The heatmap for percentage host memory used is displayed in Figure 9.3.





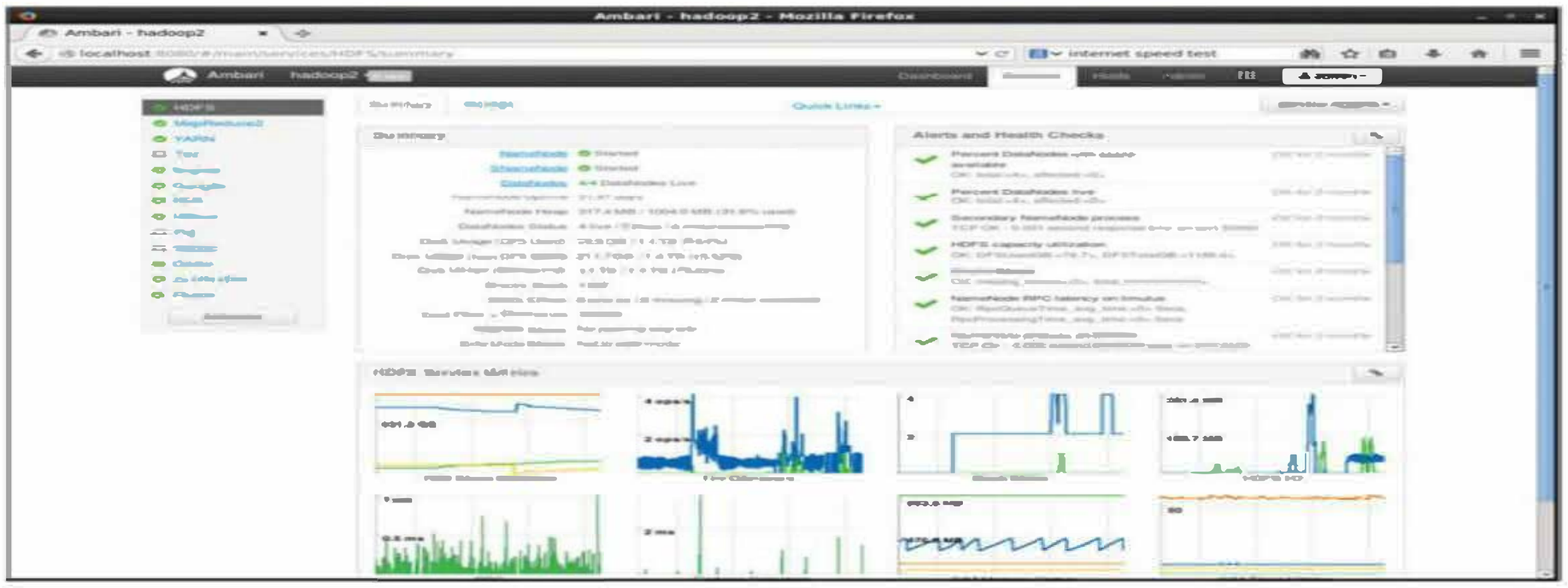
Configuration history is the final tab in the dashboard window. This view provides a list of configuration changes made to the cluster. As shown in Figure 9.4, Ambari enables configurations to be sorted by Service, Configuration Group, Data, and Author. To find the specific configuration settings, click the service name.58

Services View

- The Services menu provides a detailed look at each service running on the cluster.
- It also provides a graphical method for configuring each service (i.e., instead of hand-editing the /etc/hadoop/confXML files).
- The summary tab provides a current Summary view of important service metrics and an Alerts and Health Checks sub-window.
- Similar to the Dashboard view, the currently installed services are listed on the left-side menu.
- To select a service, click the service name in the menu.
- When applicable, each service will have its own Summary, Alerts and Health Monitoring, and Service Metrics windows.
- For example, Figure 9.5 shows the Service view for HDFS. Important information such as the status of NameNode, SecondaryNameNode, DataNodes, uptime, and available disk space is displayed in the Summary window.

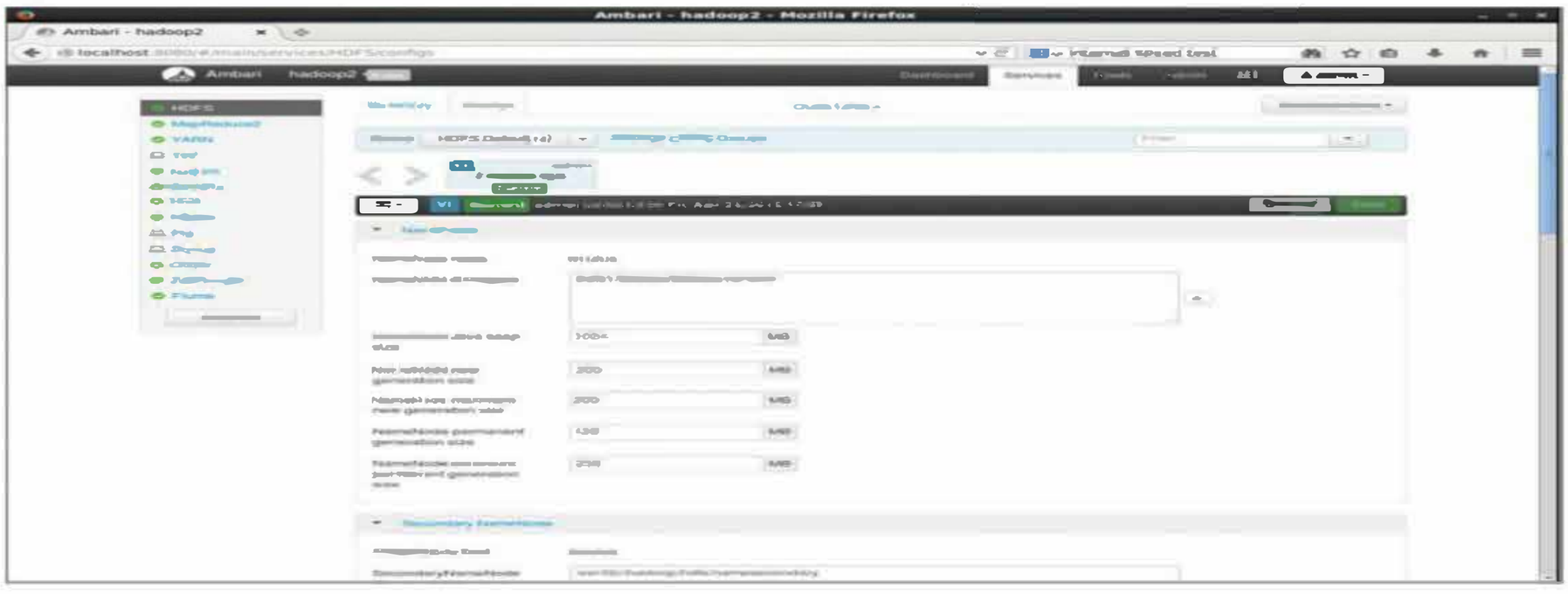
- The Alerts and Health Checks window provides the latest status of the service and its component systems.
- Finally, several important real-time service metrics are displayed as widgets at the bottom of the screen.
- As on the dashboard, these widgets can be expanded to display a more detailed view.

Figure 9.5 HDFS service summary window



- Clicking the Configs tab will open an options form, shown in Figure 9.6, for the service.
- The options (properties) are the same ones that are set in the Hadoop XML files.
- When using Ambari, the user has complete control over the XML files and should manage them only through the Ambari interface—that is, the user should not edit the files by hand.
- The current settings available for each service are shown in the form.
- The administrator can set each of these properties by changing the values in the form.
- Placing the mouse in the input box of the property displays a short description of each property.
- Where possible, properties are grouped by functionality.
- The form also has provisions for adding properties that are not listed.
- An example of changing service properties and restarting the service components is provided in the "Managing Hadoop Services" section.
- If a service provides its own graphical interface (e.g., HDFS, YARN, Oozie), then that interface can be opened in a separate browser tab by using the Quick Links pulldown menu located in top middle of the window.

Figure 9.6 Ambari service options for HDFS

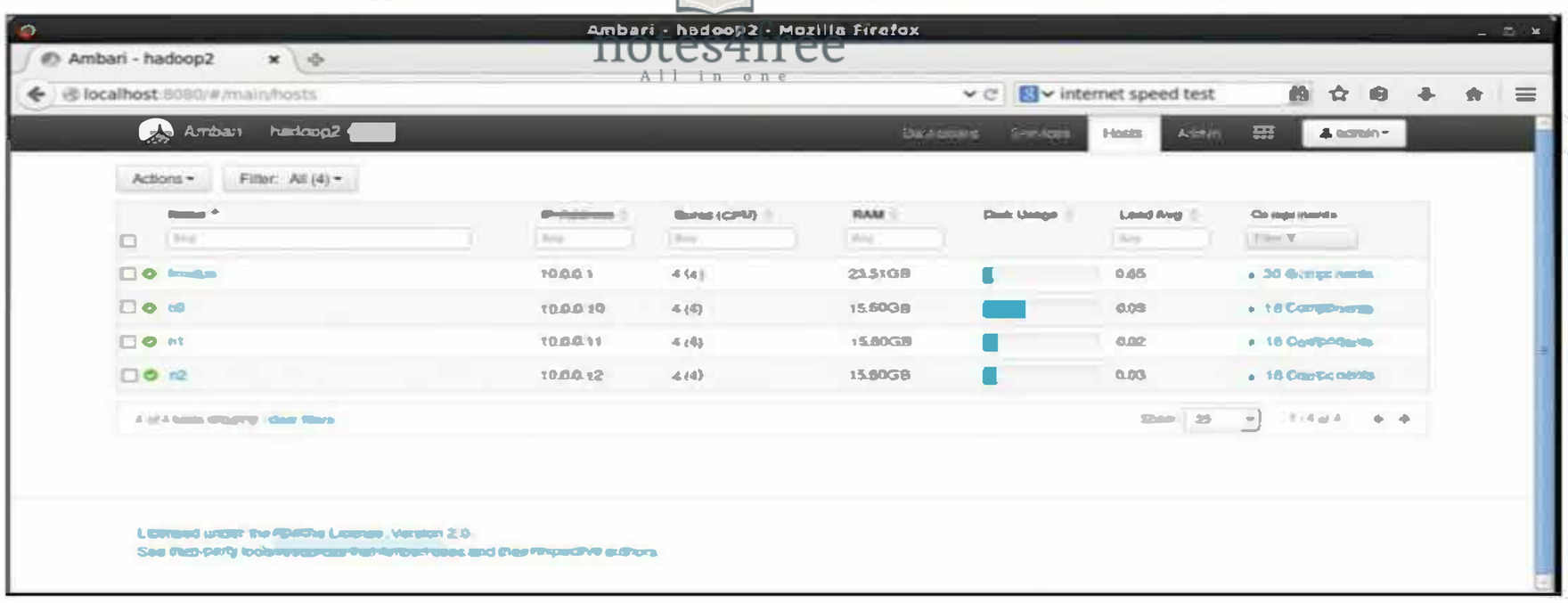


- Finally, the Service Action pull-down menu in the upperleft corner provides a method for starting and stopping each service and/or its component daemons across the cluster.
- Some services may have a set of unique actions (such as rebalancing HDFS) that apply to only certain situations.
- Finally, every service has a Service Check option to make sure the service is working property.
- The service check is initially run as part of the installation process and can be valuable when diagnosing problems.

Hosts View

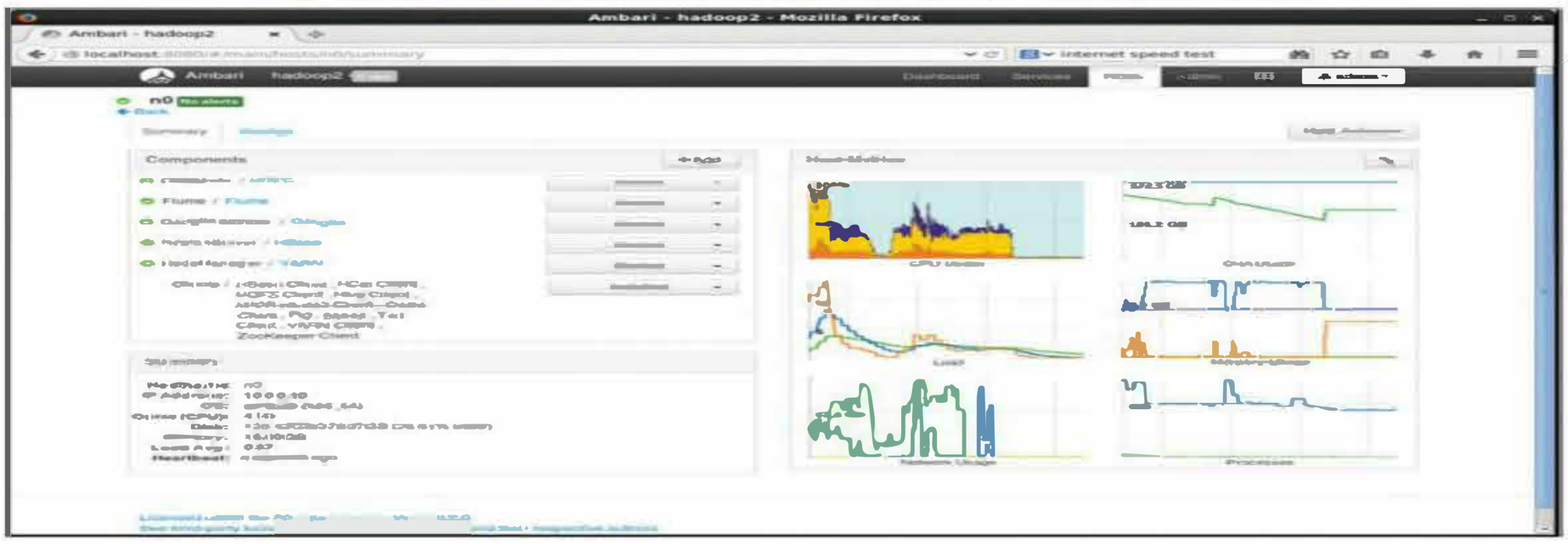
- Selecting the Hosts menu item provides the information shown in Figure 9.7.
- The host name, IP address, number of cores, memory, disk usage, current load average, and Hadoop components are listed in this window in tabular form.
- To display the Hadoop components installed on each host, click the links in the rightmost columns.
- You can also add new hosts by using the Actions pulldown menu. The new host must be running the Ambari agent (or the root SSH key must be entered) and have the base software.
- The remaining options in the Actions pull-down menu provide control over the various service components running on the hosts.

Figure 9.7 Ambari main Hosts screen



- Further details for a particular host can be found by clicking the host name in the left column. As shown in Figure 9.8, the individual host view provides three subwindows: Components, Host Metrics, and Summary information.
- The Components window lists the services that are currently running on the host. Each service can be stopped, restarted, decommissioned, or placed in maintenance mode.
- The Metrics window displays widgets that provide important metrics (e.g., CPU, memory, disk, and network usage).
- Clicking the widget displays a larger version of the graphic.
- The Summary window provides basic information about the host, including the last time a heartbeat was received.

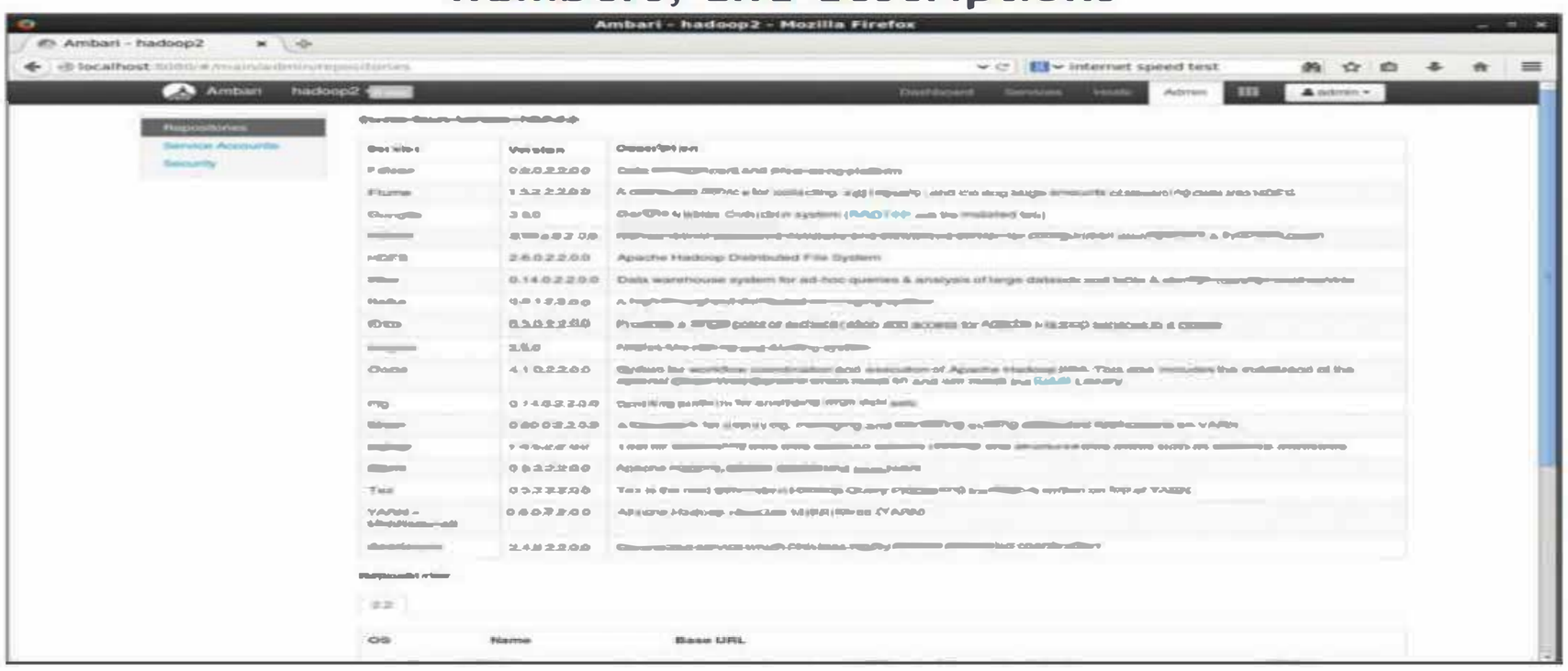
Figure 9.8 Ambari cluster host detail view



Admin View

- The Administration (Admin) view provides three options. The first, as shown in Figure 9.9, displays a list of installed software.
- This Repositories listing generally reflects the version of Horton works Data Platform (HDP) used during the installation process.
- The Service Accounts option lists the service accounts added when the system was installed.
- These accounts are used to run various services and tests for Ambari.
- The third option, Security, sets the security on the cluster.
- A fully secured Hadoop cluster is important in many instances and should be explored if a secure environment is needed.

Figure 9.9 Ambari installed packages with versions, numbers, and descriptions



Views View

- Ambari Views is a framework offering a systematic way to plug in user interface capabilities that provide for custom visualization, management, and monitoring features in Ambari.
- Views allows you to extend and customize Ambari to meet your specific needs.

Admin Pull-Down Menu

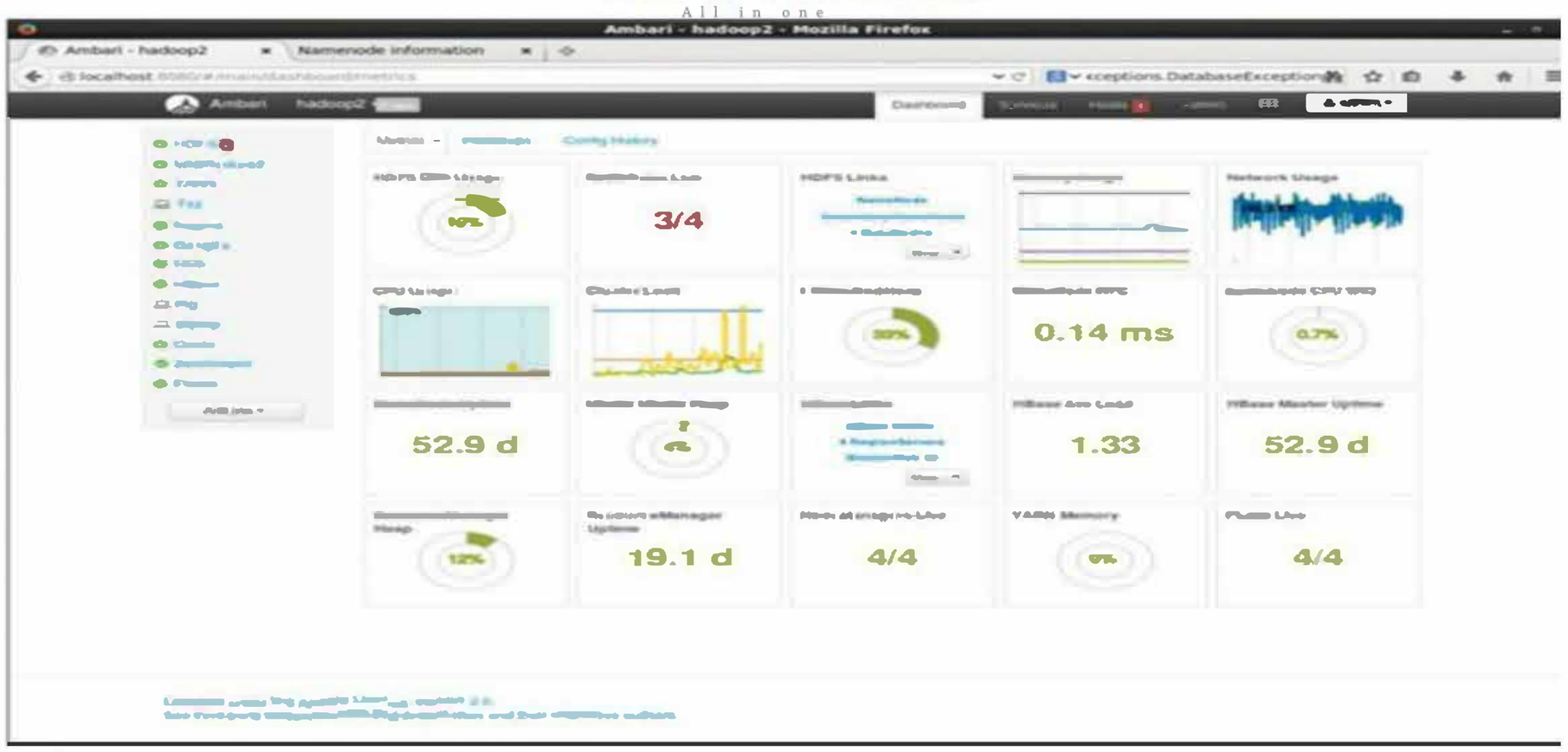
The Administrative (Admin) pull-down menu provides the following options:

- **About**—Provides the current version of Ambari.
- **Manage Ambari**—Open the management screen where Users, Groups, Permissions, and Ambari Views can be created and configured.
- **Settings**—Provides the option to turn off the progress window. (See Figure 9.15.)
- **Sign Out**—Exits the interface.—Exits the interface.

MANAGING HADOOP SERVICES

- During the course of normal Hadoop cluster operation, services may fail for any number of reasons.
- Ambari monitors all of the Hadoop services and reports any service interruption to the dashboard.
- In addition, when the system was installed, an administrative email for the Nagios monitoring system was required.
- All service interruption notifications are sent to this email address.
- Figure 9.10 shows the Ambari dashboard reporting a down DataNode.
- The service error indicator numbers next to the HDFS service and Hosts menu item indicate this condition.
- The DataNode widget also has turned red and indicates that 3/4 DataNodes are operating.

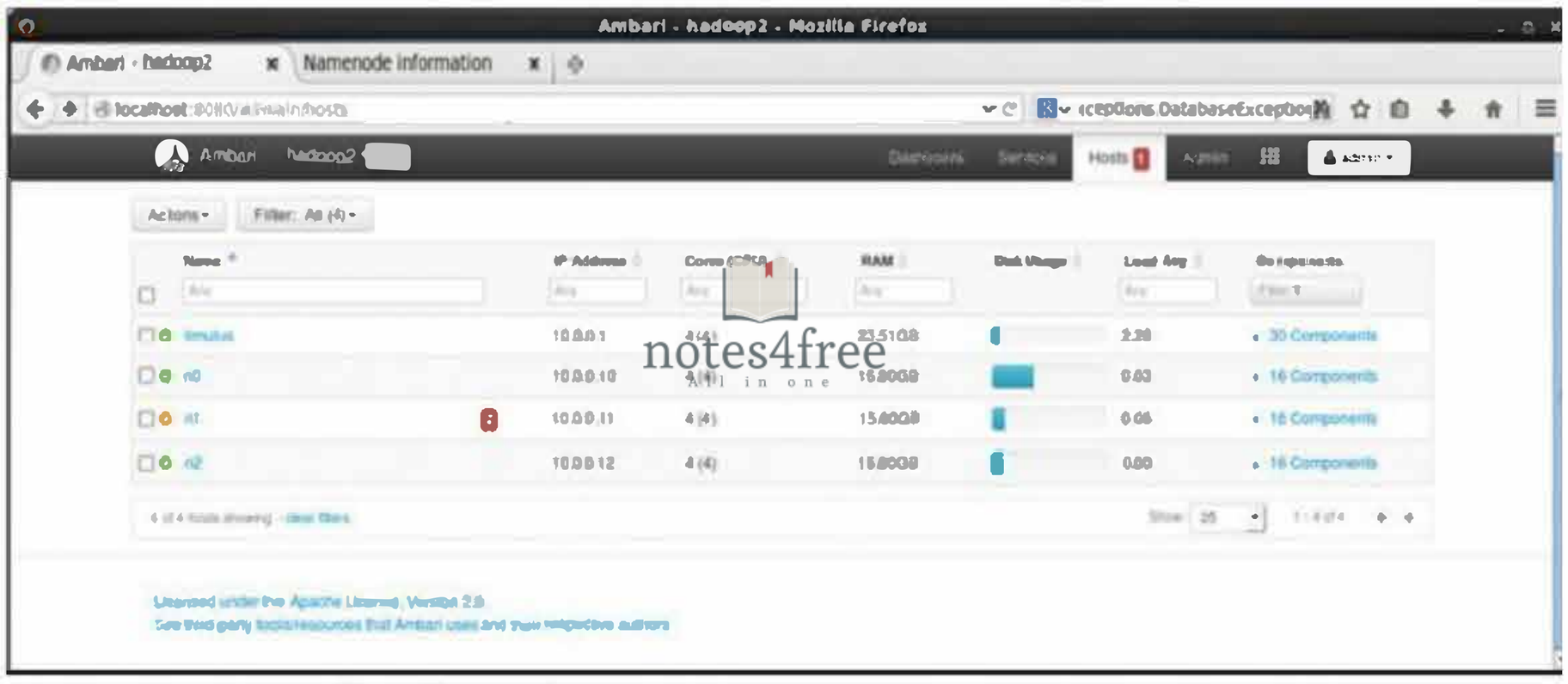
Figure 9.10 Ambari main dashboard indicating a DataNode issue



The specific host (or hosts) with an issue can be found by examining the Hosts window. As shown in Figure 9.12, the status of host n1 has changed from a green dot with a check mark inside to a yellow dot with a dash inside.

- An orange dot with a question mark inside indicates the host is not responding and is probably down.
- Other service interruption indicators may also be set as a result of the unresponsive node.

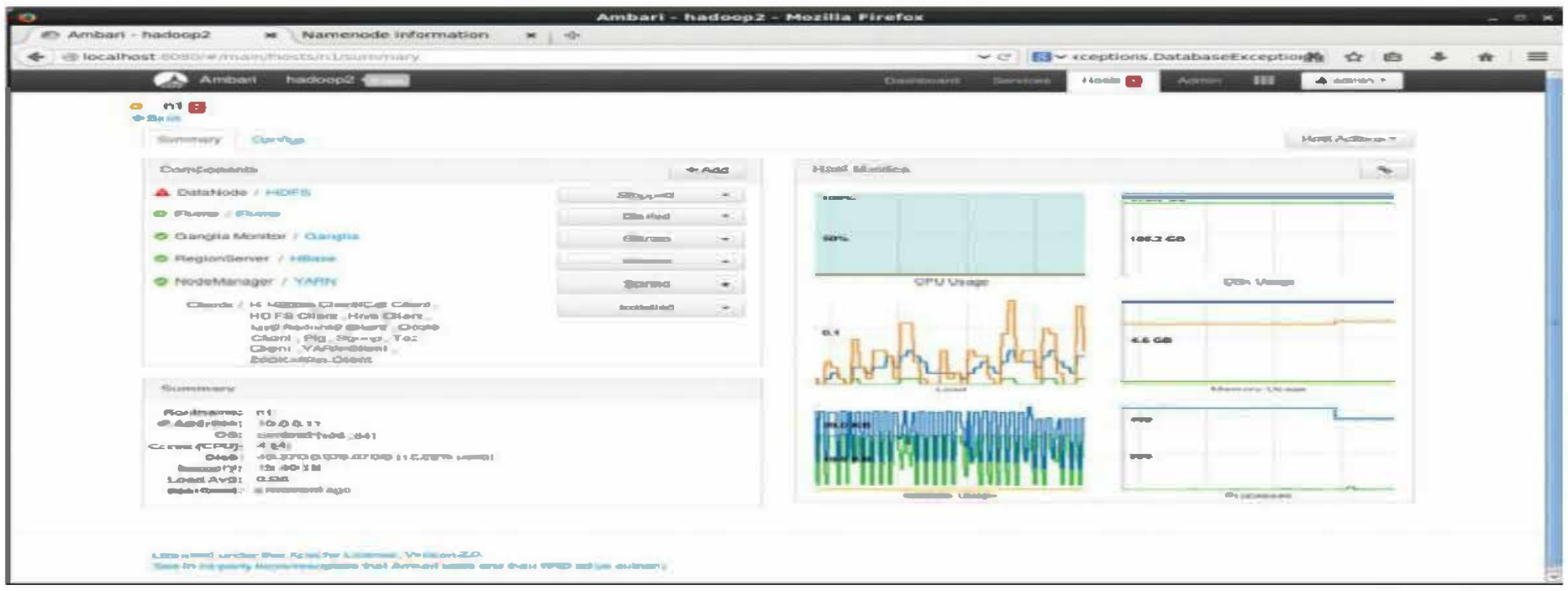
Figure 9.11 Ambari HDFS service summary window indicating a down DataNode



Clicking on the n1 host link opens the view in Figure 9.13.

- Inspecting the Components sub-window reveals that the DataNode daemon has stopped on the host.
- At this point, checking the DataNode logs on host n1 will help identify the actual cause of the failure.
- Assuming the failure is resolved, the DataNode daemon can be started using the Start option in the pull-down menu next to the service name.

Figure 9.13 Ambari window for host n1 indicating the DataNode/HDFS service has stopped



When the DataNode daemon is restarted, a confirmation similar to Figure 9.14 is required from the user.

- When a service daemon is started or stopped, a progress window similar to Figure 9.15 is opened.
- The progress bar indicates the status of each action.
- Note that previous actions are part of this window.
- If something goes wrong during the action, the progress bar will turn red.
- If the system generates a warning about the action, the process bar will turn orange.

Figure 9.15 Ambari progress window for DataNode restart

When these background operations are running, the small ops (operations) bubble on the top menu bar will indicate how many operations are running. (If different service daemons are started or stopped, each process will be run to completion before the next one starts.)

- Once the DataNode has been restarted successfully, the dashboard will reflect the new status (e.g., 4/4 DataNodes are Live).
- As shown in Figure 9.16, all four DataNodes are now working and the service error indicators are beginning to slowly disappear.

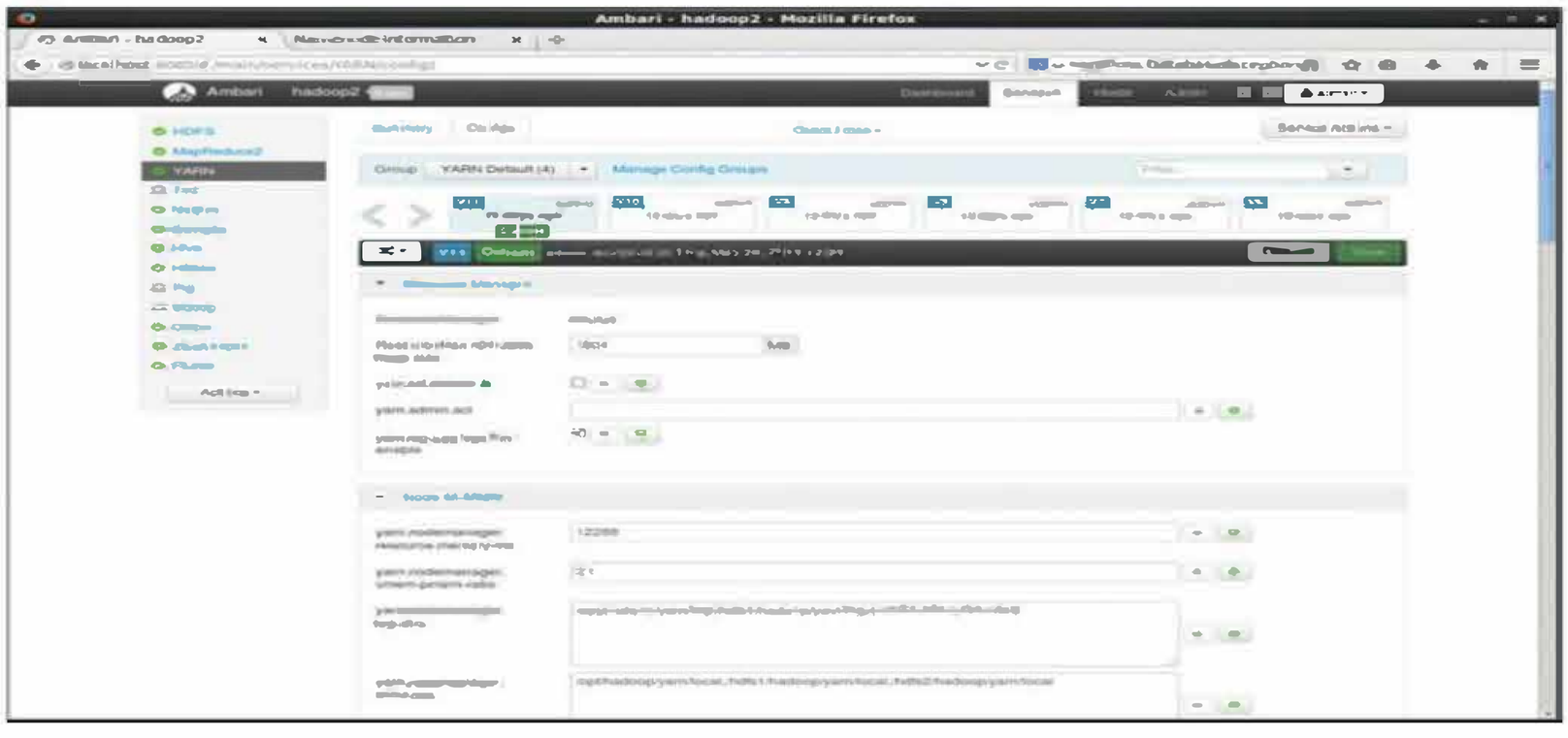
- The service error indicators may lag behind the real time widget updates for several minutes.

Figure 9.16 Ambari dashboard indicating all DataNodes are running

HANGING HADOOP PROPERTIES

- One of the challenges of managing a Hadoop cluster is managing changes to cluster-wide configuration properties.
- In addition to modifying a large number of properties, making changes to a property often requires restarting daemons (and dependent daemons) across the entire cluster.
- This process is tedious and time consuming.
- Fortunately, Ambari provides an easy way to manage this process.
- As described previously, each service provides a Configs tab that opens a form displaying all the possible service properties.
- Any service property can be changed (or added) using this interface. As an example, the configuration properties for the YARN scheduler are shown in Figure 9.17.

Figure 9.17 Ambari YARN properties view



To easily view the application logs, this property must be set to true.

- This property is normally on by default.

- As an example, for our purposes here, we will use the Ambari interface to disable this feature.
- As shown in Figure 9.18, when a property is changed, the green Save button becomes activated.

Basic Hadoop Administration Procedures

Adding Users to HDFS

To quickly create user accounts manually on a Linux-based system, perform the following steps:

1. Add the user to the group for your operating system on the HDFS client system. In most cases, the groupname should be that of the HDFS superuser, which is often hadoop or hdfs.

```
useradd -G <groupname> <username>
```

2. Create the username directory in HDFS.

```
hdfs dfs -mkdir /user/<username>
```

3. Give that account ownership over its directory in HDFS.

```
hdfs dfs -chown <username>:<groupname> /user/<username>
```

Perform an FSCK on HDFS

To check the health of HDFS, you can issue the hdfs fsck <path> (file system check) command. The entire HDFS namespace can be checked, or a subdirectory can be entered as an argument to the command. The following example checks the entire HDFS namespace.

```
$ hdfs fsck /
```

Other options provide more detail, include snapshots and open files, and management of corrupted files.

- -move moves corrupted files to /lost+found.
- -delete deletes corrupted files.
- -files prints out files being checked.
- -openforwrite prints out files opened for writes during check.
- -includeSnapshots includes snapshot data. The path indicates the existence of a snapshottable directory or the presence of snapshottabledirectories under it.
- -list-corruptfileblocks prints out a list of missing blocks andthe files to which they belong.
- -blocks prints out a block report.
- -locations prints out locations for every block.
- -racks prints out network topology for data-node locations.

Balancing HDFS

- Based on usage patterns and DataNode availability, the number of data blocks across the DataNodes may become unbalanced. To avoid over-utilized DataNodes, the HDFS balancer tool rebalances data blocks across the available DataNodes.
- Data blocks are moved from over-utilized to under-utilized nodes to within a certain percent threshold.

- Rebalancing can be done when new DataNodes are added or when a DataNode is removed from service.
- This step does not create more space in HDFS, but rather improves efficiency.

The HDFS superuser must run the balancer. The simplest way to run the balancer is to enter the following command:

```
$ hdfs balancer
```

- By default, the balancer will continue to rebalance the nodes until the number of data blocks on all DataNodes are within 10% of each other.
- The balancer can be stopped, without harming HDFS, at any time by entering a Ctrl-C.
- Lower or higher-thresholds can be set using the -threshold argument. For example, giving the following command sets a 5% threshold:

```
$ hdfs balancer -threshold 5
```

- The lower the threshold, the longer the balancer will run.
- To ensure the balancer does not swamp the cluster networks, you can set a bandwidth limit before running the balancer, as follows:

```
$ dfsadmin -setBalancerBandwidth newbandwidth
```

The newbandwidth option is the maximum amount of network bandwidth, in bytes per second, that each DataNode can use during the balancing operation.

HDFS Safe Mode

When the NameNode starts, it loads the file system state from the fsimage and then applies the edits log file. It then waits for DataNodes to report their blocks. During this time, the NameNode stays in a read-only Safe Mode. The NameNode leaves Safe Mode automatically after the DataNodes have reported that most file system blocks are available.

The administrator can place HDFS in Safe Mode by giving the following command:

```
$ hdfs dfsadmin -safemode enter
```

Entering the following command turns off Safe Mode:

```
$ hdfs dfsadmin -safemode leave
```

HDFS may drop into Safe Mode if a major issue arises within the file system (e.g., a full DataNode). The file system will not leave Safe Mode until the situation is resolved. To check whether HDFS is in Safe Mode, enter the following command:

```
$ hdfs dfsadmin -safemode get
```

HDFS Snapshots

HDFS snapshots are read-only, point-in-time copies of HDFS. Snapshots can be taken on a subtree of the file system or the entire file system. Some common use-cases for snapshots are data backup, protection against user errors, and disaster recovery.

Snapshots can be taken on any directory once the directory has been set as **snapshottable.** A snapshottable directory is able to accommodate 65,536 simultaneous snapshots. There is no limit on the number of snapshottable

directories. Administrators may set any directory to be snapshottable, but nested snapshottable directories are not allowed. For example, a directory cannot be set to snapshottable if one of its ancestors/descendants is a snapshottable directory.

The following example walks through the procedure for creating a snapshot.

The first step is to declare a directory as "snapshottable" using the following command:

```
$ hdfs dfsadmin -allowSnapshot /user/hdfs/war-and-peace-input
Allowing snapshot on /user/hdfs/war-and-peace-input succeeded
```

Once the directory has been made snapshottable, the snapshot can be taken with the following command. The command requires the directory path and a name for the snapshot—in this case, wapi-snap-1.

```
$ hdfs dfs -createSnapshot /user/hdfs/war-and-peace-input wapi-snap-1
Created snapshot /user/hdfs/war-and-peace-input/.snapshot/wapi-snap-1
```

The path of the snapshot is /user/hdfs/war-and-peaceinput/.snapshot/wapi-snap-1. The /user/hdfs/war-andpeace-input directory has one file, as shown by issuing the followingcommand:

```
$ hdfs dfs -ls /user/hdfs/war-and-peace-input/
Found 1 items
-rw-r--r-- 2 hdfs hdfs 3288746 2015-06-24 19:56 /user/hdfs/warand-
peaceinput/war-and-peace.txt
```

If the file is deleted, it can be restored from the snapshot:

```
$ hdfs dfs -rm -skipTrash /user/hdfs/war-and-peace-input/war-andpeace.
txt
Deleted /user/hdfs/war-and-peace-input/war-and-peace.txt
$ hdfs dfs -ls /user/hdfs/war-and-peace-input/
```

The restoration process is basically a simple copy from the snapshot to the previous directory (or anywhere else). Note the use of the ~/.snapshot/wapi-snap-1 path to restore the file:

```
$ hdfs dfs -cp /user/hdfs/war-and-peace-input/.snapshot/wapi-snap-1/war-
and-peace.txt /user/hdfs/war-and-peace-input
```

Confirmation that the file has been restored can be obtained by issuing the following command:

```
$ hdfs dfs -ls /user/hdfs/war-and-peace-input/
Found 1 items
-rw-r--r-- 2 hdfs hdfs 3288746 2015-06-24 21:12 /user/hdfs/warand-
peace-input/war-and-peace.txt
```

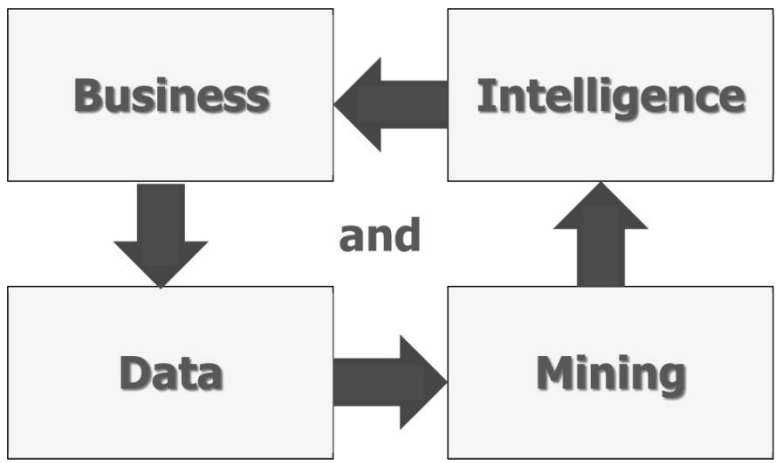

References:

- Douglas Eadline,"**Hadoop 2 Quick-Start Guide: Learn the Essentials of Big Data Computing in the Apache Hadoop 2 Ecosystem",** 1stEdition, Pearson Education, 2016. ISBN-13: 978-9332570351

Business Intelligence
Concepts and Applications

Business intelligence (BI) is an umbrella term that includes a variety of IT applications that are used to analyze an organization's data and commu-nicate the information to relevant users. Its major components are data warehousing, data mining, querying, and reporting

The nature of life and businesses is to grow. Information is the life-blood of business. Businesses use many techniques for understanding their environment and predicting the future for their own benefit and growth. Decisions are made from facts and feelings. Data-based decisions are more effective than those based on feelings alone. Actions based on accurate data, information, knowledge, experimentation, and testing, using fresh insights, can more likely succeed and lead to sustained growth.



Business intelligence and data mining cycle

One's own data can be the most effective teacher. Therefore, organizations should gather data, sift through it, analyze and mine it, find insights, and then embed those insights into their operating procedures.

There is a new sense of importance and urgency around data as it is being viewed as a new natural resource. It can be mined for value, insights, and competitive advantage. In a hyperconnected world, where everything is potentially connected to everything else, with potentially infinite correlations, data represents the impulses of nature in the form of certain events and attributes. A skilled business person is motivated to use this cache of data to harness nature, and to find new niches of unserved opportunities that could become profitable ventures.

BI Applications

BI tools are required in almost all industries and functions. The nature of the information and the speed of action may be different across busi-nesses, but every manager today needs access to BI tools to have up-to-date metrics about business performance. Businesses need to embed new insights into their operating processes to ensure that their activities continue to evolve with more efficient practices. The following are some areas of applications of BI and data mining.

Customer Relationship Management

A business exists to serve a customer. A happy customer becomes a repeat customer. A business should understand the needs and sentiments of the customer, sell more of its offerings to the existing customers, and also, expand the

pool of customers it serves. BI applications can impact many aspects of marketing.

1. *Maximize the return on marketing campaigns:* Understanding the customer's pain points from data-based analysis can ensure that the marketing messages are fine-tuned to better resonate with customers.
2. *Improve customer retention (churn analysis):* It is more difficult and expensive to win new customers than it is to retain existing custom-ers. Scoring each customer on their likelihood to quit can help the business design effective interventions, such as discounts or free ser-vices, to retain profitable customers in a cost-effective manner.
3. *Maximize customer value (cross-selling, upselling):* Every contact with the customer should be seen as an opportunity to gauge their cur-rent needs. Offering a customer new products and solutions based on those imputed needs can help increase revenue per customer. Even a customer complaint can be seen as an opportunity to wow the cus-tomer. Using the knowledge of the customer's history and value, the business can choose to sell a premium service to the customer.
4. *Identify and delight highly valued customers:* By segmenting the cus-tomers, the best customers can be identified. They can be proactively contacted, and delighted, with greater attention and better service. Loyalty programs can be managed more effectively.
5. *Manage brand image:* A business can create a listening post to listen to social media chatter about itself. It can then do sentiment analysis of the text to understand the

nature of comments and respond ap-propriately to the prospects and customers.

Health Care and Wellness

Health care is one of the biggest sectors in advanced economies. Evi-dence-based medicine is the newest trend in data-based health care man-agement. BI applications can help apply the most effective diagnoses and prescriptions for various ailments. They can also help manage public health issues, and reduce waste and fraud.

1. *Diagnose disease in patients:* Diagnosing the cause of a medical condi-tion is the critical first step in a medical engagement. Accurately diag-nosing cases of cancer or diabetes can be a matter of life and death for the patient. In addition to the patient's own current situation, many

other factors can be considered, including the patient's health history, medication history, family's history, and other environmental factors. This makes diagnosis as much of an art form as it is science. Systems, such as IBM Watson, absorb all the medical research to date and make probabilistic diagnoses in the form of a decision tree, along with a full explanation for their recommendations. These systems take away most of the guess work done by doctors in diagnosing ailments.

2. *Treatment effectiveness:* The prescription of medication and treatment is also a difficult choice out of so many possibilities. For example, there are more than 100 medications for hypertension (high blood pressure) alone. There are also interactions in terms of which drugs work well with others and which drugs do not. Decision trees can help doctors learn about and prescribe more effective treatments. Thus, the patients could recover their health faster with a lower risk of complications and cost.
3. *Wellness management:* This includes keeping track of patient health records, analyzing customer health trends, and proactively advising them to take any needed precautions.
4. *Manage fraud and abuse:* Some medical practitioners have unfortu-nately been found to conduct unnecessary tests and/or overbill the gov-ernment and health insurance companies. Exception-reporting systems can identify such providers, and action can be taken against them.

5. *Public health management:* The management of public health is one of the important responsibilities of any government. By using effec-tive forecasting tools and techniques, governments can better predict the onset of disease in certain areas in real time. They can thus be better prepared to fight the diseases. Google has been known to pre-dict the movement of certain diseases by tracking the search terms (like flu, vaccine) used in different parts of the world.

Education

As higher education becomes more expensive and competitive, it is a great user of data-based decision-making. There is a strong need for efficiency, increasing revenue, and improving the quality of student experience at all levels of education.

1. *Student enrolment (recruitment and retention):* Marketing to new po-tential students requires schools to develop profiles of the students that are most likely to attend. Schools can develop models of what kinds of students are attracted to the school, and then reach out to those students. The students at risk of not returning can be flagged, and corrective measures can be taken in time.
2. *Course offerings:* Schools can use the class enrolment data to develop models of which new courses are likely to be more popular with students. This can help increase class size, reduce costs, and improve student satisfaction.
3. *Alumni pledges:* Schools can develop predictive models of which alumni are most likely to pledge financial support to the school. Schools can create a profile for alumni more likely to pledge dona-tions to the school. This could lead to a reduction in the cost of mailings and other forms of outreach to alumni.

Retail

Retail organizations grow by meeting customer needs with quality prod-ucts, in a convenient, timely, and cost-effective manner. Understanding emerging customer shopping patterns can help retailers organize their products, inventory, store layout, and web presence in order to delight their customers, which in turn would help increase revenue and profits. Retailers generate a lot of transaction and logistics data that can be used to solve problems.

1. *Optimize inventory levels at different locations:* Retailers need to man-age their inventories carefully. Carrying too much inventory imposes carrying costs, while carrying too little inventory can cause stock-outs and lost sales opportunities. Predicting sales trends dynamically can help retailers move inventory to where it is most in demand. Retail organizations can provide their suppliers with real-time in-formation about sales of their items so that the suppliers can deliver their product to the right locations and minimize stock-outs.
2. *Improve store layout and sales promotions:* A market basket analy-sis can develop predictive models of which products sell together

often. This knowledge of affinities between products can help re-tailers co-locate those products. Alternatively, those affinity prod-ucts could be located farther apart to make the customer walk the length and breadth of the store, and thus be exposed to other products. Promotional discounted product bundles can be created to push a nonselling item along with a set of products that sell well together.

3. *Optimize logistics for seasonal effects:* Seasonal products offer tremen-dously profitable short-term sales opportunities, yet they also offer the risk of unsold inventories at the end of the season. Understand-ing which products are in season in which market can help retailers dynamically manage prices to ensure their inventory is sold during the season. If it is raining in a certain area, then the inventory of umbrella and ponchos could be rapidly moved there from nonrainy areas to help increase sales.
4. *Minimize losses due to limited shelf life:* Perishable goods offer chal-lenges in terms of disposing off the inventory in time. By tracking sales trends, the perishable products at risk of not selling before the sell-by date can be suitably discounted and promoted.

Banking

Banks make loans and offer credit cards to millions of customers. They are most interested in improving the quality of loans and reducing bad debts. They also want to retain more good customers and sell more services to them.

1. *Automate the loan application process:* Decision models can be gen-erated from past data that predict the likelihood of a loan proving successful. These can be inserted in business processes to automate the financial loan application process.
2. *Detect fraudulent transactions:* Billions of financial transactions hap-pen around the world every day. Exception-seeking models can iden-tify patterns of fraudulent transactions. For example, if money is being transferred to an unrelated account for the first time, it could be a fraudulent transaction.

3. *Maximize customer value (cross-selling, upselling):* Selling more prod-ucts and services to existing customers is often the easiest way to increase revenue. A checking account customer in good standing could be offered home, auto, or educational loans on more favorable terms than other customers, and thus, the value generated from that customer could be increased.
4. *Optimize cash reserves with forecasting:* Banks have to maintain cer-tain liquidity to meet the needs of depositors who may like to with-draw money. Using past data and trend analysis, banks can forecast how much to keep, and invest the rest to earn interest.

Financial Services

Stock brokerages are an intensive user of BI systems. Fortunes can be made or lost based on access to accurate and timely information.

1. *Predict changes in bond and stock prices:* Forecasting the price of stocks and bonds is a favorite pastime of financial experts as well as lay people. Stock transaction data from the past, along with other variables, can be used to predict future price patterns. This can help traders develop long-term trading strategies.
2. *Assess the effect of events on market movements:* Decision models using decision trees can be created to assess the impact of events on changes in market volume and prices. Monetary policy changes (such as Fed Reserve interest rate change) or geopolitical changes (such as war in a part of the world) can be factored into the predictive model to help take action with greater confidence and less risk.

3. *Identify and prevent fraudulent activities in trading:* There have un-fortunately been many cases of insider trading, leading to many prominent financial industry stalwarts going to jail. Fraud detection models can identify and flag fraudulent activity patterns.

Insurance

This industry is a prolific user of prediction models in pricing insurance proposals and managing losses from claims against insured assets.

1. *Forecast claim costs for better business planning:* When natural disas-ters, such as hurricanes and earthquakes, strike, loss of life and prop-erty occurs. By using the best available data to model the likelihood (or risk) of such events happening, the insurer can plan for losses and manage resources and profits effectively.
2. *Determine optimal rate plans:* Pricing an insurance rate plan requires covering the potential losses and making a profit. Insurers use actu-ary tables to project life spans and disease tables to project mortality rates, and thus price themselves competitively yet profitably.
3. *Optimize marketing to specific customers:* By microsegmenting potential customers, a data-savvy insurer can cherry-pick the best customers and leave the less profitable customers to its competitors. Progressive Insur-ance is a U.S.-based company that is known to actively use data mining to cherry-pick customers and increase its profitability.

4. *Identify and prevent fraudulent claim activities:* Patterns can be iden-tified as to where and what kinds of fraud are more likely to occur. Decision-tree-based models can be used to identify and flag fraudu-lent claims.

Manufacturing

Manufacturing operations are complex systems with interrelated subsys-tems. From machines working right, to workers having the right skills, to the right components arriving with the right quality at the right time, to money to source the components, many things have to go right. Toyota's famous lean manufacturing company works on just-in-time inventory systems to optimize investments in inventory and to improve flexibility in their product mix.

1. *Discover novel patterns to improve product quality:* Quality of a prod-uct can also be tracked, and this data can be used to create a predic-tive model of product quality deteriorating. Many companies, such as automobile companies, have to recall their products if they have found defects that have a public safety implication. Data mining can help with root cause analysis that can be used to identify sources of errors and help improve product quality in the future.

2. *Predict/prevent machinery failures:* Statistically, all equipment is likely to break down at some point in time. Predicting which machine is likely to shut down is a complex process. Decision models to forecast machinery failures could be constructed using past data. Preventive maintenance can be planned, and manufacturing capacity can be adjusted, to account for such maintenance activities.

Telecom

BI in telecom can help with churn management, marketing/customer profiling, network failure, and fraud detection.

1. *Churn management:* Telecom customers have shown a tendency to switch their providers in search for better deals. Telecom companies tend to respond with many incentives and discounts to hold on to customers. However, they need to determine which customers are at a real risk of switching and which others are just negotiating for a better deal. The level of risk should to be factored into the kind of deals and discounts that should be given. Millions of such customer calls happen every month. The telecom companies need to provide a consistent and data-based way to predict the risk of the customer switching, and then make an operational decision in real time while the customer call is taking place. A decision-tree- or a neural network-based system can be used to guide the customer-service call operator

to make the right decisions for the company, in a consistent manner.

2. *Marketing and product creation:* In addition to customer data, tele-com companies also store call detail records (CDRs), which pre-cisely describe the calling behavior of each customer. This unique data can be used to profile customers and then can be used for creating new products/services bundles for marketing purposes. An American telecom company, MCI, created a program called Friends & Family that allowed calls with one's friends and family on that network to be totally free and thus, effectively locked many people into their network.

Data Warehousing

A data warehouse (DW) is an organized collection of integrated, subject-oriented databases designed to support decision support functions. DW is organized at the right level of granularity to provide clean enterprise-wide data in a standardized format for reports, queries, and analysis. DW is physically and functionally separate from an operational and transac-tional database. Creating a DW for analysis and queries represents signifi-cant investment in time and effort. It has to be constantly kept up-to-date for it to be useful. DW offers many business and technical benefits.

DW supports business reporting and data mining activities. It can facilitate distributed access to up-to-date business knowledge for depart-ments and functions, thus improving business efficiency and customer service. DW

can present a competitive advantage by facilitating decision making and helping reform business processes.

DW enables a consolidated view of corporate data, all cleaned and or-ganized. Thus, the entire organization can see an integrated view of itself. DW thus provides better and timely information. It simplifies data access and allows end users to perform extensive analysis. It enhances overall IT performance by not burdening the operational databases used by Enter-prise Resource Planning (ERP) and other systems.

Design Considerations for DW

The objective of DW is to provide business knowledge to support deci-sion making. For DW to serve its objective, it should be aligned around those decisions. It should be comprehensive, easy to access, and up-to-date. Here are some requirements for a good DW:

1. *Subject-oriented:* To be effective, DW should be designed around a subject domain, that is, to help solve a certain category of problems.
2. *Integrated:* DW should include data from many functions that can shed light on a particular subject area. Thus, the organization can benefit from a comprehensive view of the subject area.
3. *Time-variant (time series):* The data in DW should grow at daily or other chosen intervals. That allows latest comparisons over time.
4. *Nonvolatile:* DW should be persistent, that is, it should not be cre-ated on the fly from the operations databases.

Thus, DW is consis-tently available for analysis, across the organization and over time.

5. *Summarized:* DW contains rolled-up data at the right level for queries and analysis. The rolling up helps create consistent granularity for effec-tive comparisons. It helps reduces the number of variables or dimensions of the data to make them more meaningful for the decision makers.
6. *Not normalized:* DW often uses a star schema, which is a rectangular central table, surrounded by some lookup tables. The single-table view significantly enhances speed of queries.
7. *Metadata:* Many of the variables in the database are computed from other variables in the operational database. For example, total daily sales may be a computed field. The method of its calculation for each variable should be effectively documented. Every element in DW should be sufficiently well-defined.
8. *Near real-time and/or right-time (active):* DWs should be updated in near real-time in many high-transaction volume industries, such as air-lines. The cost of implementing and updating DW in real time could discourage others. Another downside of real-time DW is the possibili-ties of inconsistencies in reports drawn just a few minutes apart.

DW Development Approaches

There are two fundamentally different approaches to developing DW: top down and bottom up. The top-down approach is to make a comprehensive DW that covers all the

reporting needs of the enterprise. The bottom-up approach is to produce small data marts, for the reporting needs of different departments or functions, as needed. The smaller data marts will eventually align to deliver comprehensive EDW capabilities. The top-down approach provides consistency but takes time and resources. The bottom-up approach leads to healthy local ownership and maintainability of data.

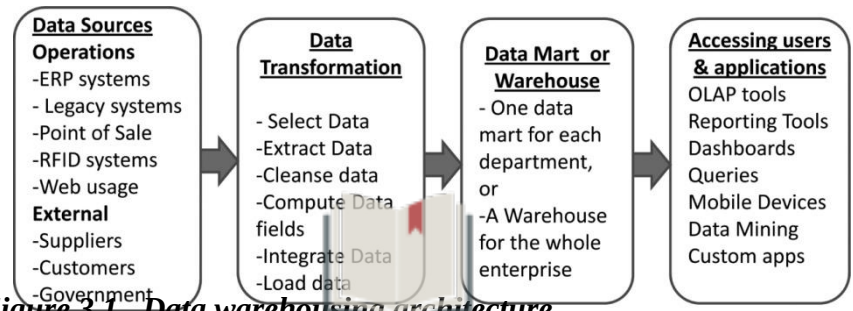


Figure 3.1 Data warehousing architecture

DW Architecture

DW has four key elements (Figure 3.1). The first element is the data sources that provide the raw data. The second element is the process of transforming that data to meet the decision needs. The third element is the methods of regularly and accurately loading of that data into EDW or data marts. The fourth element is the data access and analysis part, where devices and applications use the data from DW to deliver insights and other benefits to users.

Data Sources

DWs are created from structured data sources. Unstructured data, such as text data, would need to be structured before inserted into DW.

1. Operations data include data from all business applications, includ-ing from ERPs systems that form the backbone of an organization's IT systems. The data to be extracted will depend upon the subject matter of DW. For example, for a sales/marketing DW, only the data about customers, orders, customer service, and so on would be extracted.
2. Other applications, such as point-of-sale (POS) terminals and -e-commerce applications, provide customer-facing data. Supplier data could come from supply chain management systems. Planning and budget data should also be added as needed for making com-parisons against targets.
3. External syndicated data, such as weather or economic activity data, could also be added to DW, as needed, to provide good contextual information to decision makers.

Data Transformation Processes

The heart of a useful DW is the processes to populate the DW with good quality data. This is called the extract-transform-load (ETL) cycle.

1. Data should be extracted from many operational (transactional) da-tabase sources on a regular basis.
2. Extracted data should be aligned together by key fields. It should be cleansed of any irregularities or missing values. It should be rolled up together to the same level of granularity. Desired fields, such as daily sales totals, should be computed. The entire data should then be brought to the same format as the central table of DW.
3. The transformed data should then be uploaded into DW.

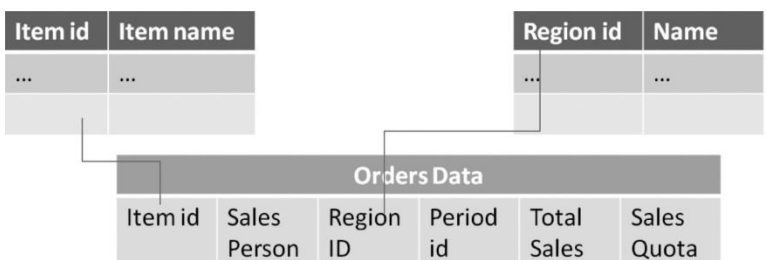
This ETL process should be run at a regular frequency. Daily trans-action data can be extracted from ERPs, transformed, and uploaded to the database the same night. Thus, DW is up-to-date next morning. If DW is needed for near-real-time information access, then the ETL pro-cesses would need to be executed more frequently. ETL work is usually automated using programing scripts that are written, tested, and then deployed for periodic updating DW.

DW Design

Star schema is the preferred data architecture for most DWs. There is a central fact table that provides most of the information of interest. There are lookup tables that provide detailed values for codes used in the central table. For example, the central table may use digits to represent a sales person. The lookup table will help provide the name for that sales person code. Here is an example of a star schema for a data mart for monitoring sales performance (Figure 3.2).

Other schemas include the snowflake architecture. The difference be-tween a star and snowflake is that in the latter, the lookup tables can have their own further lookup tables.

There are many technology choices for developing DW. This includes selecting the right database management system and the right set of data management tools. There are a few big and reliable providers of DW sys-tems. The provider of the operational DBMS may be chosen for DW also.



Alternatively, a best-of-breed DW vendor could be used. There are also a variety of tools out there for data migration, data upload, data retrieval, and data analysis.

DW Access

Data from DW could be accessed for many purposes, through many devices.

1. A primary use of DW is to produce routine management and moni-toring reports. For example, a sales performance report would show sales by many dimensions, and compared with plan. A dashboard-ing system will use data from the warehouse and present analysis to users. The data from DW can be used to populate customized performance dashboards for executives. The dashboard could in-clude drill-down capabilities to analyze the performance data for root cause analysis.
2. The data from the warehouse could be used for ad hoc queries and any other applications that make use of the internal data.

3. Data from DW is used to provide data for mining purposes. Parts of the data would be extracted, and then combined with other relevant data, for data mining.

Data Mining

Gathering and Selecting Data

The total amount of data in the world is doubling every 18 months. There is an ever-growing avalanche of data coming with higher velocity, volume, and variety. One has to quickly use it or lose it. Smart data mining re-quires choosing where to play. One has to make judicious decisions about what to gather and what to ignore, based on the purpose of the data min-ing exercises. It is like deciding where to fish; not all streams of data will be equally rich in potential insights.

To learn from data, one needs to effectively gather quality data, clean and organize it, and then efficiently process it. One requires the skills and technologies for consolidation and integration of data elements from many sources. Most organizations develop an enterprise data model (EDM), which is a unified, high-level model of all the data stored in an organiza-tion's databases. The EDM will be inclusive of the data generated from all internal systems. The EDM provides the basic menu of data to create a data warehouse for a particular decision-making purpose. Data warehouses help organize all this data in a useful manner so that it can be selected and deployed for mining. The EDM can also help imagine what relevant exter-nal data should be gathered to develop good predictive relationships with the internal data. In the United States, the governments and their agencies make a vast variety and quantity of data available at data.gov.

Gathering and curating data takes time and effort, particularly when it is unstructured or semistructured. Unstructured data can come in many forms like databases, blogs, images, videos, and chats. There are streams of unstructured social media data from blogs, chats, and tweets. There are also streams of machine-generated data from connected machines, RFID tags, the internet of things, and so on. The data should be put in rectangular data shapes with clear columns and rows before submitting it to data mining.

Knowledge of the business domain helps select the right streams of data for pursuing new insights. Data that suits the nature of the problem being solved should be gathered. The data elements should be relevant, and suitably address the problem being solved. They could directly im-pact the problem, or they could be a suitable proxy for the effect being measured. Select data will also be gathered from the data warehouse.

Industries and functions will have their own requirements and con-straints. The health care industry will provide a different type of data with different data names. The HR function would provide different kinds of data. There would be different issues of quality and privacy for these data.

Data Cleansing and Preparation

The quality of data is critical to the success and value of the data mining project. Otherwise, the situation will be of the kind of garbage in and garbage out (GIGO). Duplicate data needs to be removed. The same data may be received from multiple sources. When merging the data sets, data must be de-duped.

1. Missing values need to be filled in, or those rows should be removed from analysis. Missing values can be filled in with average or modal or default values.
2. Data elements may need to be transformed from one unit to an-other. For example, total costs of health care and the total number of patients may need to be reduced to cost/patient to allow comparabil-ity of that value.
3. Continuous values may need to be binned into a few buckets to help with some analyses. For example, work experience could be binned as low, medium, and high.
4. Data elements may need to be adjusted to make them comparable over time. For example, currency values may need to be adjusted

for inflation; they would need to be converted to the same base year for comparability. They may need to be converted to a common currency.

6. Outlier data elements need to be removed after careful review, to avoid the skewing of results. For example, one big donor could skew the analysis of alumni donors in an educational setting.
7. Any biases in the selection of data should be corrected to ensure the data is representative of the phenomena under analysis. If the data in-cludes many more members of one gender than is typical of the popu-lation of interest, then adjustments need to be applied to the data.
8. Data should be brought to the same granularity to ensure compara-bility. Sales data may be available daily, but the sales person compen-sation data may only be available monthly. To relate these variables, the data must be brought to the lowest common denominator, in this case, monthly.
9. Data may need to be selected to increase information density. Some data may not show much variability, because it was not properly recorded or for any other reasons. This data may dull the effects of other differences in the data and should be removed to improve the information density of the data.

Outputs of Data Mining

Data mining techniques can serve different types of objectives. The out-puts of data mining will reflect the objective being served. There are many representations of the outputs of data mining.

One popular form of data mining output is a decision tree. It is a hierarchically branched structure that helps visually

follow the steps to make a model-based decision. The tree may have certain attributes, such as probabilities assigned to each branch. A related format is a set of busi-ness rules, which are if-then statements that show causality. A decision tree can be mapped to business rules. If the objective function is predic-tion, then a decision tree or business rules are the most appropriate mode of representing the output.

The output can be in the form of a regression equation or mathe-matical function that represents the best fitting curve to represent the data. This equation may include linear and nonlinear terms. Regression

Evaluating Data Mining Results

There are two primary kinds of data mining processes: supervised learn-ing and unsupervised learning. In supervised learning, a decision model can be created using past data, and the model can then be used to predict the correct answer for future data instances. Classification is the main category of supervised learning activity. There are many techniques for classification, decision trees being the most popular one. Each of these techniques can be implemented with many algorithms. A common met-ric for all of classification techniques is predictive accuracy.

Predictive Accuracy 5 (Correct Predictions) / Total Predictions

Suppose a data mining project has been initiated to develop a predictive model for cancer patients using a decision tree. Using a relevant set of variables and data instances, a decision tree model has been created. The model is then used to predict other data instances. When a true positive data point is positive, that is a correct prediction, called a true positive (TP). Similarly, when a true negative data point is classified as negative, that is a true negative (TN). On the other hand, when a true-positive data

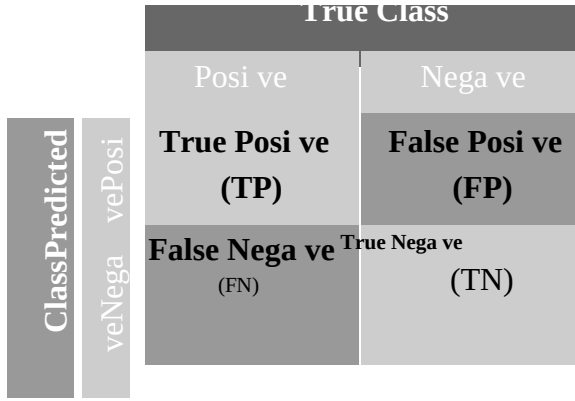


Figure Confusion matrix

point is classified by the model as negative, that is an incorrect prediction, called a false negative (FN). Similarly, when a true-negative data point is classified as positive, that is classified as a false positive (FP). This is called the confusion matrix (Figure 4.1).

Thus, the predictive accuracy can be specified by the following formula.

Predictive Accuracy = (TP + TN) / (TP + TN + FP + FN).

All classification techniques have a predictive accuracy associated with a predictive model. The highest value can be 100 percent. In practice, predictive models with more than 70 percent accuracy can be considered usable in business domains, depending upon the nature of the business.

There are no good objective measures to judge the accuracy of unsupervised learning techniques, such as cluster analysis. There is no single right answer for the results of these techniques. The value of the segmentation model depends upon the value the decision maker sees in those results.

Data Mining Techniques

Data may be mined to help make more efficient decisions in the future. Or it may be used to explore the data to find interesting associative pat-terns. The right technique depends upon the kind of problem being solved

Important Data Mining Techniques		
Supervised Learning: Classification	Machine Learning Techniques	Decision Trees
		Artificial Neural Networks
	Statistical Techniques	Regression
Unsupervised Learning: Exploration	Machine Learning Techniques	Cluster Analysis
		Association Rule Mining

Figure : Important data mining techniques

The most important class of problems solved using data mining are classification problems. These are problems where data from past deci-sions is mined to extract the few rules and patterns that would improve the accuracy of the decision-making process in the future. The data of past decisions is organized and mined for decision rules or equations, which are then codified to produce more accurate decisions. Classifica-tion techniques are called supervised learning as there is a way to super-vise whether the model's prediction is right or wrong.

A decision tree is a hierarchically organized branched, structured to help make decision in an easy and logical manner. *Decision trees* are the most popular data mining technique, for many reasons.

1. Decision trees are easy to understand and easy to use, by analysts as well as executives. They also show a high predictive accuracy.
2. They select the most relevant variables automatically out of all the available variables for decision-making.
3. Decision trees are tolerant of data quality issues and do not require much data preparation from the users.
4. Even nonlinear relationships can be handled well by decision trees.

Data Visualization

As data and insights grow in number, a new requirement is the ability of the executives and decision makers to absorb this information in real time. There is a limit to human comprehension and visualization capac-ity. That is a good reason to prioritize and manage with fewer but key variables that relate directly to the key result areas of a role.

Here are few considerations when presenting data:

1. Present the conclusions and not just report the data.
2. Choose wisely from a palette of graphs to suit the data.
3. Organize the results to make the central point stand out.
4. Ensure that the visuals accurately reflect the numbers. Inappropriate visuals can create misinterpretations and misunderstandings.
5. Make the presentation unique, imaginative, and memorable.

Module IV
Decision Trees, Regression, Artificial Neural Networks, Cluster Analysis, Association Rule Mining

Decision Trees

INTRODUCTION

- Decision Trees are a simple way to guide one's path to a decision.
- The decision may be a simple binary one, for example, whether to approve a loan or not; or it may be a complex multi-valued decision, as to what may be the diagnosis for a particular sickness.
- Decision trees are hierarchically branched structures that help one come to a decision based on asking certain questions in a particular sequence.
- Decision trees are one of the most widely used techniques for classification.
- A good decision tree should be short and ask only a few meaningful questions.
- They are very efficient to use, easy to explain, and their classification accuracy is competitive with other methods.
- Decision trees can generate knowledge from a few test instances that can then be applied to a broad population.
- Decision trees are used mostly to answer relatively simple binary decisions.

DECISION TREE PROBLEM

A decision tree would have a predictive accuracy based on how often it makes correct decisions.

- The more data available for training the decision tree, the more accurate its knowledge extraction will be, and thus, it will make more accurate decisions.
- The more variables the tree can choose from, the greater is the accuracy of the decision tree.
- In addition, a good decision tree should also be frugal so that it takes the least number of questions, and thus, the least amount of effort to get to the right decision.

Here is an exercise to create a decision tree that helps make decisions about approving the play of an outdoor game. The objective is to predict the play decision given the atmospheric conditions out there. The decision is -Should the game be allowed or not? Here is the decision problem.

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	Normal	True	??

To answer this question, one should look at the past experiences, and see what decision was made in a similar instance, if such an instance exists.

Look up the database of past decisions to find the answer. Dataset 6.1 shows a list of the decisions taken in 14 instances of past soccer game situations.

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	*No*
Sunny	Hot	High	True	*No*
Overcast	Hot	High	False	*Yes*
Rainy	Mild	High	False	*Yes*
Rainy	Cool	Normal	False	*Yes*
Rainy	Cool	Normal	True	*No*
Overcast	Cool	Normal	True	*Yes*
Sunny	Mild	High	False	*No*
Sunny	Cool	Normal	False	*Yes*
Rainy	Mild	Normal	False	*Yes*
Sunny	Mild	Normal	True	*Yes*
Overcast	Mild	High	True	*Yes*
Overcast	Hot	Normal	False	*Yes*
Rainy	Mild	High	True	*No*

- If there was a row for Sunny/Hot/Normal/Windy condition in the data table, it would match the current problem; the decision from that row could be used to answer the current problem.
- However, there is no such past instance in this case.

There are three disadvantages of looking up the data table

1. As mentioned earlier, how to decide if there isn't a row that corresponds to the exact situation today? If there is no exact matching instance available in the database, the past experience cannot guide the decision.
2. Searching through the entire past database may be time consuming, depending on the number of variables and the organization of the database.
3. What if the data values are not available for all the variables? In this instance, if the data for humidity variable was not available, looking up the past data would not help.

- A better way of solving the problem is to abstract the knowledge from the past data into decision tree or rules.
- These rules can be represented in a decision tree, and then that tree can be used to make the decisions.

- The decision tree may not need values for all the variables.

DECISION TREE CONSTRUCTION

- A decision tree is a hierarchically branched structure.
- What should be the first question asked in creating the tree? One should ask the more important questions first, and the less important questions later.
- What is the most important question that should be asked to solve the problem?
- How is the importance of the questions determined? Thus, how should the root node of the tree be determined?

Determining the Root Node of the Tree

- In this example, there are four choices based on the four variables.
- One can begin by asking one of the following questions -what is the outlook, what is the temperature, what is the humidity, and what is the wind speed?
- A criterion should he used to evaluate these choices. The key criterion would be that, which one of these questions gives the most insight about the situation?
- Another way to look at it would be the criterion of frugality. That is, which question will provide us the shortest ultimate decision tree?
- Another way to look at this is that if one is allowed to ask only one question, which one would one ask?
- In this case, the most important question should be the one that, by itself, helps make the most correct decisions with the fewest errors.
- The four questions can now be systematically compared, to see which variable by itself will help make the most correct decisions.
- One should systematically calculate the correctness of decisions based on each question.
- Then one can select the question with the most correct predictions, or the fewest errors.
- Start with the first variable in this case outlook. It can take three values, sunny, overcast, and rainy.
- Start with the sunny value of outlook.
- There are live instances where the outlook is sunny. In 2 of the 5 instances, the play decision was yes, and in the other three, the decision was no.
- Thus, if the decision rule was that Outlook: **sunny--> No**, then 3 out of 5 decisions would be correct, while 2 out of 5 such decision would be incorrect.
- There are 2 errors out of 5. This can be recorded in Row 1

Attribute	Rules	Error	Total Error
Outlook	Sunny→No	2/5	

Similar analysis can be done for other values of the outlook variable.

- There are four instances where the outlook is overcast. In all the 4 instances, the play decision was yes.
- Thus, if the decision rule was that Outlook: **overcast-->Yes**, then 4 out, of 4 decisions would be correct,
- While none of decisions would be incorrect There are 0 errors out 0f 4This can be recorded in the next row.

Attribute	Rules	Error	Total Error
Outlook	Sunny→No	2/5	
	Overcast →yes	0/4	

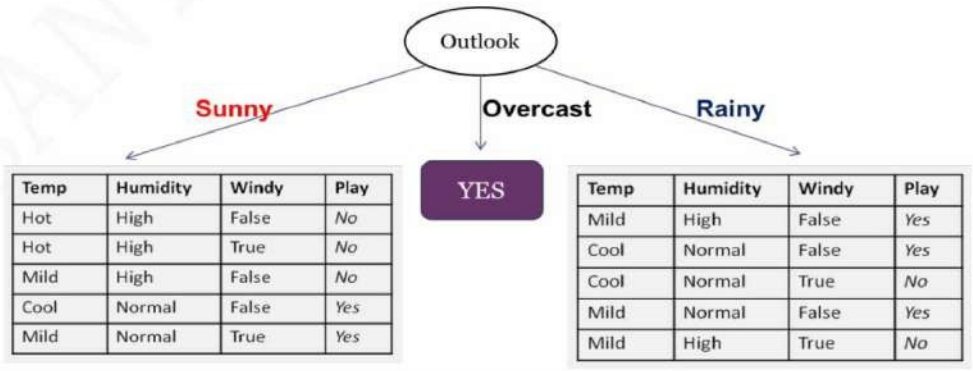
- There are five instances where the outlook is rainy. In 3 of the 5 instances, the play decision was yes, and in the other three, the decision was no.
- Thus, if the decision rule was that Outlook: r**ainy-->Yes**, then 3 out of 5 decisions would be correct, while 2 out of 5 decisions would be incorrect.
- There will be 2 out of 5 errors. This can be recorded in next row.
- Adding up errors for all values of outlook, there are 4 errors out of 14. In other words, outlook gives 10 correct decisions out of 14, and 4 incorrect ones.
- A similar analysis can be done for the other three variables. At the end of the analytical exercise, the following error table (Dataset 6.2) can be constructed.

Attribute	Rules	Error	Total Error
Outlook	Sunny→No	2/5	4/14
	Overcast →yes	0/4	
	Rainy →yes	2/5	
Temp	Hot →No	2/4	5/14
	Mild →Yes	2/6	
	Cool → Yes	1/4	
Humidity	High → No	3/7	4/14
	Normal →Yes	1/7	
Windy	False →Yes	2/8	5/14
	True →No	3/6	

- The variable that leads to the least number of errors (and thus the most number of correct decisions) should be chosen as the first node. In this case, two variables have the least number of errors.
- There is a tie between outlook and humidity, as both have 4 errors out of 14 instances. The tie can be broken using another criterion, the purity of resulting sub-trees.
- If all the errors were concentrated in few of the sub-trees and some of the branches were completely free of error, then that is preferred from a usability perspective.
- Outlook has one error-free branch, for the overcast value, while there is no such pure subclass for humidity variable.
- Thus, the tie is broken in favor of outlook.
- The decision tree will use outlook as the first node, or the first splitting variable.
- The first question that should be asked to solve the play problem is, 'What is the value of outlook'?

Splitting the Tree

- From the root node, the decision tree will be split into three branches or sub-trees, one for each of the three values of outlook.
- Data for the root node (the entire data) will be divided into three segments, one for each of the value of outlook.
- The sunny branch will inherit the data for the instances that had 'sunny' as the value of outlook.
- These will be used for further building of that sub-tree.
- Similarly the rainy branch will inherit data for the instances that had 'rainy' as the value of outlook.
- These will be used for further building of that sub-tree. The overcast branch will inherit the data for the instances that had 'overcast' as the outlook. However, there will be no need to build further on that branch.
- There is a clear decision -Yes, for all instances when outlook value is overcast.
- The decision tree will look like as follows (Figure 6.1) after the first level of splitting.



Temp	Humidity	Windy	Play
Hot	High	False	*No*
Hot	High	True	*No*
Mild	High	False	*No*
Cool	Normal	False	*Yes*
Mild	Normal	True	*Yes*

Temp	Humidity	Windy	Play
Mild	High	False	*Yes*
Cool	Normal	False	*Yes*
Cool	Normal	True	*No*
Mild	Normal	False	*Yes*
Mild	High	True	*No*

Determining the Next Nodes of the Tree

- Similar recursive logic of tree building should be applied to each branch.
- For the sunny branch on the left, error values will be calculated for the three other variables temperature, humidity and windy.
- Final comparison will look like as shown in Dataset 6.3 given below.

Attribute	Rules	Error	Total Error
Temp	Hot->No	0/2	1/5
	Mild ->No	1/2	
	Cool -> yes	0/1	
Humidity	High->No	0/3	0/5
	Normal->Yes	0/2	
Windy	False->No	1/3	2/5
	True->Yes	1/2	

Attribute	Rules	Error	Total Error
Temp	Mild->Yes	1/3	2/5
	Cool->yes	1/2	
Humidity	High->No	1/2	1/5
	Normal->Yes	1/3	
Windy	False->Yes	0/3	0/5
	True-No	0/2	

- The variable of humidity shows the least amount of error, i.e., zero error.
- The other two variables have non-zero errors. Thus the Outlook: sunny branch on the left will use humidity as the next splitting variable.
- Similar analysis should be done for the 'rainy' value of the tree. The following Dataset 6.4 depicts such analysis.
- For the rainy branch, it can similarly be seen that the variable windy gives all the correct answers, while none of the other two variables makes all the correct decisions.
- This is how the final decision tree will look like. Here it is produced using Weka open-source data mining platform (Figure 6.2). This is the model that abstracts the knowledge of the past data of decision.

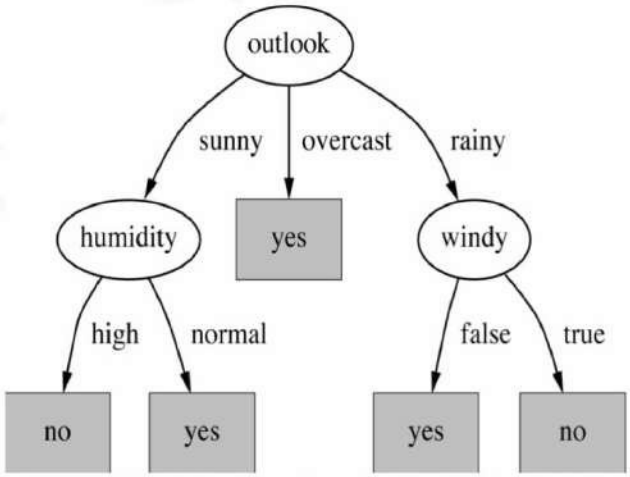


FIGURE 6.2 Decision Tree for the Weather Problem

This decision tree can be used to solve the current problem. Here is the problem again.

According to the tree, the first question to ask is about outlook. In this problem, the outlook is sunny, so the decision problem moves to the 'sunny' branch of the tree. The node in that subtree is humidity. In the problem, humidity is normal. That branch leads to an answer -Yes. Thus, the answer to the play problem is a yes.

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	Normal	True	*YES*

LESSONS FROM CONSTRUCTING TRESS

Here are some benefits of using the decision tree compared with looking up the answers from the data table (Table 6.1)

	Decision Tree	Table Lookup
Accuracy	Varied level of accuracy	100% accurate
Generality	General. Applies to all situations	Applies only when a similar case occurred before
Frugality	Only three variables needed	All four variables are needed
Simple	Only one or two questions asked	All four variable values are needed
Easy	Logical, and easy to understand	Can be cumbersome to look up; no understanding of the logic behind the decision

Here are a few observations about how the trees was constructed

- ✓ The final decision tree has zero errors in mapping to the prior data. In other words, the tree has a predictive accuracy of 100%. The tree completely fits the data. In real life situations, such perfect predictive accuracy is not possible when making decision trees. When there are larger, complicated datasets, with many more variables, a perfect fit is unachievable. This is especially true in business and social contexts, where things are not always fully clear and consistent.
- ✓ The decision tree algorithm selected the minimum number of variables that are needed to solve the problem. Thus, one can start with all available data variables, and let the decision-tree algorithm select the ones that are useful, and discard the rest.

- ✓ This tree is almost symmetric with all branches being of almost similar lengths. However, in real life situations, some of the branches may be much longer than the others, and the tree may need to be pruned to make it more balanced and usable.
- ✓ It may be possible to increase predictive accuracy by making more sub-trees and making the tree longer. However, the marginal accuracy gained from each subsequent level in the tree will be less, and may not be worth the loss in ease and interpretability of the tree. If the branches are long and complicated it will be difficult to understand and use. The longer branches may need to be trimmed to keep the tree easy to use.
- ✓ A perfectly fitting tree has the danger of over-fitting the data, thus capturing all the random variations in the data. It may fit the training data well, but may not do well in predicting the future real instances.
- ✓ There was a single best tree for this data. There could however be two or more equally efficient decision trees of similar length with similar predictive accuracy for the same dataset. Decision trees are based strictly on patterns within the data, and do not rely on any underlying theory of the problem domain. When multiple candidate trees are available, one could choose whichever is easier to understand, communicate or implement.

DECISION TREE ALGORITHMS

As we saw, decision trees employ the divide and conquer method. The data is branched at each node according to certain criteria until all the data is assigned to leaf nodes. It recursively divides a training set until each division consists of examples from one class.

The following is a pseudo code for making decision trees

1. Create a root node and assign all of the training data to it.
2. Select the best splitting attribute according to certain criteria.
3. Add a branch to the root node for each value of the split.
4. Split the data into mutually exclusive subsets along the lines of the specific split.
5. Repeat steps 2 and 3 for each and every leaf node until a stopping criteria is reached.

There are many algorithms for making decision trees. Decision tree algorithms differ on three key elements

Splitting Criteria

1. Which variable to use for the first split? How should one determine the most important variable for the first branch and subsequently for each subtree?
   - ➢ Algorithms use different measures like least errors, information gain, Gini'9 coefficient etc., to compute the splitting variable that provides the most benefit.
   - ➢ Information gain is a mathematical construct to compute the reduction in information entropy from a prior state to the next state that takes some information as given.

- The greater the reduction in entropy, the better it is.
- The Gini coefficient is a statistical concept that measures the inequality among values of a frequency distribution. The lower the Gini's coefficient; the better it is.

2. What values to use for the split? If the variables have continuous values such as for age or blood pressure, what value-ranges should be used to make bins?
3. How many branches should be allowed for each node? There could be binary trees, with just two branches at each node. Or there could be more branches allowed.

Stopping Criteria

- When to stop building the tree? There are two major ways to make this determination.
- The tree building can be stopped when a certain depth of the branches has been reached and the tree becomes unreadable after that.
- The tree can also be stopped when the error level at any node is within predefined tolerable levels.

Pruning

- It is the act of reducing the size of decision trees by removing sections of the tree that provide little value.
- The decision tree could be trimmed to make it more balanced, more general and more easily usable.
- The symptoms of an over-fitted tree are that it is too deep with too many branches which may reflect anomalies due to random noise or outliers instead of the underlying relationship.
- Pruning is often done after the tree is constructed. There are two approaches to avoid over-fitting.

1. Pre-pruning means to halt the tree construction early, when certain criteria are met. The downside is that, it is difficult to decide what criteria to use for halting the construction, because we do not know what may happen subsequently if we keep growing the tree.
2. Post-pruning means removing branches or sub-trees from a "fully grown" tree. This method is commonly used. C4.5 algorithm uses a statistical method to estimate the errors at each node for pruning. A validation set may be used for pruning as well.

The most popular decision tree algorithms are C5, CART and CHAID (Table 6.2)

Decision-Tree	C4.5	CART	CHAID
Full Name	Iterative Dichotomiser (ID3)	Classification and Regression Trees	Chi-square Automatic Interaction Detector
Basic algorithm	Hunt's algorithm	Hunt's algorithm	adjusted significance testing
Developer	Ross Quinlan	Bremman	Gordon Kass
When developed	1986	1984	1980
Types of trees	Classification	Classification & Regression trees	Classification & regression
Serial implementation	Tree-growth & Tree-pruning	Tree-growth & Tree-pruning	Tree-growth & Tree-pruning
Type of data	Discrete & Continuous; Incomplete data	Discrete and Continuous	Non-normal data also accepted
Types of splits	Multi-way splits	Binary splits only; Clever surrogate splits to reduce tree depth	Multi-way splits as default
Splitting criteria	Information gain	Gini's coefficient, and others	*Chi*-square test
Pruning Criteria	Clever bottom-up technique avoids overfitting		Trees can become very large
Implementation	Publicly available	Publicly available in most packages	Popular in market research, for segmentation

Regression

INTRODUCTION

- Regression is a well-known statistical technique to model the predictive relationship between several independent variables (DVs) and one dependent variable.
- The objective is to find the best-fitting curve for a dependent variable in a multidimensional space, with each independent variable being a dimension.
- The curve could be a straight line, or it could be a nonlinear curve.
- The quality of fit of the curve to the data can be measured by a coefficient of correlation (r), which is the square root of the amount of variance explained by the curve.

The key steps for regression are simple

1. List all the variables available for making the model.
2. Establish a Dependent Variable (DV) of interest.
3. Examine visual (if possible) relationships between variables of interest.
4. Find a way to predict DV using other variables.

CORRELATIONS AND RELATIONSHIPS

- Statistical relationships are about which elements of data hang together and which ones hang separately.

- It is about categorizing variables that have a relationship with one another and categorizing variables that are distinct and unrelated to other variables and describing significant positive relationships and significant negative differences.
- The first and foremost measure of the strength of a relationship is co-relation (or correlation).
- The strength of a correlation is a quantitative measure that is measured in a normalized range between 0 and 1.
- A correlation of 1 indicates a perfect relationship, where the two variables are in perfect sync.
- A correlation of 0 indicates that there is no relationship between the variables.
- The relationship can be positive or it can be an inverse relationship, that is, the variables may move together in the same direction or in the opposite direction.
- Therefore, a good measure of correlation is the correlation coefficient, which is the square root of correlation. This coefficient, called r, can thus range from -1 to +1.
- An r value of 0 signifies no relationship. An r value of -1 shows perfect relationship in the same direction, and an 7' value of -1 shows a perfect relationship but moving in opposite directions.
- Given two numeric variables x and y, the coefficient of correlation r is mathematically computed by the following equation. X- (called x-bar) is the mean of x, and y-(y-bar) is the mean of y.

$$r = \frac{[(x-\bar{x})][y-\bar{y}]}{\sqrt{[(x-\bar{x})^2][(y-\bar{y})^2]}}$$

$$r = \frac{(x-\bar{x})(y-\bar{y})}{\sqrt{[(x-\bar{x})^2][(y-\bar{y})^2]}}$$

VISUAL LOOK AT RELATIONSHIPS

- A scatter plot (or scatter diagram) is a simple exercise for plotting all the data pints between two variables on a two-dimensional graph.
- It provides a visual layout of all the data points placed in that two-dimensional space.
- The scatter plot can be useful for graphically intuiting the relationship between the two variables.
- Figure 7.1 shows many possible patterns in scatter diagrams.
- Chart (a) shows a very strong linear relationship between the variables x and y. This means the value of y increases proportionally with x.
- Chart (b) also shows a strong linear relationship between the variables x and y. Here, it is an inverse relationship. That means the value of y decreases proportionally with x.
- Chart (c) shows a curvilinear relationship. It is an inverse relationship, which means that the value of y decreases proportionally with x.

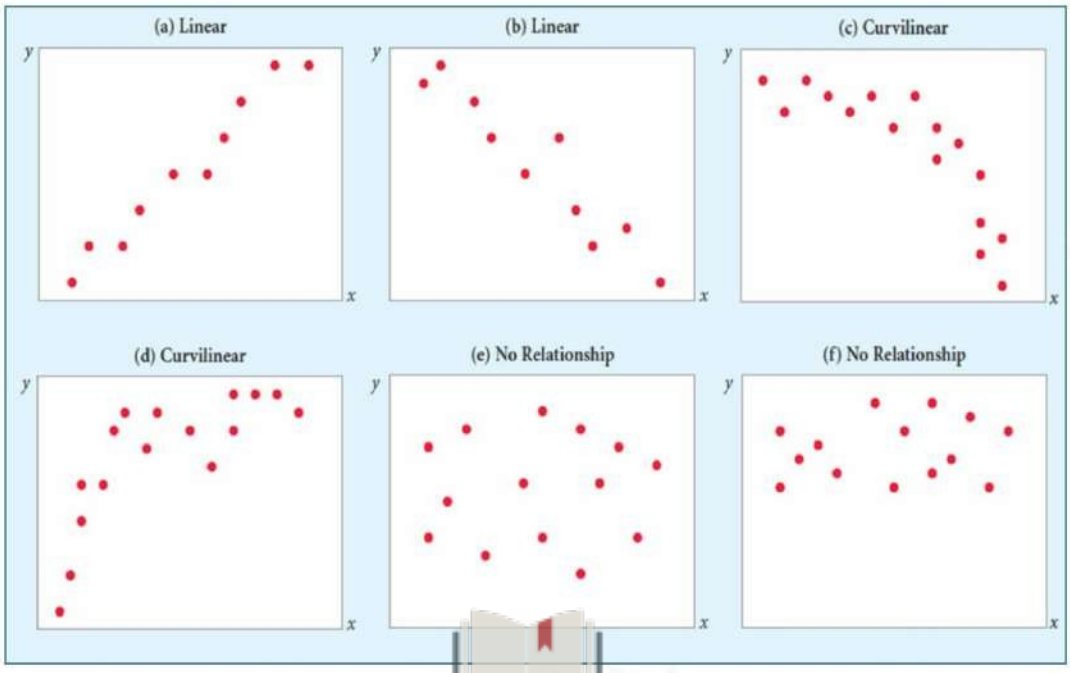


FIGURE 7.1 Scatter Plots showing Types of Relationships among Two Variables

- However, it seems a relatively well-defined relationship, like an arc of a circle, which can be represented by a simple quadratic equation (quadratic means the power of two that is, using terms like ac2 and y2).

Regression Exercise

The regression model is described as a linear equation that follows. y is the dependent variable, that is, the variable being predicted. x is the independent variable, or the predictor variable. There could be many predictor variables (such as x1, x2, ...) in a regression equation. However, there can be only one dependent variable (y) in the regression equation.

$$y=\beta_0 + \beta_1 x + \dot{\varepsilon}$$

- Where β0 and β1 are the constant, and the co-efficient for the x variable; and ε̇ is the random error variable.
- A simple example of a regression equation would be to predict a house price from the size of the house. Dataset 7.1 shows a sample house prices data:
- The two dimensions (one predictor and one outcome variable) of the data can be plotted on a scatter diagram.

House Price	Size (sqft)
$229,500	1850
$273,300	2190
$247,000	2100
$195,100	1930
$261,000	2300
$179,700	1710
$168,500	1550
$234,400	1920
$168,800	1840
$180,400	1720
$156,200	1660
$288,350	2405
$186,750	1525
$202,100	2030
$256,800	2240

- A scatter plot with a best-fitting line looks like the graph that follows (Figure 7.2).

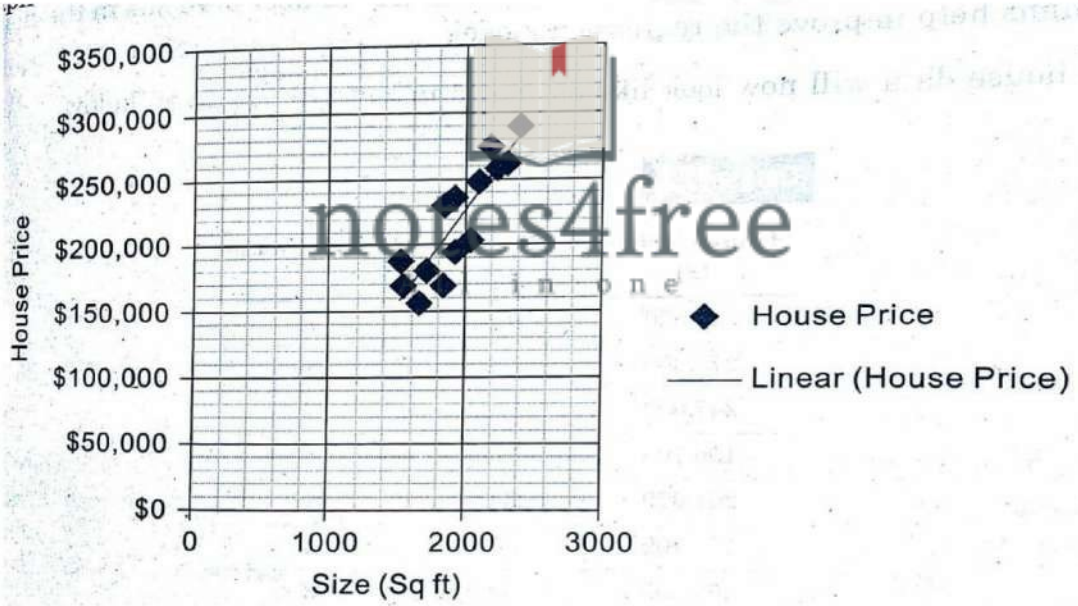


FIGURE 7.2. Scatter Plot and Regression Equation between House Price and House Size

- Visually, one can see a positive correlation between house price and size (Sq ft).
- However, the relationship is not perfect. Running a regression model between the two variables produces the following output (truncated).
- It shows the coefficient of correlation to be 0.891. r2, the measure of total variance explained by the equation, is 0.794 or 79%.

- That means the two variables are moderately and positively correlated. Regression coefficients help create the following equation for predicting house prices.

Regression Statistics	
Multiple R	0.891
R Square	0.794
	Coefficients
Intercept	-54191
Size (sqft)	139.48

House Price ($) = 139.48 * Size (Sq ft) -54191

- This equation explains only 79% of the variance in house prices. Suppose other predictor variables are made available, such as the number of rooms in the house, It might help improve the regression model.
- The house data will now look like as shown in Dataset 7.2 given below.

House Price	Size (sqft)	#Rooms
$229,500	1850	4
$273,300	2190	5
$247,000	2100	4
$195,100	1930	3
$261,000	2300	4
$179,700	1710	2
$168,500	1550	2
$234,400	1920	4
$168,800	1840	2
$180,400	1720	2
$156,200	1660	2
$288,350	2405	5
$186,750	1525	3
$202,100	2030	2
$256,800	2240	4

While it is possible to make a three dimensional scatter plot, one can alternatively examine the correlation matrix among the variables.

- It shows that the house price has a strong correlation with number of rooms (0.944) as well. Thus, it is likely that adding this variable to the regression model will add to the strength of the model.

	House Price	Size (sqft)	#Rooms
House Price	1		
Size (sqft)	0.891	1	
Rooms	0.944	0.748	1

- Running a regression model between these three variables produces the following output (truncated).

Regression Statistics	
Multiple R	0.984
R Square	0.968
Coefficients	
Intercept	12923
Size(sqft)	65.60
Rooms	23613

- It shows that the coefficient of correlation of this regression model is 0.984. R2, the total variance explained by the equation, is 0.968 or 97%.
- That means the variables are positively and very strongly correlated. Adding a new relevant variable has helped improve the strength of the regression model.
- Using the regression coefficients helps create the following equation for predicting house prices.

House Price($) = 65.6 * Size (Sq ft) + 23613 * Rooms + 12924

- This equation shows a 97% goodness of fit with the data, which is very good for business and economic data.
- There is always some random variation in naturally occurring business data, and it is not desirable to over fit the model to the data.
- This predictive equation should be used for future transactions. Given a situation as below, it will be possible to predict the price of the house with 2000 Sq ft and 3 rooms.

House Price	Size (Sq ft)	#No. of Rooms
??	2000	3

House Price ($) '= 65.6 2: 2000 (Sq ft) + 23613 23 + 12924 5 $214,963

- The predicted values should be compared with the actual values to see how close the model is able to predict the actual value.
- As new data points become available, there are opportunities, to fine-tune and improve the model.

NON-LINEAR REGRESSION EXERCISE

- The relationship between the variables may also be curvilinear.
- For example, given past data from electricity consumption (kWh) and temperature(K), the objective is to predict the electrical consumption from the temperature value, Dataset 7.3 shows a dozen past observations.

Temp	Kwatts
59.2	9,730
61.9	9,750
55.1	10,180
66.2	10,230
52.1	10,800
69.9	11,160
46.8	12,530
76.8	13,910
79.7	15,110
79.3	15,690
80.2	17,020
83.3	17,880

- In two dimensions (one predictor and one outcome variable), data can be plotted on a scatter diagram. A scatter plot with a best-fitting line looks like the graph on next page (Figure 7 .3).
- It is visually clear that the first line does not fit the data well. The relationship between temperature and Kwatts follows a curvilinear model, where it hits bottom at a certain value of temperature.
- The regression model confirms the relationship since R 1s only 0.77 and Rsquare 1s also only 60%. Thus, only 60% of the variance is explained
- This regression model can be enhanced by introducing a nonlinear variable (such as a quadratic variable Temp2) in the equation. The second line is the relationship between kWh and Temp2.
- The scatter plot shows that energy consumption has a strong linear relationship With Temp2. Computing the regression model after adding the Temp2 variable leads to the following results the total variance explained by the equation is 0.985 or 98.5%.

- That means the variables are very strongly and positively correlated. The regression coefficients help create the following equation.

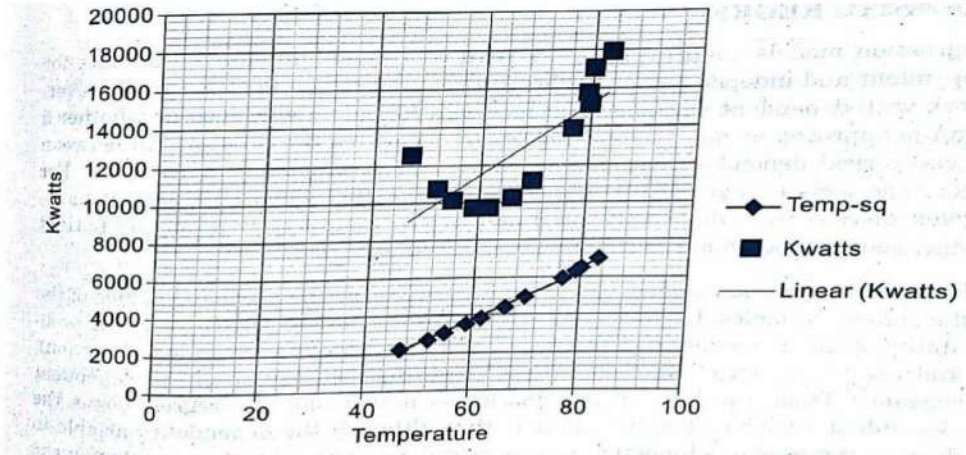


FIGURE 7.3 Scatter Plots showing Regression between (a) Kwatts and Temperature, and (b) Kwatts and Temperature Square

Regression Statistics	
r	0.992
r^2	0.984
	Coefficients
Intercept	67245
Temp (F)	1911
Temp Sq	15.87

It shows that the coefficient of correlation of the regression model is now.0.99. R2,

Energy Consumption (Kwatts) = 15.87 * Temp² –1911 * Temp + 67245

This equation shows a 98.5% fit which is very good for business and economic contexts. Now one can predict the Kwatts value when the temperature is 72 degree.

Energy consumption = (15.87 * 72 * 72) (1911 * 72) + 67245 = 11923 Kwatts

LOGISTIC REGRESSION

- Regression models traditionally work with continuous numeric value data for dependent and independent variables.
- Logistic regression models can, however Work with dependent variables that have categorical values, such as whether a loan IS approved or not.

- Logistic regression measures the relationship between a categorical dependent variable and one or more independent variables.
- For example, logistic regression might be used to predict whether a patient has a given disease (e.g., diabetes), based on observed characteristics of the patient (age, gender, body mass index, results of blood tests, etc.).
- Logistical regression models use probability scores as the predicted values of the dependent variables.
- Logistic regression takes the natural logarithm of the prob. ability of the dependent variable being a case (referred to as the logit function), and creates a continuous criterion as a transformed version of the dependent variable.
- Thus, the logit transformation is used in logistic regression as the dependent variable. The net effect is that although the dependent variable in logistic regression is binomial (or categorical, i.e., has only two possible values), the logit is the continuous function upon which linear regression is conducted.
- Here is the general logistic function with independent variable on the horizontal axis and, the logit dependent variable on the vertical axis (Figure 7.4).

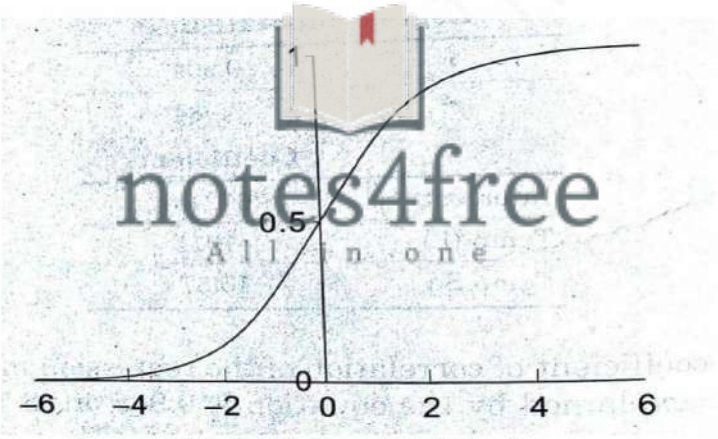


FIGURE 7.4 General Logit Function

All popular data mining platforms provide support for regular multiple regression models, as well as options for Logistic Regression.

ADVANTAGES AND DISADVANTAGES OF REGRESSION MODELS

Regression models are very popular because they offer many advantages. Few are as follows.

1. Regression models are easy to understand as they are built upon basic statistical principles such as correlation and least square error.
2. Regression models provide simple algebraic equations that are easy to understand and use.

3. The strength (or the goodness of fit) of the regression model is measured in terms of the correlation coefficients, and other related statistical parameters that are well understood.
4. Regression models can match and beat the predictive power of other modeling techniques.
5. Regression models can include all the variables that one wants to, include in the model.
6. Regression modeling tools are pervasive. They are found in statistical packages as well as data mining packages. MS-Excel spreadsheets can also provide simple regression modeling capabilities.

Regression models can however prove inadequate under many circumstances.

1. Regression models cannot cover for poor data quality issues. If the data is not prepared well to remove missing values, or is not well-behaved in terms of a normal distribution, the validity of the model suffers.
2. Regression models suffer from collinearity problems (meaning strong linear correlations among some independent variables). If the independent variables have strong correlations among themselves, then they will eat into each other's predictive power and the regression coefficients will lose their ruggedness.
3. Regression models will not automatically choose between highly collinear variables, although some packages attempt to do that.
4. Regression models can be unwieldy and unreliable if a large number of variables are included in the model. All variables entered into the model will be reflected in the regression equation, irrespective of their contribution to the predictive power of the model. There is no concept of automatic pruning of the regression model.
5. Regression models do not automatically take care of nonlinearity. The user needs to imagine the kind of additional terms that might be needed to be added to the regression model to improve its fit.
6. Regression models work only with numeric data and not with categorical Variables. There are ways to deal with categorical variables though by creating multiple new variables with a yes or no value.

Artificial Neural Networks

INTRODUCTION

Artificial Neural Networks (ANNs) are inspired by the information processing model of the brain. The human brain consists of billions of neurons that link with one another in an intricate pattern. Every neuron receives information from many other neurons, processes it, gets excited or not, and passes its state information to other neurons. Just like the brain is a multipurpose system, so also the ANNs are very versatile systems.

- They can be used for many kinds of pattern recognition and prediction.

- They are used in finance, marketing, manufacturing operations, information systems applications, and so on.
- ANNs are composed of a large number of highly interconnected processing 919' ments (neurons) working in multilayered structures that receive inputs, process the inputs, and produce an output.

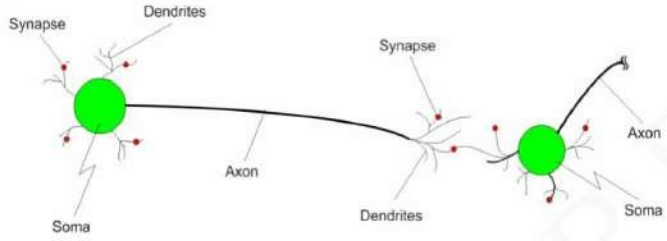


Figure: Two interconnected brain cells (neurons)

- An ANN is designed for a specific application, such as pattern recognition or data classification, and trained through a learning process. Just like in biological systems, ANNs make adjustments to the synaptic connections with each learning instance.
- ANNs are like a black box trained into solving a particular type of problem and they can develop high predictive powers.
- Their intermediate synaptic parameter values evolve as the system obtains feedback on its predictions and thus an AN learns from more training data (Figure 8.1).

FIGURE 8.1 General ANN Model

BUSINESS APPLICATIONS OF ANN

Neural networks are used most often when the objective function is complex and where there exists plenty of data and the model is expected to improve over a period of time. A few sample applications are as follows

1. They are used in stock price prediction where the rules of the game are extremely complicated, and a lot of data needs to be processed very quickly.
2. They are used for character recognition, as in recognizing hand written text, or damaged or mangled text. They are used in recognizing finger prints. These are complicated patterns and are unique for each person. Layers of neurons can progressively clarify the pattern leading to a remarkably accurate result.
3. They are also used in traditional classification problems, like approving a financial loan application.

DESIGN PRINCIPLES OF AN ARTIFICIAL NEURAL NETWORK

1. A neuron is the basic processing unit of the network. The neuron (or processing element) receives inputs from its preceding neurons (or PEs) does some nonlinear weighted computation on the basis of those inputs, transforms the result into its output value, and then passes on the output to the next neuron in the network (Figure 8.2). x's are the inputs, w's are the weights for each input, and y is the output.

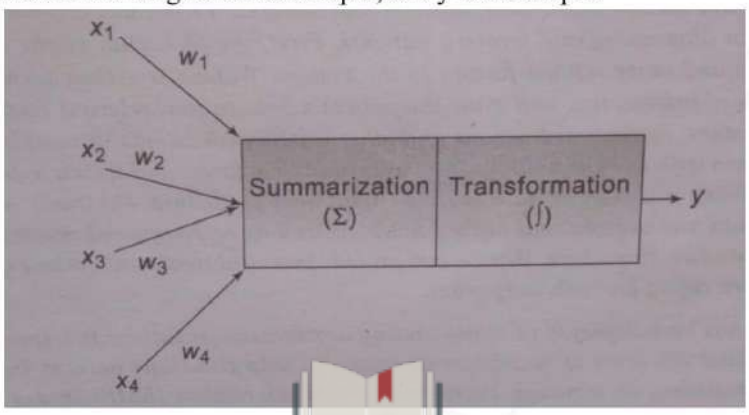


FIGURE 8.2 Model for a Single Artificial Neuron

2. A neural network is a multilayered model. There is at least one input neuron one output neuron, and at least one processing neuron. An ANN with just this basic structure would be a simple, single-stage computational unit. A simple task may be processed by just that one neuron and the result may be communicated soon. AN N s however, may have multiple layers of processing elements in sequence. There could be many neurons involved in a sequence depending upon the complexity of the predictive action. The layers of PEs could work in sequence or in parallel (Figure 8.3).

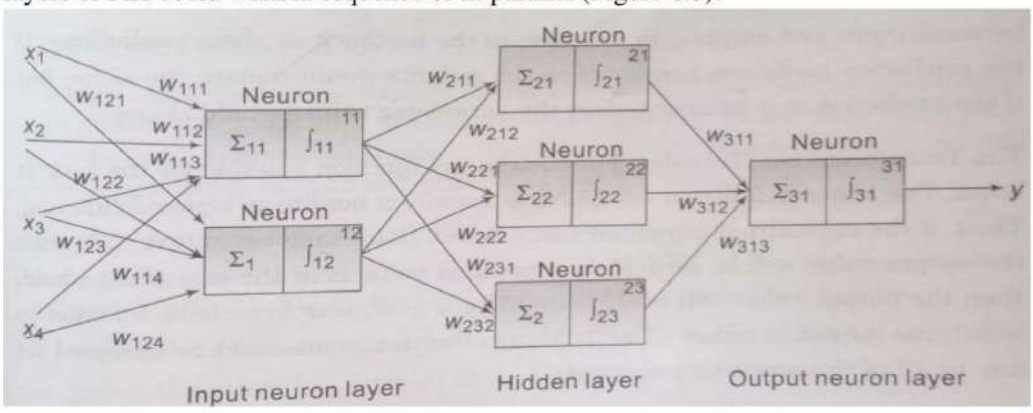


FIGURE 8.3 Model for a Multilayer ANN

3. The processing logic of each neuron may assign different weights to the various incoming input streams. The processing logic may also use nonlinear transformation, such as a sigmoid function, from the processed values to the output value. This processing logic and the intermediate weight and processing functions are just what works for the system as a whole, in its objective of solving a problem collectively. Thus, neural networks are considered to be an opaque and a black-box system.
4. The neural network can be trained by making similar decisions over and over again with many training cases. It will continue to learn by adjusting its internal computation and communication based on feedback about its previous decisions. Thus, the neural networks become better at making a decision as they handle more and more decisions.

- Depending upon the nature of the problem and the availability of good training data, at some point, the neural network will learn enough and begin to match the predictive accuracy of a human expert.
- In many practical situations, the predictions of ANN, trained over a long period of time with a large number of training data, have begun to decisively become more accurate than human experts. At that point, ANN can begin to be seriously considered for deployment in real situations in real time.

REPRESENTATION OF A NEURAL NETWORK

- A neural network is a series of neurons that receive inputs from other neurons.
- They do a weighted summation function of all the inputs, using different weights (or importance) for each input.
- The weighted sum is then transformed into an output value using a transfer-function.
- Learning in ANN occurs when the various processing elements in the neural network adjust the underlying relationship (weights, transfer function, etc.) between input and outputs, in response to the feedback on their predictions.
- If the prediction made was correct, then the weights would remain the same, but if the prediction was incorrect, then the parameter values would change.
- The Transformation (Transfer) Function is any function suitable for the task at hand.
- The transfer function for ANNs is usually a nonlinear sigmoid function.
- Thus, if the normalized computed value is less than some value (say, 0.5) then the output value will be zero.
- If the computed value is at the cut-off threshold, then the output value will be 1.
- It could be a nonlinear hyperbolic function in which the output is either -1 or 1.
- Many other functions could be designed for any or all of the processing elements.
- Thus, in a neural network, every processing element can potentially have a different number of input values, a different set of weights for those inputs, and a different transformation function.
- Those values support and compensate for one another until the neural network as a whole learns to provide the correct output, as desired by the user.

ARCHITECTING A NEURAL NETWORK

- There are many ways to architect the functioning of an ANN using fairly simple and open rules with a tremendous amount of flexibility at each stage.
- The most popular architecture is a feed forward, multilayered perceptron with back-propagation learning algorithm.
- That means there are multiple layers of PEs in the system and the output of neurons are fed forward to the PBS in the next layers; and the feedback on the prediction is fed back into the neural network for learning to occur.

The ANN architectures for different applications are shown in Table 8.1.

Table 8.1 ANN Architecture for Different Applications

Classification	Feedforward networks (MLP), radial basis function and probabilistic
Regression	Feedforward networks (MLP), radial basis function
Clustering	Adaptive resonance theory (ART), Self-organizing maps (SOMs)
Association Rule Mining	Hopfield networks

DEVELOPING AN ANN

It takes resources, training data, skill and time to develop a neural network. Most data mining platforms offer at least the Multi-Layer Perceptron (MLP) algorithm to implement a neural network. Other neural network architectures include probabilistic networks and self-organizing feature maps.

The steps required to build an ANN are as follows

1. Gather data and divide into training data and test data. The training data needs to be further divided into training data and validation data.
2. Select the network architecture, such as Feed forward network.
3. Select the algorithm, such as Multi-Layer Perception.
4. Set network parameters.
5. Train the ANN with training data.
6. Validate the model with validation data.
7. Freeze the weights and other parameters.
8. Test the trained network with test data.
9. Deploy the ANN when it achieves good predictive accuracy.

Training an ANN requires the training data be split into three parts (Table 8.2)

Table 8.2 ANN Training Datasets

Training Set	**This dataset is used to adjust the weights on the neural network (~ 60%).**
Validation Set	This dataset is used to minimize overfitting and verifying accuracy (~ 20%).
Testing Set	**This dataset is used only for testing the final solution in order to confirm the actual predictive power of the network (~ 20%).**
***k*-fold Cross Validation**	This approach means that the data is divided into k equal pieces, and the learning process is repeated k times with each piece becoming the training set. This process leads to less bias and more accuracy, but is more time consuming.

ADVANTAGES AND DISADVANTAGES OF USING ANNs

There are many benefits of using ANN. Some are given below

1. ANNs impose very little restrictions on their use. ANN can deal with (identify/model) highly nonlinear relationships on their own, without much work from the user or analyst. They help find practical data-driven solutions where algorithmic solutions are nonexistent or are too complicated.
2. There is no need to program neural networks as they learn from examples. They get better with use, without much programming effort.
3. They can handle a variety of problem types, including classification, clustering, associations, etc.
4. ANNs are tolerant of data quality issues and they do not restrict the data to follow strict normality and or independence assumptions.
5. They can handle both numerical and categorical variables.
6. ANNs can be much faster than other techniques.
7. Most importantly, they usually provide better results (prediction and/or clustering) compared to statistical counterparts, once they have been trained enough.

The key disadvantages arise from the fact that they are not easy to interpret or explain or compute.

1. They are deemed to be black-box solutions, lacking explain ability. Thus they are difficult to communicate about, except through the strength of their results.
2. Optimal design of ANN is still an art. It requires expertise and extensive experimentation.
3. It could be difficult to handle a large number of variables (especially the rich nominal attributes).
4. It takes large datasets to train an ANN.

CLUSTER ANALYSIS

INTRODUCTION

Cluster analysis is used for automatic identification of natural grouping of things. It is also known as the segmentation technique.

- In this technique, data instances that are similar to (or near) each other are categorized into one cluster and data instances that are very different (or far away) from each other are moved into different clusters.
- Clustering is an unsupervised learning technique as there is no output or dependent variable for which a right or wrong answer can be computed.
- The correct number of clusters or the definition of those clusters is not known ahead of time.
- Clustering techniques can only suggest to the user how many clusters would make sense from the characteristics of the data.
- The user can specify a different, larger or smaller, number of desired clusters based on their making business sense.
- The cluster analysis technique will then define many distinct clusters from analysis of the data, with cluster definitions for each of those clusters.
- However, there are good cluster definitions, depending on how closely the cluster parameters fit the data.

APPLICATIONS OF CLUSTER ANALYSIS

- Cluster analysis is used in almost every field where there is a large variety of transactions.
- It helps provide characterization, definition, and labels for populations. It can help identify natural grouping of customers, products, patients, and so on.
- It can also help identify outliers in a specific domain and thus decrease the size and complexity of problems.
- A prominent business application of cluster analysis is in market research.
- Customers are segmented into clusters based on their characteristics-wants and needs, geography, price sensitivity, and so on.

Here are some examples of clustering

- **Market Segmentation** Categorizing customers according to their similarities, for instance by their common wants and needs, and propensity to pay can help With targeted marketing.
- **Product Portfolio** People of similar sizes can be grouped together to make small, medium and large sizes for clothing items.
- **Text Mining** Clustering can help organize a given collection of text documents according to their content similarities into clusters of related topics.

DEFINITION OF A CLUSTER

An Operational definition of a cluster is that, given a representation of n objects find K groups based on a measure of similarity, such that objects within the same group are alike but the objects in different groups are not alike.

- The notion of similarity can be interpreted in many ways.
- Clusters can differ in terms of their shape, size, and density.
- Clusters are patterns and there can be many kinds of patterns.
- Some clusters are the traditional types, such as data points hanging together.
- However, there are other clusters, such as all points representing the circumference of a circle.
- There may be concentric circles with points of different circles representing different clusters.
- The presence of noise in the data makes the detection of the clusters even more difficult.
- An ideal cluster can be defined as a set of points that is compact and isolated.
- In reality, a cluster is a subjective entity whose significance and interpretation requires domain knowledge.
- In the sample data below (Figure 9.1), how many clusters can one visualize?

FIGURE 9.1 Visual Cluster Example

- It seems like there are two clusters of approximately equal sizes.
- However, they can be seen as three clusters, depending on how we draw the dividing lines.
- There is not a truly optimal way to calculate it.
- Heuristics are often used to define the number of clusters.

REPRESENTING CLUSTERS

- The clusters can be represented by a central or modal value.
- A cluster can be defined as the centroid of the collection of points belonging to it.
- A centroid is 8 measure of central tendency.

- It is the point from where the sum total of squared distance from all the points is the minimum.
- A real-life equivalent would be the city center as the point that is considered the most easy to use by all constituents of the city.
- Thus all cities are defined by their centers or downtown areas.
- A cluster can also be represented by the most frequently occurring value in the cluster, i.e., a cluster can be defined by its modal value.
- Thus, a particular cluster representing a social point of view could be called the 'soccer moms', even though not all members of that cluster need currently be a mom with soccer-playing children.

CLUSTERING TECHNIQUES

The quality of a clustering result depends on the algorithm, the distance function, and the application.

- First, consider the distance function. Most cluster analysis methods use a distance measure to calculate the closeness between pairs of items.
- There are two major measures of distances: Euclidian distance ("as the crow flies" or straight line) is the most intuitive measure; the other popular measure is the Manhattan (rectilinear) distance, where one can go only on orthogonal directions.
- The Euclidian distance is the hypotenuse of a right triangle, while the Manhattan distance is the sum of the two legs of the right triangle.
- There are other measures of distance like Jacquard distance (to measure similarity of sets), or Edit distance (similarity of texts), and others.
- In either case, the key objective of the clustering algorithm is the same, i.e., inter-cluster distance is maximized and intra-clusters distance is minimized.
- There are many algorithms to produce clusters. There are top-down, hierarchical methods that start with creating a given number of best-fitting clusters. There are also bottom-up methods that begin with identifying naturally occurring clusters.
- The most popular clustering algorithm is the K-means algorithm. It is a top down, statistical technique, based on the method of minimizing the least squared distance from the center points of the clusters.
- Other techniques, such as neural networks, are also used for clustering.
- Comparing cluster algorithms is a difficult task as there is no single right number of clusters. However, the speed of the algorithm and its versatility in terms of different dataset are important criteria.

Here is the generic pseudo-code for clustering

1. Pick an arbitrary number of groups/segments to be created.
2. Start with some initial randomly chosen center values for groups.
3. Classify instances to closest groups.

4. Compute new values for the group centers.
5. Repeat steps 3 and 4 till groups converge.
6. If clusters are not satisfactory, go to step 1 and pick a different number 0f groups/segments.

CLUSTERING EXERCISE

Here is a simple exercise to visually and intuitively identify clusters from the data as shown in Dataset 9.1. X and Y are the two dimensions of interest. The objective is to determine the number of clusters and the center points of those clusters.

Dataset 9.1

X	Y
2	4
2	6
5	6
4	7
8	3
6	6
5	2
5	7
6	3
4	4

A scatter plot of 10 items in 2 dimensions shows them distributed fairly randomly. As a bottom-up technique, the number of cluster and their centroids can be intuited (Figure 9.2).

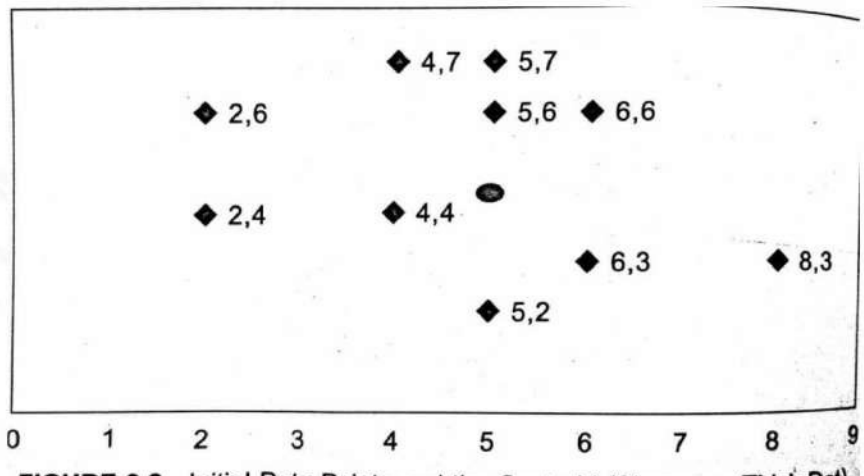


FIGURE 9.2 Initial Data Points and the Centroid (Shown as Thick Dot)

- The points are distributed randomly enough such that it is considered as one cluster.
- The solid circle represents the central point (centroid) of these points.
- However, there is a big distance between the points (2, 6) and (8, 3). So, this, data can be broken into 2 clusters.
- The 3 points at the bottom right can form one cluster and the other 7 forms the other cluster.
- The two clusters look like as follows (Figure 9.3). The two circles are the new centroids.

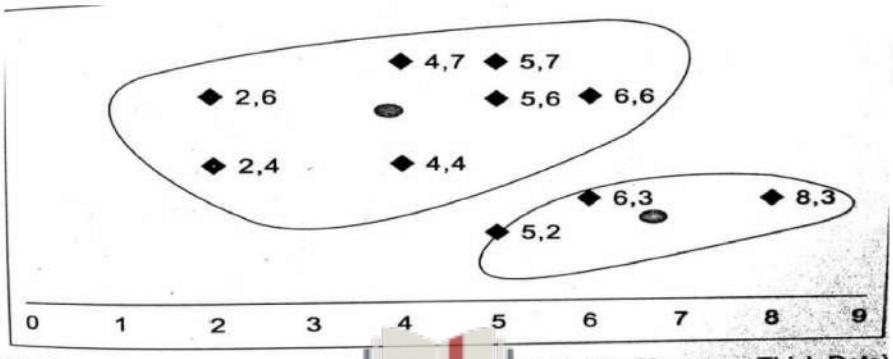


FIGURE 9.3 Dividing into Two Clusters (Centroids Shown as Thick Dots)

The bigger cluster seems too far apart. So, it seems like the 4 points on the top form a separate cluster. The three clusters look like as follows (Figure 9.4).

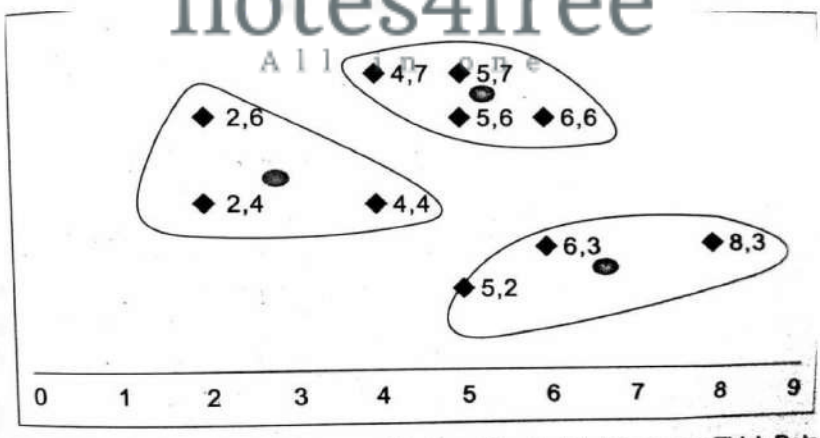


FIGURE 9.4 Dividing into Three Clusters (Centroids Shown as Thick Dots

- This solution has 3 clusters. The cluster on the right is far from the other 2 clusters.
- However, its centroid is not too close to all the data points.
- The cluster at the top looks very tight-fitting, with a nice centroid. The third cluster, at the left, is spread out and may not be of much usefulness.

- This was a bottom-up exercise in visually producing 3 best-fitting cluster definitions from the given data.
- The right number of clusters will depend on the data and the application for which the data would be used.

K-MEANS ALGORITHM FOR CLUSTERING

- K-means iteratively computes the clusters and their centroids.
- It is a top down approach to clustering.
- Starting with a given number of K clusters, say 3 clusters, that means 3 random centroids will be created as starting points of the centers.
- The circles are initial cluster centroids (Figure 9.5).

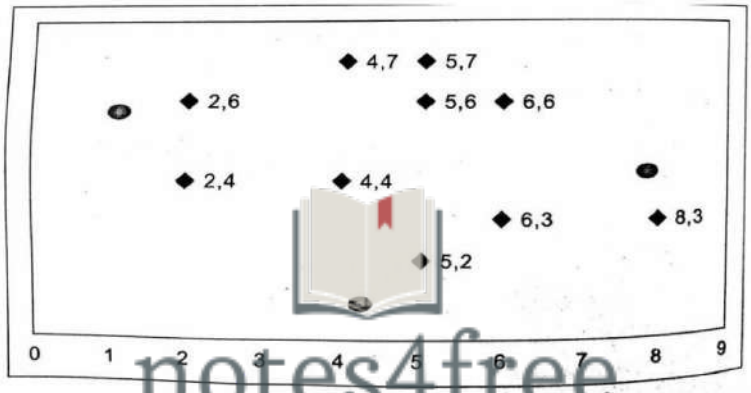


FIGURE 9.5 Randomly Assigning Three Centroids for Three Data Clusters

Step 1 For a data point, distance values will be from each of the three centroids. The data point will be assigned to the cluster with the shortest distance to the centroid. All data points will thus, be assigned to one data point or the other (Figure 9.6). The arrows from each data element show the centroid that the point is assigned to.

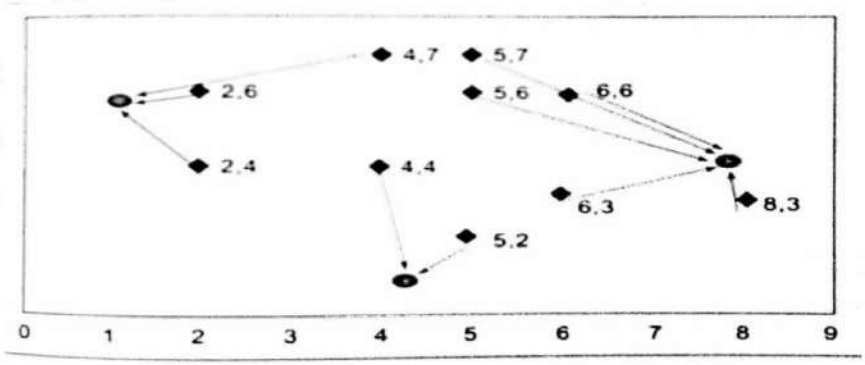


FIGURE 9.6 Assigning Data Points to Closest Centroid

Step 2 The centroid for each cluster will now be recalculated such that it is closest to all the data points allocated to that cluster the dashed arrows show the centroids being moved from their old (shaded) values to the revised new values (Figure 9.7)

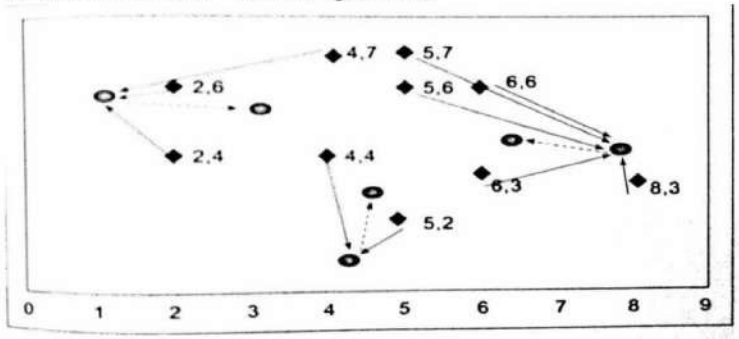


FIGURE 9.7 Recomputing Centroids for Each Cluster

Step 3 Once again, data points are assigned to the three centroids closest t it(Figure 9. 8) .

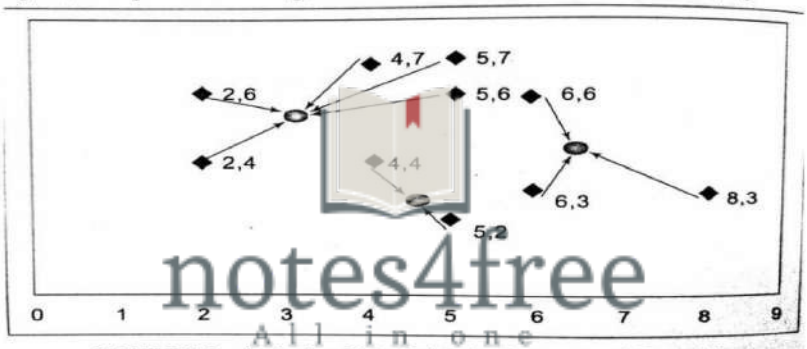


FIGURE 9.8 Assigning Data Points to Recomputed Centroids

The new centroids will be computed from the data points in the cluster until finally, the centroids stabilize in their locations. These are the three clusters computed by this algorithm.
The three clusters shown are a 3-datapoints cluster with centroid (6.5, 4.5), a 3-datapoint cluster with centroid (4.5, 3) and a 5-datapoint cluster with centroid (3.5, 3) (Figure 9.9).

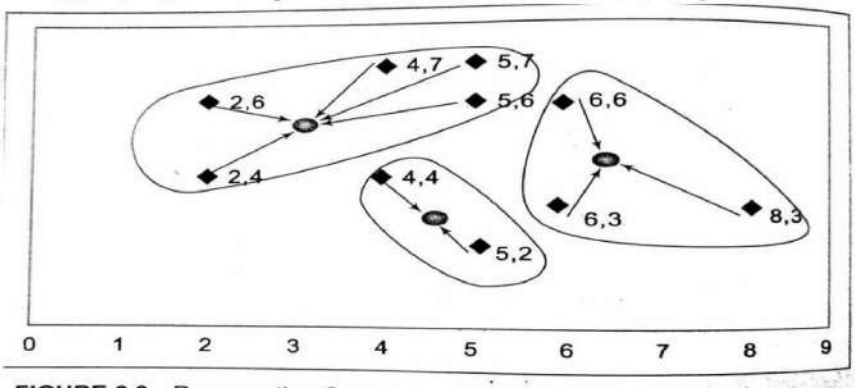


FIGURE 9.9 Recomputing Centroids for Each Cluster till Clusters Stabilize

Here is the pseudo code for implementing a K-means algorithm.

Algorithm K-Means (K number of clusters, D list of data points)

1. Choose K number of random data points as initial centroids (cluster-centers)
2. Repeat till cluster-centers stabilize
   (a) {Allocate each point in D to the nearest of K centroids;
   (b) Compute centroid for the cluster using all points in the cluster}

SELECTING THE NUMBER OF CLUSTERS

- The correct choice of the value of K is often ambiguous.
- It depends on the shape and scale of the distribution points in a dataset and the desired clustering resolution of the user.
- Heuristics are needed to pick the right number. One can graph the percentage of variance explained by the clusters against the number of clusters (Fig. 9.10).

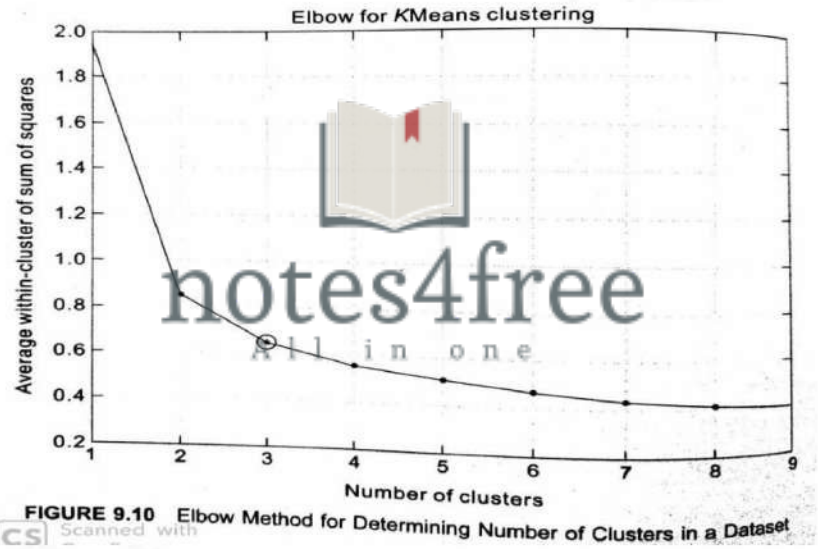


FIGURE 9.10 Elbow Method for Determining Number of Clusters in a Dataset

- The first cluster will add more information (explain a lot of variance), but at some point the marginal gain in variance will fall, giving A sharp angle to the graph, looking like an elbow.
- Beyond that elbow point, adding more clusters will not add much incremental value. That would be the desired K.
- To engage with the data and to understand the clusters better, it is often better to start with a small number of clusters such as 2 or 3, depending upon the dataset and the application domain.
- The number can be increased subsequently, as needed from an application point of view. This helps understand the data and the clusters progressively better.

ADVANTAGES AND DISADVANTAGES OF K-MEANS ALGORITHM

There are many advantages of the K-means algorithm.

1. K-means algorithm is simple, easy to understand and easy to implement.
2. It is also efficient, in that, the time taken to cluster K-means rises linearly with the number of data points.
3. No other clustering algorithm performs better than K-means, in general

There are a few disadvantages too

1. The user needs to specify an initial value of K.
2. The process of finding the clusters may not converge
3. It is not suitable for discovering cluster Shapes that are not hyper ellipsoids (or hyper spheres).

Association Rule Mining

INTRODUCTION

Associate Rule Mining is a popular, unsupervised learning technique, used in businesses to help identify shopping patterns. It is also known as Market Basket Analysis.

- It helps find interesting relationships (affinities) between variables (items or events). Thus, it can help cross-sell related items and increase the size of a sale.
- All data used in this technique is of categorical type.
- There is no dependent variable.
- This technique accepts the raw point-of-sale transaction data as input.
- The output produced is the description of the most frequent affinities among the items.

BUSINESS APPLICATIONS OF ASSOCIATION RULES

- In business environments, a pattern or knowledge can be used for many purposes.
- In sales and marketing, it is used for cross-marketing and cross-selling, catalog design, e-commerce site design, online advertising optimization, product pricing, and sales/promotion configurations.
- This analysis suggests not to put one item on sale at a time, and instead to create a bundle of products promoted as a package to sell other, non-selling items.
- In retail environments, it can be used for store design.
- Strongly associated items can be kept close together for customer convenience.
- Or they could be placed far from each other so that the customer has to walk the aisles and by doing so is potentially exposed to other items.
- In medicine, this technique can be used for relationships between symptoms and illnesses; diagnosis and patient characteristics/treatments; genes and their functions, etc.

REPRESENTING ASSOCIATION RULES

A generic association rule is represented between a set X and Y: X => Y [S%,C%] X, Y Products and/or services
X Left hand side (LHS)
Y Right hand side (RHS)
S Support how often X and Y go together in the dataset, i.e., P(X U Y)
C Confidence how often Y is found, given X, i.e., P(Y l X)
Example {Hotel booking, Flight booking} => {Rental Car} [30%, 60%]
[Note P(X) IS the mathematical representation of the probability or chance of X occurring in the dataset]

Computation Example

Suppose there are 1000 transactions in a dataset. There are 300 occurrences of X and 150 occurrences of (X, Y) in the dataset.
Support S for X => Y will be P(X u Y) = 150/1000 = 15%
Confidence for X = Y will be P(Y l X) or P(X u Y)/P(X) = 150/300 = 50%

ALGORITHMS FOR ASSOCIATION RULE

There are many algorithms that are available for generating association rules. The most popular algorithms are Apriori, Eclat, FP-Growth, along with various derivatives and hybrids of the three. All the algorithms help identify the frequent itemsets, which are then converted to association rules.

APRIORI ALGORITHM

- This is the most popular algorithm used for association rule mining.
- The objective is to find subsets that are common to at least a minimum number of the itemsets.
- A frequent itemset is the one whose support is greater than or equal to minimum support threshold.
- The Apriori property is a downward closure property, which means that any subset of a frequent itemset is also a frequent itemset.
- Thus, if (A, B, C, D) IS a frequent itemset, then any subset such as (A, B C) 0r (B, D) 1s also a frequent itemset.
- It uses a bottom-up approach and the size of frequent subsets is gradually increased, from 1-item subsets to 2-item subsets, then 3-item subsets, and so on.
- Groups of candidates at each level are tested against the data for minimum support.

ASSOCIATION RULES EXERCISE

Dataset 10.1 shows a dozen sales transactions. There are six products being sold-Milk, Bread, Butter, Eggs, Cookies, and Ketchup.

- Transaction#1 sold Milk, Eggs, Bread and Butter.
- Transaction#2 sold Milk, Butter, Eggs and Ketchup and so on.
- The objective is to use this transaction data to find affinities between products, i.e., which products sell together often.
- The support level will be set at 33 percent and the confidence level will be set at 50 percent.
- That means that we have decided to consider rules from only those itemsets that occur at least 33 percent of the time in the total set of transactions.
- Confidence level means that within those itemsets, the rules of the form X -> Y should be such that there is at least 50 percent chance of Y occurring based on X occurring.

Transactions List				
1	Milk	Egg	Bread	Butter
2	Milk	Butter	Egg	Ketchup
3	Bread	Butter	Ketchup	
4	Milk	Bread	Butter	
5	Bread	Butter	Cookies	
6	Milk	Bread	Butter	Cookies
7	Milk	Cookies		
8	Milk	Bread	Butter	
9	Bread	Butter	Egg	Cookies
10	Milk	Butter	Bread	
11	Milk	Bread	Butter	
12	Milk	Bread	Cookies	Ketchup

First step is to compute item itemsets, i.e., how often does any product sells individually.

1-itemSets	Frequency
Milk	9
Bread	10
Butter	10
Egg	3
Ketchup	3
Cookies	5

Thus, Milk sells in 9 out of 12 transactions, Bread sells in 10 out of 12 transactions, and so on.
At every point, there is an opportunity to select itemsets of interest, and thus further analysis. Other itemsets that occur infrequently may be removed. If itemsets that occur 4 or more times out of 12 are selected, that corresponds to meeting a minimum support level of 33 percent (4 out

of 12). Only 4 items make the cut. The frequent items that meet the support level of 33 percent are

Frequent 1-item Sets	Frequency
Milk	9
Bread	10
Butter	10
Cookies	5

The next step is to go for the next level of itemsets using items selected earlier, i.e., 2-item itemsets

2-item Sets	Frequency
Milk, Bread	7
Milk, Butter	7
Milk, Cookies	3
Bread, Butter	9
Butter, Cookies	3
Bread, Cookies	4

Thus the sale of (Milk, Bread) is 7 times out of 12, (Milk, Butter) is 7 times, (Bread, Butter) is 9 times, and (Bread, Cookies) is 4 times.

However, only four of these transactions meet the minimum support level of 33 percent.

2-item Sets	Frequency
Milk, Bread	7
Milk, Butter	7
Bread, Butter	9
Bread, Cookies	4

The next step is to list the next higher level of itemsets, i.e., 3 items item itemsets.

3-item Sets	Frequency
Milk, Bread, Butter	6
Milk, Bread, Cookies	1
Bread, Butter, Cookies	3

Thus the sale of(Milk, Bread, Butter)is 6 times out of 12 and (Bread, Butter, Cookies) is 3 times out of 12 .One 3 item itemsets meets the minimum support requirements.

3-item Sets	Frequency
Milk, Bread, Butter	6

There is no room to create a 4 item itemset for this support level.

CREATING ASSOCIATION RULES

- The most interesting and complex rules at higher size itemsets start top-down with the most frequent itemsets of higher size-numbers.
- Association rules are created that meet the support level (>33 percent) and confidence levels (> 50 percent).
- The highest level itemset that meets the support requirements is the 3-item itemset. The following itemset has a support level of 50 percent (6 out of 12).

Milk, Bread, Butter	6

- This itemset could lead to multiple candidates association rules.

Start with the following rule

- (Bread, Butter) -> Milk
- There are a total of 12 transactions.
- X (in this case Bread, Butter) occurs 9 times; X, Y (in this case Bread, Butter, Milk) occurs 6 times.
- The support level for this rule is 6/12 == 50 percent.
- The confidence level for this rule is 6/9 = 67 percent. This rule meets our thresholds for support (>33 percent) and confidence (>50 percent).
- Thus, the first valid association rule from this data is (Bread, Butter) --> Milk {S =.50%, C = 67%}.
- In exactly the same way, other rules can be considered for their validity.
- Consider the rule (Milk, Bread) --> Butter. Out of total 12 transactions, (Milk, Bread) occurs 7 times and (Milk, Bread, Butter) occurs 6 times.
- The support level for this rule is 6/12 = 50 percent.
- The confidence level for this rule is 6/7 = 86 percent. This rule meets our thresholds for support (>33 percent) and confidence (>50 percent).
- Thus, the second valid association rule from this data is (Milk, Bread) -> Butter {S = 50%, C = 67%}.
- Consider the rule (Milk, Butter) --> Bread. Out of total 12 transactions, (Milk, Butter) occurs 7 times, while (Milk, Butter, Bread) occur 6 times.

- The support level for this rule is 6/ 12 = 50 percent. The confidence level for this rule is 6/ 7 = 86 percent. This rule meets our thresholds for support (>33 percent) and confidence (>50 percent).
- Thus, the next valid association rule is Milk, Butter -> Bread{S = 50%, C = 86%}.
- Thus, there. were only three possible rules at the 3-item itemset level and all were found to be valid.
- One can get to the next lower level and generate association rules at the 2-item itemset level.
- Consider the following rule
- Milk --> Bread; out of total 12 transactions, Milk occurs 9 times while (Milk, Bread) occurs 7 times.
- The support level for this rule is 7/12 = 58 percent. The confidence level for this rule is 7/9 = 78 percent. This rule meets our thresholds for support (>33 percent), and confidence (>50 percent)
- Thus, the next valid association rule is Milk --> Bread{58%, 78%}
- Many such rules could be derived if needed.

Reference text book:

Anil Maheshwari, "Data Analytics", 1st Edition, McGraw Hill Education, 2017.
ISBN-13: 978-9352604180

Module V
Text Mining, Naïve-Bayes Analysis, Support Vector Machines, Web Mining, Social Network Analysis

Text Mining

INTRODUCTION

Text Mining is the art and science of discovering knowledge, insights and patterns from an organized collection of textual databases.

- Textual mining can help with frequency analysis of important terms and their semantic relationships.
- Text mining can be applied to large scale social media data for gathering preferences, and measuring emotional sentiments.
- It can also be applied to societal, organizational and individual scales.

TEXT MINING APPLICATIONS

- Text mining is a useful tool in the hands of chief knowledge officers to extract knowledge relevant to an organization.
- Text mining can be used across industry sectors and application areas, including decision support, sentiment analysis, fraud detection, survey analysis, and many more.

1. **Marketing** The voice of the customer can be captured in its native and raw format and then analyzed for customer preferences and complaints.
   a. Social personas are a clustering technique to develop customer segments of interest. Consumer input from social media sources, such as reviews, blogs, and tweets, contain numerous leading indicators that can be used toward5 anticipating and predicting consumer behavior.
   b. A listening platform is a text mining application that in real time, gathers social media, blogs, and other textual feedback, and filters out the chatter to extract true consumer sentiments. The insights can lead to more effective product marketing and better customer service.
   c. The customer call center conversations and records can be analyzed for patterns of customer complaints. Decision trees can organize this data to create decision choices that could help with product management activities and to become proactive in avoiding those complaints.
2. **Business Operation** Many aspects of business functioning can be accurately gauged from analyzing text.

a. Social network analysis and text mining can be applied to emails, blogs, social media and other data to measure the emotional states and the mood of employee populations. Sentiment analysis can reveal early signs of employee dissatisfaction which can then be proactively managed.
b. Studying people as emotional investors and using text analysis of the social Internet to measure mass psychology can help in obtaining superior investment returns.

3. **Legal** In legal applications, lawyers and paralegals can more easily search case histories and laws for relevant documents in a particular case to improve their chances of winning.
   a. Text mining is also embedded in e-discovery platforms that help in minimizing risk in the process of sharing legally mandated documents.
   b. Case histories, testimonies, and client meeting notes can reveal additional information, such as morbidities in healthcare situations that can help better predict high-cost injuries and prevent costs.
4. **Governance and Politics** Government can be overturned based on a tweet originating from a self-immolating fruit vendor in Tunisia.
   a. Social network analysis and text mining of large-scale social media data can be used for measuring the emotional states and the mood of constituent populations. Micro targeting constituents with specific messages gleaned from social media analysis can be a more efficient use of resources when fighting democratic elections.
   b. In geopolitical security, internet chatter can be processed for real-time information and to connect the dots on any emerging threats.
   c. In academics, research streams could be meta-analyzed for underlying research trends.

TEXT MINING PROCESS

As the amount of social media and other text data grows, there is a need for efficient abstraction and categorization of meaningful information from the text.

1. The first level of analysis is identifying frequent words. This creates a bag of important words. Texts documents or smaller messages can then be ranked on how they match to a particular bag-of-words. However, there are challenges with this approach. For example, the words may be spelled a little differently or there may be different words with similar meanings.
2. The next level is identifying meaningful phrases from words. Thus 'ice' and 'cream' will be two different key words that often come together. However, there is a more meaningful phrase by combining the two words into 'ice cream'. There might be similarly meaningful phrases like 'Apple Pie'.
3. The next higher level is that of Topics. Multiple phrases can be combined into Topic area. Thus the two phrases above can be put into a common basket, and this bucket is called 'Desserts'.

Text mining is a semi-automated process. Text data needs to be gathered, structured, and then mined, in a 3 step process (Figure 11.1)

1. The text and documents are first gathered into a corpus and organized
2. The corpus is then analyzed for structure. The result is a matrix mapping important terms to source documents.
3. The structured data is then analyzed for word structures, sequences, and frequency.

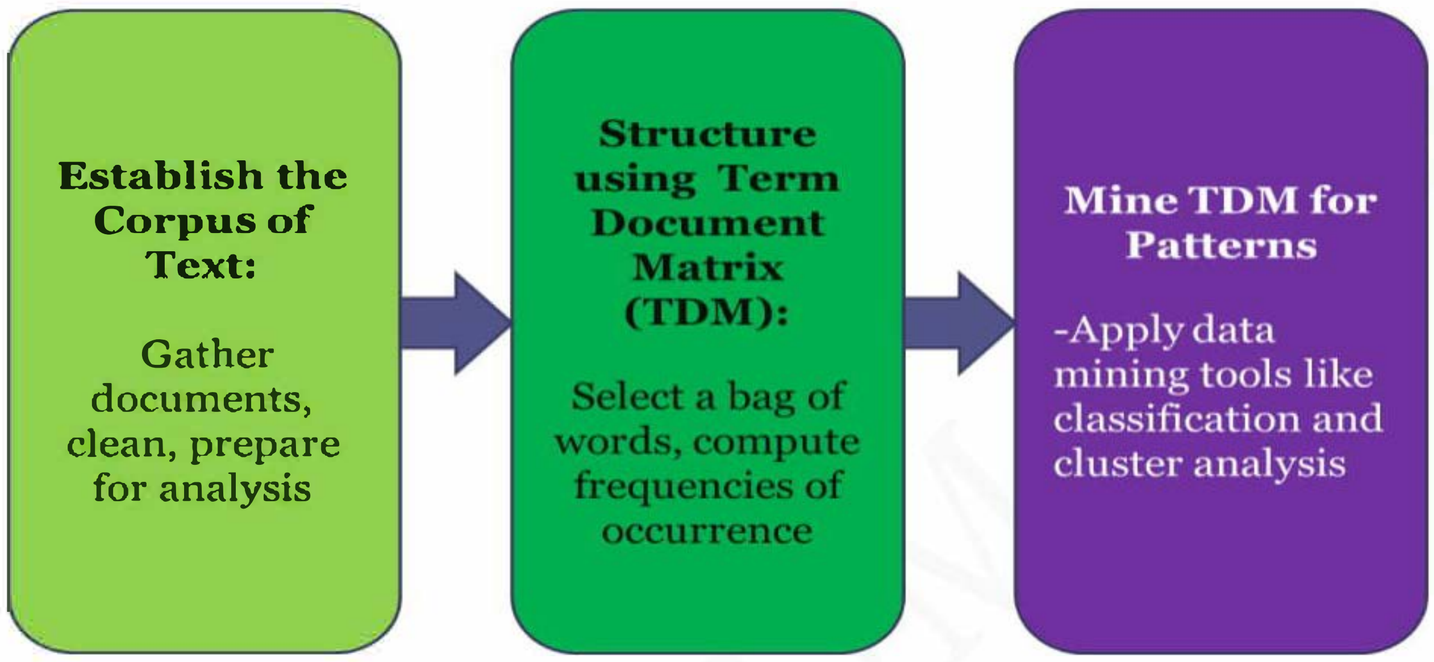


FIGURE 11.1 Text Mining Architecture

TERM DOCUMENT MATRIX

This is the heart of the structuring process. Free flowing text can be transformed into numeric data in a TDM, which can then be mined using regular data mining techniques.

	Term Document Matrix				
Document / Terms	investment	Profit	happy	Success	…
Doc 1	10	4	3	4	
Doc 2	7	2	2		
Doc 3			2	6	
Doc 4	1	5	3		
Doc 5		6		2	
Doc 6	4		2		
…					

- There are several efficient techniques for identifying key terms from a text.
- There are less efficient techniques available for creating topics out of them.

- This approach measures the frequencies of select important terms occurring in each document.
- This creates a t x d Term-by-Document Matrix (TDM), where t is the number of terms and d is the number of documents (Table 11.1).
- Creating a TDM requires making choices of which terms to include.
- The terms chosen should reflect the stated purpose of the text mining exercise.
- The list of terms should be as extensive as needed, but should not include unnecessary stuff that will serve to confuse the analysis or slow the computation.

Here are some considerations in creating a TDM

- A large collection of documents mapped to a large bag of words will likely lead to a very sparse matrix if they have few common words.
- Reducing dimensionality of data will help improve the speed of analysis and meaningfulness of the results.
- Synonyms or terms with similar meanings should be combined and should be counted together as a common term. This would help reduce the number of distinct terms of words or 'tokens'.
- Data should be cleaned for spelling errors. Common spelling errors should be ignored and the terms should be combined. Uppercase-lowercase terms should also be combined.
- When many variants of the same term are used, just the stem of the word would be used to reduce the number of terms. For instance, terms like customer order, ordering, order data, should be combined into a single token Word, called 'order'.
- On the other side, homonyms (terms with the same spelling but different meanings) should be counted separately. This would enhance the quality of analysis.

For example, the term order can mean a customer order or the ranking of certain choices. These two should be treated separately "The boss ordered that the customer orders data analysis be presented in chronological order". This statement shows three different meanings for the word 'order thus, there will be a need for manual review of the TD matrix

- Terms with very few occurrences in the documents should be eliminated from the matrix. This would help increase the density of the matrix and the quality of analysis.
- The measures in each cell of the matrix could be one of the several possibilities. It could be a simple count of the number of occurrences of each term in a document. It could also be the log of that number. It could be the fraction number computed by dividing the frequency count by the total number of words in the document. Or there may be binary values in the matrix to represent whether a term is mentioned or not.
- The choice of value in the cells will depend upon the purpose of the text analysis.

- At the end of this analysis and cleansing, a well formed, densely populated, rectangular, TDM will be ready for analysis. The TDM can be mined using all the available data mining techniques.

MINING THE TDM

- The TDM can be mined to extract patterns/knowledge. A variety of techniques could be applied to the TDM to extract new knowledge.
- A simple application is to Visualize the highest frequency terms. This can be done very attractively and colorfully in the form of a 'word-cloud'.
- The word-cloud can be created after removing the common words like prepositions.
- It can be done for the top n words such as top 100 words, to focus on the key terms used in the document. The attached word-cloud represents the speech by US President Barack Obama on the topic of terrorism.

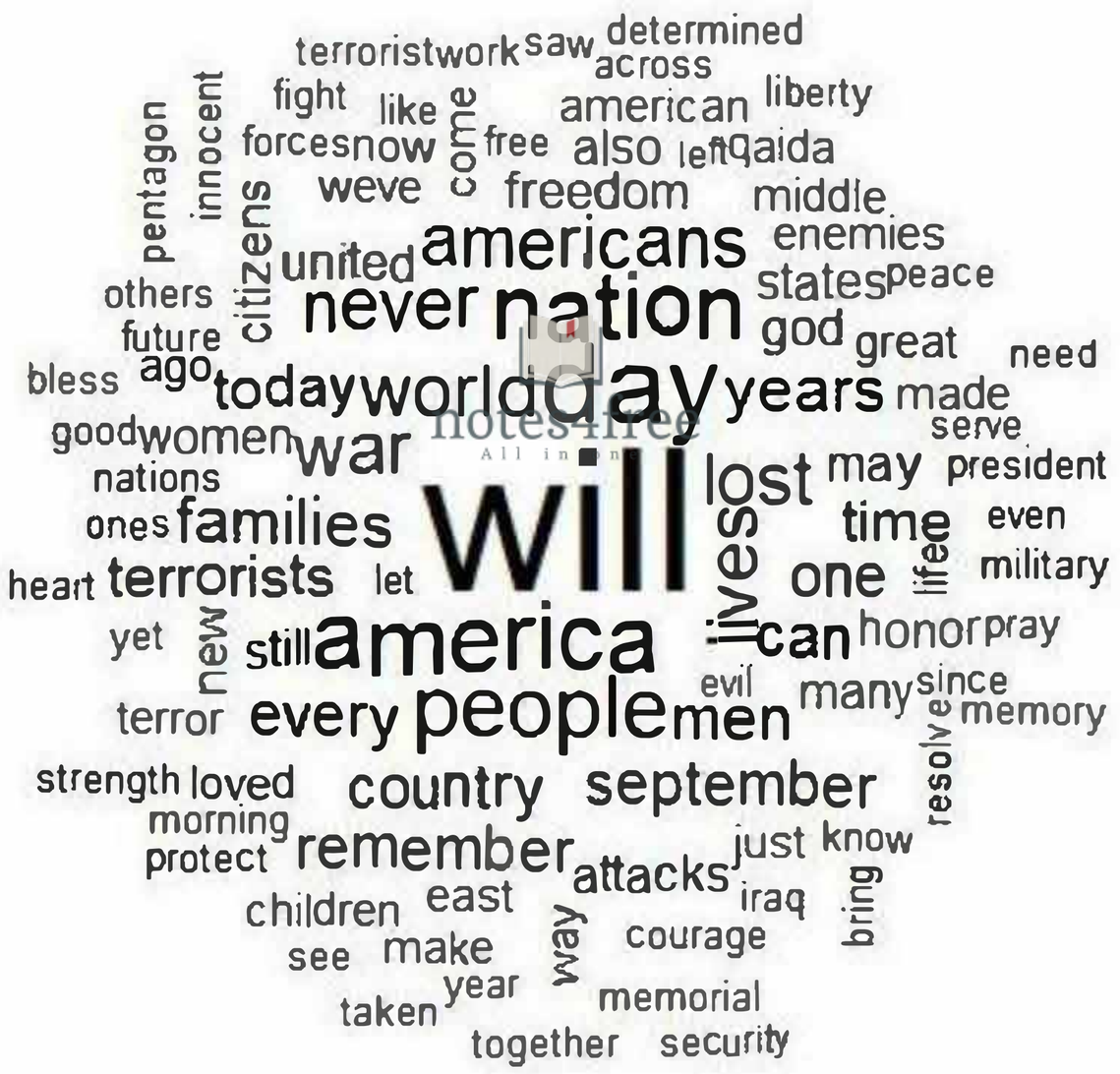


FIGURE 11.2 Word-cloud of Top 100 Words from US President's Speech

COMPARING TEXT MINING AND DATA MINING

- Text Mining is a form of data mining. There are many common elements between Text and Data Mining.
- However, there are some differences (Table 11.2). The key difference is that text mining requires conversion of text data into frequency data, before data mining techniques can be applied.

Table 11.2 Comparing Text Mining and Data Mining

Dimension	Text Mining	Data Mining
Nature of Data	Unstructured data: words, phrases, sentences	Numbers, alphabetical and logical values
Language Used	Many languages and dialects used in the world; many languages are extinct, new documents are discovered	Similar numerical systems across the world
Clarity and Precision	Sentences can be ambiguous; sentiment may contradict the words	Numbers are precise
Consistency	Different parts of the text can contradict each other	Different parts of data can be inconsistent, thus requiring statistical significance analysis
Sentiment	Text may present a clear and consistent or mixed sentiment, across a continuum. Spoken words add further sentiment	Not applicable
Quality	Spelling errors. Differing values of proper nouns such as names. Varying quality of language translation	Issues with missing values, outliers, etc.
Nature of Analysis	Keyword based search; co-existence of themes; sentiment mining;	A full wide range of statistical and machine learning analysis for relationships and differences

TEXT MINING BEST PRACTICES

Many of the best practices that apply to the use of data mining techniques will also apply to text mining.

- The first and most important practice is to ask the right question. A good question is the one which gives an answer and would lead to large payoffs for the organization. The purpose and the key question will define how and at what levels of granularity the TDM would be made.
  - **For example**, TDM defined for simpler searches would be different from those used for complex semantic analysis or network analysis.
- A second important practice is to be creative and open in proposing imaginative hypotheses for the solution. Thinking outside the box is important, both in the quality of the proposed solution as well as in finding the high quality datasets required to test the hypothesized solution.
  - **For example**, a TDM of consumer sentiment data should be combined with customer order data in order to develop a comprehensive view of customer behavior. It's important to assemble a team that has a healthy mix of technical and business skills.

- Another important element is to pursue the problem iteratively. Too much data can overwhelm the infrastructure and also befuddle the mind. It is better to divide and conquer the problem with a simpler TDM, with fewer terms and fewer documents and data sources. Expand as needed, in an iterative sequence of steps. In the future, add new terms to help improve predictive accuracy.
- A variety of data mining tools should be used to test the relationships in the TDM. Different decision tree algorithms could be run alongside cluster analysis and other techniques. Triangulating the findings with multiple techniques, and many what-if scenarios helps build confidence in the solution. Test the solution in many ways before committing to deploy it.

NAIVE-BAYES ANALYSIS

INTRODUCTION

- Naive-Bayes (NB) technique is a supervised learning technique that uses probability-theory-based analysis.
- It is a machine-learning technique that computes the probabilities of an instance belonging to each one of many target classes, given the prior probabilities of classification using individual factors.
- Naive-Bayes technique is used often in classifying text documents into one of multiple predefined categories.

PROBABILITY

- Probability is defined as the chance of something happening.
- The probability Values thus range from zero to one; with a value of zero representing no chance, and to one representing total certainty.
- Using past event records, the probability of something happening in the future can be reliably assessed.

For example, one can assess the probability of dying from an airline accident, by dividing the total number of airline accident related deaths in a time period by the total number of People flying during that period. These probabilities can then be compared to come to the conclusions, such as the safety levels of various event types.

For example, past data may show that the probability of dying from airline accident is less than that of dying from being hit by lightning.

- The Naive-Bayes algorithm is special in that it takes into consideration the prior probability of an instance belonging to a class, in addition to the recent track record of the instance belonging to that class.

- The word Bayes refers to Bayesian analysis (based on the work of mathematician Thomas Bayes) which computes the probability of a new occurrence not only on the recent record, but also on the basis of prior experience.
- The word Naive represents the strong assumption that all parameters/features of an instance are independent variables with little or no correlation Thus if people are identified by their height, weight, age, and gender; all these variables are assumed to be independent of each other.
- NB algorithm is easy to understand and works fast. It also performs well in multiclass prediction, such as when the target class has multiple options beyond binary yes/no classification.
- NB can perform well even in case of categorical input variables compared to numerical variable(s).

NAIVE-BAYES MODEL

- In the abstract, Naive-Bayes is a conditional probability model for classification purposes.
- The goal is to find a way to predict the class variable (Y) using a vector of independent variables (X), i.e., finding the function

$$f: \mathbf{X} \rightarrow \mathbf{Y}$$

- In probability terms, the goal is to find $P(Y \mid X)$, i.e., the probability of Y belonging to a certain class X. Y is generally assumed to be a categorical variable with two or more discrete values.
- Given an instance to be classified, represented by a vector $x = (x_1, \ldots, x_n)$ representing 'n' features (independent variables), the Naive-Bayes model assigns, to an instance, probabilities of belonging to any of the K classes. The class K with the highest posterior probability is the label assigned to the instance.
- The posterior probability (of belonging to a Class K) is calculated as a function of prior probabilities and current likelihood value, as shown in the equation below

$$p(C_k \mid x) = \frac{p(C_k)\, p(x \mid C_k)}{p(x)}$$

- $P(C_k \mid x)$ is the posterior probability of class K, given predictor X.
- $P(C_k)$ is the prior probability of class K.
- P(x) is the prior probability of predictor.
- $P(x \mid C_k)$ is the current likelihood of predictor given class.

SIMPLE CLASSIFICATION EXAMPLE

Suppose a salon needs to predict the service required by the incoming customer If there are only two services offered -Hair cut (R) and Manicure-Pedicure (M) then the value to be predicted is whether the next customer will be for R or M. The number of classes (K) is 2.

- The first step is to compute prior probability. Suppose the data gathered for the last one year showed that during that period there were 2500 customers for R and 1500 customers for M.
- Thus, the default (or prior) probability for the next customer to be for R is 2500/4000 or 5/8.
- Similarly, the default probability for the next customer to be for M is 1500/4000 or 3/8.
- Based on this information alone, the next customer would likely be for R.

Another way to predict the service requirement by the next customer is to look at the most recent data. One can look at the last few (choose a number) customers, to predict the next customer. Suppose the last five customers were for the services R, M, R, M, M order.

- Thus, the data shows the recent probability of R is 2/5 and that of M is 3/5.
- Based on just this information, the next customer will likely to be for M.
- Thus in this example, the NB posterior probability P(R) is (5/8 x 2/5) = 10/40.
- Similarly, the NB probability P (M) is (3/8 x 3/5) = 9/40.
- Since P(R) is greater than P (M), it follows that there is a greater probability of the next customer to be for R. Thus the expected class label assigned to the next customer would be R.
- Suppose, however the next customer coming in was for M service. The last five customer sequence now becomes M, R, M, M, M.
- Thus, the recent data shows the probability for R to be 1/5 and that of M to be 4/5.
- Now the N B probability for R is (5/8 x 1/5) = 5/40.
- Similarly, the NB probability for M is (3/8 x 4/5) = 12/40.

Since P (M) is greater than P(R), it follows that there is a greater probability of the next customer to be for M. Thus the expected class label assigned to the next customer is M.

The NB predictor thus dynamically changes its prediction value based on the recent data.

TEXT CLASSIFICATION EXAMPLE

The probability of the document 'd' being in class 'c' is computed as follows

$$P(c \mid d) \propto P(c) \prod_{1 \le k \le n_d} P(t_k \mid c)$$

Where, P (t_k | c) is the conditional probability of term tk occurring in a document of class c.
Dataset 12.1 shows the text classification training and test data. The goal is to classify the test data into the right class as h or ~h (read as not h).

Dataset 12.1

Training Set	Document ID	Keywords in the Document	Class = *h* (Healthy)
	1	Love Happy Joy Joy Love	Yes
	2	Happy Love Kick Joy Happy	Yes
	3	Love Move Joy Good	Yes
	4	Love Happy Joy Pain Love	Yes
	5	Joy Love Pain Kick Pain	No
	6	Pain Pain Love Kick	No
Test Data	7	Love Pain Joy Love Kick	?

- The prior probabilities of a document being classified using the six documents are

$$P(h) = 4/6 = 2/3$$

$$P(\sim h) = 2/6 = 1/3$$

- i.e., there is 2/3 prior probability that a document will be classified as h and 1/3 probability of not h.
- The conditional probability for each term is the relative frequency of the term occurring in each class of the documents 'h class' and 'not h class'.

Conditional Probabilities	
Class *h*	Class ~*h*
$P(\text{Love} \mid h) = 5/19$	$P(\text{Love} \mid \sim h) = 2/9$
$P(\text{Pain} \mid h) = 1/19$	$P(\text{Pain} \mid \sim h) = 4/9$
$P(\text{Joy} \mid h) = 5/19$	$P(\text{Joy} \mid \sim h) = 1/9$
$P(\text{Kick} \mid h) = 1/19$	$P(\text{Kick} \mid \sim h) = 2/9$

The probability of the test instance belonging to class h can be computed by multiplying the prior probability of the instance belonging to class h, with the conditional probabilities for each of the terms in the document for class h. Thus,

$$P(h \mid d7) = P(h) * (P(\text{Love} \mid h))^2 * P(\text{Pain} \mid h) * P(\text{Joy} \mid h) * P(\text{Kick} \mid h)$$
$$= (2/3) * (5/19) * (5/19) * (1/19) * (5/19) * (1/19) = \sim 0.0000067$$

The NB probability of the test instance being 'not h' is much higher than its being h. Thus the test document will be classified as 'not h'.

$$P(\sim h \mid d7) = P(\sim h) * P(\text{Love} \mid \sim h) * P(\text{Love} \mid \sim h) * P(\text{Pain} \mid \sim h) * P(\text{Joy} \mid \sim h) * P(\text{Kick} \mid \sim h)$$
$$= (1/3) * (2/9) * (2/9) * (4/9) * (1/9) * (2/9) = 0.00018$$

ADVANTAGES AND DISADVANTAGES OF NAIVE-BAYES

1. The NB logic is simple and so is the NB posterior probability computation.
2. Conditional probabilities can be computed for discrete data and for probabilistic distributions. When there are number of variables in the vector X. then the problem can be modeled using probability functions to simulate the incoming values. A variety of methods exist for modeling the conditional distributions of the X variables, including normal, lognormal, gamma, and Poisson.
3. Naive-Bayes assumes that all the features are independent for most instances that work fine. However, it can be a limitation. If there are no joint occurrences at all of a class label with a certain attribute, then the frequency-based conditional probability will be zero. When all the probabilities are multiplied, it will make the entire posterior probability estimate to be zero. This can be rectified by adding 1 to all the numerators and adding n, the number of variables in X, to all the denominators. This will make those probabilities very small but not zero.
4. A limitation of NB is that the posterior probability computations are good for comparison and classification of the instances. However, the probability values themselves are not good estimates of the event happening.

Support Vector Machines

INTRODUCTION

- Support Vector Machine (SVM) is a mathematically rigorous, machine learning technique to build a linear binary classifier.
- It creates a hyperplane in a high-dimensional space that can accurately slice a dataset into two segments according to the desired objective.
- The algorithms for developing the classifier can be mathematically challenging though.
- SVMs are popular since they are state-of-the-art for many practical problems, such as identifying spam emails and other text mining applications.

SVM MODEL

- An SVM is a classifier function in a high dimensional space that defines the decision boundary between two classes.
- The support vectors are the data points that define the 'gutters' or the boundary condition on either side of the hyperplane, for each of the two classes.

- The SVM model is thus conceptually easy to understand.
- Suppose there is a labeled set of points classified into two classes. The goal is to find the best classifier between the points of the two types.

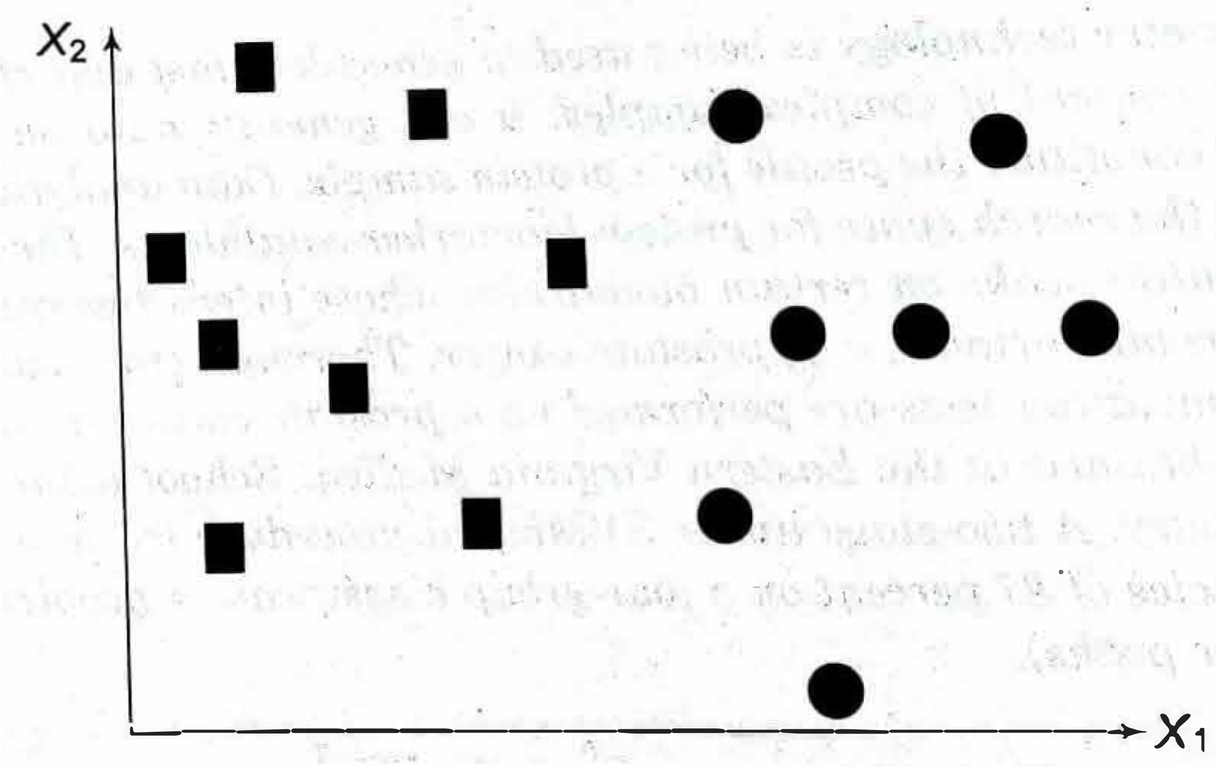


FIGURE 13.1 Data Points for Classification

SVM takes the widest street (a vector) approach to demarcate the two classes and thus finds the hyperplane that has the widest margin, i.e., largest distance to the nearest training data points of either class (Figure 13.2).

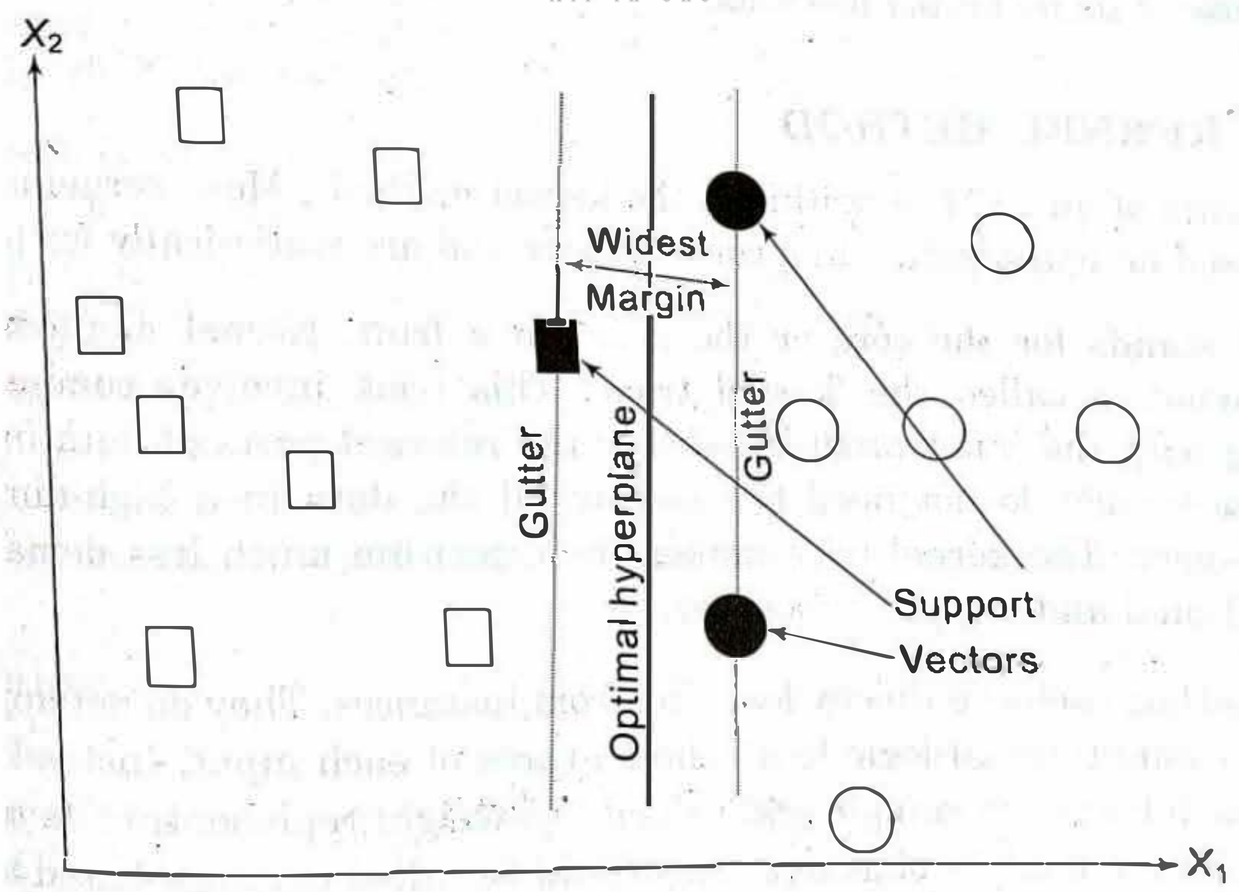


FIGURE 13.2 Support Vector Machine. Classifier

- In Figure 13.2, the hard line is the optimal hyperplane. The dotted lines are the gutters on the sides of the two classes.
- The gap between the gutters is the maximum or widest margin. The classifier (hyperplane) is defined by only those points that fall on the gutters on both sides.

- These points are called the support vectors (shown in their bold shape). The rest of the data points in their class are irrelevant for defining the classifier (shown unfilled).
- Abstractly, suppose that the training data of n points is

$$(X_1, y_1), \ldots\ldots, (X_i, y_i)$$

Where X_i represents the p-value vector for point i and y_i is its binary class value of 1 or -1. Thus there are two classes represented as 1 and -1.

Assuming that the data is indeed linearly separable, the classifier hyperplane is defined as a set of points (which is a subset of the training data) that satisfy the equation

$$W \cdot X + b = 0$$

Where W is the normal vector to the hyperplane.

The hard margins can be defined by the following hyperplanes

$$W.X+b=1 \text{ and } W.X+b= -1$$

The width of the hard margin is $(2/|W|)$.

- For all points not on the hyperplane, they will be safely in their own class.
- Thus, the y values will have to be either greater than 1 (for point in class 1) or less than -1 (for points in class -1).
- The SVM algorithm finds the weights vector (W) for the features, such that there is a widest margin between the two categories.
- Computing an SVM using these equations is a hill-climbing process problem in a convex space.
- However, by working with points nearest to the classifying boundary only, it reduces sharply the number of data instances to work with this approach reduces its memory requirements for computation.
- This is possible because of using kernel methods.

THE KERNEL METHOD

- The heart of an SVM algorithm is the kernel method. Most kernel algorithms are based on optimization in a convex space and are statistically well-founded.
- Kernel stands for the core or the germ in a fruit. Kernel methods operate using what is called the *'kernel trick'*.
- This trick involves computing and working with the inner products of only the relevant pairs of data in the feature space; they do not need to compute all the data in a high-dimensional feature space.
- The kernel trick makes the algorithm much less demanding in computational and memory resources.
- Kernel methods achieve this by learning from instances. So it is called instance-based learning.

- There are several types of support vector models including linear, polynomial, RBF, and sigmoid.
- SVMs have evolved to be more flexible and be able to tolerate some amount of misclassification the margin of separation between the categories is thus a *'soft margin'* as against a *hard margin*.

ADVANTAGES AND DISADVANTAGES OF SVMs

1. The main strength of SVMs is that they work well even when the number of features is much larger than the number of instances. It can work on datasets with huge feature space; such is the case in spam filtering, where a large number of words are the potential signifiers of a message being spam.
2. Another advantage of SVMs is that even when the optimal decision boundary is a nonlinear curve, the SVM transforms the variables to create new dimensions such that the representation of the classifier is a linear function of those transformed dimensions of the data.
3. SVMs are conceptually easy to understand. They create an easy-to-understand linear classifier. By working on only a subset of relevant data, they are computationally efficient. SVMs are now available with almost all data analytics toolsets.
4. The SVM technique has two major constraints
   a. It works well only with real numbers, i.e. all the data points in all the dimensions must be defined by numeric values only
   b. It works only with binary classification problems. One can make a series of cascaded SVMs to get around this constraint.
5. Training the SVMs is an inefficient and time consuming process, when the data is large. It does not work well when there is much noise in the data, and thus has to compute soft margins. The SVMs will also not provide a probability estimate of classification, i.e., the confidence level for classifying an instance.

WEB MINING

INTRODUCTION

Web Mining is the art and science of discovering patterns and insights from the World Wide Web so as to improve it.

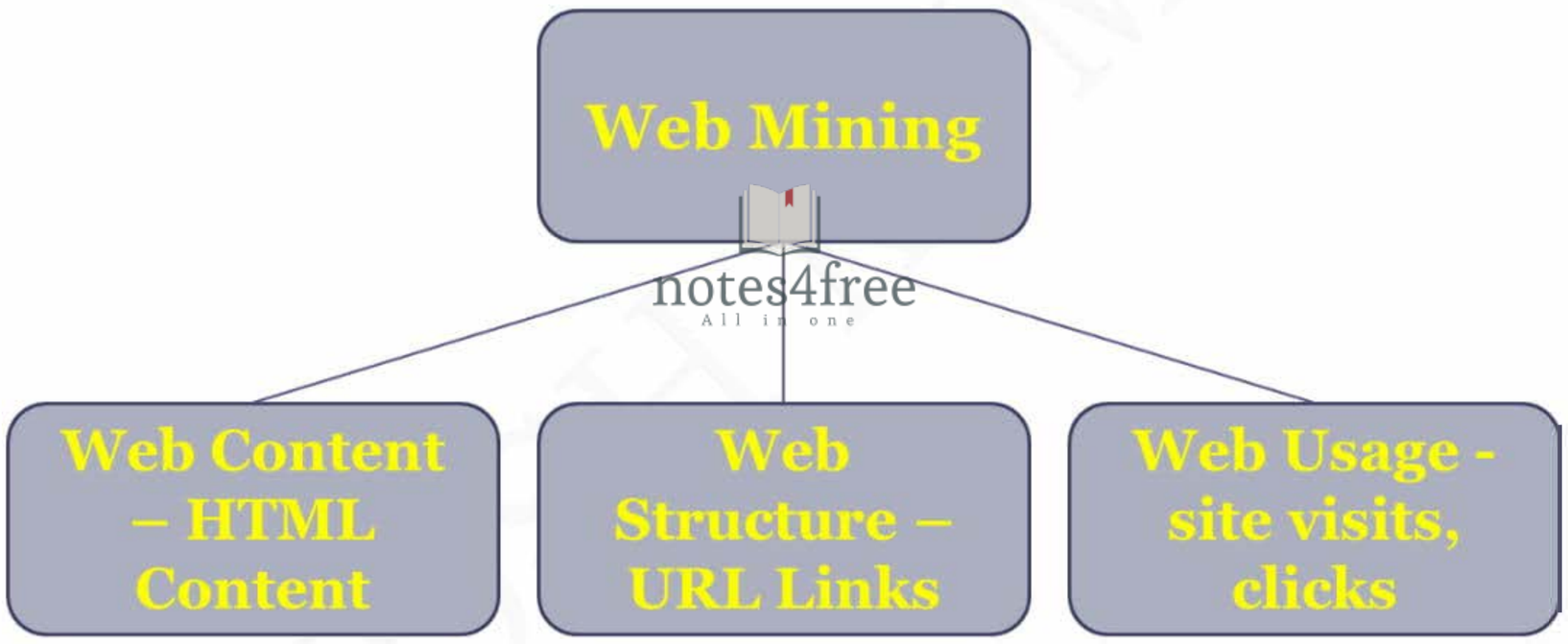
- Web mining analyzes data from the web and helps find insights that could optimize the web content and improve the user experience.
- Data for web mining is collected via web crawlers, web logs, and other means.

Here are some characteristics of optimized websites

1. **Appearance** Aesthetic design, well-formatted content, easy to scan and navigate, good color contrasts
2. **Content** well-planned information architecture with useful content, Fresh content, search engine optimized, links to other good sites
3. **Functionality** accessible to all authorized users, fast loading times, usable forms, mobile enabled.

- The analysis of web usage provides feedback on the web content and also the consumer's browsing habits.
- This data can be of immense use for commercial advertising and even for social engineering.
- The web could be analyzed for its structure as well as content.
- The usage pattern of web pages could also be analyzed.
- Depending upon objectives, web mining can be divided into three different types-web usage mining.

Web content mining and web structure mining (Figure 14.1)



WEB CONTENT MINING

- A website is designed in the form of pages with a distinct URL (Universal Resource Locator).
- A large website may contain thousands of pages.
- These pages (and their content) are managed using specialized software systems called Content Management Systems.
- Every page can have text, graphics, audio, video, forms, applications, and more kinds of content including user generated content.
- The websites keep a record of all requests received for its page/URLs, including the requester information using 'cookies'.
- The log of these requests could be analyzed to gauge the popularity of those pages among different segments of the population.

- The text and application content on the pages could be analyzed for its usage by visit counts.
- The pages on a website themselves could be analyzed for quality of content that attracts most users.
- Thus, the unwanted or unpopular pages could be weeded out or they can be transformed with different content and style. Similarly, more resources could be assigned to keep the more popular pages fresh and inviting.

WEB STRUCTURE MINING

- The web works through a system of hyperlinks using the hypertext protocol (http).
- Any page can create a hyperlink to any other page, i.e., it can be linked to by another page.
- The intertwined or self-referral nature of web lends itself to some unique network analytical algorithms.
- The structure of web pages could also be analyzed to examine the pattern of hyperlinks among pages.

There are two basic strategic models for successful websites -Hubs and Authorities.

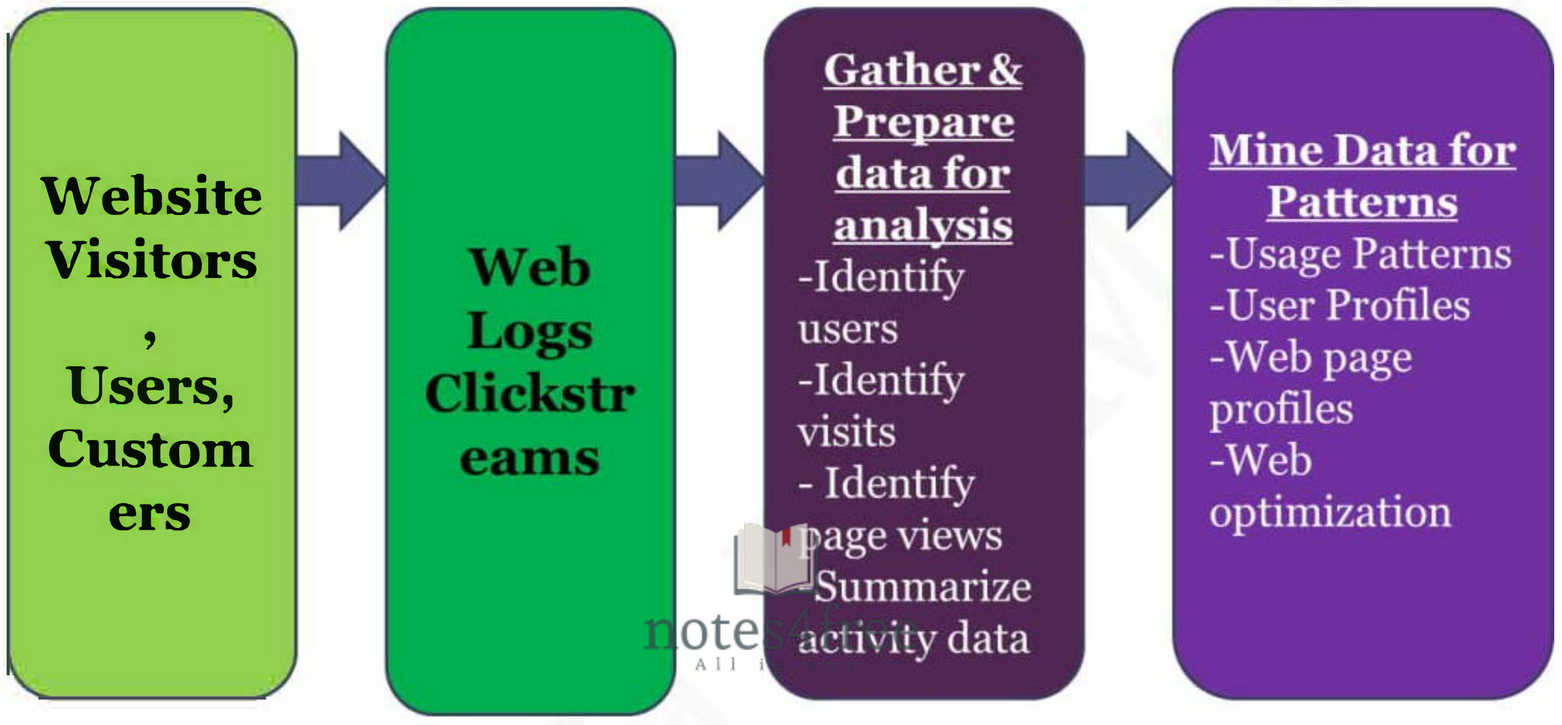
1. **Hubs**
   - These are pages with a large number of interesting links.
   - They serve as a hub or a gathering point, where people visit to access a variety of information.
   - Media sites like yahoo.com or government sites could serve that purpose.
   - More focused sites like traveladvisor.com and yelp.com could aspire to becoming hubs for new emerging areas.
2. **Authorities**
   - Ultimately, people would gravitate towards pages that provide the most complete and authoritative information on a particular subject. This could be factual information, news, advice, user reviews etc.
   - These websites have the most number of inbound links from other websites.
   - Thus, mayoclinic.com could serve as an authoritative page for expert medical opinion; NYtimes.com could serve as an authoritative page for daily news.

WEB USAGE MINING

- As a user clicks anywhere on a webpage or application, the action is recorded by many entities in many locations.
- The browser at the client machine will record 'the click and the web server providing the content would also make a record of the pages served and the user activity on those pages.

- The entities between the client and the server, such as the router, proxy server, or ad server too would record that click.
- The goal of web usage mining is to extract useful information and patterns from data generated through web page visits and transactions.
- The activity data comes from data stored in server access logs, referrer logs, agent logs, and client-side cookies.
- The user characteristics and usage profiles are also gathered directly or indirectly through syndicated data. Further, metadata such as page attributes, content attributes, and usage data are also gathered.

The Web content could be analyzed at multiple levels (Figure 14.2).



- The server side analysis would show the relative popularity of the web pages accessed. Those websites could be hubs and authorities.
- The client side analysis would focus on the usage pattern or the actual content consumed and created by users.
    a) Usage pattern could be analyzed using 'clickstream' analysis, i.e., analyzing web activity for patterns of sequence of clicks, and the location duration of visits on websites. Clickstream analysis can be useful forweb activity analysis, software testing, market research, and analyzing employee productivity.
    b) Textual information accessed on the pages retrieved by users could be analyzed using text mining techniques. The text would be gathered and structured using the bag-of-words technique to build a term-document matrix. This matrix could then be mined using cluster analysis and association rules for patterns such as popular t0pics, user segmentation, and sentiment analysis.
- Web usage mining has many business applications. It can help predict user behavior based on previously learned rules and users' profiles, and can help determine lifetime value of clients. It can also help design cross-marketing strategies across products by observing association rules among the pages on the website.

- Web usage can help evaluate promotional campaigns and see if the users were attracted to the website and used the pages relevant to the campaign.
- Web usage mining could be used to present dynamic information to users based on their interests and profiles. This includes targeted online ads and coupons at user groups based on user access patterns.

WEB MINING ALGORITHMS

- Hyperlink-Induced Topic Search (HITS) is a link analysis algorithm that rates web pages as being hubs or authorities.
- The most famous and powerful of these algorithms is the *Page Rank algorithm*. Invented by Google cofounder Larry Page, this algorithm i9 used by Google to organize the results of its search function.
- This algorithm helps determine the relative importance of any particular web page by counting the number and quality of links to a page.
- The websites with more number of links, and/or more links from higher quality websites, will be ranked higher.
- It works in a similar way as determining the status of a person in a society of people.
- Those with relations to more people and/or relations to people of higher status will be accorded a higher status.
- Page Rank is the algorithm that helps determine the order of pages listed upon a Google search query.
- The original Page Rank algorithm formulation has been updated in many ways and the latest algorithm is kept a secret so other websites cannot take advantage of the algorithm and manipulate their website according to it.
- However, there are many standard elements that remain unchanged. These elements lead to the principles for a good website. This process is also called Search Engine Optimization (SEO).

Social Network Analysis

INTRODUCTION

Social Networks are a graphical representation of relationships among people and or entities.

Social Network Analysis (SNA) is the art and science of discovering patterns of interaction and influence within the participants in' a network.

- These participants could be people, organizations, machines, concepts, or any other kinds of entities.
- An ideal application of social network analysis will discover essential characteristics of a network including its Central nodes and its subnetwork structure.

- Subnetworks are clusters of nodes, Where the within subnetwork connections are stronger than the connections with nodes outside the subnetwork.
- SNA is accomplished through graphically representing social relationships into a network of nodes and links and applying iterative computational techniques to measure the strengths of relationships.
- The social network analysis ultimately helps relate the totality of network to the Unified Field which is the ultimate entity with infinite relationships among everything.

Applications of SNA

1. **Self-awareness** Visualizing his/her social network can help a person organiz their relationships and support network.
2. **Communities** Social Network Analysis can help identification, construction and strengthening of networks within communities to build wellness, comfort and resilience. Analysis of joint authoring relationships and citations help identify subnetworks of specializations of knowledge in an academic field. Researchers at Northwestern University found that the most determinant factor in the success of a Broadway play was the strength of relationships amongst the crew and cast.
3. **Marketing** There is a popular network insight that any two people are related to each other through at most seven degrees of links. Organizations can use this insight to reach out with their message to large number of people and also to listen actively to opinion leaders as ways to understand their customers' needs and behaviors. Politicians can reach out to opinion leaders to get their message out.
4. **Public Health** Awareness of networks can help identify the paths that certain diseases take to spread. Public health professionals can isolate and contain diseases before they expand to other networks.

Network Topologies

There are two primary types of network topologies *ring-type* and *hub-spoke* topologies. Each of the topologies has different characteristics and benefits.

- In the *ring network*, nodes typically connect to their adjacent nodes in the network and all nodes can be connected to each other.
- A ring network could be dense where every node has a direct connection with practically every node or it could be sparse where every node connects to a small subset of the nodes.
- A dense network, with more connections, will allow many direct connections between pairs of nodes.
- In a sparse network, one may need to traverse many connections to reach the desired destination. A peer-to-peer email (or messaging) network is an example on the ring model, as anyone can potentially directly connect with anyone else.

- In the hub-spoke model, there is one central hub node to which all the other node are connected and no direct relationships between the nodes.
- Nodes connect with each other through the hub node.
- This is a hierarchical network structure since the hub node is central to the network.
- The hub node is structurally more important as it is central to all communications between other peripheral nodes,
- The hub-spoke network is that one could predictably reach from any node to any other node through traversing exactly just two connections.
- As an example, modern airlines operate on this model to maintain hub networks from which they operate flights to a large number of airports.
- The density of a network can be defined as the average number of connections per node.
- The cohesiveness of the network is a related concept, which is the average number of connections needed to reach from one node to the other.
- Another way of analyzing networks is to define the centrality (or importance) of wary node. The number of links associated with a node is a sign of centrality of the node.
- In the ring network in the figure below, each node has exactly 2 links. Thus, there is no central node. However, in the hub-spoke network, the hub-node N has 8 links while all other nodes have only 1 link each. Thus, node N has a high centrality.

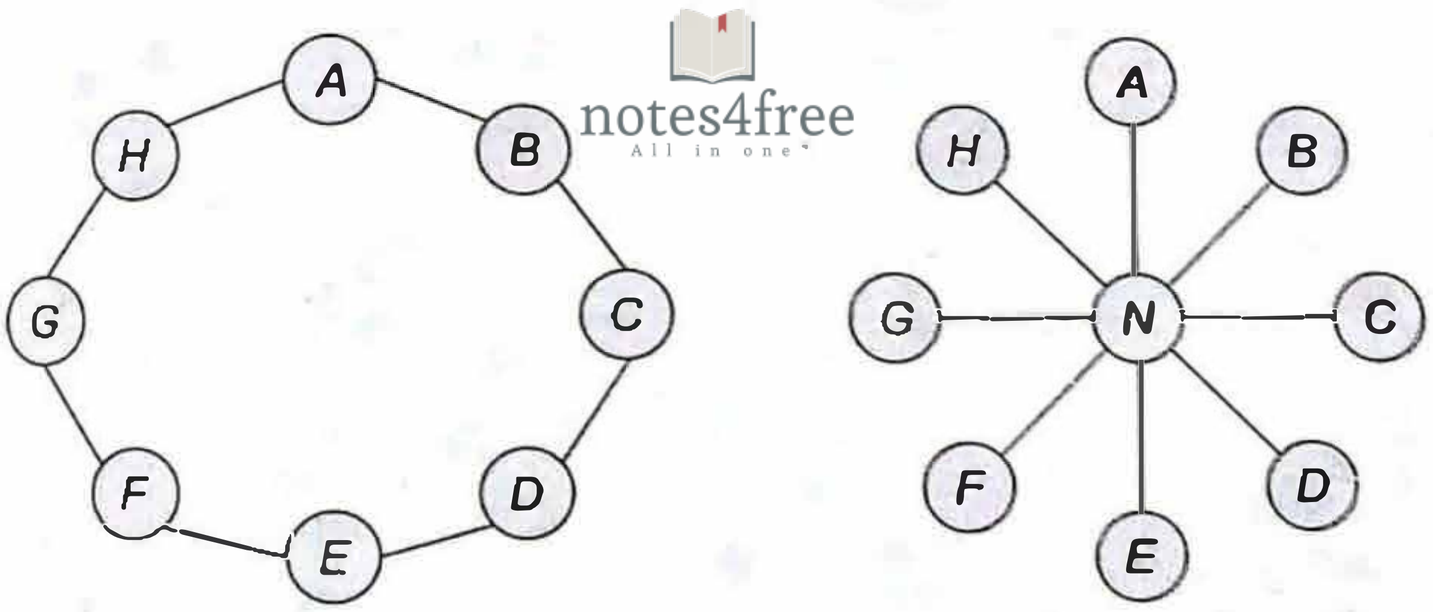


FIGURE 15.1 Network Topologies: Ring (left) and Hub-spoke (right)

A variant that combines both the above types is the network of networks in which each participating network will connect with other networks at selected points of contact.

For example, the internet is a network of networks. Here, all the commercial, university, government and similar computer networks connect to one other at certain designated nodes (called gateways) to exchange information and conduct business.

TECHNIQUES AND ALGORITHMS

There are two major levels of social network analysis *discovering subnetworks* within the network *and ranking the nodes* to find more important nodes or hubs

Finding Subnetworks

- A large network could be better analyzed and managed if it can be seen as an interconnected set of distinct subnetworks each with its own distinct identity and unique characteristics.
- This is like doing a cluster analysis of nodes. Nodes with strong ties between them would belong to the same subnetwork, while those with weak or no ties would belong to separate subnetworks.
- This is unsupervised learning technique, as in Apriori there is no correct number of subnetworks in a network.
- The usefulness of the subnetwork structure for decision-making is the main criterion for adopting a particular structure.

FIGURE 15.2 A Network with Distinct Subnetworks

Computing Importance of Nodes

- When the connections between nodes in the network have a direction to them, then the nodes can be compared for their relative influence or rank.
- This is done using *'Influence Flow model'*. Every outbound link from a node can be considered an outflow of influence.
- Every incoming link is similarly an inflow of influence. More in-links to a node means greater importance.
- Thus there will be many direct and indirect flows of influence between any two nodes in the network.
- Computing the relative influence of each node is done on the basis of an input output matrix of flows of influence among the nodes.

- Assume each node has an influence value. The computational task is to identify a set of rank values that satisfies the set of links between the nodes.
- It is an iterative task where we begin with some initial values and continue to iterate till the rank values stabilize.

Consider the following simple network with 4 A ' B nodes (A, B, C, D) and 6 directed links between them as shown in Figure 15.3. Note that there is a bidirectional link. Here are the links

Node A links into B
Node B links into C
Node C links into D
Node D links into A
Node A links into C
Node B links into A.

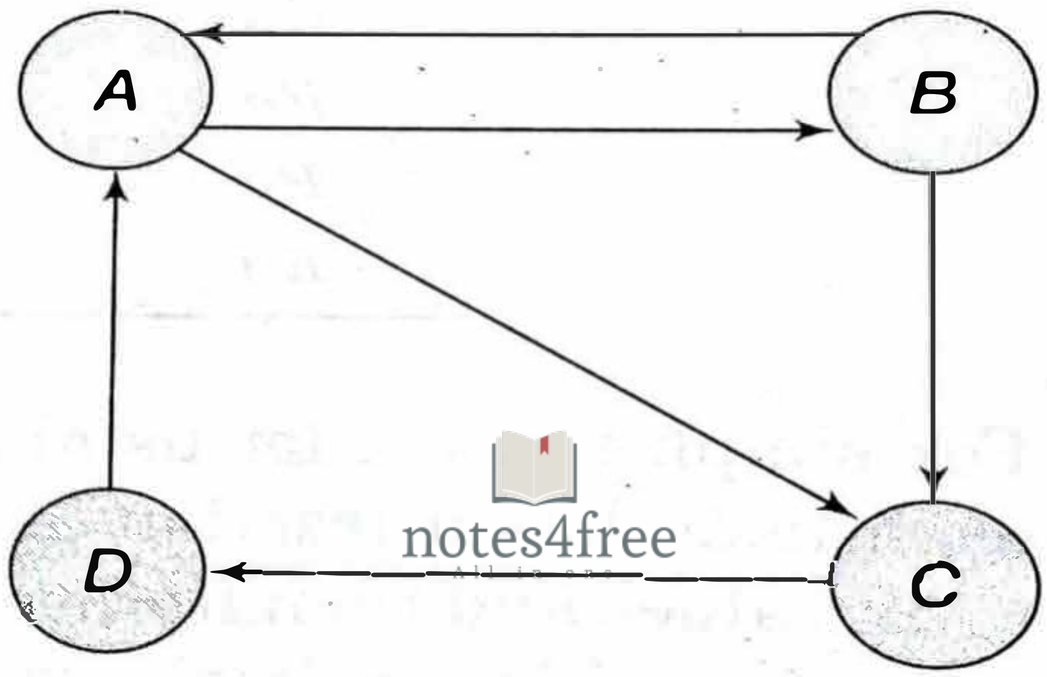


FIGURE 15-3

- The goal is to find the relative importance, or rank, or every node in the network. This will help identify the most important node(s) in the network.
- We begin by assigning the variables for influence (or rank) value for each node s **R_a, R_b, R_c, and R_d**. The goal is to find the relative values of these variables
- There are two outbound links from node A to nodes B and C. Thus, both B and C receive half of node A's influence. Similarly, there are two outbound links from node B to nodes C and A, so both C and A receive half of node B's influence.
- There is only outbound link from node D into node A. Thus, node A gets all the influence of node D. There is only outbound link from node C into node D and hence, node D gets all the influence of node C.
- Node A gets all of the influence of node D and half the influence of node B,

Thus, $$R_a = 0.5 \times R_b + R_d$$

Node B gets half the influence of node A.

Thus, $$R_b = 0.5 \times R_a$$

Node C gets half the influence of node A and half the influence of node B.

Thus, $R_c = 0.5 \times R_a + 0.5 \times R_b$

Node D gets all of the influence of node C and half the influence of node B.

Thus, $R_d = R_c$

We have 4 equations using 4 variables. These can be solved mathematically.

We can represent the coefficients of these 4 equations in a matrix form as shown in Dataset 15.1 given below. This is the Influence Matrix. The zero values represent that the term is not represented in an equation.

Dataset 15.1

	Ra	*Rb*	*Rc*	*Rd*
Ra	0	0.50	0	1.00
Rb	0.50	0	0	0
Rc	0.50	0.50	0	0
Rd	0	0	1.00	0

For simplification, let us also state that all the rank values add up to 1. Thus, each node has a fraction as the rank value. Let us start with an initial set of rank values and then iteratively compute new rank values till they stabilize. One can start with any initial rank values, such as 1/n or 1/4 for each of the nodes.

Variable	Initial Value
Ra	0.250
Rb	0.250
Rc	0.250
Rd	0.250

Computing the revised values using the equations stated earlier, we get a revised set of values shown as Iterationl. (This can be computed easily by creating formulae using the influence matrix in a spreadsheet such as Excel.)

Variable	Initial Value	Iteration1
Ra	0.250	0.375
Rb	0.250	0.125
Rc	0.250	0.250
Rd	0.250	0.250

Using the rank values from Iterationl as the new starting values, we can compute new values for these variables, shown as Iteration2 Rank values will continue to change.

Variable	Initial Value	Iteration1	Iteration2
Ra	0.250	0.375	0.3125
Rb	0.250	0.125	0.1875
Rc	0.250	0.250	0.250
Rd	0.250	0.250	0.250

Working from values of Iteration2 and so, we can do a few more iterations till the values stabilize. Dataset 15.2 shows the final values after the 8th iteration.

Dataset 15.2

Variable	Initial Value	Iteration1	Iteration2	...	Iteration8
Ra	0.250	0.375	0.313	...	0.333
Rb	0.250	0.125	0.188	...	0.167
Rc	0.250	0.250	0.250	...	0.250
Rd	0.250	0.250	0.250	...	0.250

- The final rank shows that rank of node A is the highest at 0.333. Thus, the most important node is A. The lowest rank is 0.167 of Rb.
- Thus, B is the least important node. Nodes C and D are in the middle. In this case, their ranks did not change at all.
- The relative scores of the nodes in this network would have been the same irrespective of the initial values chosen for the computations.
- It may take longer or shorter number of iterations for the results to stabilize for different sets of Initial values.

PAGERANK

- PageRank is a particular application of the social network analysis techniques above to compute the relative importance of websites in the overall World Wide Web.
- The data on websites and their links is gathered through web crawler bots that traverse through the webpages at frequent intervals.
- Every webpage is a node in a social network and all the hyperlinks from that page become directed links to other webpages.
- Every outbound link from a webpage is considered an outflow of influence of that webpage.

- An iterative computational technique is applied to compute a relative importance to each page.
- That score is called PageRank according to an eponymous algorithm invented by the founders of Google, the web search company.
- PageRank is used by Google for ordering the display of websites in response to search queries.
- To be shown higher in the search results, many website owners try to artificially boost their PageRank by creating many dummy websites whose ranks can be made to flow into their desired website.
- Also, many websites can be designed to cyclical sets of links from where the web crawler may not be able to break out. These are called spider traps.
- To overcome these and other challenges, Google includes a Teleporting factor into computing the PageRank.
- Teleporting assumed that there is a potential link from any node to any other node, irrespective of whether it actually exists.
- Thus, the influence matrix is multiplied by a weighting factor called Beta with a typical value of 0.85 or 85 percent.
- The remaining weight of 0.15 or 15 percent is given to teleportation.
- In teleportation matrix, each cell is given a rank of 1/n. where n is the number of nodes the web.
- The two matrices are added to compute the final influence matrix. This matrix can be used to iteratively compute the PageRank of all the nodes, just as shown in the example earlier.

PRACTICAL CONSIDERATIONS

1. **Network Size** Most SNA research is done using small networks. Collecting data about large networks can be very challenging. This is because the number of links is the order of the square of the number of nodes. Thus, in a network of 1000 nodes there are potentially 1 million possible pairs of links.
2. **Gathering Data** Electronic communication records (emails, chats, etc.) can be harnessed to gather social network data more easily. Data on the nature and quality of relationships need to be collected using survey documents. Capturing and cleansing and organizing the data can take a lot of time and effort, just like in a typical data analytics project.
3. **Computation and Visualization** Modeling large networks can be computationally challenging and visualizing them also would require special skills. Big data analytical tools may be needed to compute large networks.
4. **Dynamic Networks** Relationships between nodes in a social network can be fluid. They can change in strength and functional nature. For example, there could be multiple relationships between two people they could simultaneously be coworkers, coauthors,

and spouses. The network should be modeled frequently to see the dynamics of the network.

Table 15.1 Social Network Analysis vs Traditional Data Analytics

Dimension	Social Network Analysis	Traditional Data Mining
Nature of learning	Unsupervised learning	Supervised and unsupervised learning
Analysis of goals	Hub nodes, important nodes, and subnetworks	Key decision rules, cluster centroids
Dataset structure	A graph of nodes and (directed) links	Rectangular data of variables and instances
Analysis techniques	Visualization with statistics; iterative graphical computation	Machine learning, statistics
Quality measurement	Usefulness is key criterion	Predictive accuracy for classification techniques

Reference text book:

Anil Maheshwari, "Data Analytics", 1st Edition, McGraw Hill Education, 2017.
ISBN-13: 978-9352604180