الجمعية المصرية لهندسة اللغة

THE EGYPTIAN SOCIETY
OF LANGUAGE ENGINEERING

Keras: The Python Deep Learning Library

Prof. Aly Fahmy
aly.fahmy@gmail.com

Dr. Wael Gomaa
wael.goma@gmail.com

The Seventeenth Conference
On Language Engineering
December 6-7, 2017, Cairo, Egypt

Outline

- **Introduction**
- **Keras Documentation**
- **Life-Cycle for Models in Keras**
- **Practical Examples**

Introduction

Introduction

- Keras is a deep-learning framework for Python that provides a convenient way to define and train almost any kind of deep-learning model
- Keras was initially developed for researchers, with the aim of enabling fast experimentation

Introduction

- It allows the same code to run seamlessly on CPU or GPU.
- It has a user-friendly API that makes it easy to quickly prototype deep-learning models.
- It has built-in support for convolutional networks, recurrent networks, and any combination of both.
- It supports arbitrary network architectures: multi-input or multi-output models, layer sharing, model sharing, and so on.
- It's compatible with any version of Python from 2.7 to 3.6 .

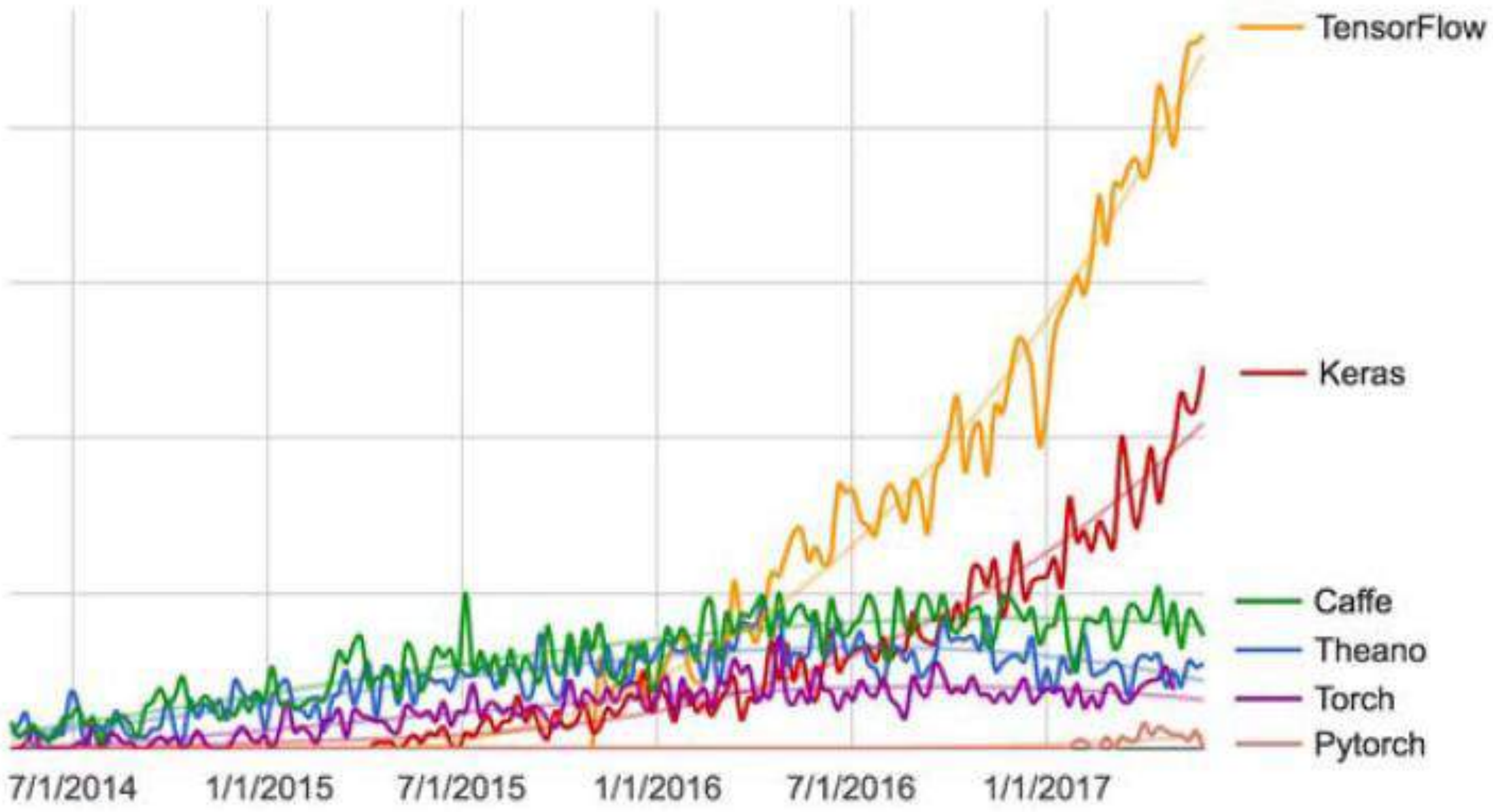
Introduction

- Keras has well over 200,000 users, ranging from academic researchers and engineers at both startups and large companies to graduate students and hobbyists.

- Keras is used at Google, Netflix, Uber, CERN, Yelp, Square, and hundreds of startups working on a wide range of problems.

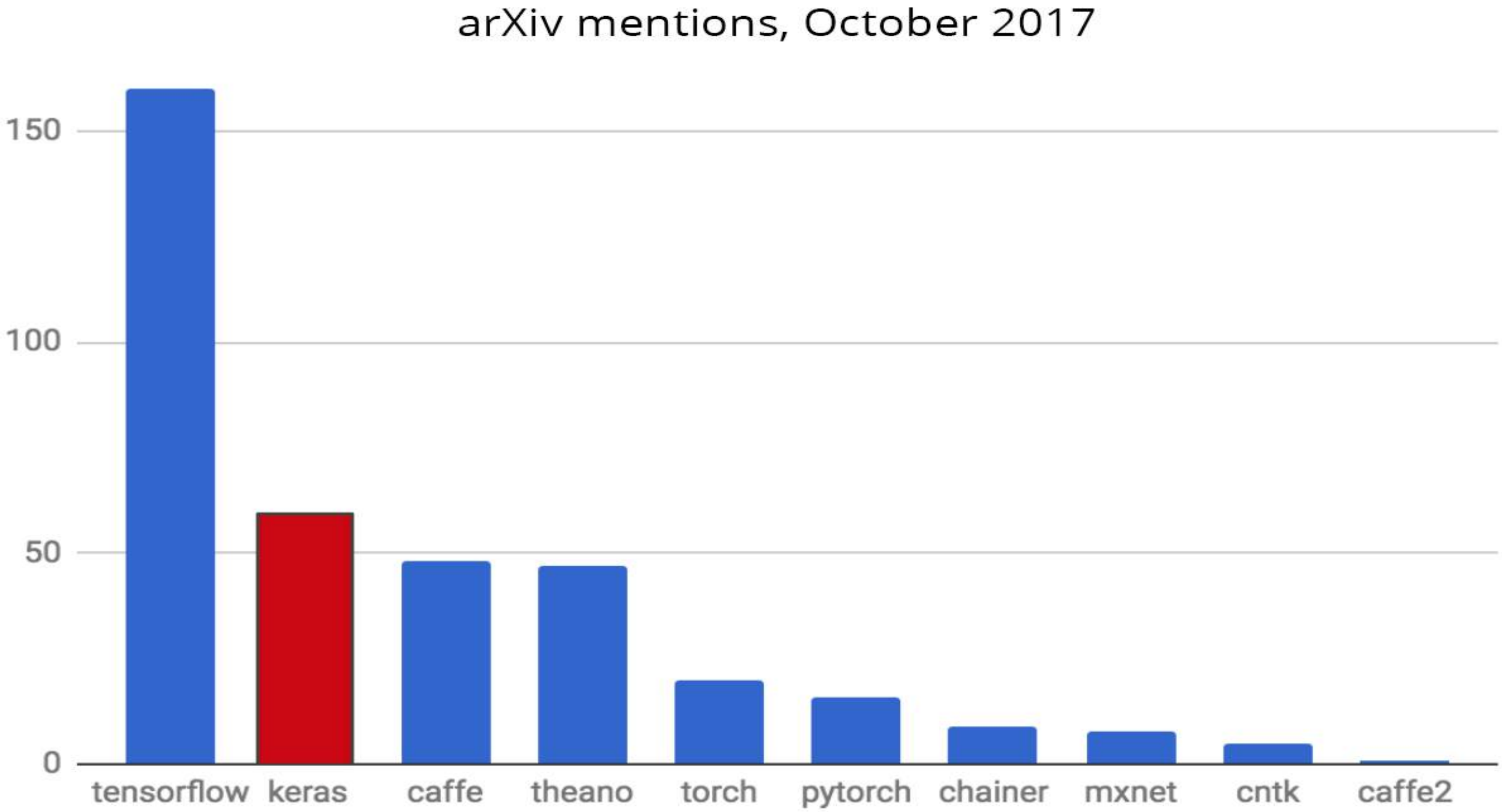
- Keras is also a popular framework on Kaggle, the machine-learning competition website, where almost every recent deep-learning competition has been won using Keras models.

Introduction



Google web search interest for different deep-learning frameworks over time

Introduction

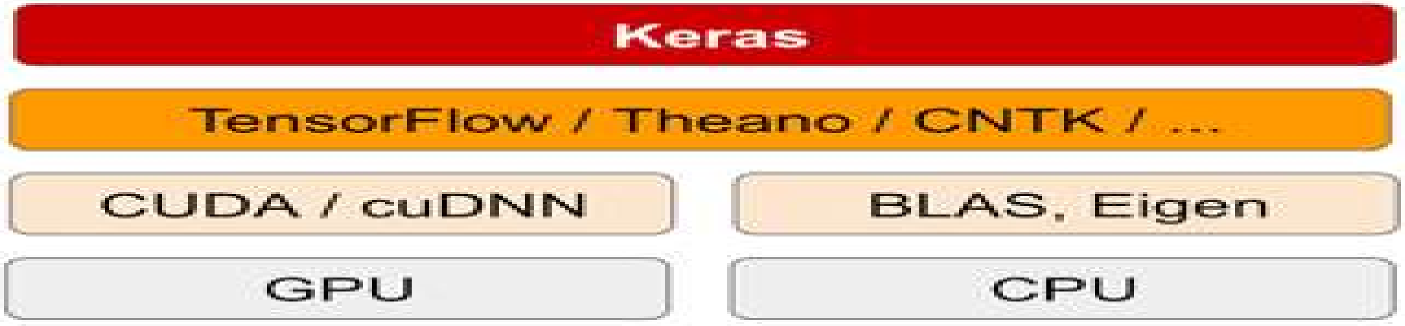


Keras is also a favorite among deep learning researchers, coming in #2 in terms of mentions in scientific papers uploaded to the preprint server arXiv.org

Introduction

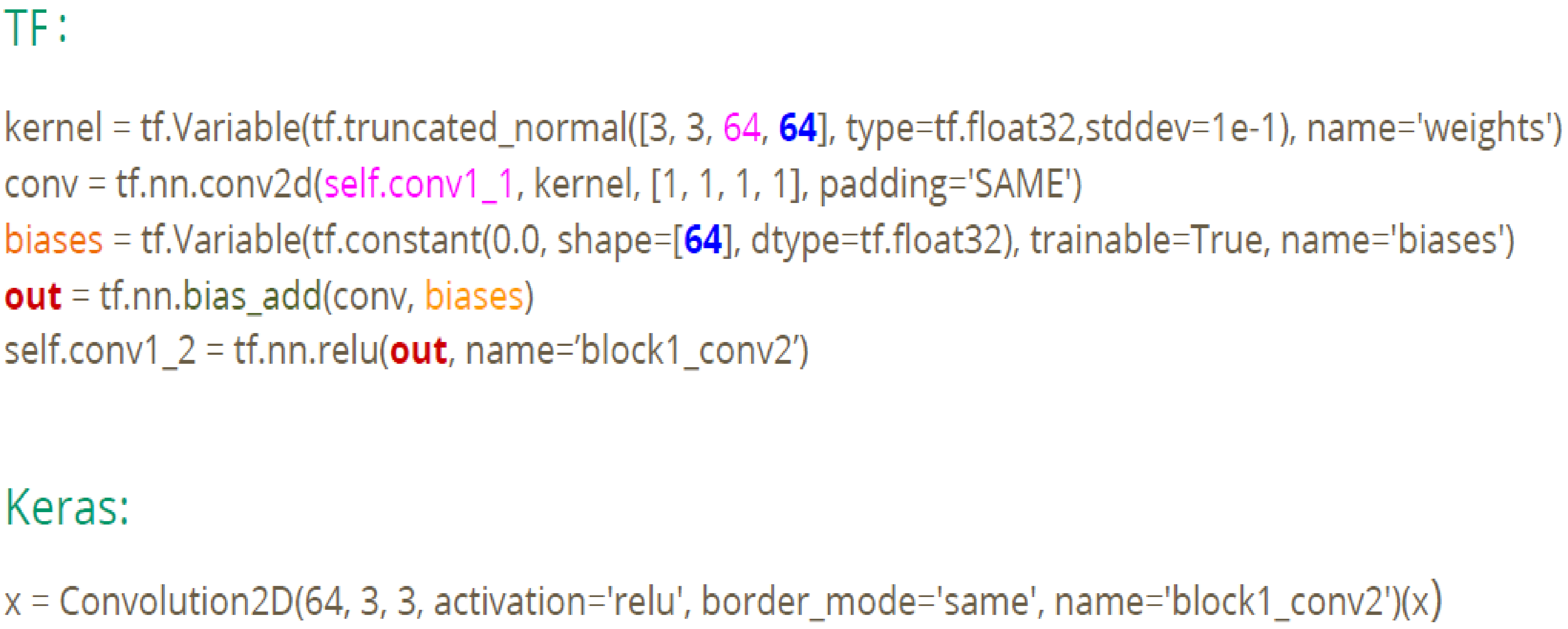
- Currently, the three existing backend implementations are the TensorFlow backend, the Theano backend, and the Microsoft Cognitive Toolkit (CNTK) backend.

- Any piece of code that you write with Keras can be run with any of these backends without having to change anything in the code.
- Keras compatibility module introduced in TensorFlow: tf.keras



Introduction

- Keras is a model-level library, providing high-level building blocks for developing deep-learning models.
- It doesn’t handle low-level operations such as tensor manipulation and differentiation.



TF :

```
kernel = tf.Variable(tf.truncated_normal([3, 3, 64, 64], type=tf.float32,stddev=1e-1), name='weights')
conv = tf.nn.conv2d(self.conv1_1, kernel, [1, 1, 1, 1], padding='SAME')
biases = tf.Variable(tf.constant(0.0, shape=[64], dtype=tf.float32), trainable=True, name='biases')
out = tf.nn.bias_add(conv, biases)
self.conv1_2 = tf.nn.relu(out, name=’block1_conv2’)
```

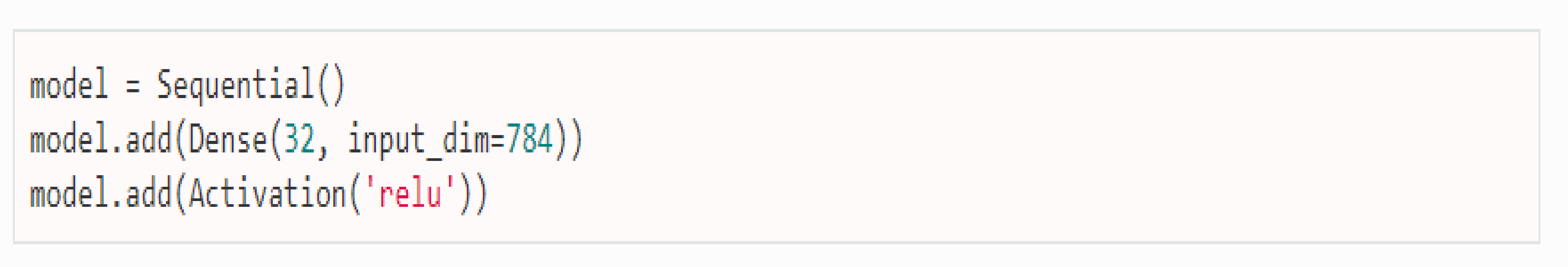
Keras:

```
x = Convolution2D(64, 3, 3, activation='relu', border_mode='same', name='block1_conv2')(x)
```

Keras Documentation

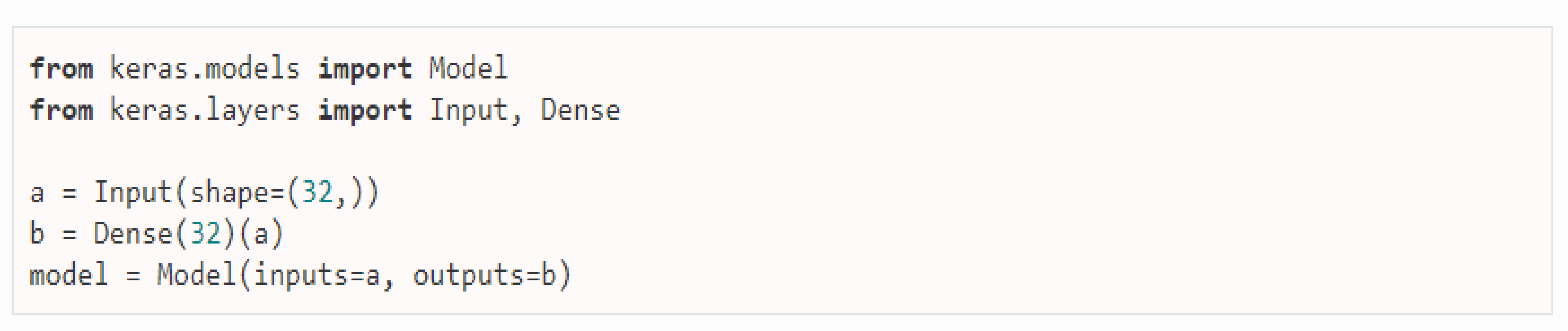
Keras Documentation

- Keras Models: https://keras.io/models/
  - The Sequential model: is a linear stack of layers



```
model = Sequential()
model.add(Dense(32, input_dim=784))
model.add(Activation('relu'))
```

  - The Model class used with functional API : given some input tensor(s) and output tensor(s), you can instantiate a Model.



```
from keras.models import Model
from keras.layers import Input, Dense

a = Input(shape=(32,))
b = Dense(32)(a)
model = Model(inputs=a, outputs=b)
```

Keras Documentation

- Keras Layers: https://keras.io/layers/
  - Core Layers (Dense, Dropout ..)
  - Convolutional Layers (Conv1D, Conv2D ..)
  - Pooling Layers (MaxPooling1D, MaxPooling2D ..)
  - Recurrent Layers (RNN, GRU, LSTM ..)
  - Embedding Layers
  - Merge Layers (Add, Concatenate ..)
  - Noise Layers (Gaussian Noise, Gaussian Dropout)

Keras Documentation

- Keras Preprocessing: https://keras.io/preprocessing/
    - Sequence Preprocessing (pad sequence, skipgrams ..)
    - Text Preprocessing (one_hot, Tokenizer ..)
    - Image Preprocessing (Image Data Generator)

Keras Documentation

- Keras Losses: https://keras.io/losses/
  - Mean_squared_error
  - Mean_asolute_error
  - Binary_crossentropy
  - Categorical_crossentropy
  - Cosine_proximity

Keras Documentation

- Keras Metrics: https://keras.io/metrics/
    - Binary accuracy
    - Categorical accuracy
    - Sparse categorical accuracy
    - Top K categorical accuracy ...

Keras Documentation

- Keras Optimizers: https://keras.io/optimizers/
  - SGD
  - RMSprop
  - Adam
  - Adamax
  - Nadam
  - Adagrad
  - TFOptimizer ...

Keras Documentation

- Keras Activations: https://keras.io/activations/
  - softmax
  - softplus
  - elu
  - selu
  - relu
  - sigmoid
  - tanh
  - linear ...

Keras Documentation

- Keras Datasets: https://keras.io/datasets/
  - CIFAR100 small image classification
  - MNIST database of handwritten digits
  - Boston housing price regression dataset
  - Reuters newswire topics classification
  - IMDB Movie reviews sentiment classification …

Keras Documentation

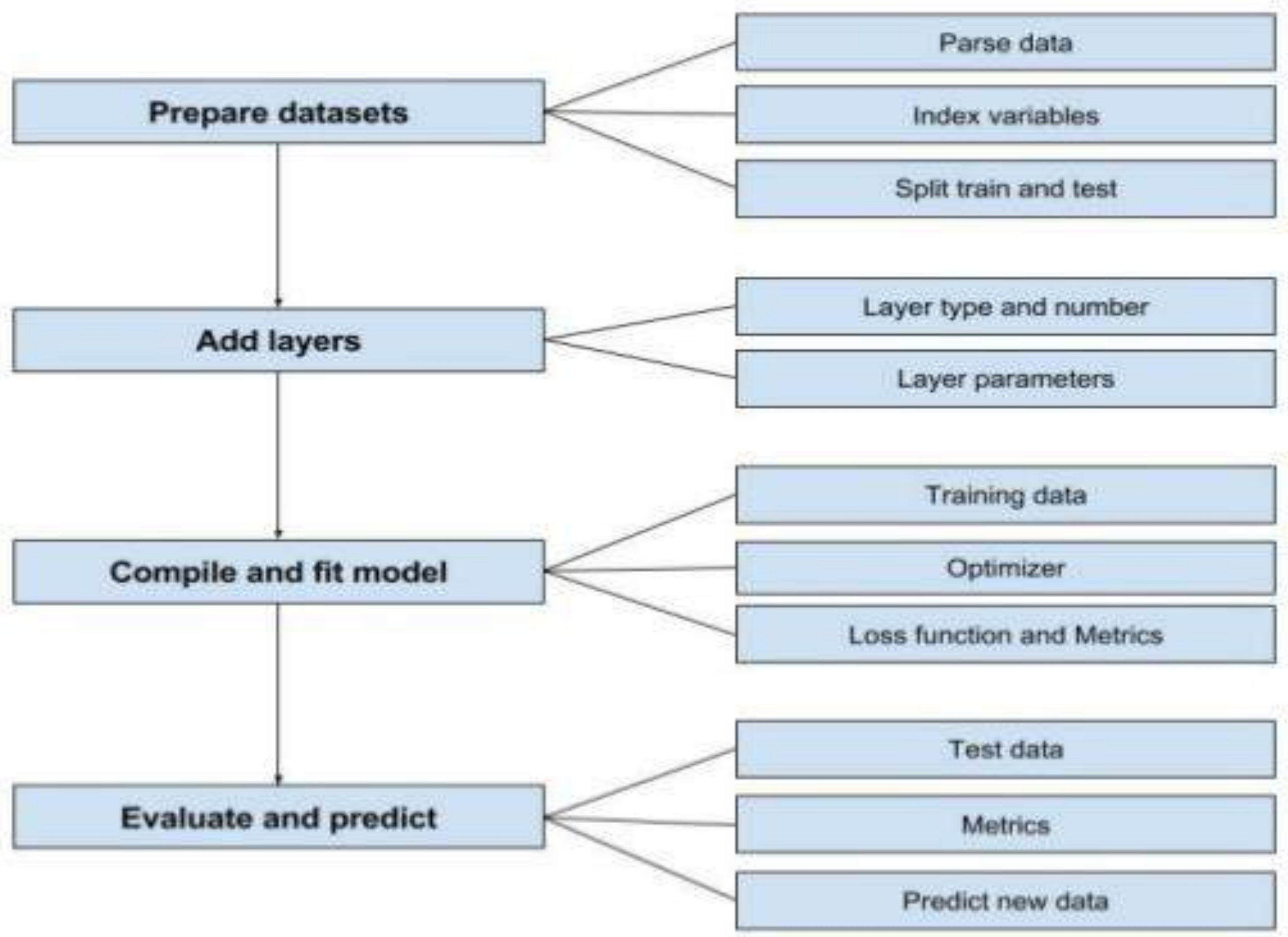
- Keras Applications: https://keras.io/applications/
  - Xception
  - VGG16 – VGG19
  - ResNet50
  - MobileNet
  - InceptionV3…

Keras Documentation

- Backend https://keras.io/backend/
- Initializers https://keras.io/initializers/
- Regularizers https://keras.io/regularizers/
- Constraints https://keras.io/constraints/
- Visualization https://keras.io/visualization/
- Scikit-learn API https://keras.io/scikit-learn-api/
- Utils https://keras.io/utils/
- Contributing https://keras.io/contributing/

Life-Cycle for Models in Keras

Life-Cycle



Prepare datasets

Parse data

Index variables

Split train and test

Add layers

Layer type and number

Layer parameters

Compile and fit model

Training data

Optimizer

Loss function and Metrics

Evaluate and predict

Test data

Metrics

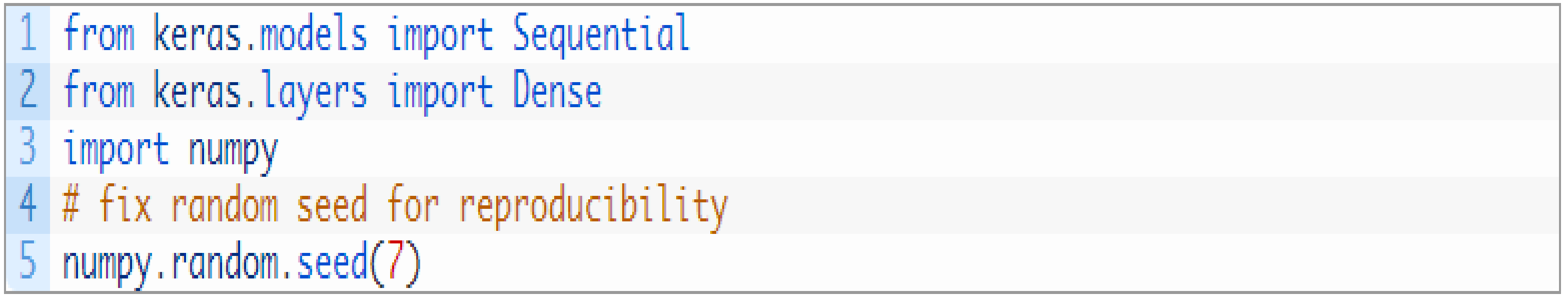
Predict new data

Life-Cycle

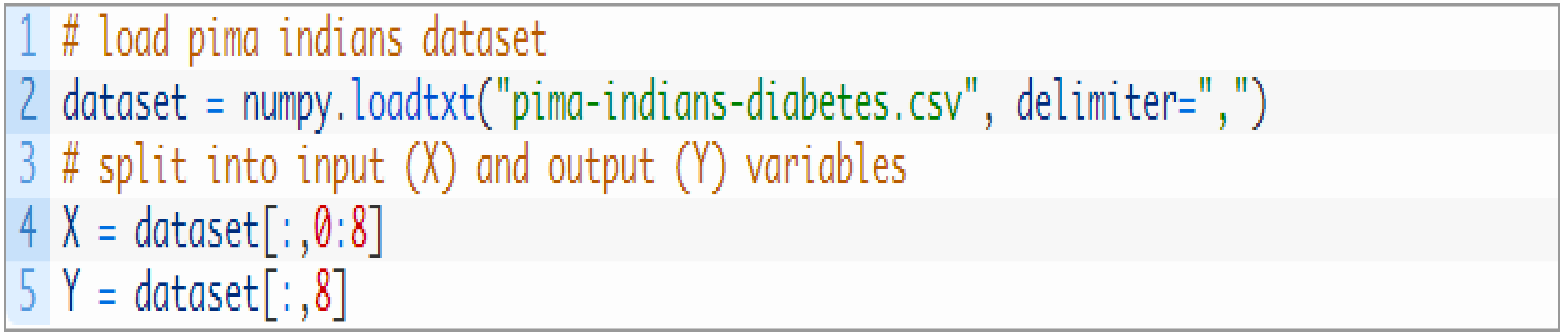
- Load Data
- Define Model
- Compile Model
- Fit Model
- Evaluate Model
- Make Predictions

Life-Cycle

- Load Data



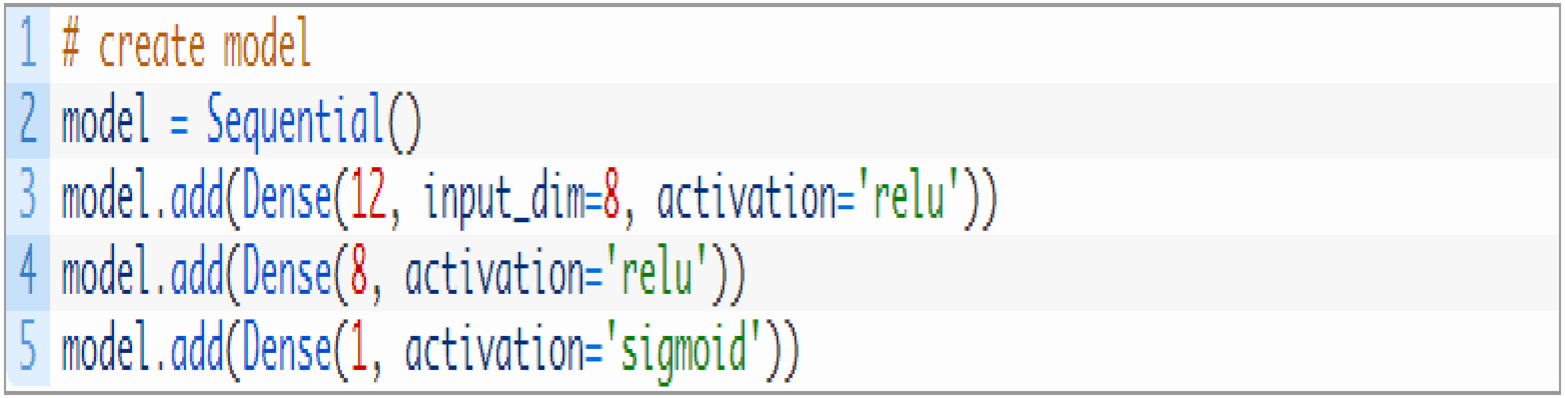
```python
from keras.models import Sequential
from keras.layers import Dense
import numpy
# fix random seed for reproducibility
numpy.random.seed(7)
```



```python
# load pima indians dataset
dataset = numpy.loadtxt("pima-indians-diabetes.csv", delimiter=",")
# split into input (X) and output (Y) variables
X = dataset[:,0:8]
Y = dataset[:,8]
```

Life-Cycle

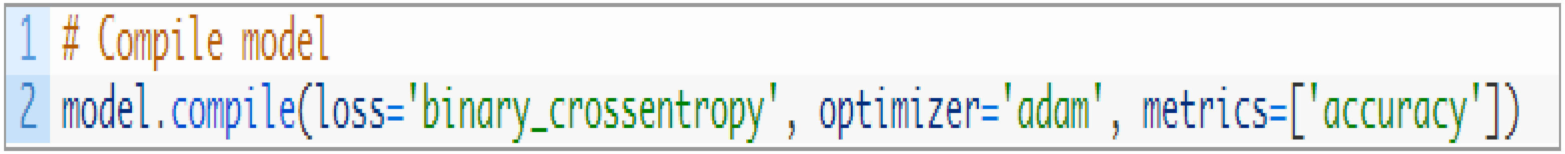
- Define Model



```
# create model
model = Sequential()
model.add(Dense(12, input_dim=8, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

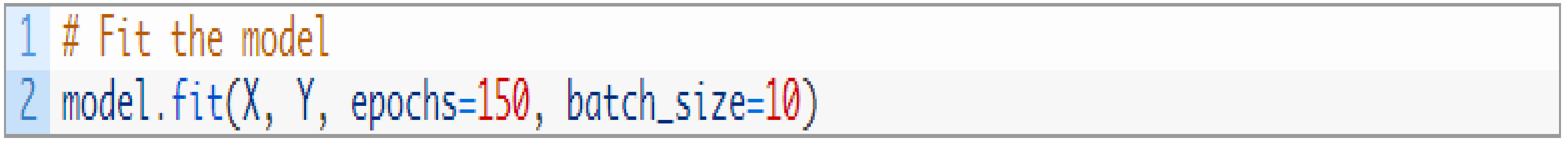
Life-Cycle

- Compile Model



```
# Compile model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

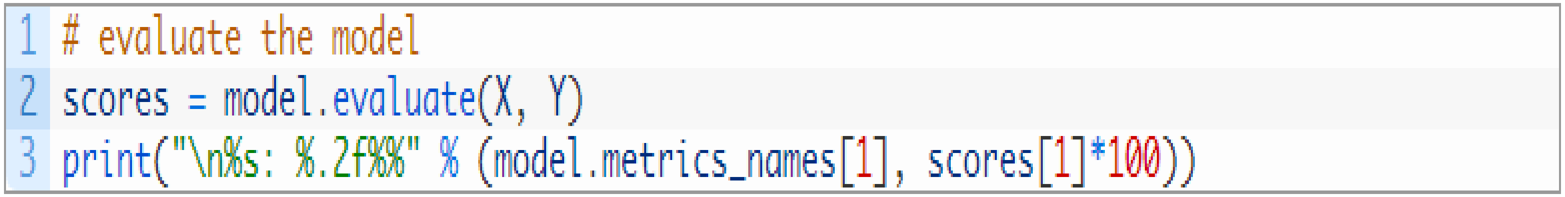
- Fit Model



```
# Fit the model
model.fit(X, Y, epochs=150, batch_size=10)
```

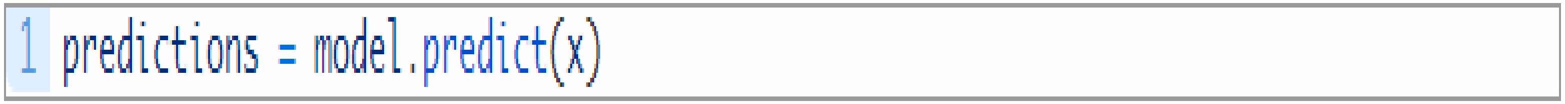
Life-Cycle

- Evaluate Model



```
# evaluate the model
scores = model.evaluate(X, Y)
print("\n%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
```

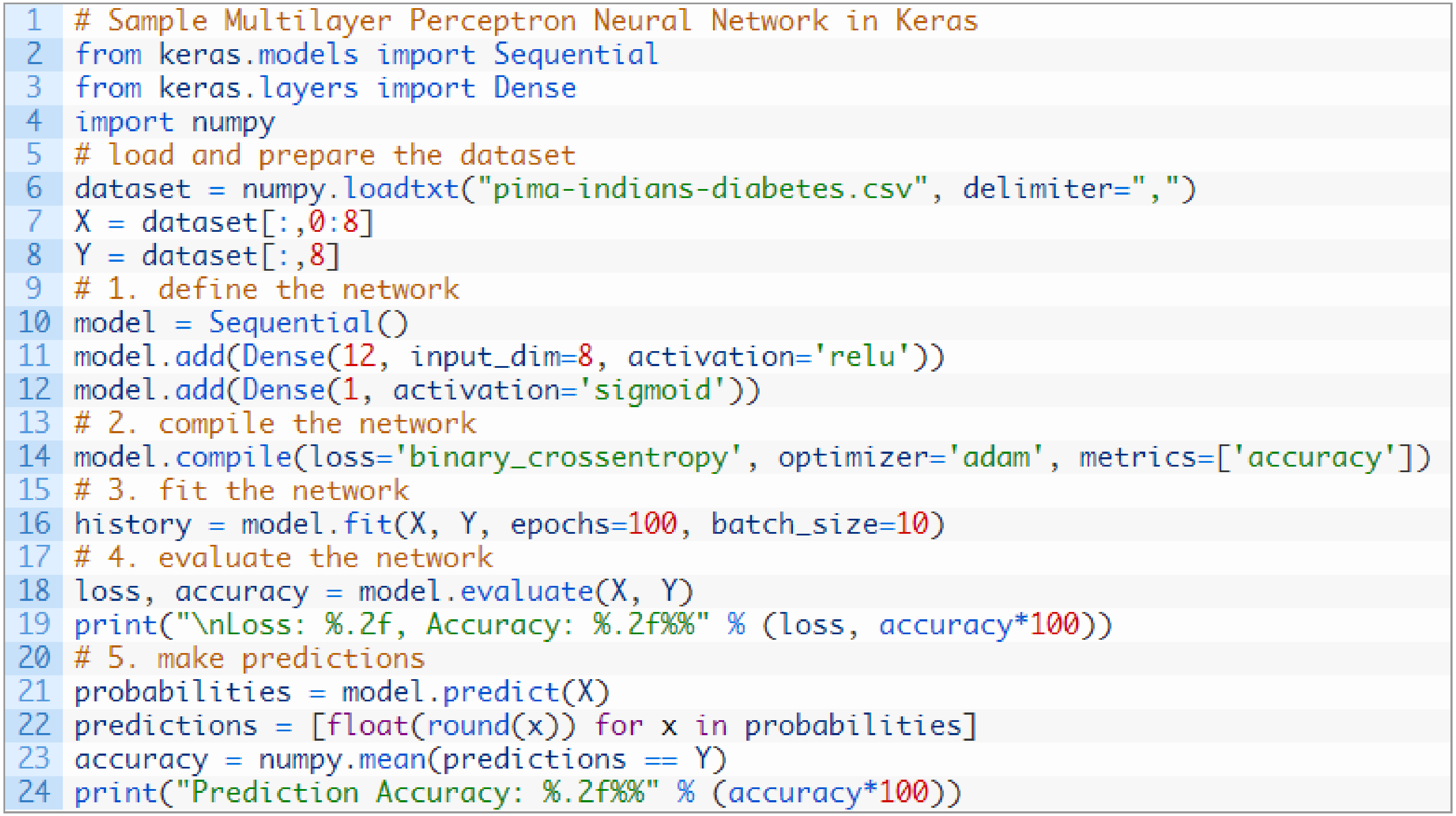
- Make Predictions



```
predictions = model.predict(x)
```

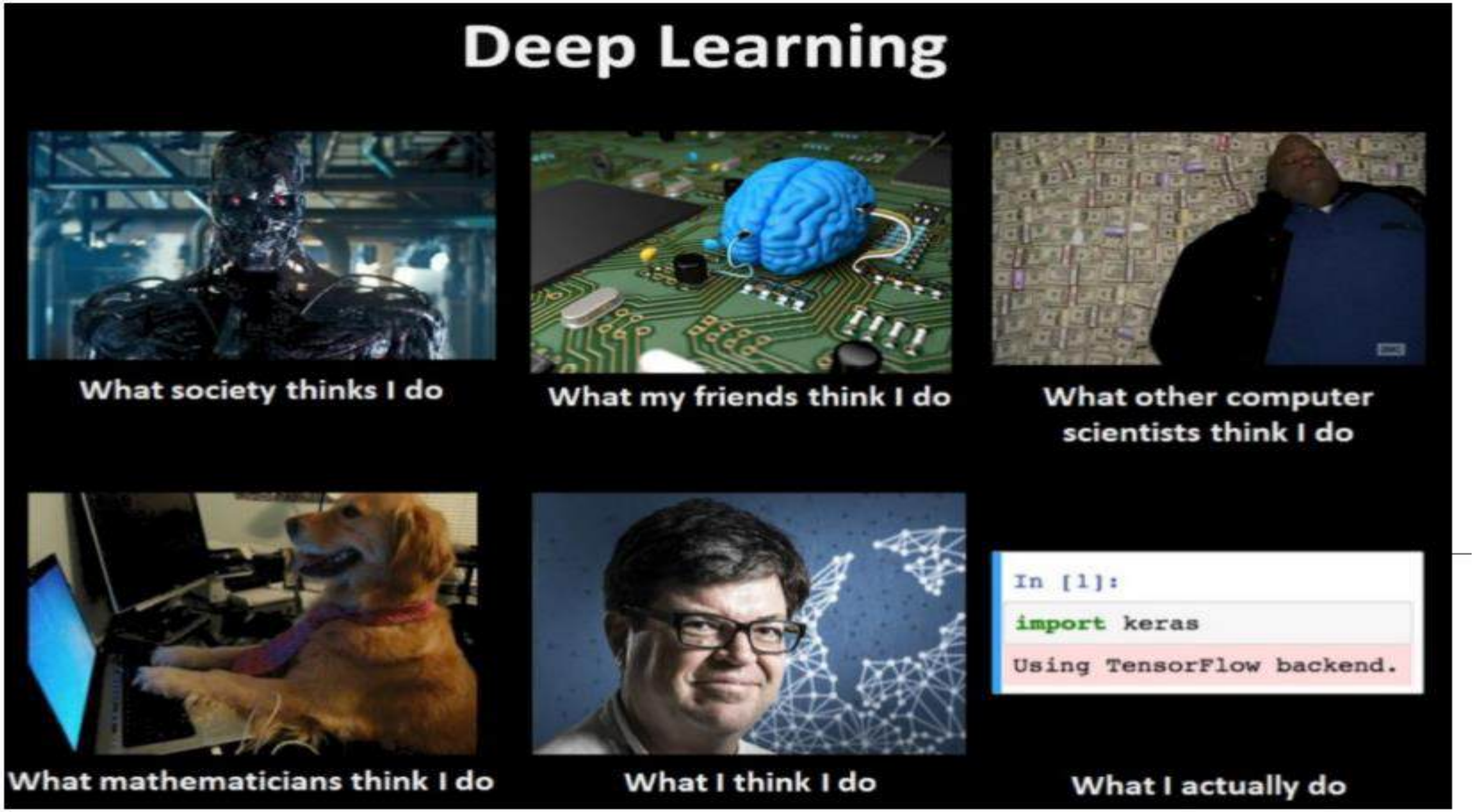
Life-Cycle

- Tie it all together



```
# Sample Multilayer Perceptron Neural Network in Keras
from keras.models import Sequential
from keras.layers import Dense
import numpy
# load and prepare the dataset
dataset = numpy.loadtxt("pima-indians-diabetes.csv", delimiter=",")
X = dataset[:,0:8]
Y = dataset[:,8]
# 1. define the network
model = Sequential()
model.add(Dense(12, input_dim=8, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
# 2. compile the network
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
# 3. fit the network
history = model.fit(X, Y, epochs=100, batch_size=10)
# 4. evaluate the network
loss, accuracy = model.evaluate(X, Y)
print("\nLoss: %.2f, Accuracy: %.2f%%" % (loss, accuracy*100))
# 5. make predictions
probabilities = model.predict(X)
predictions = [float(round(x)) for x in probabilities]
accuracy = numpy.mean(predictions == Y)
print("Prediction Accuracy: %.2f%%" % (accuracy*100))
```

Practical Examples


Deep Learning
What society thinks I do
What my friends think I do
What other computer scientists think I do
What mathematicians think I do
What I think I do
What I actually do

```
In [1]:
import keras
Using TensorFlow backend.
```

HTK Tool Kit

HTK Tool Kit

What is HTK tool kit

The HTK language modeling tools are a group of programs designed for constructing and testing statistical *n-gram* language models

HTK Tool Kit

What to prepare

Training & Test Text

Dictionary

HTK Tool Kit

Training & Test Text

- Plain text sentences
- One sentence per line
- Sentence starts with <s>
- Sentence ends with </s>

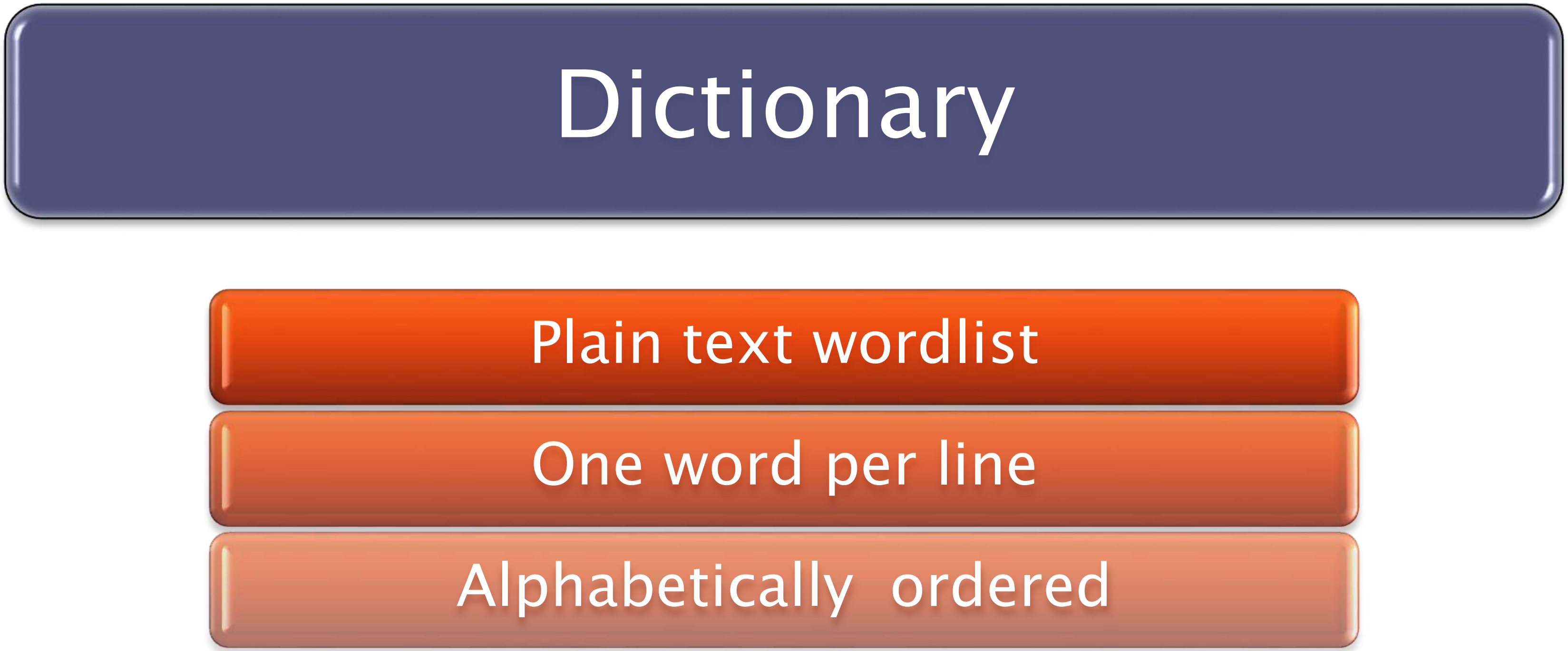
HTK Tool Kit

Training Text Sample

<s> IT WAS ON A BITTERLY COLD NIGHT AND FROSTY MORNING TOWARDS THE END OF THE WINTER OF NINETY SEVEN THAT I WAS AWAKENED BY A TUGGING AT MY SHOULDER </s>

<s> IT WAS HOLMES </s>

HTK Tool Kit



Dictionary

Plain text wordlist

One word per line

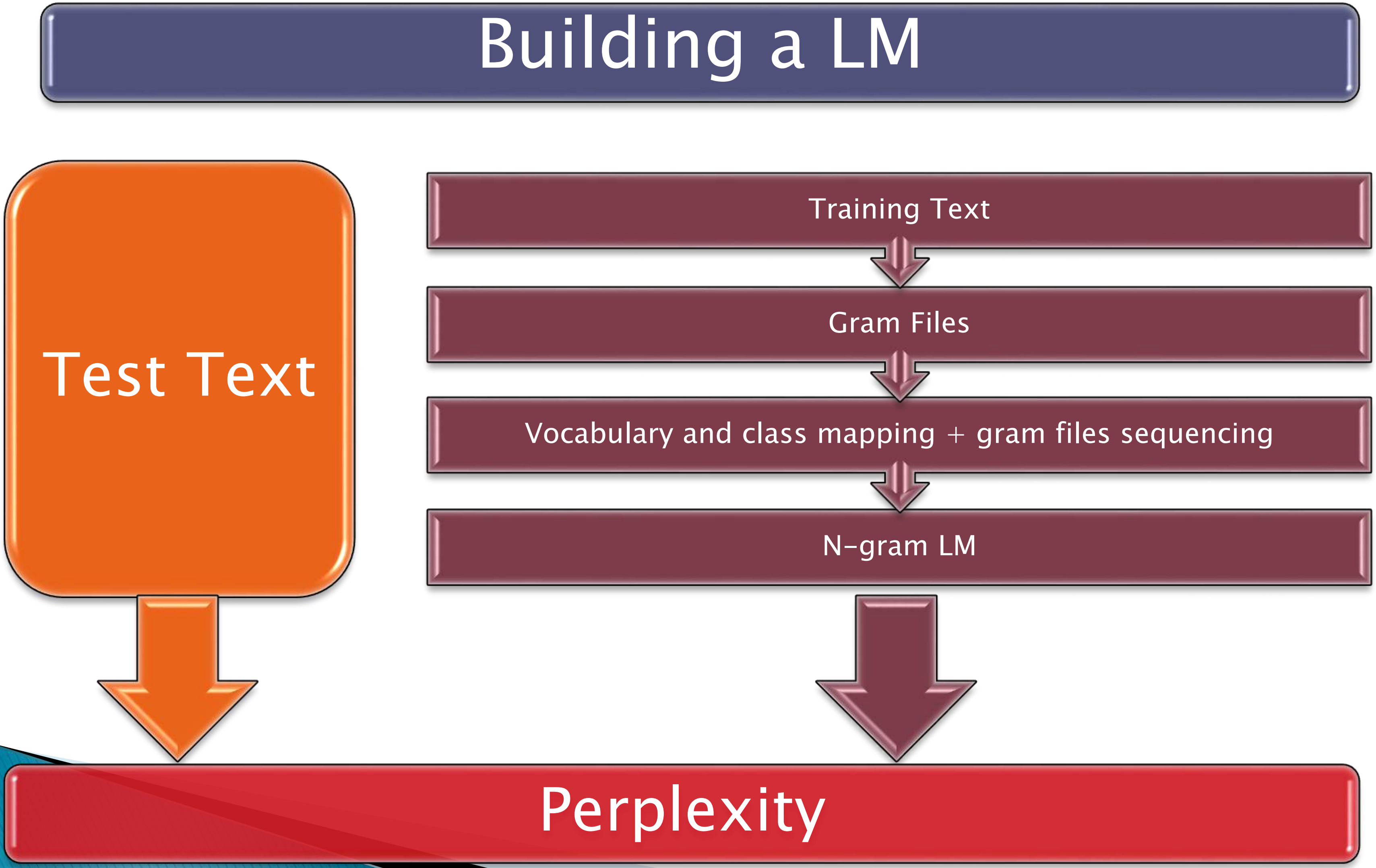
Alphabetically ordered

HTK Tool Kit

Dictionary Sample

```
</s>
<s>
A
A.
ABANDON
ABANDONED
ABBEY
ABDULLAH
ABE
```

HTK Tool Kit



Building a LM

Test Text

Training Text

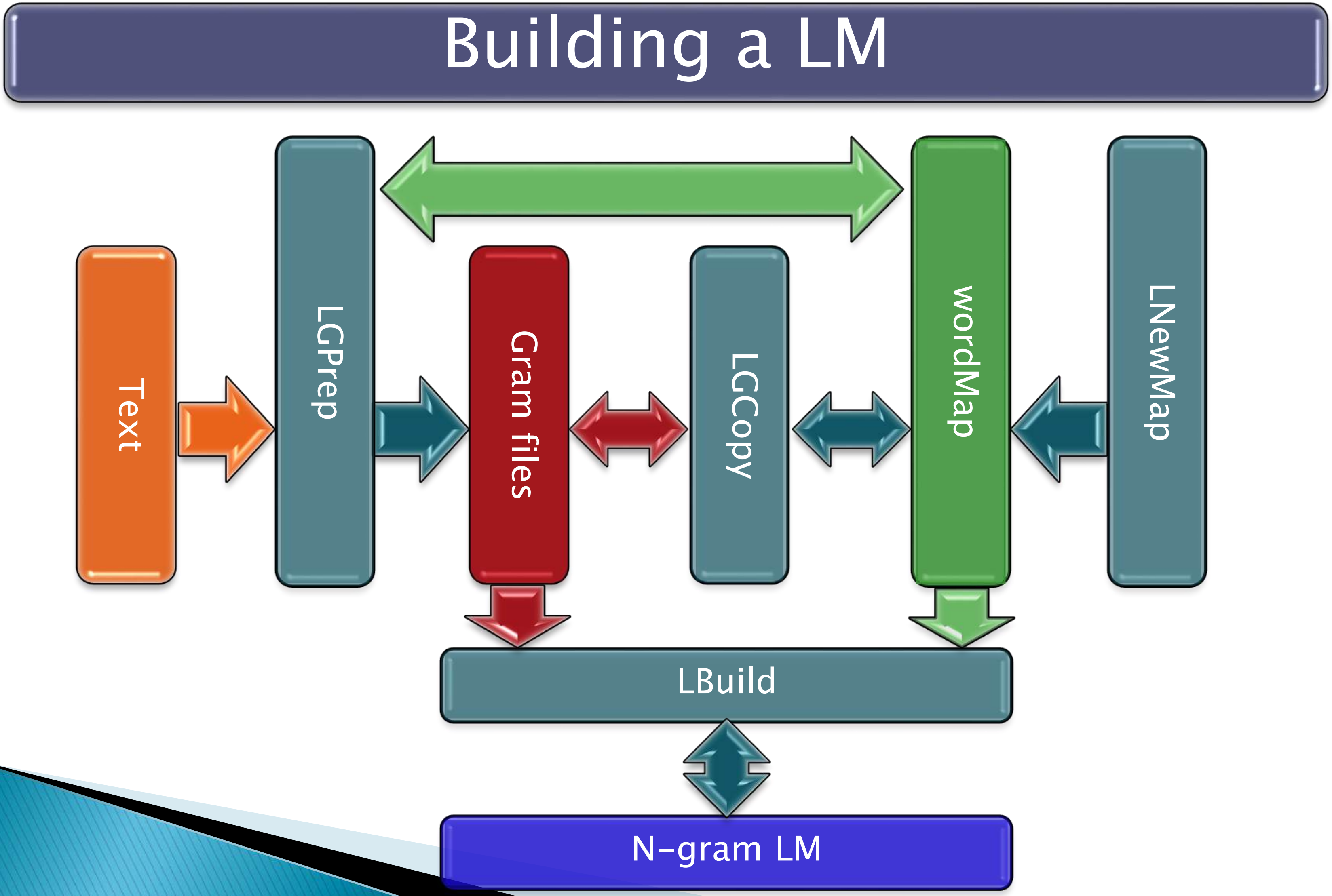
Gram Files

Vocabulary and class mapping + gram files sequencing

N-gram LM

Perplexity

HTK Tool Kit


Building a LM
Text
LGPrep
Gram files
LGCopy
wordMap
LNewMap
LBuild
N-gram LM

HTK Tool Kit

LNewMap

LNewMap [options] name mapfn

-e esc Change the contents of the EscMode header to esc. Default is RAW.

-f fld Add the field fld to the Fields header.

HTK Tool Kit

LNewMap

Example:

```
LNewMap –f WFC Holmes empty.wmap
```

```
  Name = Holmes
  SeqNo = 0
  Entries = 0
  EscMode = RAW
  Fields = ID,WFC
  \Words\
```

HTK Tool Kit

LGPrep

```
LGPrep [options] wordmap [textfile ...]
```

-a n Allow upto n new words in input texts (default 100000).

-b n Set the internal gram buffer size to n (default 2000000). LGPrep stores incoming n-grams in this buffer. When the buffer is full, the contents are sorted and written to an output gram file. Thus, the buffer size determines the amount of process memory that LGPrep will use and the size of the individual output gram files.

HTK Tool Kit

LGPrep cont’d

LGPrep [options] wordmap [textfile ...]

-d Directory in which to store the output gram files (default current directory).

-i n Set the index of the first gram file output to be n (default 0).

-n n Set the output n-gram size to n (default 3).

-r s Set the root name of the output gram files to s (default “gram”).

HTK Tool Kit

LGPrep cont’d

LGPrep [options] wordmap [textfile ...]

-s s Write the string s into the source field of the output gram files. This string should be a comment describing the text source.

-z Suppress gram file output. This option allows LGPrep to be used just to compute a word frequency map. It is also normally applied when applying edit rules to the input.

HTK Tool Kit

LGPrep cont’d

Example:

```
LGPrep –T 1 –a 100000 –b 2000000 –d holmes.0 –n 4
–s "Sherlock Holmes" empty.wmap
D:\train\abbey_grange.txt, D:\train\beryl_coronet.txt,...
```

HTK Tool Kit

LGPrep cont’d

WMAP file

```
Name = Holmes
SeqNo = 1
Entries = 18080
EscMode = RAW
Fields = ID,WFC
\Words\
<s>      65536   33669
IT       65537   8106
WAS      65538   7595
...
```

HTK Tool Kit

LGCopy

LGCopy [options] wordmap [mult] gramfiles

-b n Set the internal gram buffer size to n (default 2000000). LGPrep stores incoming n-grams in this buffer. When the buffer is full, the contents are sorted and written to an output gram file. Thus, the buffer size determines the amount of process memory that LGPrep will use and the size of the individual output gram files.

-d Directory in which to store the output gram files (default current directory).

HTK Tool Kit

LGCopy cont’d

LGCopy [options] wordmap [mult] gramfiles

-o n Output class mappings only. Normally all input *n*–grams are copied to the output, however, if a class map is specified, this options forces the tool to output only *n*–grams containing at least one class symbol.

HTK Tool Kit

LGCopy cont’d

Example:

```
LGCopy –T 1 –b 2000000 –d D:\holmes.1
D:\ holmes.0\wmap  D:\ holmes.0\gram.1 D:\
holmes.0\gram.2.....
```

HTK Tool Kit

LBuild

LBuild [options] wordmap outfile [mult] gramfile ..

-c n c Set cutoff for n-gram to c.

-n n Set final model order to n.

HTK Tool Kit

LBuild cont’d

Example:

```
LBuild –T 1 –c 2 1 –c 3 1 –n 3 D:\lm_5k\5k.wmap
D:\lm_5k\tg2-1_1 D:\holmes.1\data.1
D:\holmes.1\data.2…  D:\lm_5k\data.1 D:\lm_5k\data.12
```

HTK Tool Kit

LPlex

LPlex [options] langmodel labelFiles

-n n Perform a perplexity test using the n-gram component of the model. Multiple tests can be specified. By default the tool will use the maximum value of n available.

-t Text stream mode. If this option is set, the specified test files will be assumed to contain plain text.

HTK Tool Kit

LPlex cont’d

Example:

```
Lplex –n 3 –t D:\lm_5k\tg1_1 D:\test\red–headed_league.txt
```

Statistical Language Modeling using SRILM Toolkit

Presented by:
Kamal Eldin Mahmoud

AGENDA

- Introduction
- Basic SRILM Tools
  - ngram-count
  - ngram
  - ngram-merge
- Basic SRILM file format
  - ngram-format
  - nbest-format

AGENDA

Basic SRILM Scripts

- **Training-scripts**
- **lm-scripts**
- **ppl-scripts**

Introduction

- SRILM is a collection of C++ libraries, executable programs, and helper scripts.

- The toolkit supports creation and evaluation of a variety of language model types based on N-gram statistics.

- The main purpose of SRILM is to support language model estimation and evaluation.

- Since most LMs in SRILM are based on N-gram statistics, the tools to accomplish these two purposes are named ngram-count and ngram, respectively.

Introduction

- A standard LM (trigram with Good-Turing discounting and Katz backoff for smoothing) would be created by

```
ngram-count -text TRAINDATA -lm LM
```

- The resulting LM may then be evaluated on a test corpus using

```
ngram -lm LM -ppl TESTDATA -debug 0
```

Basic SRILM Tools

ngram-count

ngram-count generates and manipulates N-gram counts, and estimates N-gram language models from them.

Syntax:

***Ngram-count** [**-help**] option ...*

ngram-count options

Each filename argument can be an ASCII file, or a compressed file (name ending in .Z or .gz)

-help
Print option summary.
-version
Print version information.
-order n
Set the maximal order (length) of N-grams to count. This also determines the order of the estimated LM, if any. The default order is 3.
-memuse
Print memory usage statistics.

ngram-count options

-vocab *file*
Read a vocabulary from file.

-vocab-aliases *file*
Reads vocabulary alias definitions from file, consisting of lines of the form

alias word

This causes all tokens alias to be mapped to word.

-**write-vocab** *file*
-write-vocab-index *file*
Write the vocabulary built in the counting process to file.

ngram-count counting options

-tolower
Map all vocabulary to lowercase.

-text *textfile*
Generate N-gram counts from text file.

-no-sos
Disable the automatic insertion of start-of-sentence tokens in N-gram counting.
-no-eos
Disable the automatic insertion of end-of-sentence tokens in N-gram counting.

-read *countsfile*
Read N-gram counts from a file.

ngram-count counting options

-**read-google** *dir*
Read N-grams counts from an indexed directory structure rooted in dir, in a format developed by Google. The corresponding directory structure can be created using the script *make-google-ngrams* .

-**write** *file*
-**write-binary** *file*
-write-order n
-writen *file*
Write total counts to file.

-sort
Output counts in lexicographic order, as required for ngram-merge.

ngram-count lm options

-lm *lmfile*
-write-binary-lm
Estimate a backoff N-gram model from the total counts, and write it to *lmfile* .

-unk
Build an ``open vocabulary'' LM.

-map-unk *word*
Map out-of-vocabulary words to *word*.

ngram-count lm options

-cdiscount*n* *discount*
Use Ney's absolute discounting for N-grams of order *n*, using *discount* as the constant to subtract.

-wbdiscount*n*
Use Witten-Bell discounting for N-grams of order *n*.

-ndiscount*n*
Use Ristad's natural discounting law for N-grams of order *n*.

-addsmooth*n* *delta*
Smooth by adding *delta* to each N-gram count.

ngram-count lm options

-kndiscount*n*
Use Chen and Goodman's modified Kneser-Ney discounting for N-grams of order *n*.

-kn-counts-modified
Indicates that input counts have already been modified for Kneser-Ney smoothing.

-interpolate*n*
Causes the discounted N-gram probability estimates at the specified order *n* to be interpolated with lower-order estimates. Only Witten-Bell, absolute discounting, and (original or modified) Kneser-Ney smoothing currently support interpolation.

ngram

Ngram performs various operations with N-gram-based and related language models, including sentence scoring, and perplexity computation.

Syntax:

***ngram** [**-help**] option ...*

ngram options

-help
Print option summary.

-version
Print version information.

-order n
Set the maximal N-gram order to be used, by default 3.

-memuse
Print memory usage statistics for the LM.

ngram options

The following options determine the type of LM to be used.

-null

Use a `null' LM as the main model (one that gives probability 1 to all words).

-use-server *S*

Use a network LM server as the main model.

-lm *file*

Read the (main) N-gram model from *file*.

ngram options

-tagged
Interpret the LM as containing word/tag N-grams.

-skip
Interpret the LM as a ``skip'' N-gram model.

-classes *file*
Interpret the LM as an N-gram over word classes.

-factored
Use a factored N-gram model.

-unk
Indicates that the LM is an open-class LM.

ngram options

-ppl *textfile*
Compute sentence scores (log probabilities) and perplexities from the sentences in *textfile*.
The **-debug** option controls the level of detail printed.

-debug 0
Only summary statistics for the entire corpus are printed.

-debug 1
Statistics for individual sentences are printed.

ngram options

-debug 2
Probabilities for each word, plus LM-dependent details about backoff used etc., are printed.

-debug 3
Probabilities for all words are summed in each context, and the sum is printed.

ngram options

-nbest *file*
Read an N-best list in nbest-format and rerank the hypotheses using the specified LM. The reordered N-best list is written to stdout.

-nbest-files *filelist*
Process multiple N-best lists whose filenames are listed in *filelist*.

-write-nbest-dir *dir*
Deposit rescored N-best lists into directory *dir*, using filenames derived from the input ones.

ngram options

-decipher-nbest
Output rescored N-best lists in Decipher 1.0 format, rather than SRILM format.

-no-reorder
Output rescored N-best lists without sorting the hypotheses by their new combined scores.

-max-nbest *n*
Limits the number of hypotheses read from an N-best list.

ngram options

-no-sos
Disable the automatic insertion of start-of-sentence tokens for sentence probability computation.

-no-eos
Disable the automatic insertion of end-of-sentence tokens for sentence probability computation.

ngram-merge

ngram-merge reads two or more lexicographically sorted N-gram count files and outputs the merged, sorted counts.

Syntax:

ngram-merge [-help] [-write outfile] [-float-counts] \ [--] infile1 infile2 ...

Ngram-merge options

-write *outfile*
Write merged counts to *outfile*.

-float-counts
Process counts as floating point numbers.

--
Indicates the end of options, in case the first input filename begins with ``-''.

Basic SRILM file format

ngram-format

ngram-format File format for ARPA backoff N-gram models

```
\data\
ngram 1=n1
ngram 2=n2.
..
ngram N=nN
\1-grams:
p           w                       [bow]
...\
2-grams:
p           w1 w2                   [bow]
...
\N-grams:
p           w1 ... wN
...
\end\
```

nbest-format

SRILM currently understands three different formats for lists of N-best hypotheses for rescoring or 1-best hypothesis extraction. The first two formats originated in the SRI Decipher(TM) recognition system, the third format is particular to SRILM.

The first format consists of the header

NBestList1.0

followed by one or more lines of the form

(score) w1 w2 w3 ...

where *score* is a composite acoustic/language model score from the recognizer, on the bytelog scale.

nbest-format

The second Decipher(TM) format is an extension of the first format that encodes word-level scores and time alignments. It is marked by a header of the form

NBestList2.0

The hypotheses are in the format

(*score*) *w1* (st: *st1* et: *et1* g: *g1* a: *a1*) *w2* ...

where words are followed by start and end times, language model and acoustic scores (bytelog-scaled), respectively.

nbest-format

The third format understood by SRILM lists hypotheses in the format

ascore lscore nwords w1 w2 w3 ...

where the first three columns contain the acoustic model log probability, the language model log probability, and the number of words in the hypothesis string, respectively. All scores are logarithms base 10.

Basic SRILM Scripts

Training-scripts

These scripts perform convenience tasks associated with the training of language models.

get-gt-counts

Syntax

get-gt-counts max=*K* **out=***name* [*counts* ...] **>** *gtcounts*

Computes the counts-of-counts statistics needed in Good-Turing smoothing. The frequencies of counts up to *K* are computed (default is 10). The results are stored in a series of files with root *name*, ***name*****.gt1counts**,..., ***name*****.gt*****N*****counts**.

Training-scripts

make-gt-discounts

Santax:
make-gt-discounts min=*min* max=*max gtcounts*
Takes one of the output files of get-gt-counts and computes the corresponding Good-Turing discounting factors. The output can then be passed to **ngram-count** via the **-gt*n*** options to control the smoothing during model estimation.

Training-scripts

make-abs-discount

Syntax

make-abs-discount *gtcounts*

Computes the absolute discounting constant needed for the **ngram-count -cdiscount***n* options. Input is one of the files produced by **get-gt-counts**.

Training-scripts

make-kn-discount

Syntax

make-kn-discounts min=*min gtcounts*

Computes the discounting constants used by the modified Kneser-Ney smoothing method. Input is one of the files produced by **get-gt-counts**.

Training-scripts

make-batch-counts

Syntax

make-batch-counts *file-list* \ [*batch-size* [*filter* [*count-dir* [*options* ...]]]]

Performs the first stage in the construction of very large N-gram count files. *file-list* is a list of input text files. Lines starting with a `#' character are ignored. These files will be grouped into batches of size *batch-size* (default 10). The N-gram count files are left in directory *count-dir* (``counts'' by default), where they can be found by a subsequent run of **merge-batch-counts**.

Training-scripts

merge-batch-counts

Syntax

merge-batch-counts *count-dir* [*file-list*|*start-iter*]

Completes the construction of large count files. Optionally, a *file-list* of count files to combine can be specified. A number as second argument restarts the merging process at iteration *start-iter*.

Training-scripts

make-google-ngrams

Syntax

make-google-ngrams [**dir=***DIR*] [**per_file=***N*] [**gzip=0**] \ [**yahoo=1**] [*counts-file* ...]

Takes a sorted count file as input and creates an indexed directory structure, in a format developed by Google to store very large N-gram collections. Optional arguments specify the output directory *dir* and the size *N* of individual N-gram files (default is 10 million N-grams per file). The **gzip=0** option writes plain. The **yahoo=1** option may be used to read N-gram count files in Yahoo-GALE format.

Training-scripts

tolower-ngram-counts

Syntax

tolower-ngram-counts [*counts-file* ...]

Maps an N-gram counts file to all-lowercase. No merging of N-grams that become identical in the process is done.

Training-scripts

reverse-ngram-counts

Syntax

reverse-ngram-counts [*counts-file* ...]

Reverses the word order of N-grams in a counts file or stream.

reverse-text

Syntax

reverse-text [*textfile* ...]

Reverses the word order in text files, line-by-line.

Training-scripts

compute-oov-rate

Syntax

compute-oov-rate *vocab* [*counts* ...]

Determines the out-of-vocabulary rate of a corpus from its unigram *counts* and a target vocabulary list in *vocab*.

lm-scripts

add-dummy-bows

Syntax

add-dummy-bows [*lm-file*] **>** *new-lm-file*

Adds dummy backoff weights to N-grams, even where they are not required, to satisfy some broken software that expects backoff weights on all N-grams (except those of highest order).

lm-scripts

change-lm-vocab

Syntax

change-lm-vocab -vocab *vocab* **-lm** *lm-file* **-write-lm** *new-lm-file* \ [**-tolower**] [**-subset**] [*ngram-options* ...]

Modifies the vocabulary of an LM to be that in *vocab*. Any N-grams containing OOV words are removed, new words receive a unigram probability, and the model is renormalized. The **-tolower** option causes case distinctions to be ignored. **-subset** only removes words from the LM vocabulary, without adding any.

lm-scripts

make-lm-subset

Syntax

make-lm-subset *count-file*|**-** [lm-file |**-**] **>** *new-lm-file*

Forms a new LM containing only the N-grams found in the *count-file*. The result still needs to be renormalized with **ngram -renorm** .

lm-scripts

get-unigram-probs

Syntax

get-unigram-probs [**linear=1**] [*lm-file*]

Extracts the unigram probabilities in a simple table format from a backoff language model. The **linear=1** option causes probabilities to be output on a linear (instead of log) scale.

ppl-scripts

These scripts process the output of the ngram option **-ppl** to extract various useful information.

add-ppls

Syntax

add-ppls [*ppl-file* ...]

Takes several ppl output files and computes an aggregate perplexity and corpus statistics.

ppl-scripts

subtract-ppls

Syntax

subtract-ppls *ppl-file1* [*ppl-file2* ...]

Similarly computes an aggregate perplexity by removing the statistics of zero or more *ppl-file2* from those in *ppl-file1*.

ppl-scripts

compare-ppls

Syntax

compare-ppls [**mindelta=***D*] *ppl-file1 ppl-file2*

Tallies the number of words for which two language models produce the same, higher, or lower probabilities. The input files should be **ngram -debug 2 -ppl** output for the two models on the same test set. The parameter *D* is the minimum absolute difference for two log probabilities to be considered different.

ppl-scripts

compute-best-mix

Syntax

compute-best-mix [**lambda='***l1 l2* ...**'**] **[precision=***P*] \ *ppl-file1* [*ppl-file2* ...]

Takes the output of several **ngram -debug 2 –ppl** runs on the same test set and computes the optimal interpolation weights for the corresponding models. Initial weights may be specified as *l1 l2 ...*. The computation is iterative and stops when the interpolation weights change by less than *P* (default 0.001).

ppl-scripts

compute-best-sentence-mix

Syntax

compute-best-sentence-mix [**lambda='***l1 l2* ...**'**] [**precision=***P*] \ *ppl-file1* [*ppl-file2* ...]

similarly optimizes the weights for sentence-level interpolation of LMs. It requires input files generated by **ngram -debug 1 -ppl**.

THANK YOU ☺

Introduction to language modeling

Dr. Mohamed Waleed Fakhr

AAST

Language Engineering Conference

22 December 2009

Topics

- Why a language model?
- Probability in brief
- Word prediction task
- Language modeling (N-grams)
  - N-gram intro.
  - Model evaluation
  - Smoothing
- Other modeling approaches

Why a language model?

- Suppose a machine is required to translate: “The human Race”.
- The word “Race” has at least 2 meanings, which one to choose?
- Obviously, the choice depends on the “history” or the “context” preceding the word “Race”. E.g., “the human race” versus “the dogs race”.
- A statistical language model can solve this ambiguity by giving higher probability to the correct meaning.

Probability in brief

- Joint probability: P(A,B) is the probability that events A and B are simultaneously true (observed together).
- Conditional probability: P(A|B): is the probability that A is true given that B is true (observed).

Relation between joint and conditional probabilities

- **BAYES RULE:**

P(A|B) = P(A,B)/P(B)

P(B|A) = P(A,B)/P(A)

Or;

P(A,B)= P(A).P(B|A) = P(B).P(A|B)

Chain Rule

- The joint probability:
  $P(A,B,C,D)=P(A).P(B|A).P(C|A,B).P(D|A,B,C)$
- This will lend itself to the language modeling paradigm as we will be concerned by the joint probability of the occurrence of a word-sequence $(W_1,W_2,W_3,\dots.W_n)$:

  $P(W_1,W_2,W_3,\dots.W_n)$

  which will be put in terms of conditional probability terms:
- $P(W1).P(W2|W1).P(W3|W1,W2)\dots\dots\dots$

  (More of this later)

Language Modeling?

In the narrow sense, statistical language modeling is concerned by estimating the joint probability of a word sequence . $P(W_1,W_2,W_3,\ldots.W_n)$

This is always converted into conditional probs: P(Next Word | History)

e.g., P(W3|W1,W2)

i.e., can we predict the next word given the previous words that have been observed?

In other words, if we have a History, find the Next-Word that gives the highest prob.

Word Prediction

- Guess the next word...

  ... It is too late I want to go ???

  ... I notice three guys standing on the ???

- There are many sources of knowledge that can be used to inform this task, including arbitrary world knowledge and deeper history (It is too late)
- But it turns out that we can do pretty well by simply looking at the preceding words and keeping track of some fairly simple counts.

Word Prediction

- We can formalize this task using what are called *N*-gram models.
- *N*-grams are token sequences of length *N*.
- Our 2nd example contains the following 2-grams (Bigrams)
  - (I notice), (notice three), (three guys), (guys standing), (standing on), (on the)
- Given knowledge of counts of N-grams such as these, we can guess likely next words in a sequence.

N-Gram Models

- More formally, we can use knowledge of the counts of *N*-grams to assess the conditional probability of candidate words as the next word in a sequence.
- In doing so, we actually use them to assess the joint probability of an entire sequence of words. (chain rule).

Applications

- It turns out that being able to predict the next word (or any linguistic unit) in a sequence is an extremely useful thing to be able to do.
- As we’ll see, it lies at the core of the following applications
  - Automatic speech recognition
  - Handwriting and character recognition
  - Spelling correction
  - Machine translation
  - Information retrieval
  - And many more.

ASR

$$\underset{wordsequence}{\arg\max}\, P(wordsequence \mid acoustics) =$$

$$\underset{wordsequence}{\arg\max}\, \frac{P(acoustics \mid wordsequence) \times P(wordsequence)}{P(acoustics)}$$

$$\underset{wordsequence}{\arg\max}\, P(acoustics \mid wordsequence) \times P(wordsequence)$$

Source Channel Model for Machine Translation

$$\underset{wordsequence}{\arg\max}\, P(wordsequence \mid acoustics) =$$

$$\underset{wordsequence}{\arg\max}\, \frac{P(acoustics \mid wordsequence)' \;\; P(wordsequence)}{P(acoustics)}$$

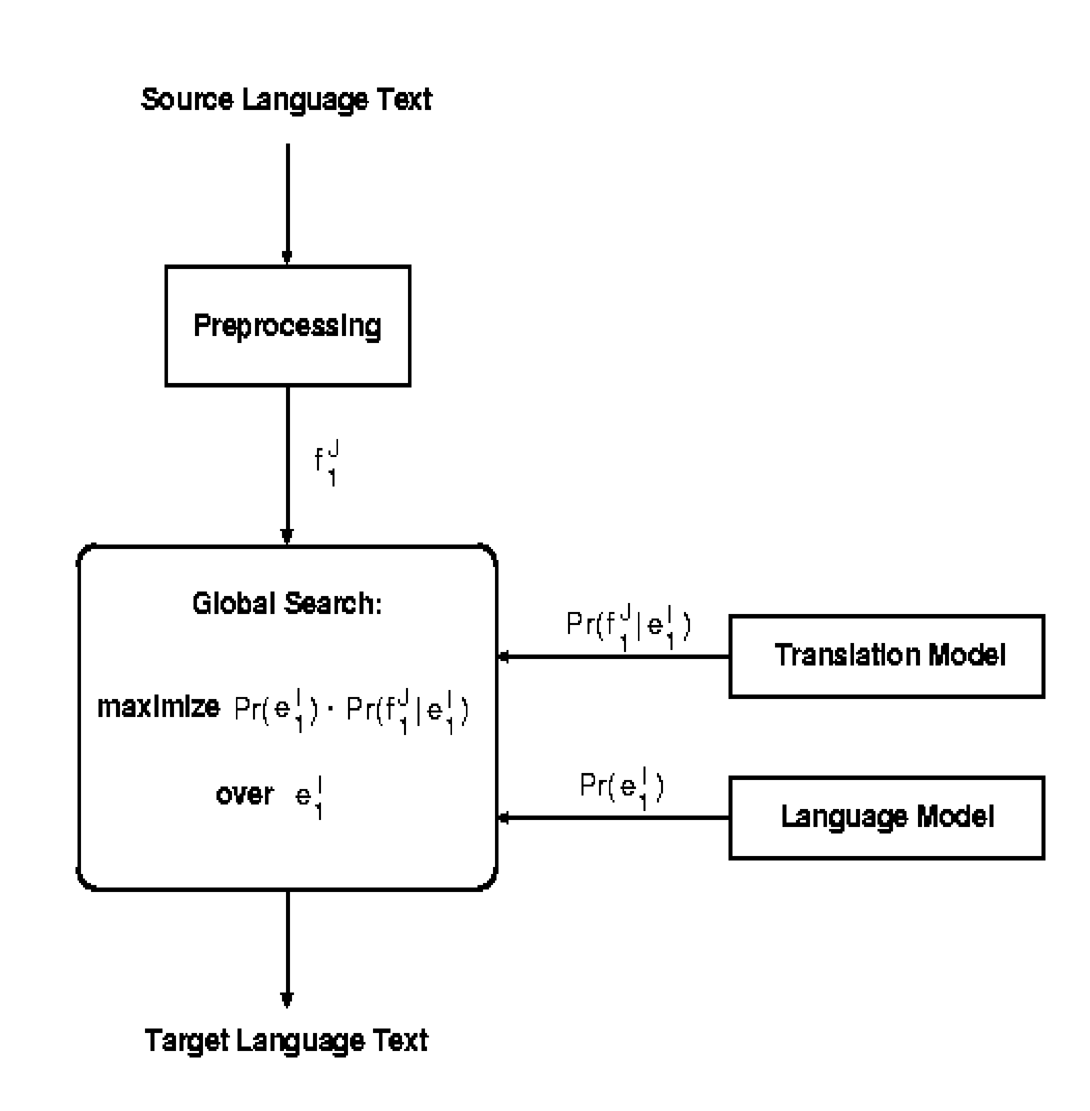
$$\underset{wordsequence}{\arg\max}\, P(acoustics \mid wordsequence)' \;\; P(wordsequence)$$

$$\underset{wordsequence}{\arg\max}\, P(english \mid french) =$$

$$\underset{wordsequence}{\arg\max}\, \frac{P(french \mid english)' \;\; P(english)}{P(french)}$$

$$\underset{wordsequence}{\arg\max}\, P(french \mid english)' \;\; P(english)$$

SMT Architecture



Based on Bayes´ Decision Rule:

$$\hat{e} = \text{argmax}\{ p(e \mid f) \}$$
$$= \text{argmax}\{ p(e)\, p(f \mid e) \}$$

Counting

- Simple counting lies at the core of any probabilistic approach. So let’s first take a look at what we’re counting.
  - *He stepped out into the hall, was delighted to encounter a water brother.*
    - 13 tokens, 15 if we include “,” and “.” as separate tokens.
    - Assuming we include the comma and period, how many bigrams are there?

Counting

- Not always that simple
  - *I do uh main- mainly business data processing*
- Spoken language poses various challenges.
  - Should we count “uh” and other fillers as tokens?
  - What about the repetition of “mainly”? Should such do-overs count twice or just once?
  - The answers depend on the application.
    - If we’re focusing on something like ASR to support indexing for search, then “uh” isn’t helpful (it’s not likely to occur as a query).
    - But filled pauses are very useful in dialog management, so we might want them there.

Counting: Types and Tokens

- How about
  - *They picnicked by the pool, then lay back on the grass and looked at the stars.*
    - 18 tokens (again counting punctuation)
- But we might also note that “*the*” is used 3 times, so there are only 16 unique types (as opposed to tokens).
- In going forward, we’ll have occasion to focus on counting both types and tokens of both words and *N*-grams.

Counting: Wordforms

- Should “cats” and “cat” count as the same when we’re counting?
- How about “geese” and “goose”?
- Some terminology:
  - Lemma: a set of lexical forms having the same stem, major part of speech, and rough word sense: (car, cars, automobile)
  - Wordform: fully inflected surface form
- Again, we’ll have occasion to count both lemmas, morphemes, and wordforms

Counting: Corpora

- So what happens when we look at large bodies of text instead of single utterances?
- Brown et al (1992) large corpus of English text
  - 583 million wordform tokens
  - 293,181 wordform types
- Google
  - Crawl of 1,024,908,267,229 English tokens
  - 13,588,391 wordform types
    - That seems like a lot of types… After all, even large dictionaries of English have only around 500
      here?

•Numbers
•Misspellings
•Names
•Acronyms
•etc

Language Modeling

- Back to word prediction
- We can model the word prediction task as the ability to assess the conditional probability of a word given the previous words in the sequence
  - $P(w_n|w_1,w_2\ldots w_{n-1})$
- We'll call a statistical model that can assess this a *Language Model*

Language Modeling

- How might we go about calculating such a conditional probability?
  - One way is to use the definition of conditional probabilities and look for counts. So to get
  - P(*the* | *its water is so transparent that*)
- By definition that’s

$$\frac{\text{Count(its water is so transparent that the)}}{\text{Count(its water is so transparent that)}}$$

We can get each of those counts in a large corpus.

Very Easy Estimate

- According to Google those counts are 5/9.
  - Unfortunately... 2 of those were to these slides... So maybe it’s really   3/7
  - In any case, that’s not terribly convincing due to the small numbers involved.

Language Modeling

- Unfortunately, for most sequences and for most text collections we won’t get good estimates from this method.
  - What we’re likely to get is 0. Or worse 0/0.
- Clearly, we’ll have to be a little more clever.
  - Let’s use the chain rule of probability
  - And a particularly useful independence assumption.

The Chain Rule

- Recall the definition of conditional probabilities

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

- Rewriting:

$$P(A,B) = P(B).P(A|B)$$

- For sequences...
  - *P(A,B,C,D) = P(A)P(B|A)P(C|A,B)P(D|A,B,C)*
- In general
  - $P(x_1,x_2,x_3,\ldots x_n) =$
    $P(x_1)P(x_2|x_1)P(x_3|x_1,x_2)\ldots P(x_n|x_1\ldots x_{n-1})$

The Chain Rule

$$\begin{aligned} P(w_1^n) &= P(w_1)P(w_2|w_1)P(w_3|w_1^2)\ldots P(w_n|w_1^{n-1}) \\ &= \prod_{k=1}^{n} P(w_k|w_1^{k-1}) \end{aligned}$$

P(its water was so transparent)=

P(its)*

P(water|its)*

P(was|its water)*

P(so|its water was)*

P(transparent|its water was so)

Unfortunately

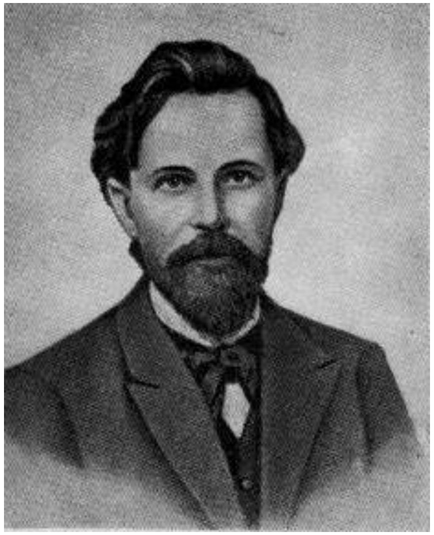
- There are still a lot of possible sentences
- In general, we'll never be able to get enough data to compute the statistics for those longer prefixes
  - Same problem we had for the strings themselves

Independence Assumption

- Make the simplifying assumption
  - P(lizard|the,other,day,I,was,walking,along,and ,saw,a) = P(lizard|a)
- Or maybe
  - P(lizard|the,other,day,I,was,walking,along,and ,saw,a) = P(lizard|saw,a)
- That is, the probability in question is independent of its earlier history.

Independence Assumption

- This particular kind of independence assumption is called a *Markov assumption* after the Russian mathematician Andrei Markov.



Markov Assumption

So for each component in the product replace with the approximation (assuming a prefix of N)

$$P(w_n \mid w_1^{n-1}) \approx P(w_n \mid w_{n-N+1}^{n-1})$$

Bigram version

$$P(w_n \mid w_1^{n-1}) \approx P(w_n \mid w_{n-1})$$

Estimating Bigram Probabilities

The picture can't be displayed.

- The Maximum Likelihood Estimate (MLE):

$$P(w_i \mid w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})}$$

Normalization

- For N-gram models to be probabilistically correct they have to obey prob. Normalization constraints:

$$\sum_{over-all-j} P(W_j \,|\, Context_i) = 1$$

- The sum over all words for the same context (history) must be 1.
- The context may be one word (bigram) or two words (trigram) or more.

An Example: bigrams

- <s> I am Sam </s>
- <s> Sam I am </s>
- <s> I do not like green eggs and ham </s>

$$P(\texttt{I} \mid \texttt{<s>}) = \frac{2}{3} = .67 \qquad P(\texttt{Sam} \mid \texttt{<s>}) = \frac{1}{3} = .33 \qquad P(\texttt{am} \mid \texttt{I}) = \frac{2}{3} = .67$$

$$P(\texttt{</s>} \mid \texttt{Sam}) = \frac{1}{2} = 0.5 \qquad P(\texttt{Sam} \mid \texttt{am}) = \frac{1}{2} = .5 \qquad P(\texttt{do} \mid \texttt{I}) = \frac{1}{3} = .33$$

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

estimates depend on the corpus

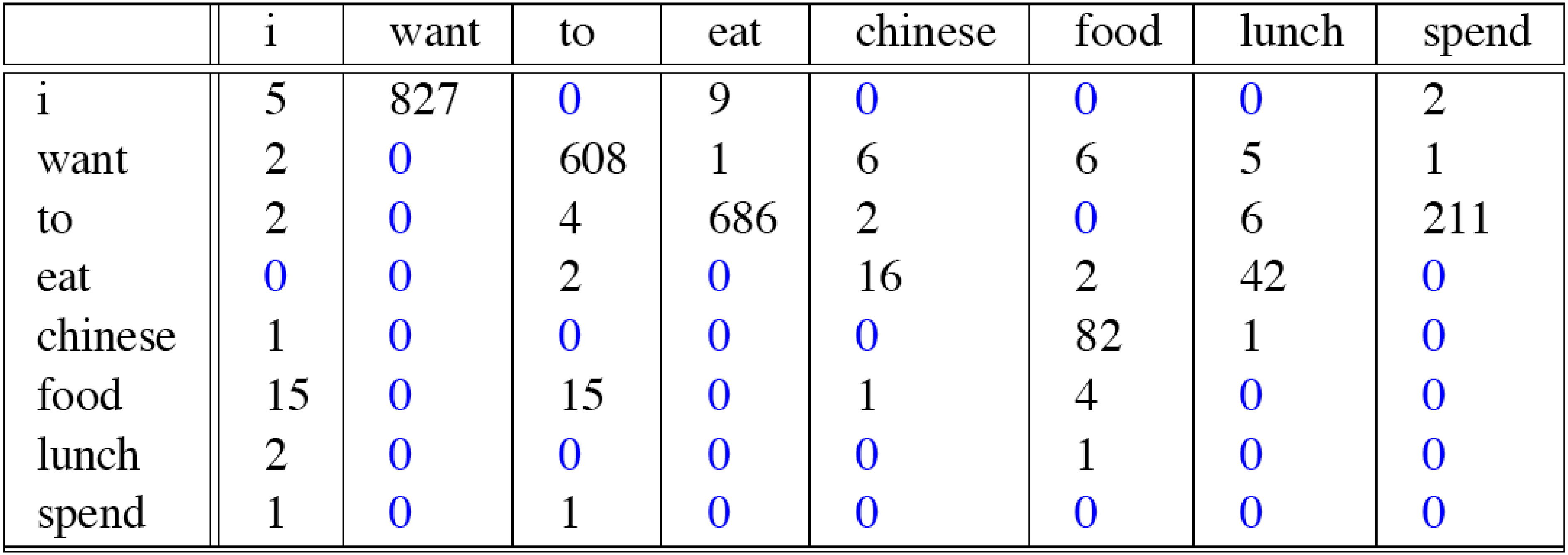
- The maximum likelihood estimate of some parameter of a model M from a training set T
  - Is the estimate that maximizes the likelihood of the training set T given the model M
- Suppose the word Chinese occurs 400 times in a corpus of a million words (Brown corpus)
- What is the probability that a random word from some other text from the same distribution will be “Chinese”
- MLE estimate is 400/1000000 = .004
  - This may be a bad estimate for some other corpus

Berkeley Restaurant Project Sentences examples

- *can you tell me about any good cantonese restaurants close by*
- *mid priced thai food is what i’m looking for*
- *tell me about chez panisse*
- *can you give me a listing of the kinds of food that are available*
- *i’m looking for a good place to eat breakfast*
- *when is caffe venezia open during the day*

Bigram Counts

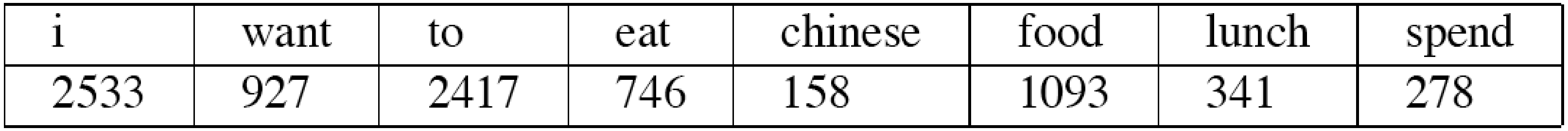
- Out of 9222 sentences
  - – e.g. “I want” occurred 827 times



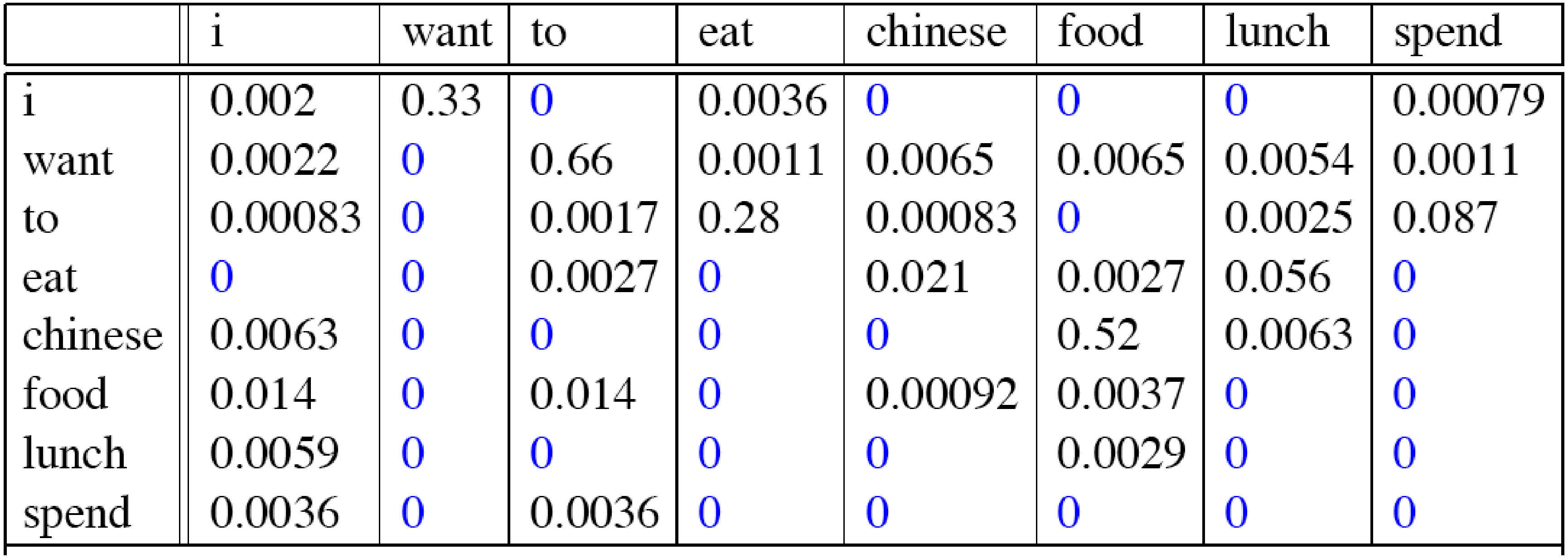
	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Bigram Probabilities

- Divide bigram counts by prefix unigram counts to get probabilities.



i	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278



	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

examples

- P(Want | I) = C(I Want) / C(I)

 = 827/2533 = 0.33

P(Food | Chinese) = C(Chinese Food) / C(Chinese)

= 82/158 = 0.52

Bigram Estimates of Sentence Probabilities

- P(<s> I want english food </s>) =

  P(i|<s>)*

  P(want|I)*

  P(english|want)*

  P(food|english)*

  P(</s>|food)*

  =.000031

Evaluation

- How do we know if our models are any good?
  - And in particular, how do we know if one model is better than another?

Evaluation

- Standard method
  - Train parameters of our model on a **training set**.
  - Look at the models performance on some new data
    - This is exactly what happens in the real world; we want to know how our model performs on data we haven’t seen
  - So use a **test set**. A dataset which is different than our training set, but is drawn from the same source
  - Then we need an **evaluation metric** to tell us how well our model is doing on the test set.
    - One such metric is **perplexity**

Unknown Words

- But once we start looking at test data, we'll run into words that we haven't seen before (pretty much regardless of how much training data you have) (zero unigrams)
- With an ***Open Vocabulary* task**
  - Create an unknown word token <UNK>
  - Training of <UNK> probabilities
    - Create a fixed lexicon L, of size V
      - From a dictionary or
      - A subset of terms from the training set
    - At text normalization phase, any training word not in L changed to **<UNK>**
    - Now we count that like a normal word
  - At test time
    - Use <**UNK>** counts for any word not in training

Perplexity

- Perplexity is the probability of the test set (assigned by the language model), normalized by the number of words:

$$\begin{aligned}\mathrm{PP}(W) &= P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} \\ &= \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N}})\end{aligned}$$

- Chain rule:

$$\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_1 \ldots w_{i-1})}}$$

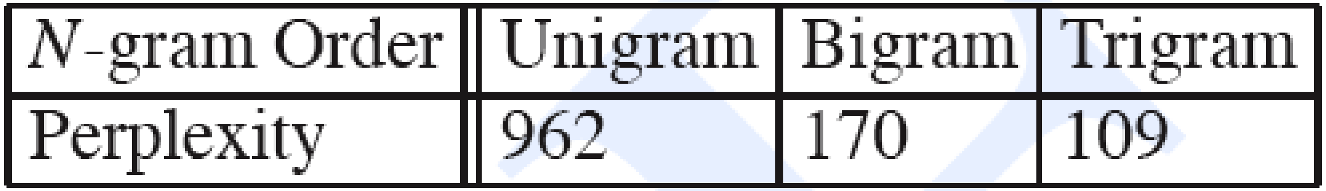
- For bigrams:

$$\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-1})}}$$

- Minimizing perplexity is the same as maximizing probability
  - **The best language model is one that best predicts an unseen test set**

Lower perplexity means a better model

- Training 38 million words, test 1.5 million words, WSJ (Wall-Street Journal)



N-gram Order	Unigram	Bigram	Trigram
Perplexity	962	170	109

Evaluating *N*-Gram Models

- Best evaluation for a language model
  - Put model *A* into an application
    - For example, a speech recognizer
  - Evaluate the performance of the application with model *A*
  - Put model *B* into the application and evaluate
  - Compare performance of the application with the two models
  - ***Extrinsic evaluation***

Difficulty of extrinsic (in-vivo) evaluation of N-gram models

- Extrinsic evaluation
  - This is really time-consuming
  - Can take days to run an experiment
- So
  - To evaluate N-grams we often use an **intrinsic** evaluation, an approximation called **perplexity**
  - But perplexity is a poor approximation unless the test data looks **similar to** the training data
  - So is **generally only useful in pilot experiments**
  - **But still, there is nothing like the real experiment!**

N-gram Zero Counts

- For the English language,
  - V^2= 844 million possible bigrams...
  - So, for a medium size training data, e.g., Shakespeare novels, 300,000 bigrams were found Thus, 99.96% of the possible bigrams were never seen (have zero entries in the table)
  - Does that mean that any ***test*** sentence that contains one of those bigrams should have a probability of 0?

N-gram Zero Counts

- Some of those zeros are really zeros...
  - Things that really can’t or shouldn’t happen.
- On the other hand, some of them are just rare events.
  - If the training corpus had been a little bigger they would have had a count (probably a count of 1).
- Zipf’s Law (long tail phenomenon):
  - A small number of events occur with high frequency
  - A large number of events occur with low frequency
  - You can quickly collect statistics on the high frequency events
  - You might have to wait an arbitrarily long time to get valid statistics on low frequency events
- Result:
  - Our estimates are sparse ! We have no counts at all for the vast bulk of things we want to estimate!
- Answer:
  - ***Estimate*** the likelihood of unseen (zero count) N-grams!
  - **N-gram Smoothing techniques**

Laplace Smoothing

- Also called add-one smoothing
- Just add one to all the counts!
- This adds extra *V* observations
  (*V* is vocab. Size)

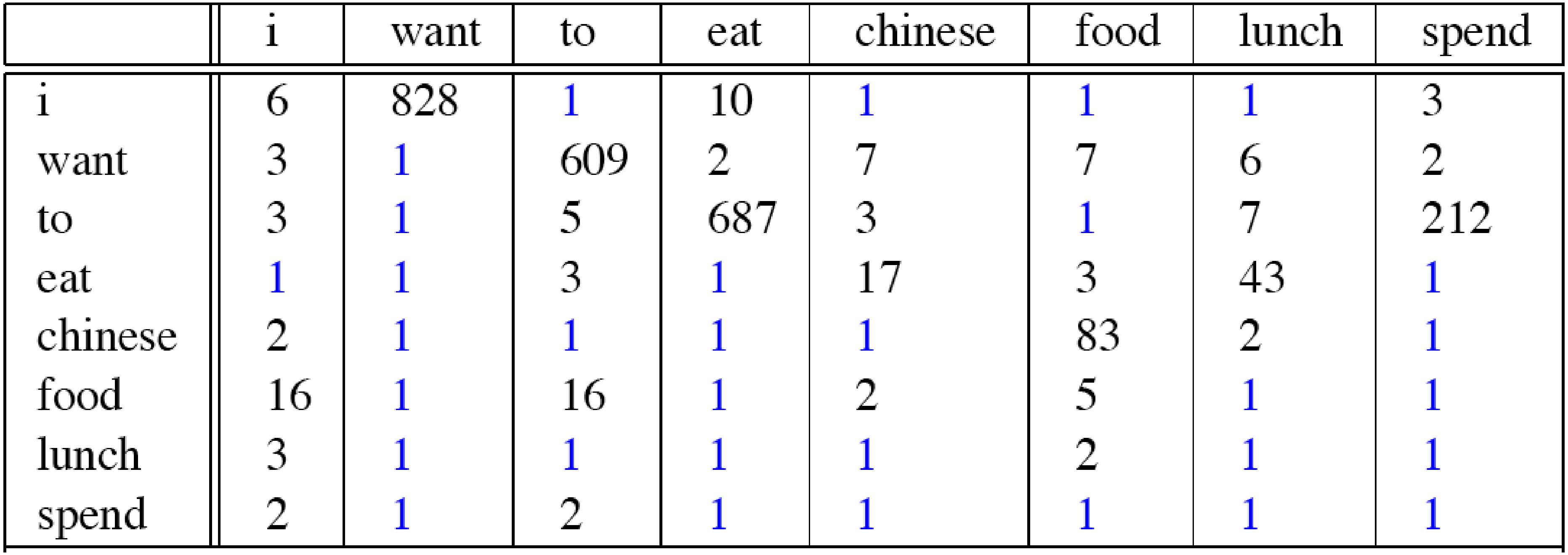
- MLE estimate: $P(w_i) = \frac{c_i}{N}$

- Laplace estimate: $P_{\text{Laplace}}(w_i) = \frac{c_i + 1}{N + V}$ $\boldsymbol{P_{Laplace}} = \frac{1}{N}\frac{(ci+1).N}{(N+V)}$

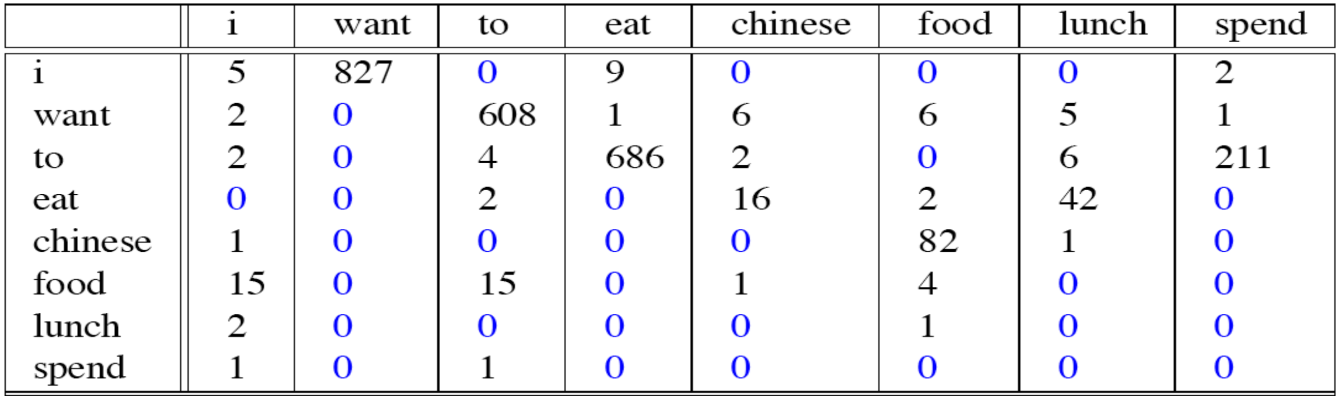
- Reconstructed counts:

(making the volume N again) $c_i^* = (c_i + 1)\frac{N}{N + V}$

Laplace-Smoothed Bigram Counts



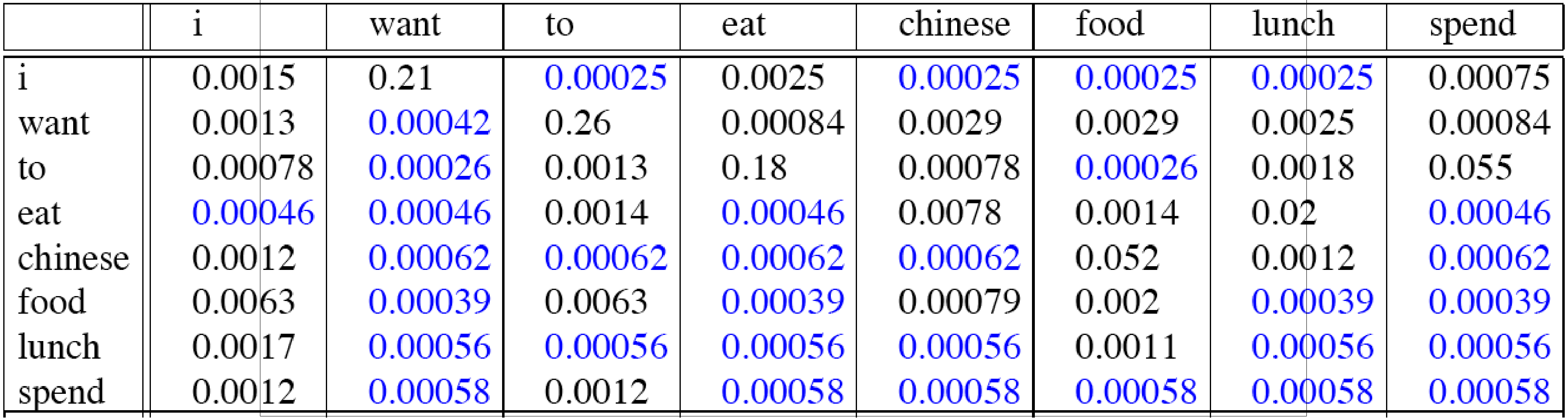
	i	want	to	eat	chinese	food	lunch	spend
i	6	828	1	10	1	1	1	3
want	3	1	609	2	7	7	6	2
to	3	1	5	687	3	1	7	212
eat	1	1	3	1	17	3	43	1
chinese	2	1	1	1	1	83	2	1
food	16	1	16	1	2	5	1	1
lunch	3	1	1	1	1	2	1	1
spend	2	1	2	1	1	1	1	1



	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

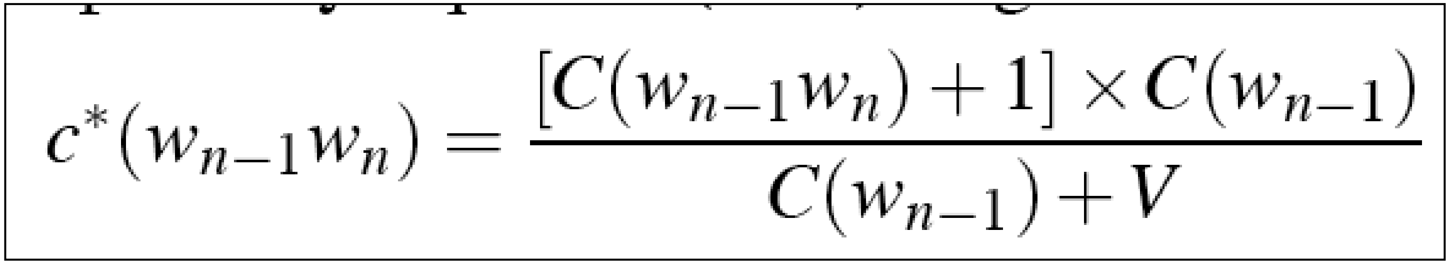
Laplace-Smoothed Bigram Probabilities

$$P^*(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)+1}{C(w_{n-1})+V}$$

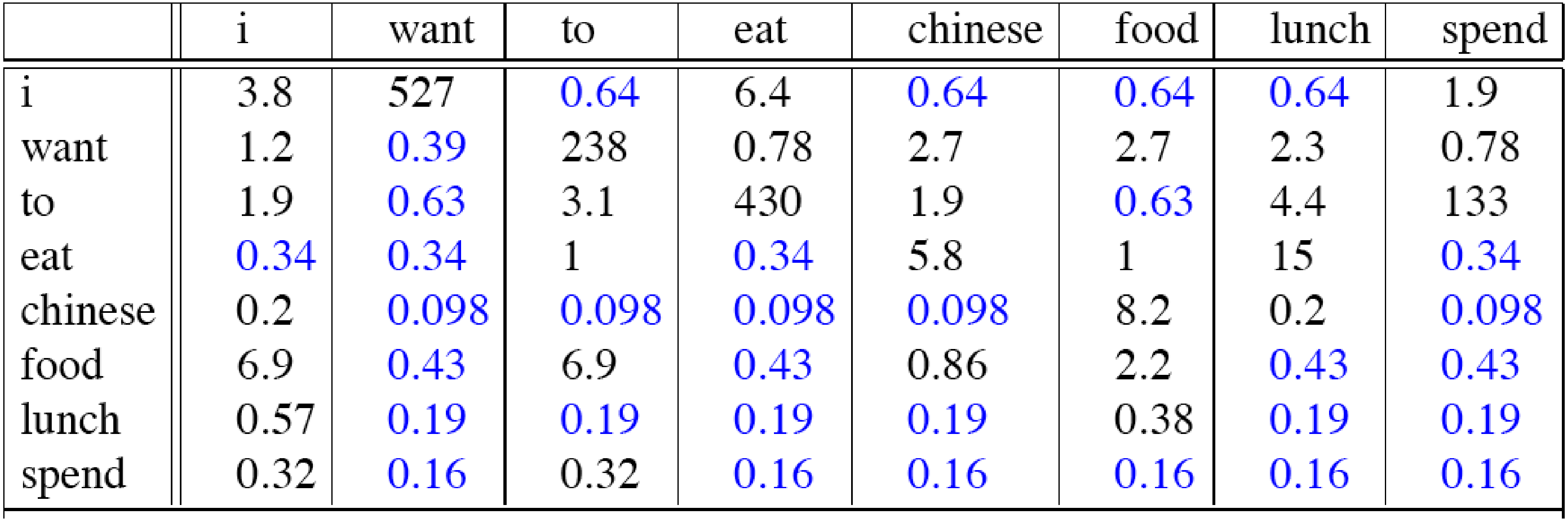


	i	want	to	eat	chinese	food	lunch	spend
i	0.0015	0.21	0.00025	0.0025	0.00025	0.00025	0.00025	0.00075
want	0.0013	0.00042	0.26	0.00084	0.0029	0.0029	0.0025	0.00084
to	0.00078	0.00026	0.0013	0.18	0.00078	0.00026	0.0018	0.055
eat	0.00046	0.00046	0.0014	0.00046	0.0078	0.0014	0.02	0.00046
chinese	0.0012	0.00062	0.00062	0.00062	0.00062	0.052	0.0012	0.00062
food	0.0063	0.00039	0.0063	0.00039	0.00079	0.002	0.00039	0.00039
lunch	0.0017	0.00056	0.00056	0.00056	0.00056	0.0011	0.00056	0.00056
spend	0.0012	0.00058	0.0012	0.00058	0.00058	0.00058	0.00058	0.00058

Reconstructed Counts



$$c^*(w_{n-1}w_n) = \frac{[C(w_{n-1}w_n)+1]\times C(w_{n-1})}{C(w_{n-1})+V}$$



	i	want	to	eat	chinese	food	lunch	spend
i	3.8	527	0.64	6.4	0.64	0.64	0.64	1.9
want	1.2	0.39	238	0.78	2.7	2.7	2.3	0.78
to	1.9	0.63	3.1	430	1.9	0.63	4.4	133
eat	0.34	0.34	1	0.34	5.8	1	15	0.34
chinese	0.2	0.098	0.098	0.098	0.098	8.2	0.2	0.098
food	6.9	0.43	6.9	0.43	0.86	2.2	0.43	0.43
lunch	0.57	0.19	0.19	0.19	0.19	0.38	0.19	0.19
spend	0.32	0.16	0.32	0.16	0.16	0.16	0.16	0.16

$$P(w1\,|\,w2) = \frac{C(w2w1)+1}{C(w2)+V} = \frac{C(w2)}{C(w2)}\,\frac{C(w2w1)+1}{C(w2)+V} = \frac{1}{C(w2)}\,\frac{C(w2).[C(w2w1)+1]}{[C(w2)+V]}$$

Big Change to the Counts!

- C(want to) went from 608 to 238!
- P(to|want) from .66 to .26!
- Discount d= c*/c
  - d for “Chinese food” = 0.1 !!! A 10x reduction
  - So in general, Laplace is a blunt instrument
  - Could use more fine-grained method (add-k)
- But Laplace smoothing not used for N-grams, as we have much better methods
- Despite its flaws, Laplace (add-k) is however still used to smooth other probabilistic models in NLP, especially
  - For pilot studies
  - in domains where the number of zeros isn’t so huge.

Better Smoothing

- Intuition used by many smoothing algorithms, for example;
  - Good-Turing
  - Kneyser-Ney
  - Witten-Bell
- Is to use the count of things we’ve seen ***once*** to help estimate the count of things we’ve never seen

Good-Turing
Josh Goodman Intuition

- Imagine you are fishing
  - There are 8 species in this waters: carp, perch, whitefish, trout, salmon, eel, catfish, bass
- You have caught
  - 10 carp, 3 perch, 2 whitefish, 1 trout, 1 salmon, 1 eel = 18 fish
- How likely is it that the next fish caught is from a new species (one not seen in our previous catch)?
  - 3/18 (3 is number of events that seen once)
- Assuming so, how likely is it that next species is trout?
  - Must be less than 1/18 because we just stole 3/18 of our probability mass to use on unseen events

Good-Turing

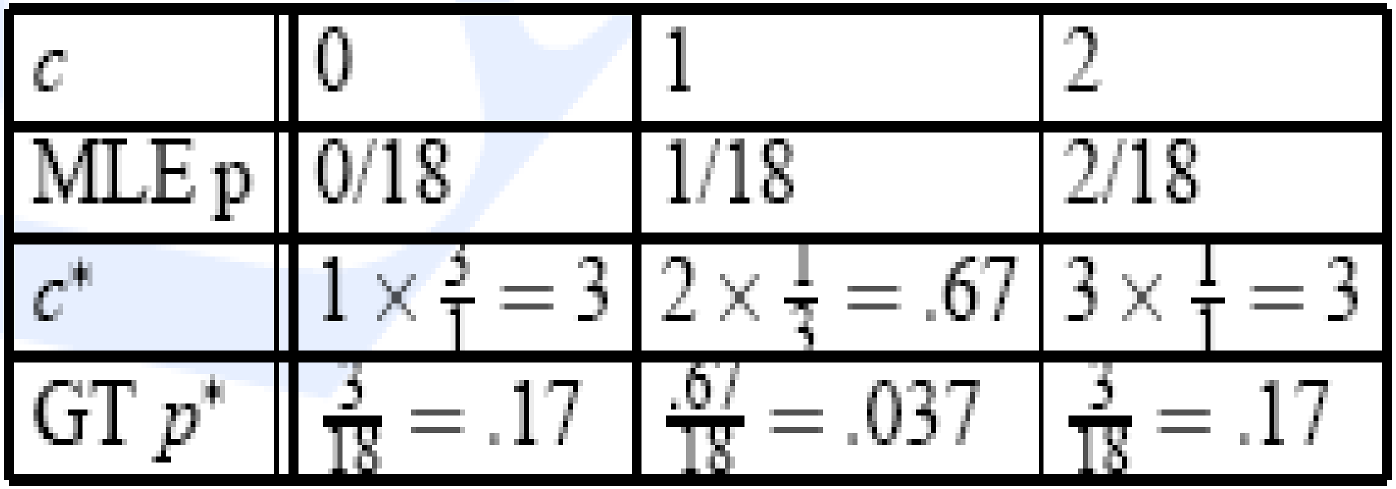
Notation: Nx is the frequency-of-frequency-x

- So N**10**=1
  - Number of fish species seen 10 times is 1 (carp)
- N**1**=3
  - Number of fish species seen 1 time is 3 (trout, salmon, eel)

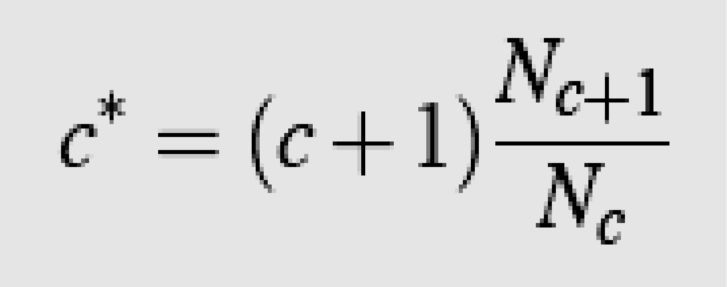
  - To estimate total number of unseen species (seen 0 times)
- Use number of species (bigrams) we’ve seen once (i.e. 3)
- So, the estimated count c* for <unseen> is 3.
- All other estimates are adjusted (down) to account for the stolen mass given for the unseen events, using the formula:

$$c^* = (c+1)\frac{N_{c+1}}{N_c}$$

GT Fish Example

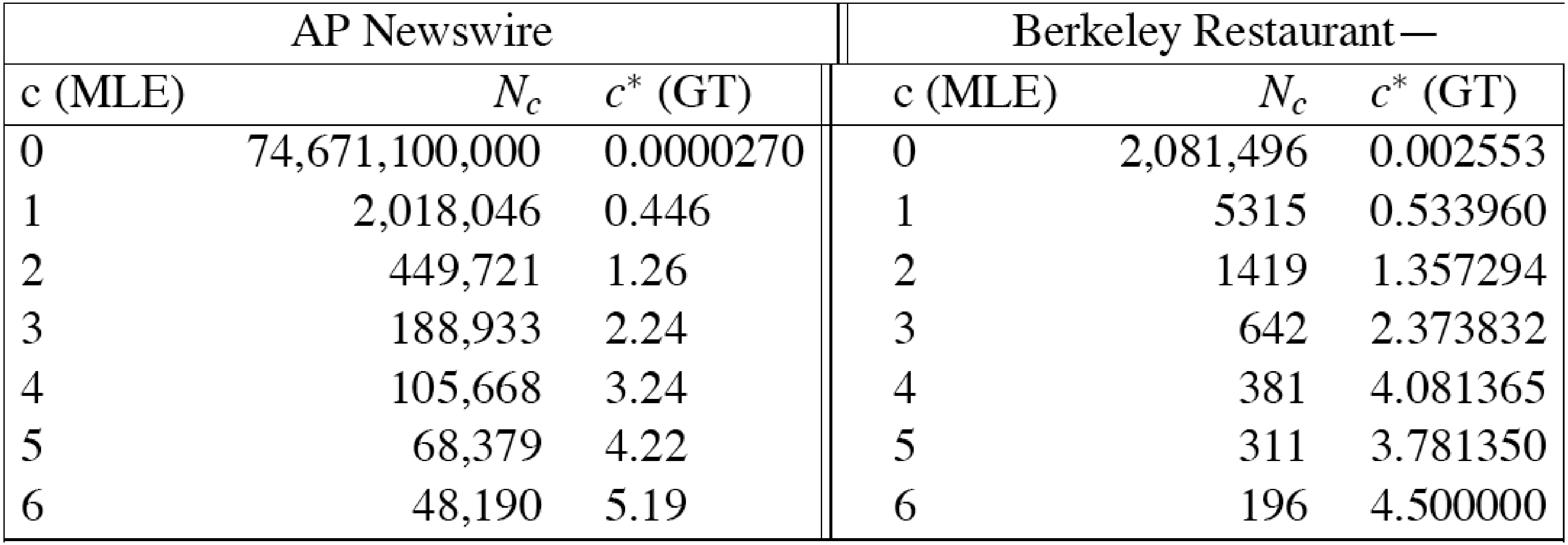


c	0	1	2
MLE p	0/18	1/18	2/18
c^*	$1 \times \frac{3}{1} = 3$	$2 \times \frac{1}{3} = .67$	$3 \times \frac{1}{1} = 3$
GT p^*	$\frac{3}{18} = .17$	$\frac{.67}{18} = .037$	$\frac{3}{18} = .17$



$$c^* = (c+1)\frac{N_{c+1}}{N_c}$$

Bigram Frequencies of Frequencies and GT Re-estimates



AP Newswire			Berkeley Restaurant—		
c (MLE)	N_c	c^* (GT)	c (MLE)	N_c	c^* (GT)
0	74,671,100,000	0.0000270	0	2,081,496	0.002553
1	2,018,046	0.446	1	5315	0.533960
2	449,721	1.26	2	1419	1.357294
3	188,933	2.24	3	642	2.373832
4	105,668	3.24	4	381	4.081365
5	68,379	4.22	5	311	3.781350
6	48,190	5.19	6	196	4.500000

AP Newswire: 22million words, Berkeley: 9332 sentences

Backoff and Interpolation

- Another really useful source of knowledge
- If we are estimating:
  - trigram p(z|x,y)
  - but count(xyz) is zero
- Use info from:
  - Bigram p(z|y)
- Or even:
  - Unigram p(z)
- How to combine this trigram, bigram, unigram info in a valid fashion?

Backoff Vs. Interpolation

1. **Backoff**: use trigram if you have it, otherwise bigram, otherwise unigram
2. **Interpolation**: mix all three by weights

Interpolation

- Simple interpolation

$$\begin{aligned}\hat{P}(w_n|w_{n-1}w_{n-2}) &= \lambda_1 P(w_n|w_{n-1}w_{n-2}) \\ &\quad +\lambda_2 P(w_n|w_{n-1}) \\ &\quad +\lambda_3 P(w_n)\end{aligned}$$

$$\sum_i \lambda_i = 1$$

- Lambdas conditional on context:

$$\begin{aligned}\hat{P}(w_n|w_{n-2}w_{n-1}) &= \lambda_1(w_{n-2}^{n-1}) P(w_n|w_{n-2}w_{n-1}) \\ &\quad +\lambda_2(w_{n-2}^{n-1}) P(w_n|w_{n-1}) \\ &\quad +\lambda_3(w_{n-2}^{n-1}) P(w_n)\end{aligned}$$

How to Set the Lambdas?

- Use a **held-out, or development** corpus
- Choose lambdas which maximize the probability of some held-out data
  - I.e. fix the *N*-gram probabilities
  - Then search for lambda values that when plugged into previous equation give largest probability for held-out set
  - Can use EM to do this search
  - Can use direct search methods (Genetic, Swarm, etc…)

Katz Backoff (very popular)

$$P_{\text{katz}}(w_n|w_{n-N+1}^{n-1}) = \begin{cases} P^*(w_n|w_{n-N+1}^{n-1}), & \text{if } C(w_{n-N+1}^{n}) > 0 \\ \alpha(w_{n-N+1}^{n-1})P_{\text{katz}}(w_n|w_{n-N+2}^{n-1}), & \text{otherwise.} \end{cases}$$

$$P_{\text{katz}}(z|x,y) \;=\; \begin{cases} P^*(z|x,y), & \text{if } C(x,y,z) > 0 \\ \alpha(x,y)P_{\text{katz}}(z|y), & \text{else if } C(x,y) > 0 \\ P^*(z), & \text{otherwise.} \end{cases}$$

$$P_{\text{katz}}(z|y) \;=\; \begin{cases} P^*(z|y), & \text{if } C(y,z) > 0 \\ \alpha(y)P^*(z), & \text{otherwise.} \end{cases}$$

Why discounts P* and alpha?

- MLE probabilities sum to 1

$$\sum_i P(w_i|w_j w_k) = 1$$

- So if we used MLE probabilities but backed off to lower order model when MLE prob is zero we would be adding extra probability mass (it is like in smoothing), and total probability would be greater than 1. So, we have to do discounting.

OOV words: <UNK> word

- **Out Of Vocabulary** = OOV words
- create an unknown word token <UNK>
  - Training of <UNK> probabilities
    - Create a fixed lexicon L of size V
    - At text normalization phase, any training word not in L changed to <UNK>
    - Now we train its probabilities like a normal word
  - At decoding time
    - If text input: Use UNK probabilities for any word not in training

Other Approaches

Class-based LMs
Morpheme-based LMs
Skip LMs

Class-based Language Models

- Standard word-based language models

$$p(w_1, w_2, ..., w_T) = \prod_{t=1}^{T} p(w_t \mid w_1, ..., w_{t-1})$$

$$\approx \prod_{t=1}^{T} p(w_t \mid w_{t-1}, w_{t-2})$$

- How to get robust n-gram estimates ($p(w_t \mid w_{t-1}, w_{t-2})$)?
  - – Smoothing
    - E.g. Kneyser-Ney, Good-Turing
  - – Class-based language models

$$p(w_t \mid w_{t-1}) \approx p(w_t \mid C(w_t)) p(C(w_t) \mid C(w_{t-1}))$$

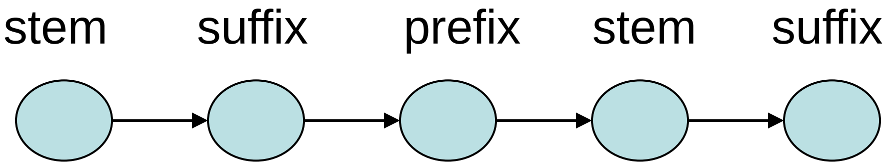
Limitation of Word-based Language Models

- **Words are inseparable whole units**.
  - E.g. “book” and “books” are distinct vocabulary units
- Especially problematic in **morphologically-rich languages**:
  - E.g. Arabic, Finnish, Russian, Turkish
  - Many unseen word contexts
  - High out-of-vocabulary rate
  - High perplexity

Arabic k-t-b	
Kitaab	A book
Kitaab-iy	My book
Kitaabu-hum	Their book
Kutub	Books

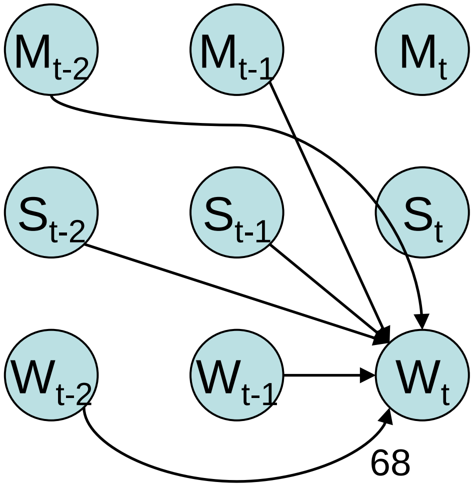
Solution: Word as Factors

- Decompose words into “factors” (e.g. stems)
- Build language model over factors: P(w|factors)
- Two approaches for decomposition
  - Linear
    - [e.g. Geutner, 1995]
  - Parallel



[Kirchhoff et. al., JHU Workshop 2002]

[Bilmes & Kirchhoff, NAACL/HLT 2003]



Different Kinds of Language Models

- cache language models (constantly adapting to a floating text)
- trigger language models (can handle long distance effects)
- POS-based language models, LM over POS tags
- class-based language models based on semantic classes
- multilevel *n*-gram language models (mix many LM together)
- interleaved language models (different LM for different parts of text)
- morpheme-based language models (separate words into core and modifyers)
- context free grammar language models (use simple and efficient LM-definition)
- decision tree language models (handle long distance effects, use rules)
- HMM language models (stochastic decision for combination of independent LMs)

Tutorial on Statistics, Probability and Information Theory for Language Engineers

Prof. Ibrahim F. Imam

Full Professor and Assistant Dean,
College of Computing and Information Technology
Arab Academy for Science, Technology & Maritime Transport, Cairo

Email: ifi05@yahoo.com Phone: 012-2242929

OUTLINE

Tutorial on Text Mining

Part 0

Supporting Tools
WordNet & SUMO

The WordNet

- WordNet is a semantic network encoding the words of a single (or multiple) language(s) using:
  - Synsets encoding the meanings for each word
  - Relations synonymy, antonymy, hypernymy, hyponymy, holonymy, meronymy, homonymy, troponymy, . . .
  - The English WordNet (v3) encodes 155287 words

POS	Unique Strings	Synsets	Total Word-Sense Pairs
Noun	117798	82115	146312
Verb	11529	13767	25047
Adjective	21479	18156	30002
Adverb	4481	3621	5580 .
Totals	155287	117659	206941

- WordNet is organized by the concept of synonym sets (synsets), e.g.:

  musician, instrumentalist, player

  person, individual, someone

http://wordnet.princeton.edu/

The WordNet Relations

Relation	Definition	Example
Hypernym	From lower to higher concepts	breakfast -> meal
Hyponym	From concepts to subordinates	meal -> lunch
Has-Member	From groups to their members	faculty -> professor
Member-Of	From members to their groups	copilot -> crew
Has-Part	From wholes to parts	table -> leg
Part-Of	From parts to wholes	course -> meal
Antonym	Opposites	leader -> follower

The WordNet

Word: Cool

Noun

S: (n) **cool** (the quality of being at a refreshingly low temperature) *"the cool of early morning"*

S: (n) aplomb, assuredness, **cool**, poise, sang-froid (great coolness and composure under strain) *"keep your cool"*

Verb

S: (v) **cool**, chill, cool down (make cool or cooler) *"Chill the food"*

S: (v) **cool**, chill, cool down (loose heat) *"The air cooled considerably after the thunderstorm"*

S: (v) **cool**, cool off, cool down (lose intensity) *"His enthusiasm cooled considerably"*

Adjective

S: (adj) **cool** (neither warm nor very cold; giving relief from heat) *"a cool autumn day"; "a cool room"; "cool summer dresses"; "cool drinks"; "a cool breeze"*

S: (adj) **cool**, coolheaded, nerveless (marked by calm self-control (especially in trying circumstances); unemotional) *"play it cool"; "keep cool"; "stayed coolheaded in the crisis"; "the most nerveless winner in the history of the tournament"*

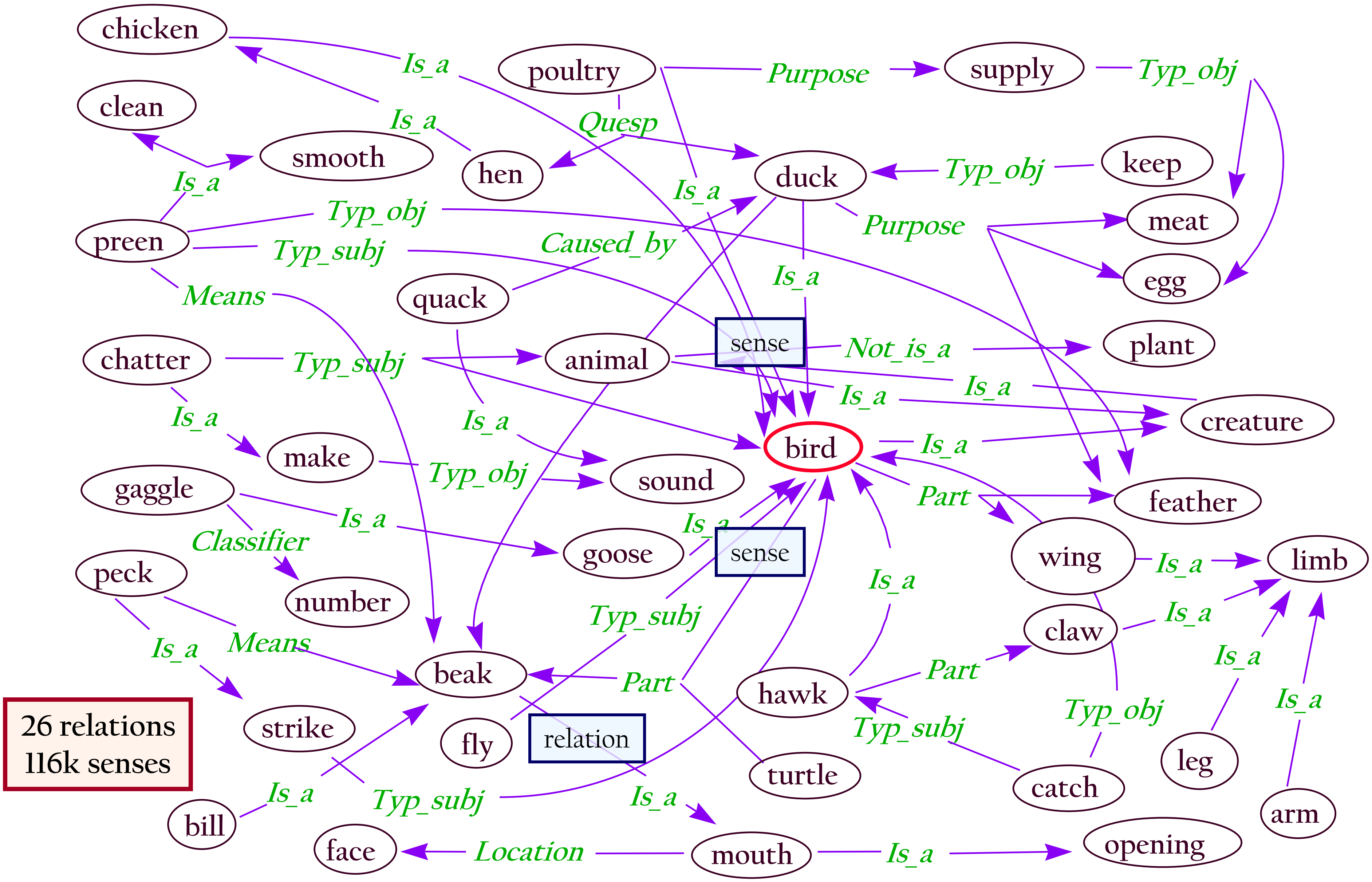
S: (adj) **cool** ((color) inducing the impression of coolness; used especially of greens and blues and violets) *"cool greens and blues and violets"*

S: (adj) **cool** (psychologically cool and unenthusiastic; unfriendly or unresponsive or showing dislike) *"relations were cool and polite"; "a cool reception"; "cool to the idea of higher taxes"*

S: (adj) **cool** ((used of a number or sum) without exaggeration or qualification) *"a cool million bucks"*

S: (adj) **cool** (fashionable and attractive at the time; often skilled or socially adept) *"he's a cool dude"; "that's cool"; "Mary's dress is really cool"; "it's not cool to arrive at a party too early"*

Sample Graph from The WordNet



26 relations
116k senses

Suggested Upper Merged Ontology (SUMO)

Suggested S

It is large, open source, and formal

+ Upper U

Focusing on *The most general* and reusable terms and definitions

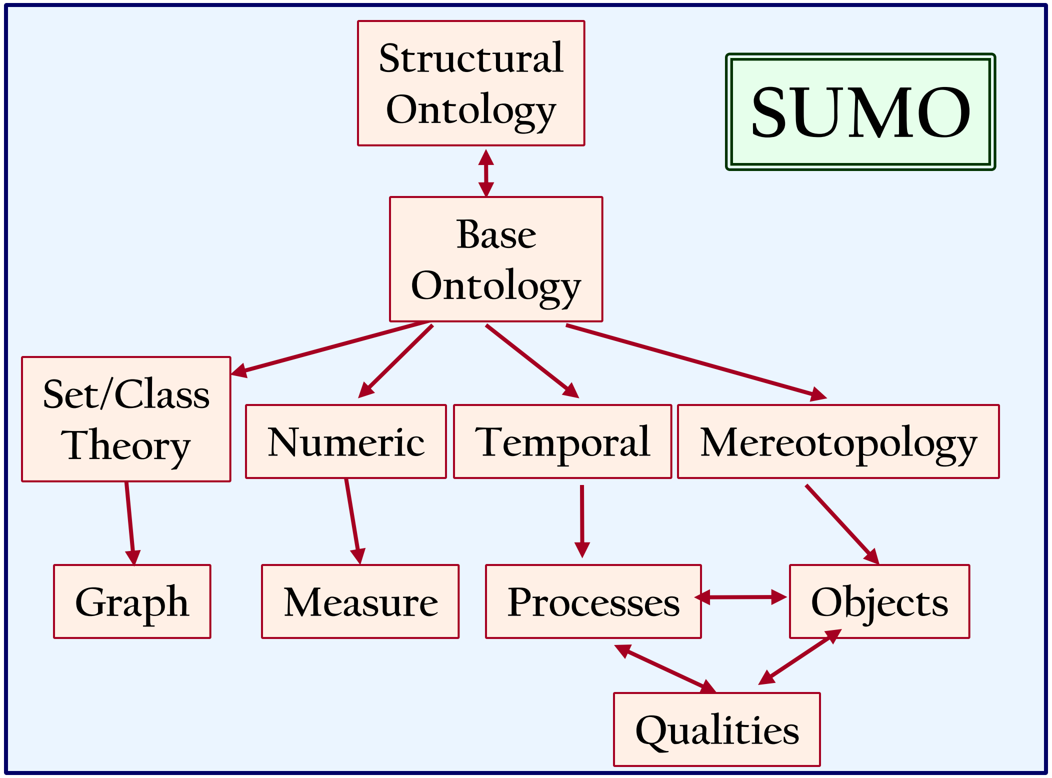
+ Merged M

Mapped with large multi-lingual lexicon

+ Ontology O = SUMO

Ontology is a set of term definitions in a formal language describing the world

Suggested Upper Merged Ontology (SUMO)



SUMO

Structural Ontology

Base Ontology

Set/Class Theory

Numeric

Temporal

Mereotopology

Graph

Measure

Processes

Objects

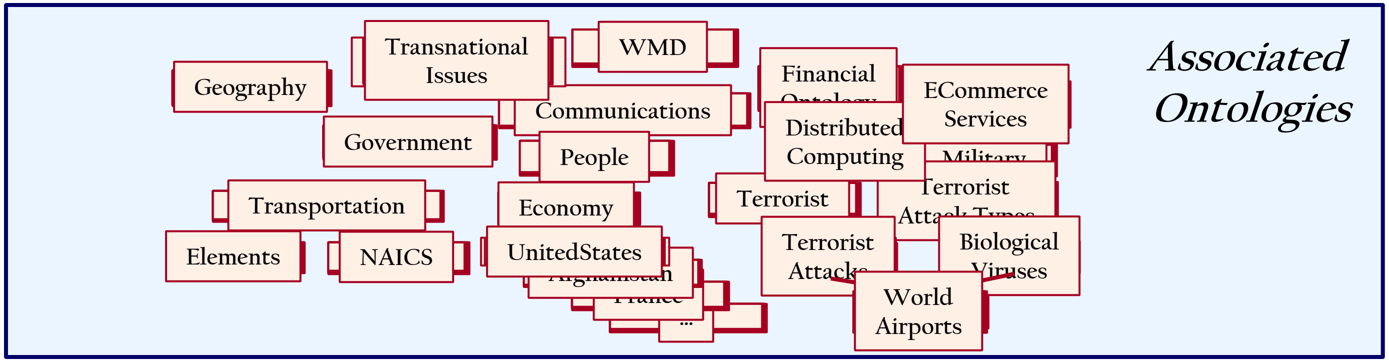
Qualities

Total Terms = 20399

Total Axioms = 67108

Rules = 2500

www.ontologyportal.org



Associated Ontologies

Geography

Transnational Issues

WMD

Communications

Government

People

Transportation

Economy

Elements

NAICS

UnitedStates

Afghanistan

France

...

Financial Ontology

ECommerce Services

Distributed Computing

Military

Terrorist

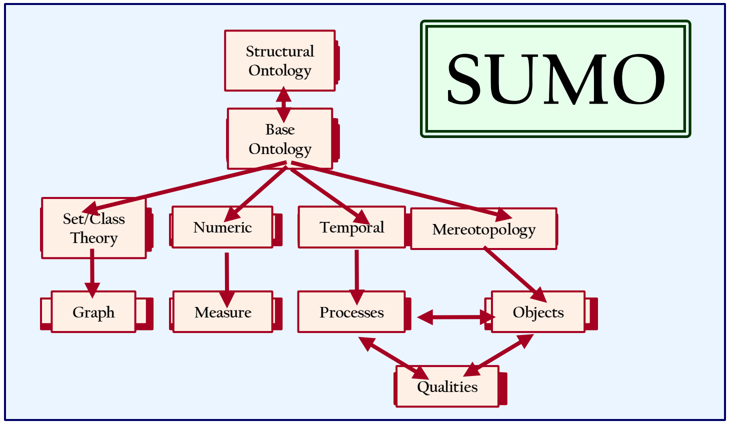
Terrorist Attack Types

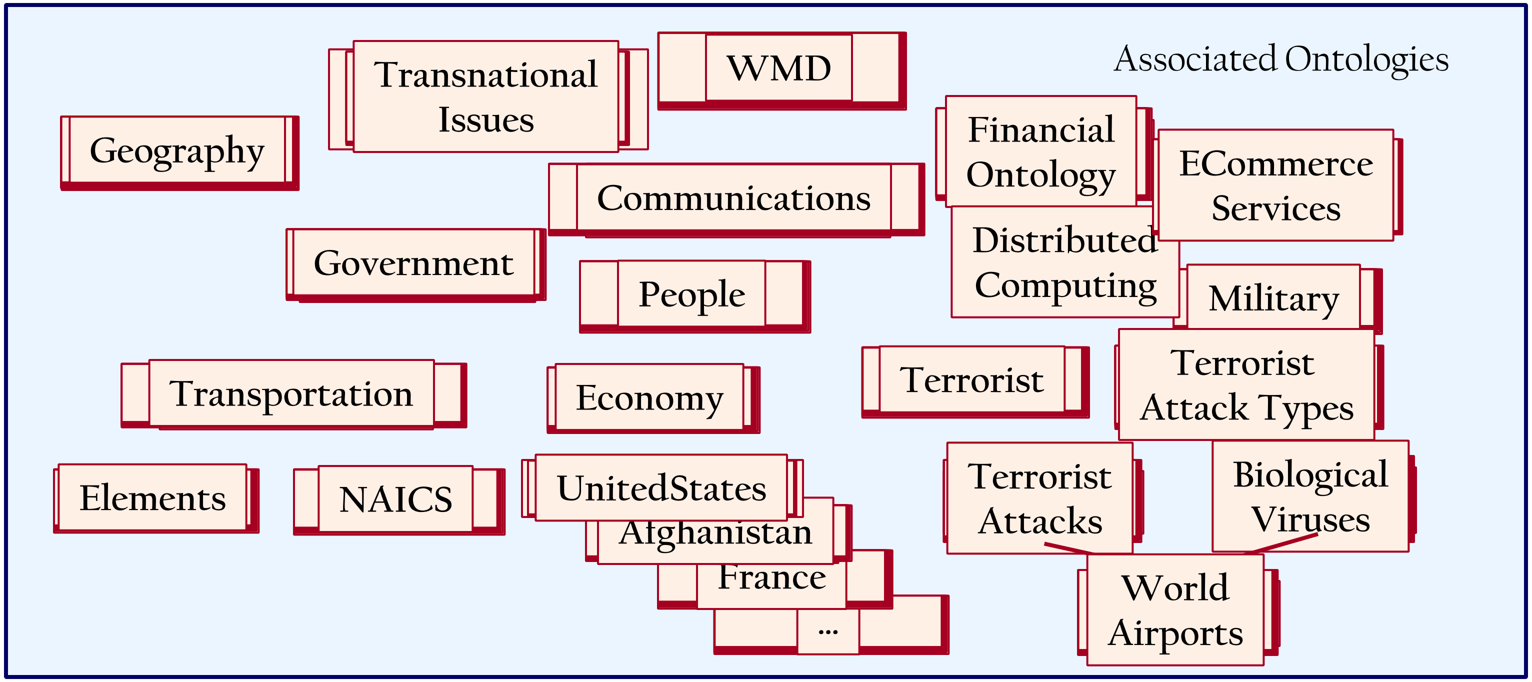
Terrorist Attacks

Biological Viruses

World Airports

Suggested Upper Merged Ontology (SUMO)





Suggested Upper Merged Ontology (SUMO)

SUMO Search Tool

This tool relates English terms to concepts from the SUMO ontology by means of mappings to WordNet synsets.

English Word: *According to WordNet, the noun "**table**" has **6 sense(s).***

104379243 a piece of furniture having a smooth flat top that is usually supported by one or more vertical legs; "it was a sturdy table".

SUMO Mappings: Table (equivalent mapping)

104379964 a piece of furniture with tableware for a meal laid out on it; "I reserved a table at my favorite restaurant".

SUMO Mappings: Table (subsuming mapping)

107565259 food or meals in general; "she sets a fine table"; "room and board".

SUMO Mappings: Food (subsuming mapping)

108266235 a set of data arranged in rows and columns; "see table 1".

SUMO Mappings: ContentBearingObject (subsuming mapping)

108480135 a company of people assembled at a table for a meal or game; "he entertained the whole table with his witty remarks".

SUMO Mappings: Meeting (subsuming mapping)

109351905 flat tableland with steep edges; "the tribe was relatively safe on the mesa but they had to descend into the valley for water".

SUMO Mappings: Mesa (equivalent mapping)

Suggested Upper Merged Ontology

Table(table)

King Arthur's Round Table, Lord's table, Parsons table, Round Table, altar, board, booth, breakfast table, card table, cocktail table, coffee table, communion table, conference table, console, console table, council board, council table, counter, dining-room table, dining table, dinner table, dresser, dressing table, drop-leaf table, gaming table, gueridon, high table, kitchen table, operating table, pedestal table, pier table, refectory table, stand, table, tea table, toilet table, trestle table, triclinium, vanity, work table, worktable

appearance as argument number 1

(documentation Table EnglishLanguage "A piece of Furniture with four legs and a flat top. It is used either for eating, paperwork or meetings.")Mid-level-ontology.kif 1328-1329%3(externalImage Table "http://upload.wikimedia.org/wikipedia/commons/7/7a/ Table_and_chairs.jpg")

BASIC MATHEMATICS

Part 1

Basic Concepts

BASIC MATHEMATICS

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n$$

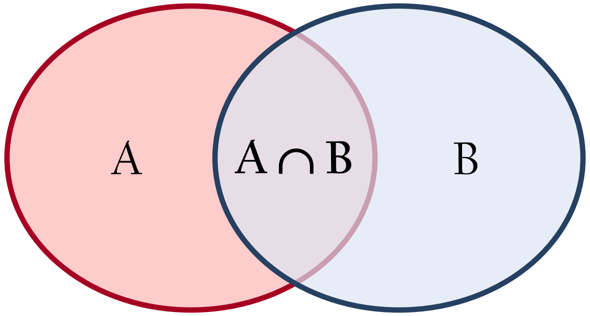
$$\prod_{i=1}^{n} i = 1 * 2 * \ldots * n$$

$$\sum_{i=1}^{n} ki = k \sum_{i=1}^{n} i$$

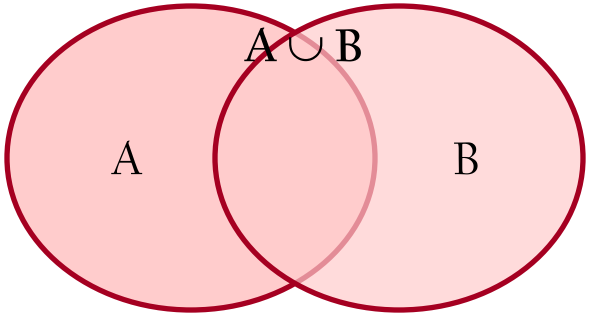
$$\prod_{i=1}^{n} ki = k \prod_{i=1}^{n} i$$

Introduction to Set Theory

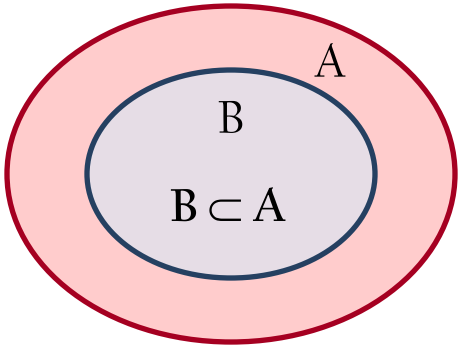
- A set is a collection of distinct items (Example: A = {1, 2, 3, 4, 5})



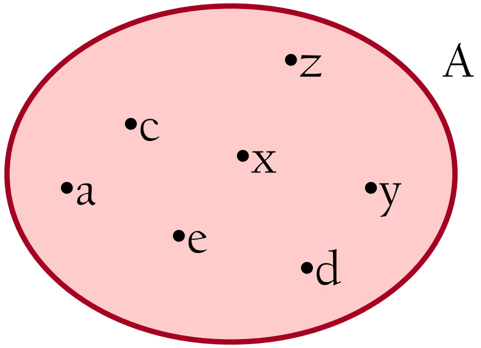
Intersection



Union



Sub-set & Super-set



x ∈ A; a∈ A; d ∈ A; ...

Introduction to Set Theory

- $A = \{a, c, e, d, x, y, z\}$ $B = \{b, c, d, y, m, n\}$ $C = \{c, d\}$

$A \cap B = \{c, d, y\}$

Intersection

$A \cup B = \{a, b, c, d, e, m, n, x, y, z\}$

Union

$A \not\subset B$ $C \subset B$ $C \subset A$

Sub-set & Super-set

$x \in A;$ $x \notin B;$ $x \notin C$

Belong Relationship

Φ/ϕ is the empty set

$\cap \cup \subset \not\subset \in \notin \neg \wedge \vee$

Introduction to Set Theory

- $A \cap (B \cap C) = (A \cap B) \cap C$ & $A \cup (B \cup C) = (A \cup B) \cup C$
- $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
- $\neg(\neg A) = A$
- $\neg(A \cap B) = \neg A \cup \neg B$

Introduction to Propositional Logic

- It is also called the Zero Order Logic
- A sentence X can be either true or false (1 or 0)

X
0
1

Y
0
1

X	Y	$X \wedge Y$
0	0	0
0	1	0
1	0	0
1	1	1

X	Y	$X \vee Y$
0	0	0
0	1	1
1	0	1
1	1	1

X	Y	$X \rightarrow Y$
0	0	1
0	1	1
1	0	0
1	1	1

X	Y	X XOR Y
0	0	1
0	1	0
1	0	0
1	1	1

$X \rightarrow Y = \neg X \vee Y$
$\neg(X \wedge Y) = \neg X \vee \neg Y$
$X \wedge X = X$ & $X \vee X = X$
$X \vee (Y \wedge Z) = (X \vee Y) \wedge (X \vee Z)$
$\neg(\neg X) = X$

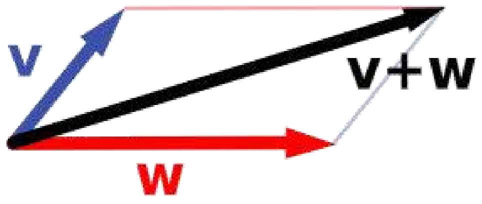
Introduction to Vectors

Part 2

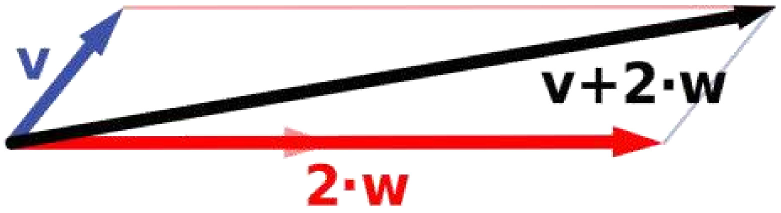
Representing Documents As Vectors

Introduction to Vectors

Adding two vectors
$(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2)$

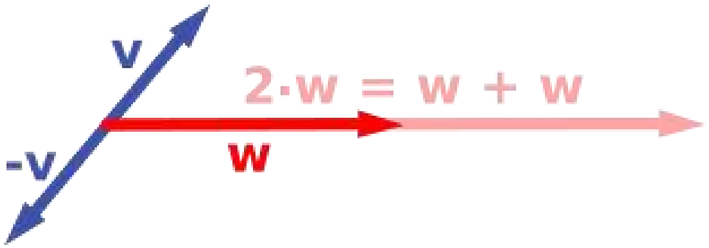


Multiplying a vector by a constant and adding it to another vector
$(x_1, y_1) + (2.x_2, 2.\ y_2) = (x_1 + 2x_2, y_1 + 2y_2)$



Multiplying a vector by -1
$-(x_1, y_1) = (-x_1, -y_1)$

Multiplying a vector by a constant
$2\ .\ (x_2, y_2) = (2x_2, 2y_2)$



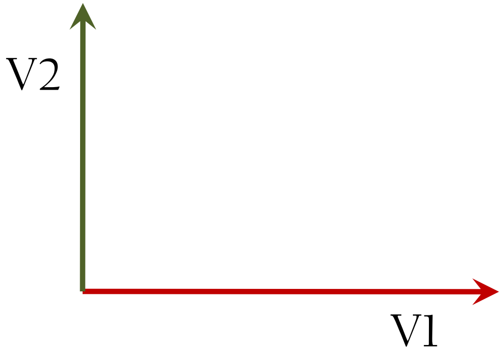
Introduction to Vectors

Multiplying two orthogonal vectors equal to zero.

Examples:

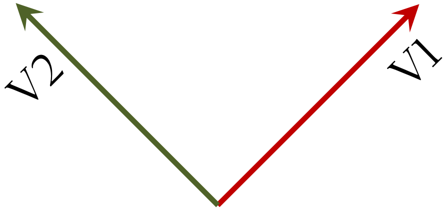
V1 =(5, 0) & V2=(0, 4)

V1 . V2 = 0



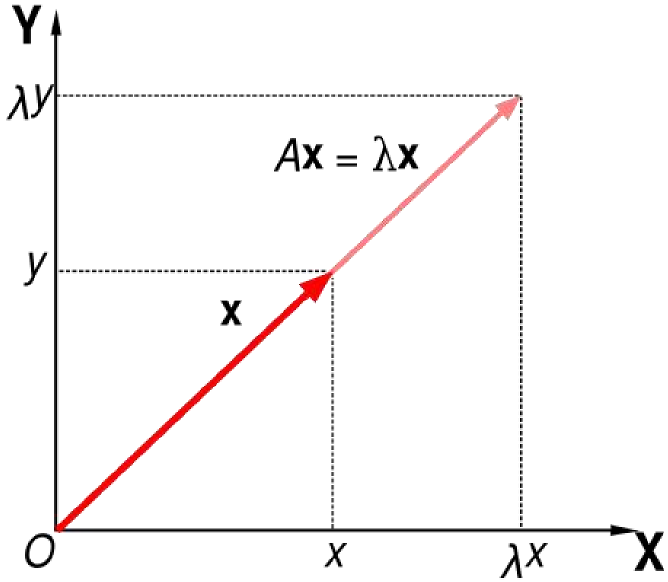
V1 =(5, 4) & V2=(-4, 5)

V1 . V2 = 0



Eigen Values & Eigen Vectors

- An eigenvector of a matrix *A* is a nonzero vector *x*, where *A.x* is similar to applying a linear transformation *λ* to *x* which, may change in length, but not direction
- *A* acts to stretch the vector *x*, not change its direction, so *x* is an eigenvector of *A*



$$Ax - \lambda I x = 0$$

$$(A - \lambda I)x = 0$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} x \\ y \end{bmatrix}$$

if there exist an inverse $(A - \lambda I)^{-1}$, *then* $x = 0$

we need $\det(A - \lambda I) = 0$ *to avoid the trevial solution* $x = 0$

$$\det(A - \lambda I) = 0$$

Example on Eigen Values & Eigen Vectors

- Suppose $\underline{A}$ is 2x2 matrix

$$A=\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}$$

$$\det\begin{bmatrix}2-\lambda & 1\\ 1 & 2-\lambda\end{bmatrix}=(2-\lambda)^2-1=0$$

$$\lambda=1 \quad or \quad \lambda=3$$

$$for\ \lambda=3,\quad \begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=3\begin{bmatrix}x\\ y\end{bmatrix}$$

$$for\ \lambda=1,\quad \begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=1\begin{bmatrix}x\\ y\end{bmatrix}$$

$$\begin{bmatrix}2x+y\\ x+2y\end{bmatrix}=\begin{bmatrix}3x\\ 3y\end{bmatrix}$$

$$2x+y=3x$$

$$\boxed{x=y}$$

$$\begin{bmatrix}2x+y\\ x+2y\end{bmatrix}=\begin{bmatrix}x\\ y\end{bmatrix}$$

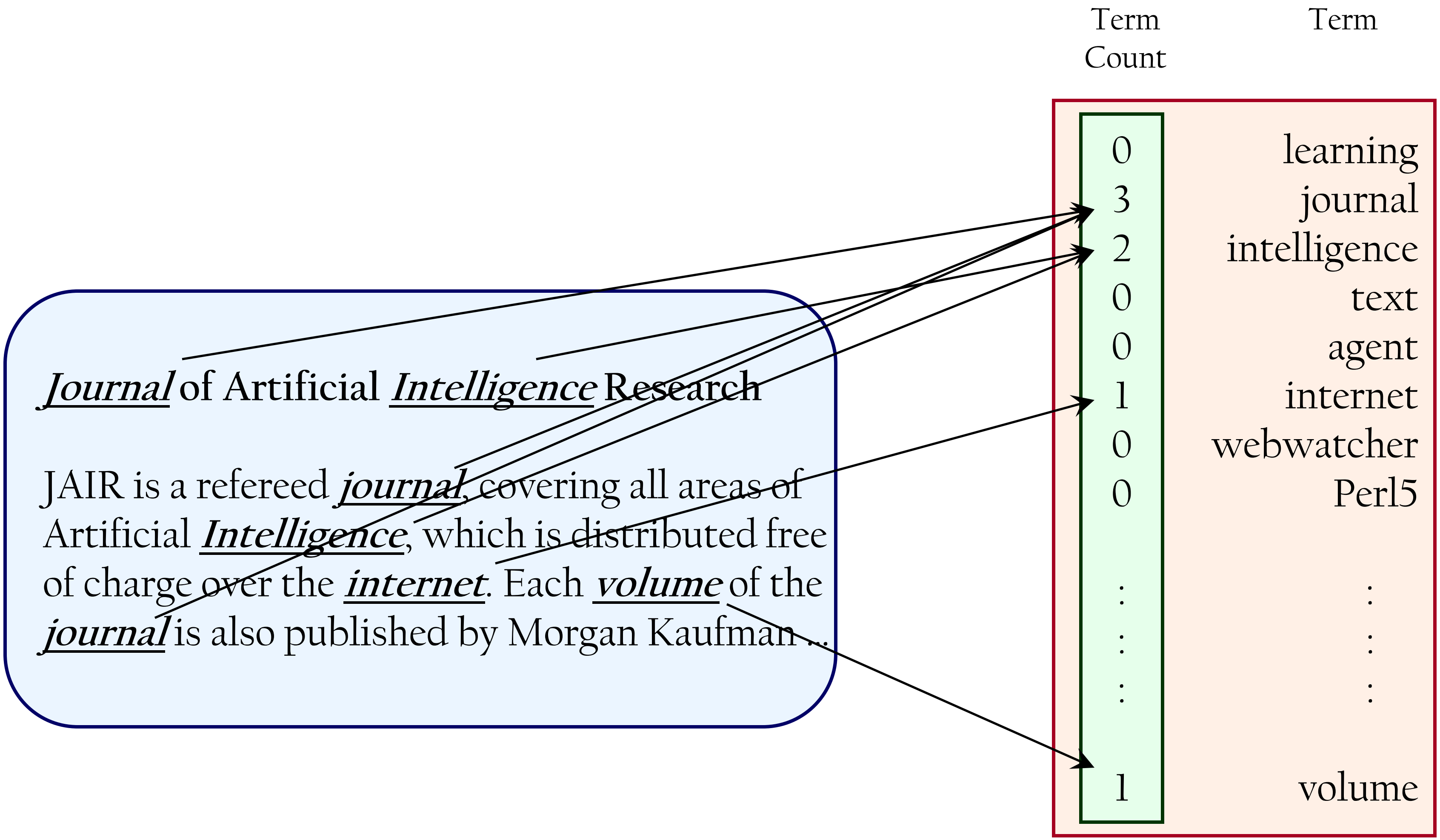
$$2x+y=x$$

$$\boxed{x=-y}$$

The eigenvectors are:

$$\boxed{\begin{bmatrix}1\\ 1\end{bmatrix}} \qquad \boxed{\begin{bmatrix}1\\ -1\end{bmatrix}}$$

Representing Documents as Vectors



Journal of Artificial *Intelligence* Research

JAIR is a refereed *journal*, covering all areas of Artificial *Intelligence*, which is distributed free of charge over the *internet*. Each *volume* of the *journal* is also published by Morgan Kaufman …

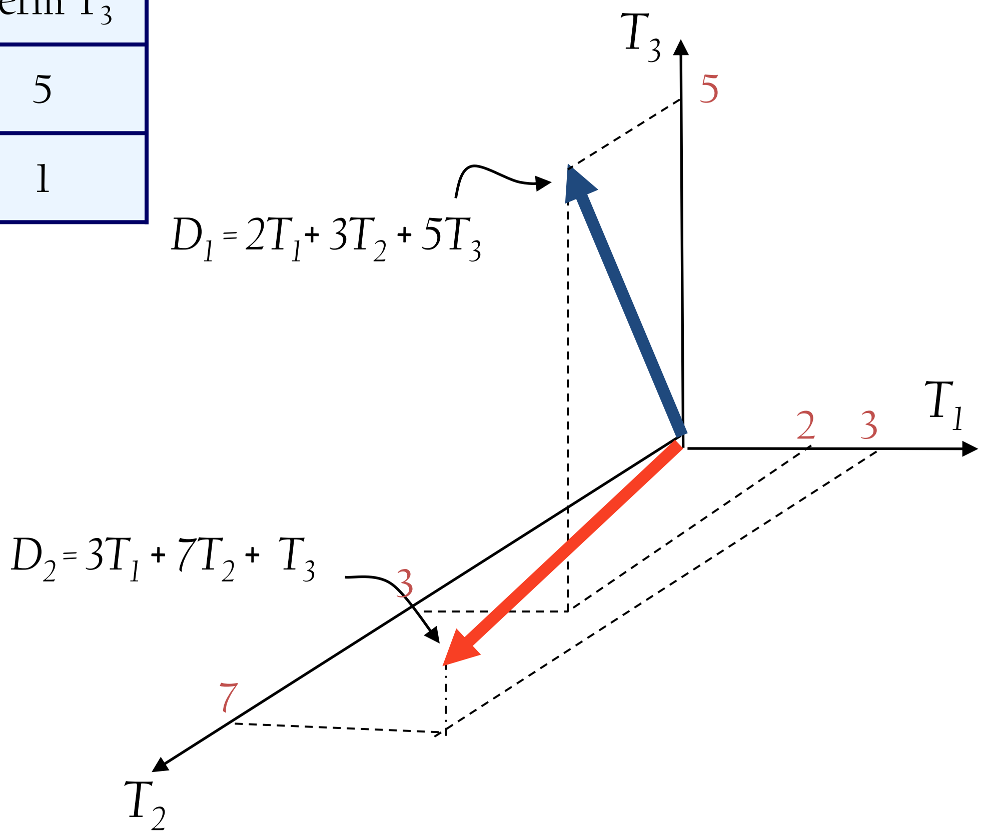
Term Count	Term
0	learning
3	journal
2	intelligence
0	text
0	agent
1	internet
0	webwatcher
0	Perl5
:	:
:	:
:	:
1	volume

Documents as Vectors

Suppose we have two documents containing three nouns only

	Term T_1	Term T_2	Term T_3
Document D_1	2	3	5
Document D_2	3	7	1

$$\begin{array}{c|c} D_1 & D_2 \\ \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} & \begin{bmatrix} 3 \\ 7 \\ 1 \end{bmatrix} \end{array}$$



$D_1 = 2T_1 + 3T_2 + 5T_3$

$D_2 = 3T_1 + 7T_2 + T_3$

Dimensionality Reduction

Term Count	Term
34	Home
32	Garden
15	Room
14	Window
11	Furniture
11	Restroom
6	Floor
5	Kitchen
5	Balcony
1	Chimney
1	Street
1	City
1	Dog
1	Lake

Dimensionality Reduction

Term Count	Term
15	Room
14	Window
11	Furniture
11	Restroom
6	Floor
5	Kitchen
5	Balcony

- Term Count
- tfidf
- Chi-Square
- Information Gain
- Gain Ratio

Term Frequency & Inverse Document Frequency

Usually a combination of the term frequency and the inverse document frequency

$$TFIDF = w_{ik} = tf_{ik} \times idf_{ik}$$

$$tf_{ik} = 1 + \log_2(tr_{ik}) \qquad \textit{and zero when} \log = 0$$

$$idf_{ik} = \log_2\left(\frac{N}{n_{ik}}\right) \qquad \textit{and zero when} \log = 0$$

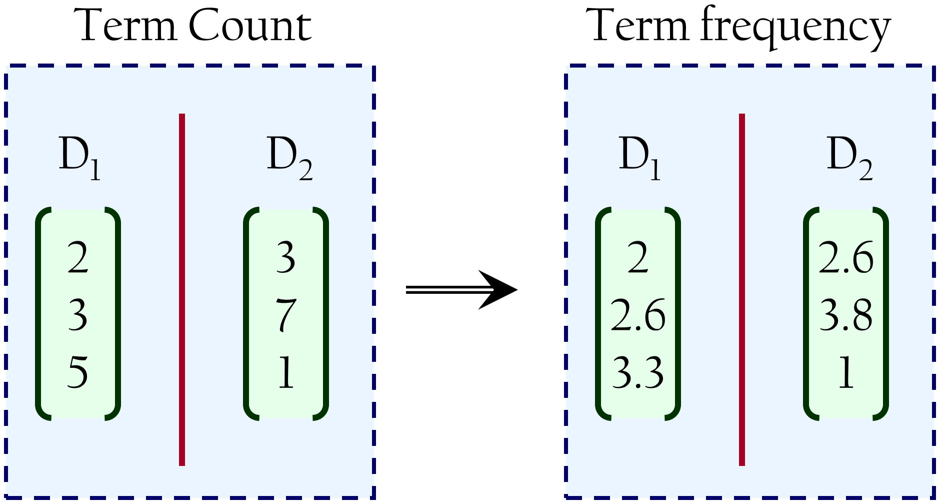
tf_{ik} is the term frequency of term *i* in document *k*, tr_{ik} is the count of term *i* in document *k*, idf_{ik} is the inverse document frequency of term *i* in document *k*, *N* is the total number of documents in the collection, n_{ik} is the number of occurrence of term *i* in document *k*, w_{ik} is the weight of term *i* in document *k*. Logarithm has been used to reduces the difference between the weight of high and low frequency terms. Logarithm of base 2 is used when vectors are full of binary TFIDF weights 0 and 1. Logarithm of base 10 is used when vectors are full of TFIDF weights except binary ones. TFIDF weights values are not normalized.

Term Frequency & Inverse Document Frequency

$$tf_{ik} = 1 + \log_2(tr_{ik}) \qquad and\ zero\ when \log = 0$$

$$idf_{ik} = \log_2\left(\frac{N}{n_{ik}}\right) \qquad and\ zero\ when \log = 0$$

$$\log_2 x = \log_{10} x / \log_{10} 2$$



Term Count

$$\begin{array}{cc} D_1 & D_2 \\ \begin{bmatrix}2\\3\\5\end{bmatrix} & \begin{bmatrix}3\\7\\1\end{bmatrix} \end{array} \Longrightarrow \begin{array}{cc} D_1 & D_2 \\ \begin{bmatrix}2\\2.6\\3.3\end{bmatrix} & \begin{bmatrix}2.6\\3.8\\1\end{bmatrix} \end{array}$$

Term frequency

The Chi-Square Distribution

$$\chi^2(t_k, c_i) = \frac{[P(t_k, c_i)P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i)P(\bar{t}_k, c_i)]^2}{P(t_k)P(\bar{t}_k)P(c_i)P(\bar{c}_i)}$$

$P(t_k, c_i)$ ➔probability document x contains term t and belongs to category c.

$P(\bar{t}_k, c_i)$ ➔probability document x does not contain term t and belongs to category c.

$P(t_k, \bar{c}_i)$ ➔probability document x contains term t and does not belong to category c.

$P(\bar{t}_k, \bar{c}_i)$ ➔probability document x does not contain term t and does not belong to category c.

$P(t)$ ➔probability of term t

$P(c)$ ➔probability of category c

The Information Gain

It measures the classification power of a term

$$IG(t_k, c_i) = \sum_{c \in \{c_i, \bar{c}_i\}} \sum_{t \in \{t_k, \bar{t}_k\}} P(t,c) \log_2 \frac{P(t,c)}{P(t)P(c)}$$

$P(t_k, c_i)$ ➔probability document x contains term t and belongs to category c.

$P(\bar{t}_k, c_i)$ ➔probability document x does not contain term t and belongs to category c.

$P(t_k, \bar{c}_i)$ ➔probability document x contains term t and does not belong to category c.

$P(\bar{t}_k, \bar{c}_i)$ ➔probability document x does not contain term t and does not belong to category c.

$P(t)$ ➔probability of term t.

$P(c)$ ➔probability of category c.

The Gain Ratio

$$GR(t_k, c_i) = \frac{\sum_{c \in \{c_i, \bar{c}_i\}} \sum_{t \in \{t_k, \bar{t}_k\}} P(t,c) \log_2 \frac{P(t,c)}{P(t)P(c)}}{-\sum_{c \in \{c_i, \bar{c}_i\}} P(c) \log_2 P(c)}$$

$P(t_k, c_i)$ ➔probability document x contains term t and belongs to category c.

$P(\bar{t}_k, c_i)$ ➔probability document x does not contain term t and belongs to category c.

$P(t_k, \bar{c}_i)$ ➔probability document x contains term t and does not belong to category c.

$P(\bar{t}_k, \bar{c}_i)$ ➔probability document x does not contain term t and does not belong to category c.

$P(t)$ ➔probability of term t.

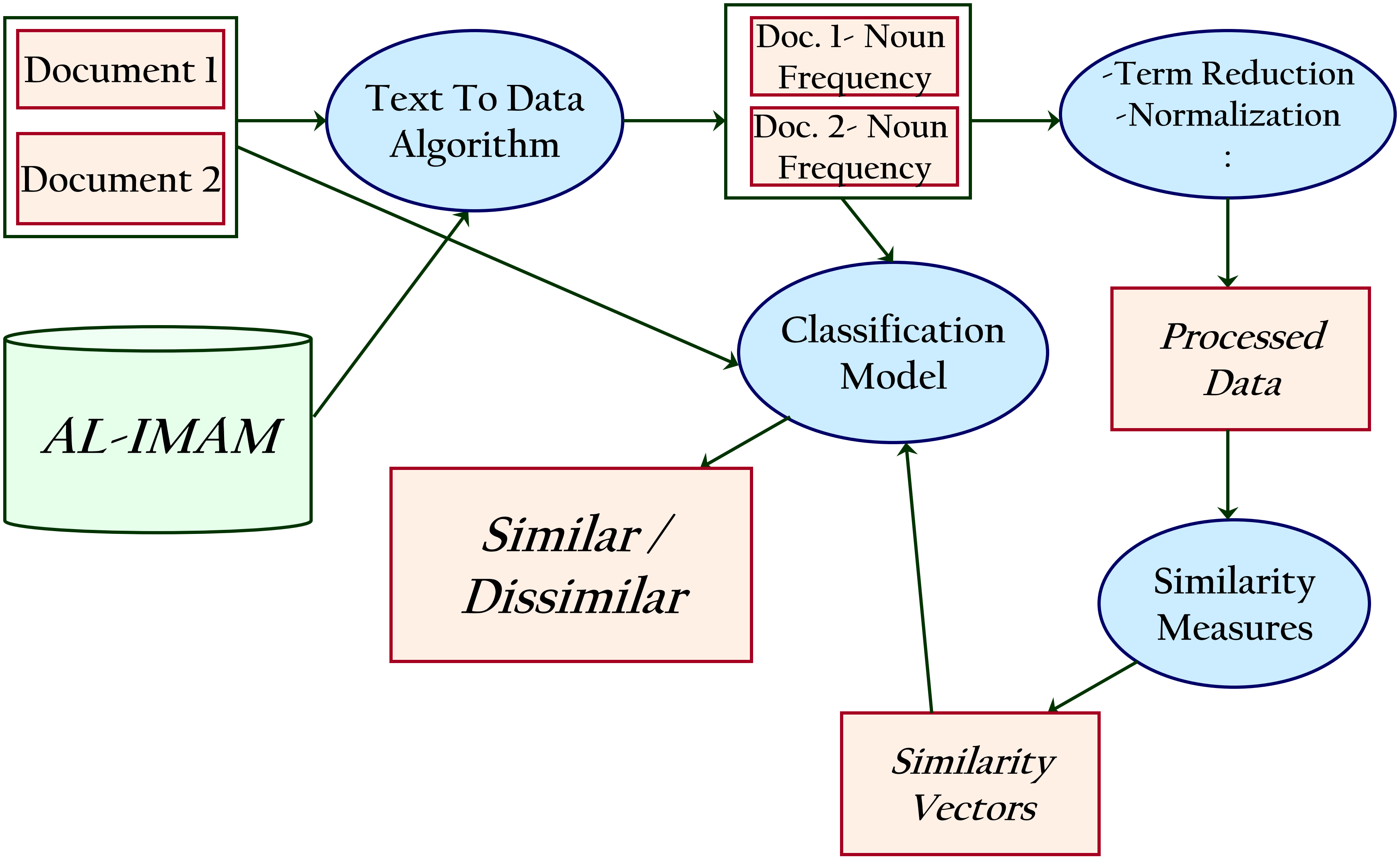
$P(c)$ ➔probability of category c.

Tutorial on Text Mining

Part 3

Text Mining Applications

Text Similarity



Document 1

Document 2

Text To Data Algorithm

Doc. 1- Noun Frequency

Doc. 2- Noun Frequency

-Term Reduction
-Normalization
:

AL-IMAM

Classification Model

Processed Data

Similar / Dissimilar

Similarity Measures

Similarity Vectors

Text Similarity

- Each Document is represented by a vector of terms
- Each Term is considered as a dimension in the space
- Terms in the space are uncorrelated so the dimensions are orthogonal on each other
- Each element of the vector has a value (**Term Weight**)

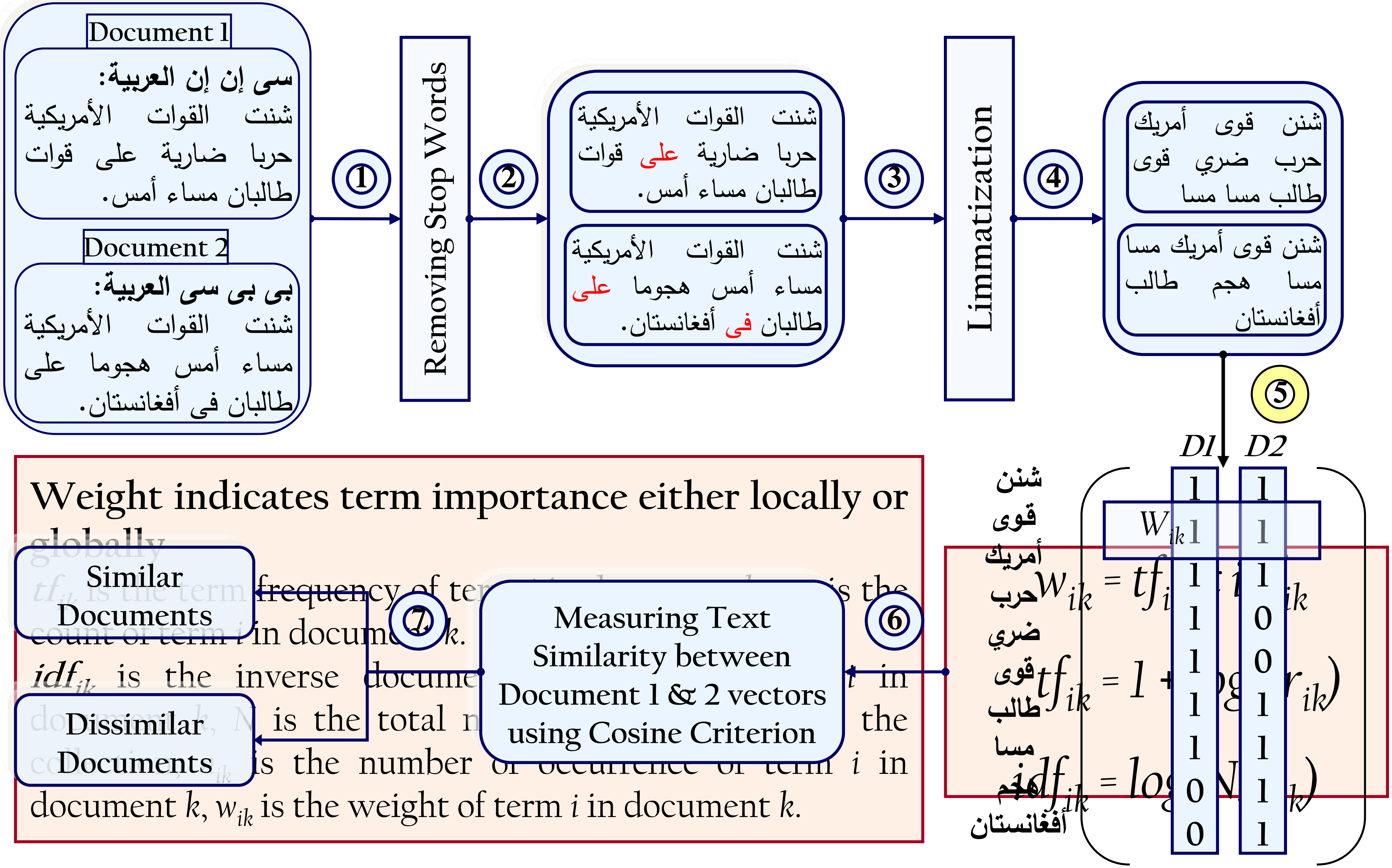
- Document A
  - "A dog and a cat."

A	Dog	and	Cat	Frog
2	1	1	1	0

- Document B
  - "A frog."

A	Dog	and	Cat	Frog
1	0	0	0	1

Text Similarity



Document 1

سى إن إن العربية:
شنت القوات الأمريكية حربا ضارية على قوات طالبان مساء أمس.

Document 2

بى بى سى العربية:
شنت القوات الأمريكية مساء أمس هجوما على طالبان فى أفغانستان.

Removing Stop Words

شنت القوات الأمريكية حربا ضارية على قوات طالبان مساء أمس.

شنت القوات الأمريكية مساء أمس هجوما على طالبان فى أفغانستان.

Limmatization

شنن قوى أمريك
حرب ضري قوى
طالب مسا مسا

شنن قوى أمريك مسا
مسا هجم طالب
أفغانستان

Measuring Text Similarity between Document 1 & 2 vectors using Cosine Criterion

Similar Documents

Dissimilar Documents

Weight indicates term importance either locally or globally

tf_{ik} is the term frequency of ter… is the count of term i in docume… k.

idf_{ik} is the inverse docume… in do… k, N is the total n… the co… r_{ik} is the number of occurrence of term i in document k, w_{ik} is the weight of term i in document k.

$w_{ik} = tf_{i}$ … $_{ik}$

$tf_{ik} = 1 +$ … $r_{ik})$

$idf_{ik} = \log$ … N … $_{k})$

Text Similarity

$$\text{Cosine}(D_j, D_k) = \frac{\sum_{i=1}^{n} w_{ij} \times w_{ik}}{\sqrt{\sum_{i=1}^{n} w_{ij}^2}\sqrt{\sum_{i=1}^{n} w_{ik}^2}}$$

$$\text{Euclidean }(D_j, D_k) = \sqrt{\sum_{i=1}^{n} (w_{i,j} - w_{i,k})^2 / n}$$

$$\text{Dice}(D_j, D_k) = \frac{2\sum_{i=1}^{n} w_{i,j} \times w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2} + \sqrt{\sum_{i=1}^{n} w_{i,k}^2}}$$

$$\text{Overlap}(D_j, D_k) = \frac{\sum_{i=1}^{n} w_{i,j} \times w_{i,k}}{\min(\sqrt{\sum_{i=1}^{n} w_{i,j}^2}, \sqrt{\sum_{i=1}^{n} w_{i,k}^2})}$$

$$\text{Jaccard}(D_j, D_k) = \frac{\sum_{i=1}^{n} w_{i,j} \times w_{i,k}}{\sum_{i=1}^{n} w_{i,j}^2 + \sum_{i=1}^{n} w_{i,k}^2 - \sum_{i=1}^{n} w_{i,j} \times w_{i,k}}$$

Term Weight (w_{ik})= tf_{ik} *x* $idf_{ik,}$

Term Frequency (tf_{ik}) = *1 + log*(tr_{ik})

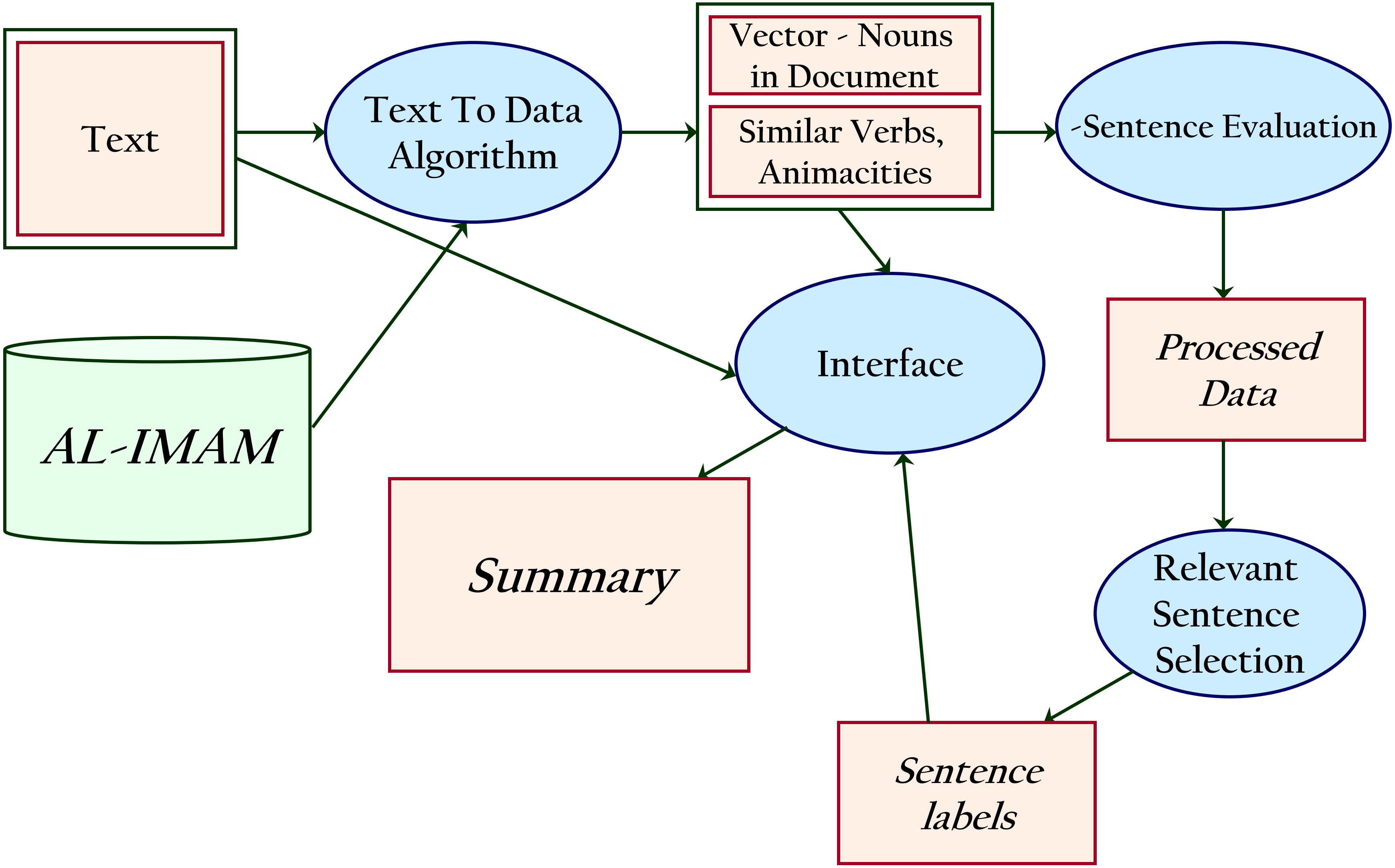
Inverse Doc. Freq. (idf_{ik}) = *log*(N/n_{ik})

(tr_{ik}) is the count of term *i* in doc. *k*.

(N) is the total # of docs

(n_{ik}) is the # of occur. of term *i* in doc. *k*

Text Summarization



Text

Text To Data Algorithm

Vector - Nouns in Document

Similar Verbs, Animacities

-Sentence Evaluation

AL-IMAM

Interface

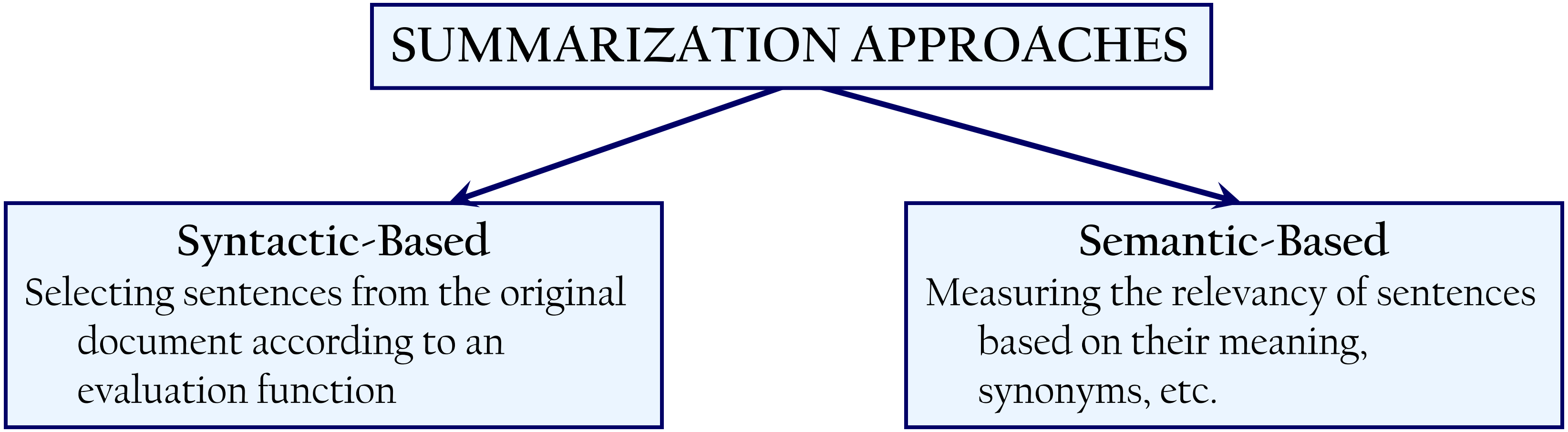
Processed Data

Summary

Relevant Sentence Selection

Sentence labels

Text Summarization Approaches



SUMMARIZATION APPROACHES

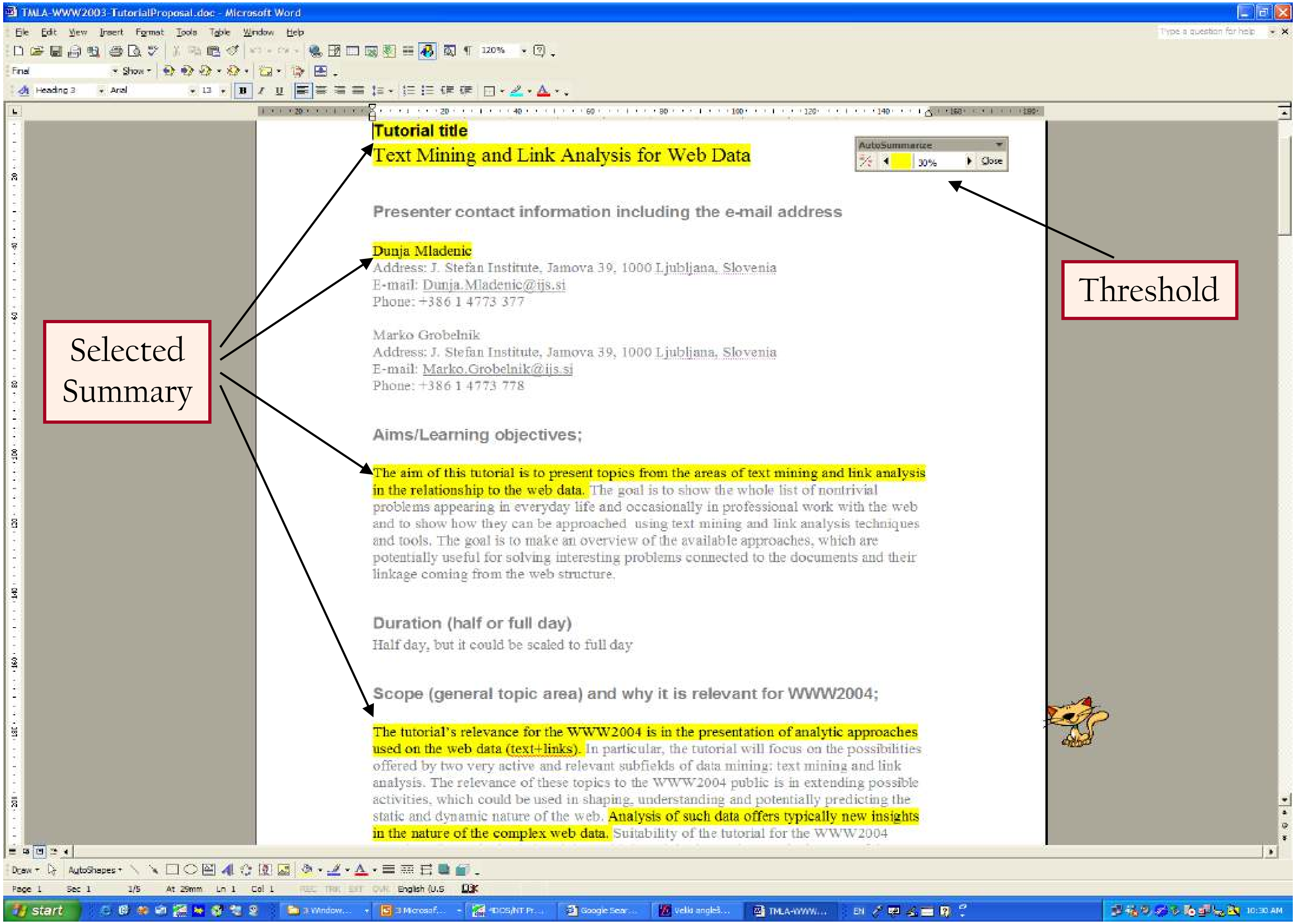
Syntactic-Based

Selecting sentences from the original document according to an evaluation function

Semantic-Based

Measuring the relevancy of sentences based on their meaning, synonyms, etc.

Microsoft Word Summarizer



TMLA-WWW2003-TutorialProposal.doc - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Type a question for help
120%
Final
Show
Heading 3
Arial
13
AutoSummarize
30%
Close
Draw
AutoShapes
Page 1 Sec 1 1/5 At 29mm Ln 1 Col 1
REC TRK EXT OVR English (U.S
start
10:30 AM

Tutorial title

Text Mining and Link Analysis for Web Data

Presenter contact information including the e-mail address

Dunja Mladenic
Address: J. Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
E-mail: Dunja.Mladenic@ijs.si
Phone: +386 1 4773 377

Marko Grobelnik
Address: J. Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
E-mail: Marko.Grobelnik@ijs.si
Phone: +386 1 4773 778

Aims/Learning objectives;

The aim of this tutorial is to present topics from the areas of text mining and link analysis in the relationship to the web data. The goal is to show the whole list of nontrivial problems appearing in everyday life and occasionally in professional work with the web and to show how they can be approached using text mining and link analysis techniques and tools. The goal is to make an overview of the available approaches, which are potentially useful for solving interesting problems connected to the documents and their linkage coming from the web structure.

Duration (half or full day)

Half day, but it could be scaled to full day

Scope (general topic area) and why it is relevant for WWW2004;

The tutorial's relevance for the WWW2004 is in the presentation of analytic approaches used on the web data (text+links). In particular, the tutorial will focus on the possibilities offered by two very active and relevant subfields of data mining: text mining and link analysis. The relevance of these topics to the WWW2004 public is in extending possible activities, which could be used in shaping, understanding and potentially predicting the static and dynamic nature of the web. Analysis of such data offers typically new insights in the nature of the complex web data. Suitability of the tutorial for the WWW2004

Selected Summary

Threshold

Example of Semantic Summarization

- Summarize the following article in 10 words

  HOUSTON – The Hubble Space Telescope got smarter and better able to point at distant astronomical targets on Thursday as spacewalking astronauts replaced two major pieces of the observatory's gear. On the second spacewalk of the shuttle Discovery's Hubble repair mission, the astronauts, C. Michael Foale and Claude Nicollier, swapped out the observatory's central computer and one of its fine guidance sensors, a precision pointing device. The spacewalkers ventured into Discovery's cargo bay, where Hubble towers almost four stories above, at 2:06 p.m. EST, about 45 minutes earlier than scheduled, to get a jump on their busy day of replacing some of the telescope's most important components. . . .

Space News: [the shuttle Discovery's Hubble repair mission, the observatory's central computer]

Taken from: Ren´e Witte, "Introduction to Text Mining", http://rene-witte.net, 2006

Example of Semantic Summarization

1. Input document is split into sentences
2. Each sentence is deep-parsed
3. Name-entities are disambiguated:
   - Determining that 'George Bush' == 'Bush' == 'U.S. president'
4. Performing Anaphora resolution:
   - Pronouns are connected with named-entities
5. Extracting of **Subject-Predicate-Object** triples
6. Constructing a **graph** from triples
7. Each triple in the graph is described with features for learning
8. Using machine learning train a model for classification of triples into the summary
9. Generate a summary graph from selected triples
10. From the summary graph generate textual summary document

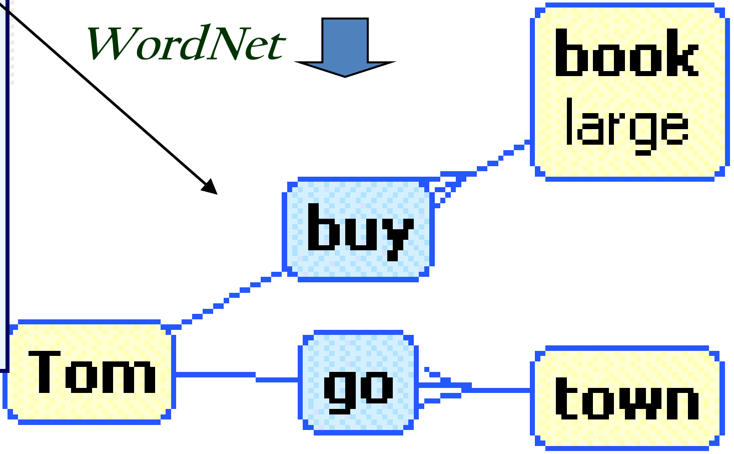
Tom went to town. In a bookstore he bought a large book.

NLPWin

Tom went to town. In a bookstore he [Tom] bought a large book.

Tom ← go → town
Tom ← buy → book

WordNet

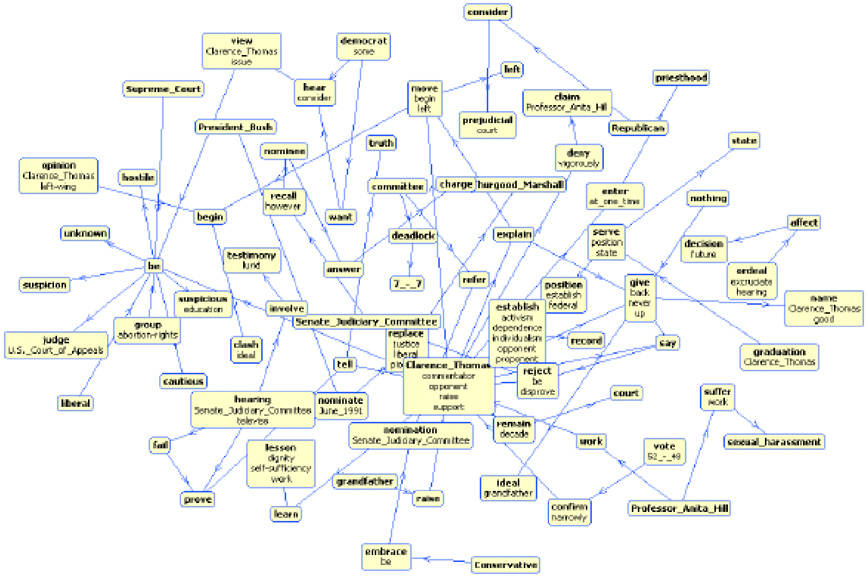


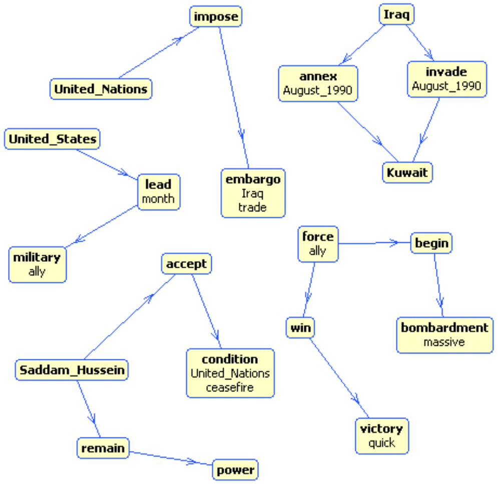
Example of Semantic Summarization (Cont.)

- A model was trained deciding which **Subject-Predicate-Object** triple belongs into the target summary
- For training was used Support Vector Machine (SVM) on 400 statistic, linguistic and graph topological features

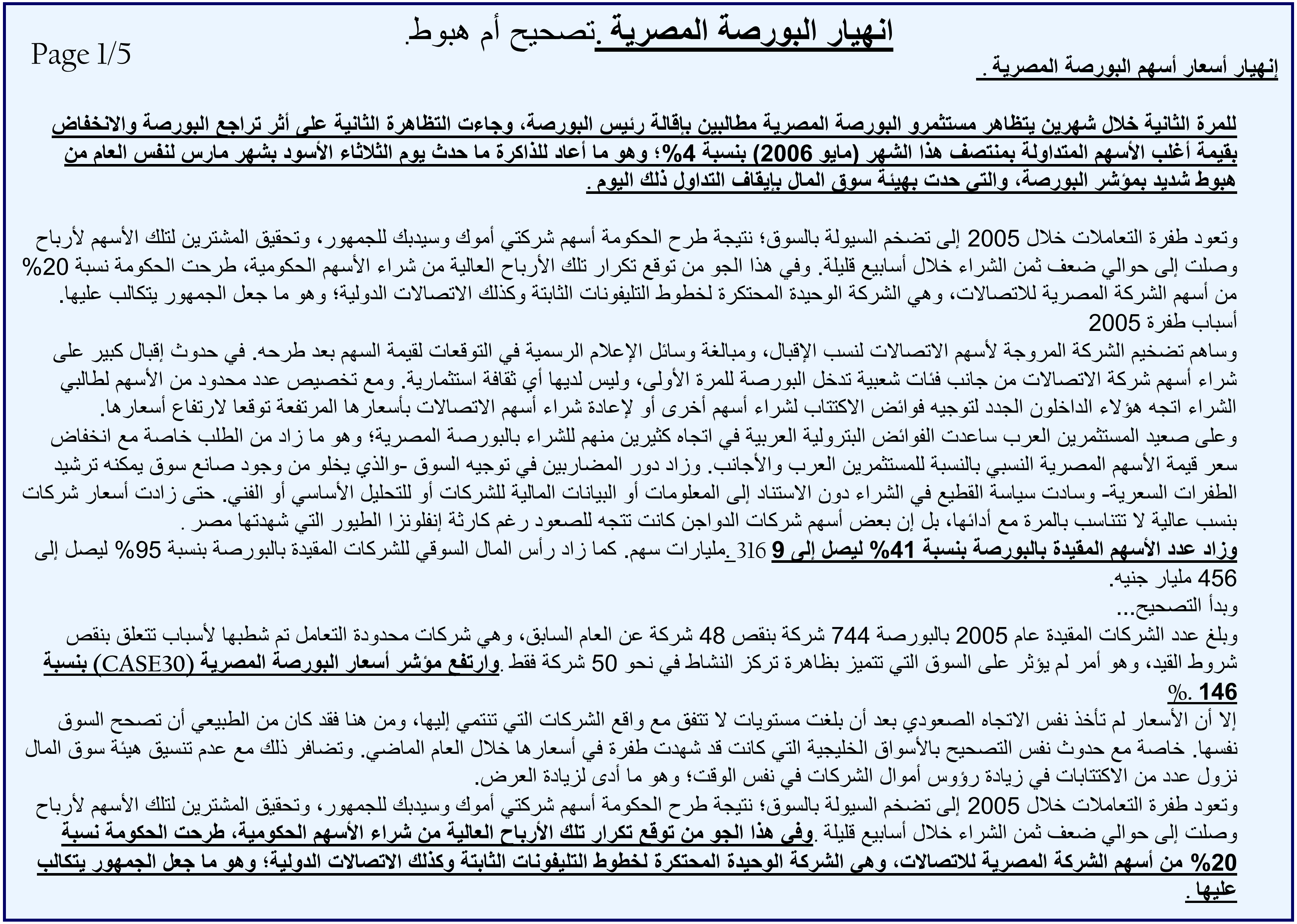
Document Semantic network

Summary semantic network





Example of Arabic Summarization



Page 1/5

انهيار البورصة المصرية .تصحيح أم هبوط

إنهيار أسعار أسهم البورصة المصرية .

للمرة الثانية خلال شهرين يتظاهر مستثمرو البورصة المصرية مطالبين بإقالة رئيس البورصة، وجاءت التظاهرة الثانية على أثر تراجع البورصة والانخفاض بقيمة أغلب الأسهم المتداولة بمنتصف هذا الشهر (مايو 2006) بنسبة 4%؛ وهو ما أعاد للذاكرة ما حدث يوم الثلاثاء الأسود بشهر مارس لنفس العام من هبوط شديد بمؤشر البورصة، والتي حدت بهيئة سوق المال بإيقاف التداول ذلك اليوم .

وتعود طفرة التعاملات خلال 2005 إلى تضخم السيولة بالسوق؛ نتيجة طرح الحكومة أسهم شركتي أموك وسيدبك للجمهور، وتحقيق المشترين لتلك الأسهم لأرباح وصلت إلى حوالي ضعف ثمن الشراء خلال أسابيع قليلة. وفي هذا الجو من توقع تكرار تلك الأرباح العالية من شراء الأسهم الحكومية، طرحت الحكومة نسبة 20% من أسهم الشركة المصرية للاتصالات، وهي الشركة الوحيدة المحتكرة لخطوط التليفونات الثابتة وكذلك الاتصالات الدولية؛ وهو ما جعل الجمهور يتكالب عليها.

أسباب طفرة 2005

وساهم تضخيم الشركة المروجة لأسهم الاتصالات لنسب الإقبال، ومبالغة وسائل الإعلام الرسمية في التوقعات لقيمة السهم بعد طرحه. في حدوث إقبال كبير على شراء أسهم شركة الاتصالات من جانب فئات شعبية تدخل البورصة للمرة الأولى، وليس لديها أي ثقافة استثمارية. ومع تخصيص عدد محدود من الأسهم لطالبي الشراء اتجه هؤلاء الداخلون الجدد لتوجيه فوائض الاكتتاب لشراء أسهم أخرى أو لإعادة شراء أسهم الاتصالات بأسعارها المرتفعة توقعا لارتفاع أسعارها.

وعلى صعيد المستثمرين العرب ساعدت الفوائض البترولية العربية في اتجاه كثيرين منهم للشراء بالبورصة المصرية؛ وهو ما زاد من الطلب خاصة مع انخفاض سعر قيمة الأسهم المصرية النسبي بالنسبة للمستثمرين العرب والأجانب. وزاد دور المضاربين في توجيه السوق -والذي يخلو من وجود صانع سوق يمكنه ترشيد الطفرات السعرية- وسادت سياسة القطيع في الشراء دون الاستناد إلى المعلومات أو البيانات المالية للشركات أو للتحليل الأساسي أو الفني. حتى زادت أسعار شركات بنسب عالية لا تتناسب بالمرة مع أدائها، بل إن بعض أسهم شركات الدواجن كانت تتجه للصعود رغم كارثة إنفلونزا الطيور التي شهدتها مصر .

وزاد عدد الأسهم المقيدة بالبورصة بنسبة 41% ليصل إلى 9 316_مليارات سهم. كما زاد رأس المال السوقي للشركات المقيدة بالبورصة بنسبة 95% ليصل إلى 456 مليار جنيه.

وبدأ التصحيح...

وبلغ عدد الشركات المقيدة عام 2005 بالبورصة 744 شركة بنقص 48 شركة عن العام السابق، وهي شركات محدودة التعامل تم شطبها لأسباب تتعلق بنقص شروط القيد، وهو أمر لم يؤثر على السوق التي تتميز بظاهرة تركز النشاط في نحو 50 شركة فقط. **وارتفع مؤشر أسعار البورصة المصرية (CASE30) بنسبة 146 .%**

إلا أن الأسعار لم تأخذ نفس الاتجاه الصعودي بعد أن بلغت مستويات لا تتفق مع واقع الشركات التي تنتمي إليها، ومن هنا فقد كان من الطبيعي أن تصحح السوق نفسها. خاصة مع حدوث نفس التصحيح بالأسواق الخليجية التي كانت قد شهدت طفرة في أسعارها خلال العام الماضي. وتضافر ذلك مع عدم تنسيق هيئة سوق المال نزول عدد من الاكتتابات في زيادة رؤوس أموال الشركات في نفس الوقت؛ وهو ما أدى لزيادة العرض.

وتعود طفرة التعاملات خلال 2005 إلى تضخم السيولة بالسوق؛ نتيجة طرح الحكومة أسهم شركتي أموك وسيدبك للجمهور، وتحقيق المشترين لتلك الأسهم لأرباح وصلت إلى حوالي ضعف ثمن الشراء خلال أسابيع قليلة. **وفي هذا الجو من توقع تكرار تلك الأرباح العالية من شراء الأسهم الحكومية، طرحت الحكومة نسبة 20% من أسهم الشركة المصرية للاتصالات، وهي الشركة الوحيدة المحتكرة لخطوط التليفونات الثابتة وكذلك الاتصالات الدولية؛ وهو ما جعل الجمهور يتكالب عليها .**

Example of Arabic Summarization (Cont.)

Page 2/5

محاولات للإنعاش أسباب طفرة 2005.

كما استخدمت الحكومة سلطانها في توجيه محافظ الأوراق المالية الضخمة بالبنوك الحكومية العامة للشراء، ونفس الأمر لبعض صناديق الاستثمار التابعة للبنوك العامة.

ومن هنا تماسكت السوق بل اتجهت للارتفاع بعض الوقت. إلا أن قوى السوق كان لا بد لها من أن تؤدي دورها فاستمرت الأسعار في التراجع.

حتى إنه مع إعلان وزير الاستثمار -الذي يشرف على السوق من قبل الحكومة- في احتفال كبير عن بدء إطلاق مؤشر داو جونز الخاص بالأسهم المصرية اتجهت الأسعار للتراجع في اليوم التالي مباشرة لإطلاق المؤشر. وحدث نفس الأثر للإعلان عن تكوين محافظ في أسواق دولية تستند محافظها إلى مكونات مؤشر البورصة الذي يضم الشركات الثلاثين الأكثر نشاطا.

أسباب الانخفاض

وجاءت انفجارات مدينة دهب السياحية خلال شهر إبريل 2006 وكان من الطبيعي أن تؤثر على الأسعار بالبورصة. إلا أن الحكومة تدخلت أيضا في إطار سياستها التي تتجه إلى الدعوى بأن أحداث دهب لم تؤثر على حركة السياحة أو الطيران وبالتالي على البورصة رغم أن واقع الحال الحقيقي غير ذلك.

ومع عودة سياسة القمع الحكومية تجاه حركات المجتمع المدني كان من الطبيعي أيضا أن تتأثر البورصة باعتبارها المرآة لكل ما يحدث بالمجتمع من مؤثرات على مناخ الاستثمار. وساهمت عدة عوامل في تراجع ثقة المستثمرين بالسوق. منها تراجع سعر أسهم المصرية للاتصالات لأقل من سعر الطرح الحكومي؛ وهو ما ألحق خسائر كبيرة لحائزيه، خاصة لمن اشتروه بقيم عالية من السوق.

كذلك انخفاض سعر سهم هيرميس القابضة كسهم قائد للسوق، وزادت حالة التشاؤم لدى صغار المتعاملين الذين أصبحت لهم النسبة الكبرى من التعامل بعد ابتعاد كثير من المؤسسات المالية عن السوق توقعا لاستمرار حالة الهبوط السعري حتى شهر أكتوبر القادم.

دور بورصات الخليج

وبدأ التصحيح.

وذكر هؤلاء أن كثيرا من المستثمرين الخليجين كانوا مقترضين جانبا من قيمة مشترياتهم من الأسهم، وأنه مع انخفاض الأسعار بأسواقهم طالبتهم البنوك المقرضة لهم بسداد الفرق عن أسعار الأسهم المنخفضة. لذا اتجهوا لتسييل محافظهم في مصر لتدبير سيولة لدفعها لتلك البنوك .

ولقد استمرت كثير من مؤشرات التعامل بالبورصة في النمو مع بداية العام الحالي 2006؛ ففي الثلث الأول من العام زادت قيمة التعامل بنسبة 207% لتصل إلى 119 مليار جنيه مقابل 39 مليار تحققت خلال الثلث الأول من 2005. وارتفع المتوسط اليومي لقيمة التعامل إلى 1. 457 مليار جنيه مقابل 491 مليون جنيه عن نفس الفترة العام الماضي. كما زاد عدد الأوراق المالية المتداولة بنسبة 78% وارتفع عدد الصفقات بنسبة 117%. مع الأخذ في الاعتبار انخفاض مؤشرات التعامل تدريجيا من يناير إلى إبريل.

توقيت حرج: جاء توقيت انهيار البورصة حرجا للحكومة المصرية التي تبنت تماسك الأسعار بالبورصة، والتي تستعد لافتتاح مؤتمر دافوس الشرق الأوسط بمدينة شرم الشيخ بعد 5 أيام من التظاهر في العشرين من مايو. وهو المؤتمر الذي تريد من خلاله الحكومة أن تؤكد ثقة المستثمرين العالميين بها خاصة بعد توالي أحداث العنف تجاه السياحة والشرطة وارتفاع حالة الاحتقان السياسي من جانب قطاعات من القضاة والصحفيين والأطباء ونقابات أخرى وبعض جمعيات حقوق الإنسان. ومن هنا تدخلت الحكومة لتتجه الأسعار للارتفاع بشكل واضح في اليوم التالي للتظاهرة مباشرة.

وهذا التدخل الحكومي بسوق الأوراق المالية المصرية يمنع حركتها من التعبير الحقيقي عن آليات السوق، والبورصة الطبيعية تحركها قوى العرض والطلب والمعلومات. حتى تكون مرآة صادقة عن الاقتصاد. ونظرا لأن الاقتصاد المصري يعاني من عجز مزمن بالميزان التجاري، وعجز مزمن بالموازنة العامة، ومن دين عام متزايد، ونسب عالية من البطالة والفقر وحالة من الغلاء، هذا بالإضافة إلى حالة احتقان سياسي غير مسبوق بالمجتمع المصري. فان هذه العوامل لا بد أن تلقي بظلالها على البورصة في الأجل القصير على الأقل، ومهما تدخلت الحكومة فإن قوى السوق لا بد أن تؤدي دورها ويكون لها الكلمة الأخيرة .

Example of Arabic Summarization (Cont.)

انـهيار البورصة المصرية

إنـهيار أسعار أسهم البورصة المصرية .

للمرة الثانية خلال شهرين يتظاهر مستثمرو البورصة المصرية مطالبين بإقالة رئيس البورصة، وجاءت التظاهرة الثانية على أثر تراجع البورصة والانخفاض بقيمة أغلب الأسهم المتداولة بمنتصف هذا الشهر (مايو 2006) بنسبة 4%؛ وهو ما أعاد للذاكرة ما حدث يوم الثلاثاء الأسود بشهر مارس لنفس العام من هبوط شديد بمؤشر البورصة، والتي حدت بهيئة سوق المال بإيقاف التداول ذلك اليوم .

وزاد عدد الأسهم المقيدة بالبورصة بنسبة 41% ليصل إلى
وارتفع مؤشر أسعار البورصة المصرية (CASE30) بنسبة 146 %.

وفي هذا الجو من توقع تكرار تلك الأرباح العالية من شراء الأسهم الحكومية، طرحت الحكومة نسبة 20% من أسهم الشركة المصرية للاتصالات، وهي الشركة الوحيدة المحتكرة لخطوط التليفونات الثابتة وكذلك الاتصالات الدولية؛ وهو ما جعل الجمهور يتكالب عليها .

ولقد استمرت كثير من مؤشرات التعامل بالبورصة فى النمو مع بداية العام الحالى 2006؛ ففى الثلث الأول من العام زادت قيمة التعامل بنسبة 207% لتصل إلى 119 مليار جنيه مقابل 39 مليار تحققت خلال الثلث الأول من 2005

After Using Sentence-Base Summarization Algorithm:

Number of Pages in the Summary: ½ out of 5
Number of Paragraphs in the Summary: 7 out of 33
Number of Sentences in the Summary: 7 out of 73

Example of Arabic Summarization (Cont.)

بعض الجمل التى تم حذفها لعدم أهميتها

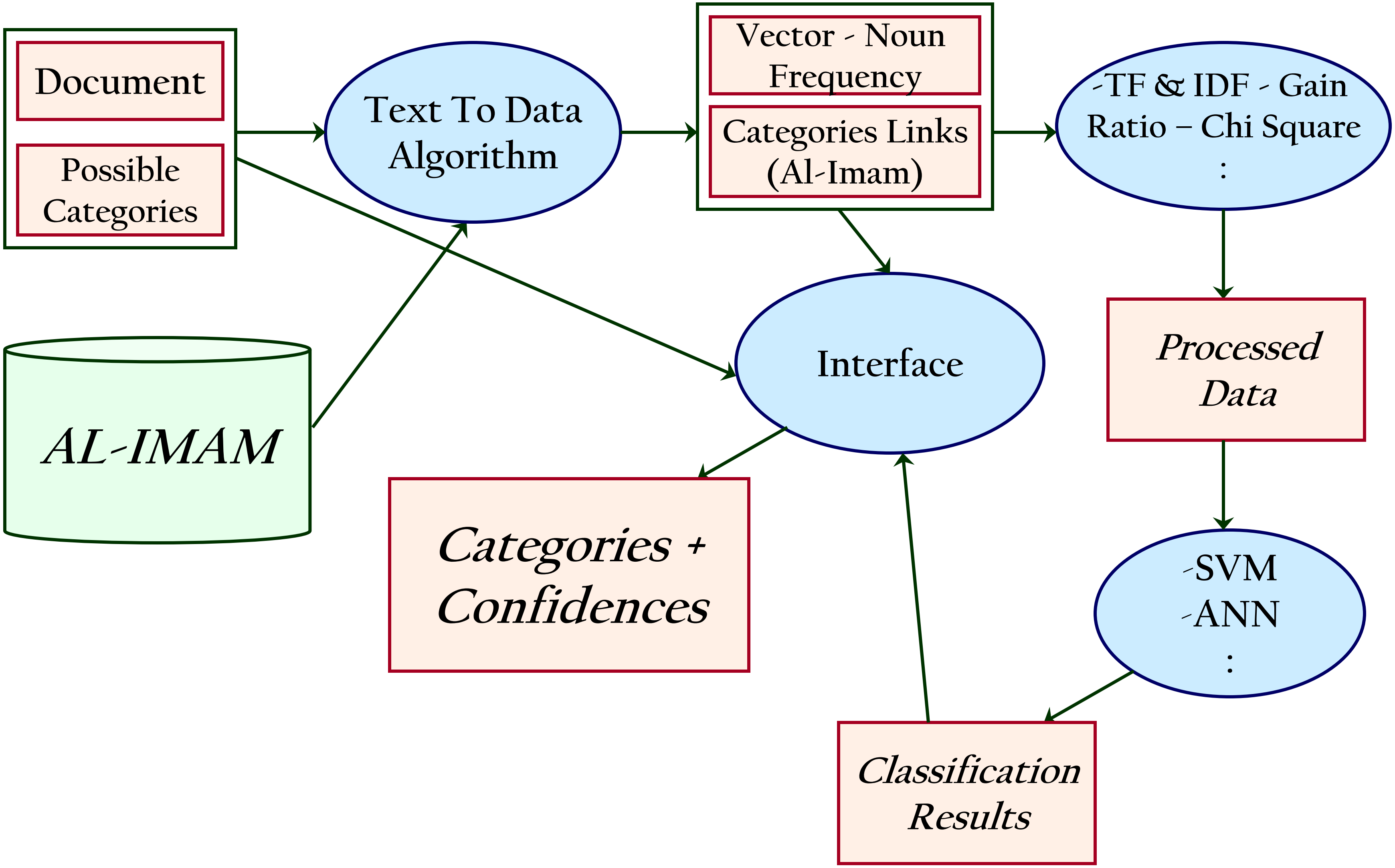
وتعود طفرة التعاملات خلال 2005 إلى تضخم السيولة بالسوق؛ نتيجة طرح الحكومة أسهم شركتي أموك وسيدبك للجمهور، وتحقيق المشترين لتلك الأسهم لأرباح وصلت إلى حوالي ضعف ثمن الشراء خلال أسابيع قليلة. وفي هذا الجو من توقع تكرار تلك الأرباح العالية من شراء الأسهم الحكومية، طرحت الحكومة نسبة 20% من أسهم الشركة المصرية للاتصالات، وهي الشركة الوحيدة المحتكرة لخطوط التليفونات الثابتة وكذلك الاتصالات الدولية؛ وهو ما جعل الجمهور يتكالب عليها.

أسباب طفرة 2005

وساهم تضخيم الشركة المروجة لأسهم الاتصالات لنسب الإقبال، ومبالغة وسائل الإعلام الرسمية في التوقعات لقيمة السهم بعد طرحه. في حدوث إقبال كبير على شراء أسهم شركة الاتصالات من جانب فئات شعبية تدخل البورصة للمرة الأولى، وليس لديها أي ثقافة استثمارية. ومع تخصيص عدد محدود من الأسهم لطالبي الشراء اتجه هؤلاء الداخلون الجدد لتوجيه فوائض الاكتتاب لشراء أسهم أخرى أو لإعادة شراء أسهم الاتصالات بأسعارها المرتفعة توقعا لارتفاع أسعارها.

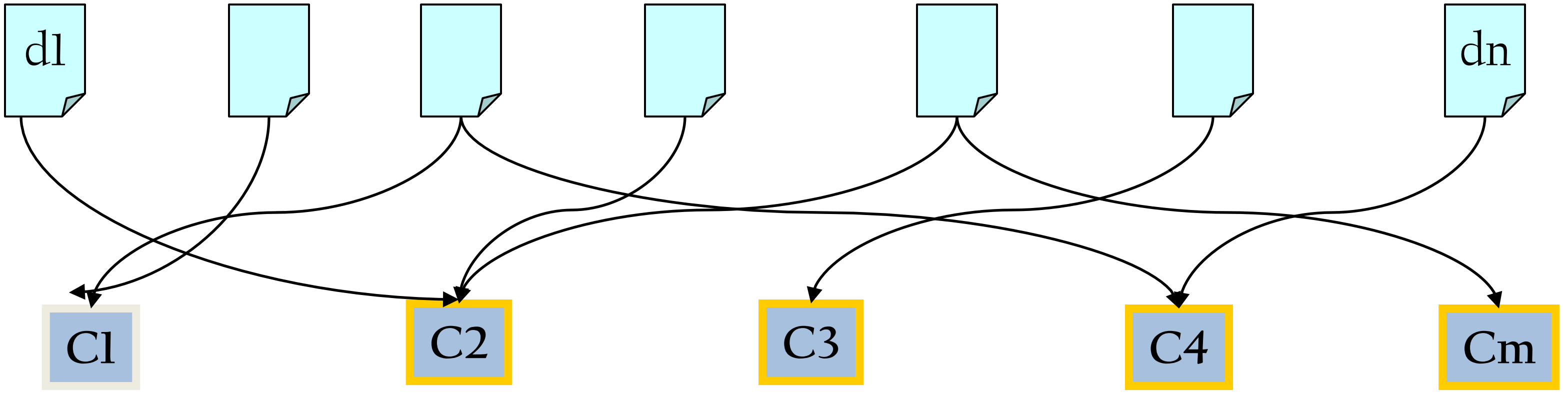
وعلى صعيد المستثمرين العرب ساعدت الفوائض البترولية العربية في اتجاه كثيرين منهم للشراء بالبورصة المصرية؛ وهو ما زاد من الطلب خاصة مع انخفاض سعر قيمة الأسهم المصرية النسبي بالنسبة للمستثمرين العرب والأجانب. وزاد دور المضاربين في توجيه السوق -والذي يخلو من وجود صانع سوق يمكنه ترشيد الطفرات السعرية- وسادت سياسة القطيع في الشراء دون الاستناد إلى المعلومات أو البيانات المالية للشركات أو للتحليل الأساسي أو الفني. حتى زادت أسعار شركات بنسب عالية لا تتناسب بالمرة مع أدائها، بل إن بعض أسهم شركات الدواجن كانت تتجه للصعود رغم كارثة إنفلونزا الطيور التي شهدتها مصر.

Supervised Text Categorization



Supervised Text Categorization

Text Categorization (TC) is the process of labeling electronic text documents with different labels



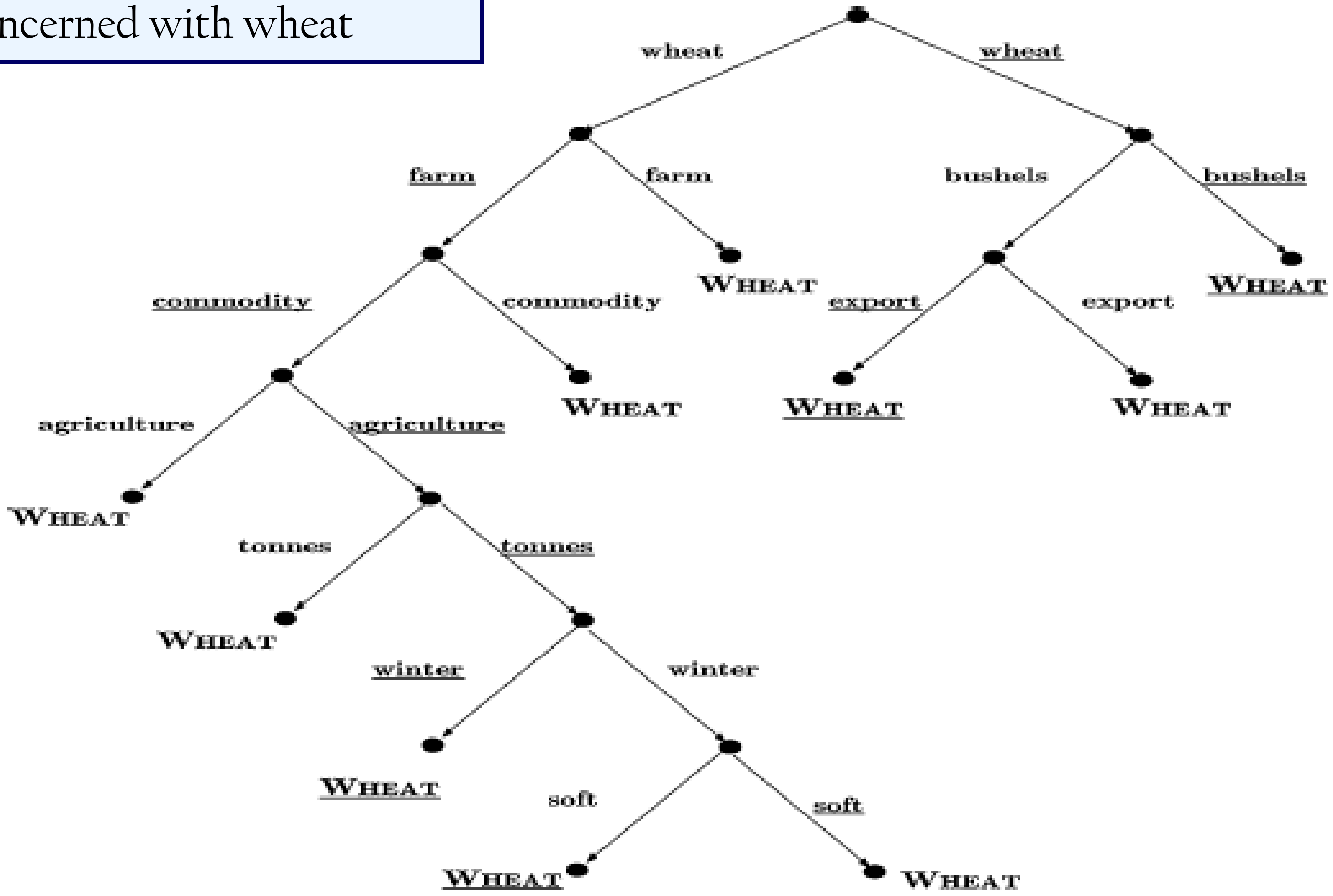
	d_1	$\dots$	$\dots$	d_j	$\dots$	$\dots$	d_n
c_1	a_{11}	$\dots$	$\dots$	a_{1j}	$\dots$	$\dots$	a_{1n}
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
c_i	a_{i1}	$\dots$	$\dots$	a_{ij}	$\dots$	$\dots$	a_{in}
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
c_m	a_{m1}	$\dots$	$\dots$	a_{mj}	$\dots$	$\dots$	a_{mn}

Supervised Text Categorization

	Supervised	Semi-supervised	Unsupervised
Input Documents	Labeled documents	Labeled and Unlabeled Documents	Unlabeled documents
Method	Machine Learning / Statistical Approaches	Clustering / Machine Learning / Statistical Approaches	Clustering / SOM / Similarity
References	*(Deng, Z. 2004)* *(Sebastiani F. ,2003)* *(Yang, Y. & Pedrson, J. 1997)*	*(Zeng, et al. 2003) (Nigam, et al.2000)*	*(Gliozzo, et al. 2005)* *(Zhao,Y.&Karypis,G. 2005)*

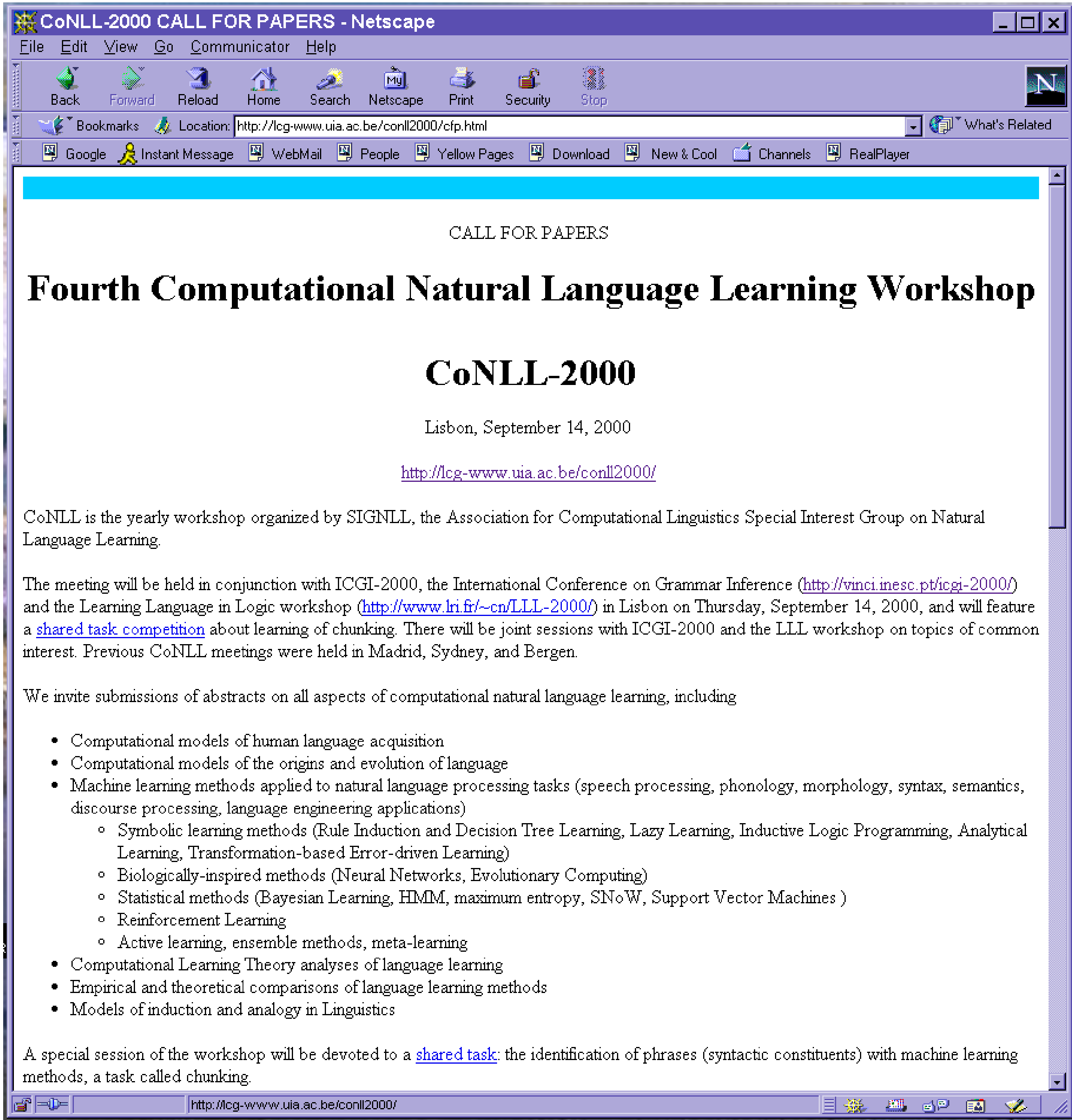
Learning Tree Categorization

Categorizing documents concerned with wheat



Document to categorize:

CFP for CoNLL-2000


CoNLL-2000 CALL FOR PAPERS - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://lcg-www.uia.ac.be/conll2000/cfp.html
What's Related
Google Instant Message WebMail People Yellow Pages Download New & Cool Channels RealPlayer
http://lcg-www.uia.ac.be/conll2000/

CALL FOR PAPERS

Fourth Computational Natural Language Learning Workshop

CoNLL-2000

Lisbon, September 14, 2000

http://lcg-www.uia.ac.be/conll2000/

CoNLL is the yearly workshop organized by SIGNLL, the Association for Computational Linguistics Special Interest Group on Natural Language Learning.

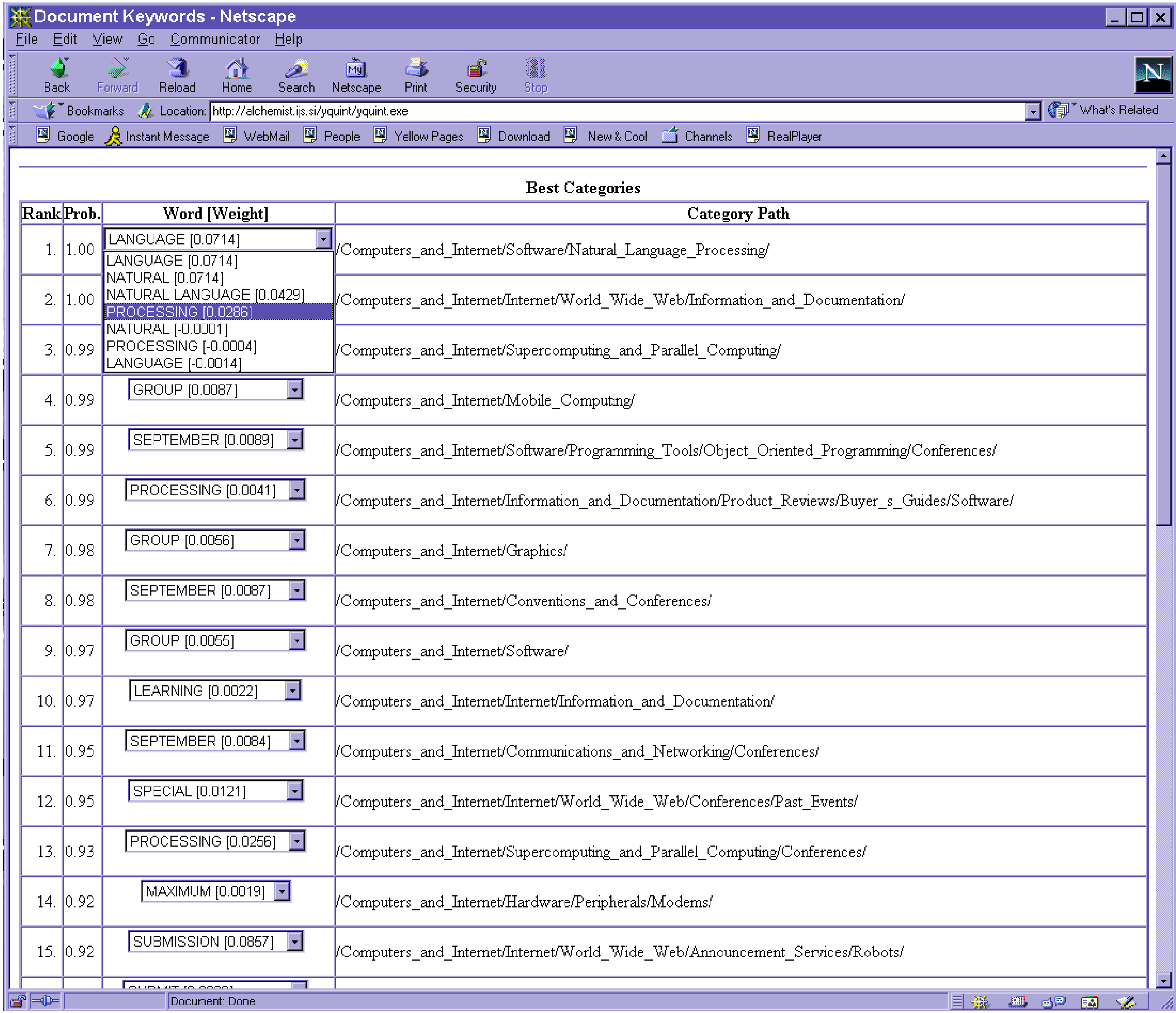
The meeting will be held in conjunction with ICGI-2000, the International Conference on Grammar Inference (http://vinci.inesc.pt/icgi-2000/) and the Learning Language in Logic workshop (http://www.lri.fr/~cn/LLL-2000/) in Lisbon on Thursday, September 14, 2000, and will feature a shared task competition about learning of chunking. There will be joint sessions with ICGI-2000 and the LLL workshop on topics of common interest. Previous CoNLL meetings were held in Madrid, Sydney, and Bergen.

We invite submissions of abstracts on all aspects of computational natural language learning, including

- Computational models of human language acquisition
- Computational models of the origins and evolution of language
- Machine learning methods applied to natural language processing tasks (speech processing, phonology, morphology, syntax, semantics, discourse processing, language engineering applications)
  - Symbolic learning methods (Rule Induction and Decision Tree Learning, Lazy Learning, Inductive Logic Programming, Analytical Learning, Transformation-based Error-driven Learning)
  - Biologically-inspired methods (Neural Networks, Evolutionary Computing)
  - Statistical methods (Bayesian Learning, HMM, maximum entropy, SNoW, Support Vector Machines)
  - Reinforcement Learning
  - Active learning, ensemble methods, meta-learning
- Computational Learning Theory analyses of language learning
- Empirical and theoretical comparisons of language learning methods
- Models of induction and analogy in Linguistics

A special session of the workshop will be devoted to a shared task: the identification of phrases (syntactic constituents) with machine learning methods, a task called chunking.

Some predicted categories


Document Keywords - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks
What's Related
Google Instant Message WebMail People Yellow Pages Download New & Cool Channels RealPlayer

Location: http://alchemist.ijs.si/yquint/yquint.exe

Best Categories

Rank	Prob.	Word [Weight]	Category Path
1.	1.00	LANGUAGE [0.0714] LANGUAGE [0.0714] NATURAL [0.0714] NATURAL LANGUAGE [0.0429] PROCESSING [0.0286] NATURAL [-0.0001] PROCESSING [-0.0004] LANGUAGE [-0.0014]	/Computers_and_Internet/Software/Natural_Language_Processing/
2.	1.00		/Computers_and_Internet/Internet/World_Wide_Web/Information_and_Documentation/
3.	0.99		/Computers_and_Internet/Supercomputing_and_Parallel_Computing/
4.	0.99	GROUP [0.0087]	/Computers_and_Internet/Mobile_Computing/
5.	0.99	SEPTEMBER [0.0089]	/Computers_and_Internet/Software/Programming_Tools/Object_Oriented_Programming/Conferences/
6.	0.99	PROCESSING [0.0041]	/Computers_and_Internet/Information_and_Documentation/Product_Reviews/Buyer_s_Guides/Software/
7.	0.98	GROUP [0.0056]	/Computers_and_Internet/Graphics/
8.	0.98	SEPTEMBER [0.0087]	/Computers_and_Internet/Conventions_and_Conferences/
9.	0.97	GROUP [0.0055]	/Computers_and_Internet/Software/
10.	0.97	LEARNING [0.0022]	/Computers_and_Internet/Internet/Information_and_Documentation/
11.	0.95	SEPTEMBER [0.0084]	/Computers_and_Internet/Communications_and_Networking/Conferences/
12.	0.95	SPECIAL [0.0121]	/Computers_and_Internet/Internet/World_Wide_Web/Conferences/Past_Events/
13.	0.93	PROCESSING [0.0256]	/Computers_and_Internet/Supercomputing_and_Parallel_Computing/Conferences/
14.	0.92	MAXIMUM [0.0019]	/Computers_and_Internet/Hardware/Peripherals/Modems/
15.	0.92	SUBMISSION [0.0857]	/Computers_and_Internet/Internet/World_Wide_Web/Announcement_Services/Robots/

Document: Done

PROBABILITY

Part 4

-Introduction
-Terminology

What Is Probability?

- A priori probability ***P(e)***: The chance that e happens
- Conditional probability ***P(f / e)***: The chance of f given e
- Joint probability ***P(e, f)***: The chance of e and f both happening; If e and f are independent, then P(e, f) = P(e) * P(f); If e and f are dependent then P(e, f) = P(e) * P(f | e)

  For example, if e stands for "the first roll of the die comes up 5" and f stands for "the second roll of the die comes up 3," then P(e,f) = P(e) * P(f) = 1/6 * 1/6 = 1/36.

$$\sum_e P(e) = 1 \qquad \sum_e P(e \mid f) = 1$$

BASIC Probabilities

$$P(A\cup B)=\begin{cases} P(A)+P(B) & A\,\&\,B\ \textit{are not dependant} \\ P(A)+P(B)-P(A,B) & A\,\&\,B\ \textit{are dependant} \end{cases}$$

- For example, when drawing a single card at random from a regular deck of cards, the chance of getting a heart or a face card (J,Q,K) (or one that is both) is

$$\frac{13}{52}+\frac{12}{52}-\frac{3}{52}=\frac{22}{52}$$

A	$P(A) \in [0,1]$
not A	$P(A') = 1 - P(A)$
A or B	$P(A\cup B) = P(A) + P(B) - P(A\cap B)$ $= P(A) + P(B)$ if A and B are mutually exclusive
A and B	$P(A\cap B) = P(A\|B)P(B)$ $= P(A)P(B)$ if A and B are independent
A given B	$P(A \mid B) = \dfrac{P(A\cap B)}{P(B)}$

Inference Using Probability

	Toothache		~Toothache	
	Catch	~Catch	Catch	~Catch
Cavity	0.108	0.012	0.072	0.008
~Cavity	0.016	0.064	0.144	0.576

$$P(Cavity \vee Toothache) = 0.108 + 0.012 + 0.072 + 0.008 + 0.016 + 0.064 = 0.28$$

$$P(Cavity) = 0.108 + 0.012 + 0.072 + 0.008 = 0.2$$

$$P(Cavity \mid Toothache) = \frac{0.108 + 0.012}{0.108 + 0.012 + 0.016 + 0.064} = 0.6$$

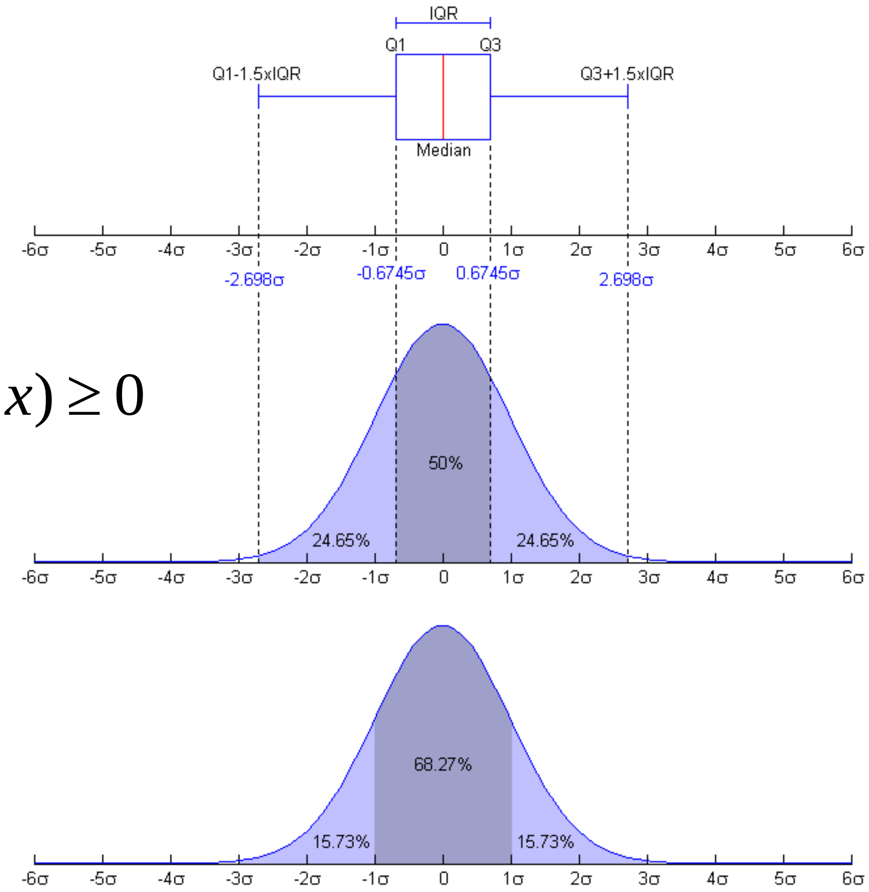
$$P(\sim Cavity \mid Toothache) = \frac{0.016 + 0.064}{0.108 + 0.012 + 0.016 + 0.064} = 0.4$$

Probability Density Function PDF

- Probability density function (pdf) is a function that represents a probability distribution in terms of integrals

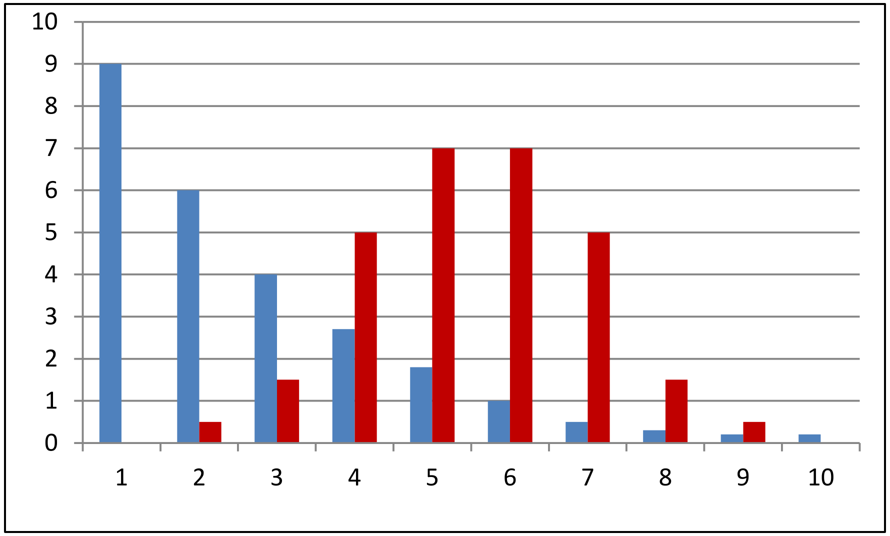
$$\int_{a}^{b} f(x)\,dx$$

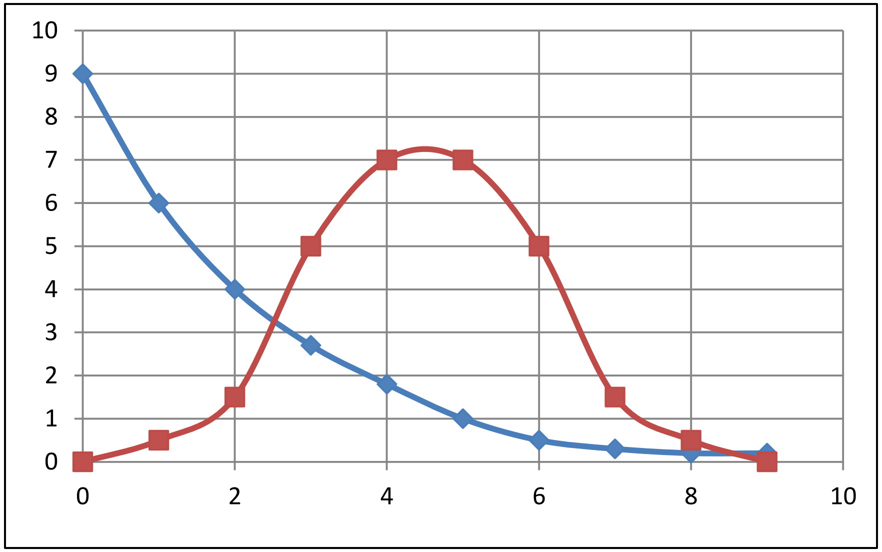
$$\int_{-\infty}^{\infty} f(x)dx = 1 \qquad \& \qquad f(x) \geq 0$$



Probability Density Function PDF

- The Summation is used with Discrete Data





Conditional & Bayesian Probability

- **Conditional probability** is the probability of some event *A*, given the occurrence of some other event *B*; *it* is written $P(A|B)$, and is read "the probability of *A*, given *B*"

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

- Bayesian probability, the probability of a hypothesis given the data (the *posterior*), is proportional to the product of the likelihood times the prior probability (often just called the *prior*)
- The likelihood brings in the effect of the data, while the prior specifies the belief in the hypothesis before the data was observed

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

- If two variables A and B are independent

$$P(A \wedge B|C) = P(A|C)P(B|C)$$

Text Mining

Part 5

Preprocessing

Text Preprocessing

- Remove “fluff” if exists (e.g., ads, navigation bars, pictures, etc.)
- Convert to plain text (i.e., from PDF, DOC, or other formats)
- Check words correctness (in case of erroneous text or using OCR)
- Handle tables, numbers, and equations

- حذف التشكيل (َ ً ِ ٍ ُ ٌ)
- حذف الرموز الخاصة (% * $ @ # & /)
- حذف الأرقام
- حذف الزوائد في بداية الكلمة وآخرها (استـ ها)
- تحويل همزات القطع إلى همزات وصل (أحمد احمد)
- تحويل الألف اللينة إلى ألف العالية
- حذف الكلمات الزائدة Stop words

Preprocessing: Sentence Splitter

Sentence Splitting

- Sentences end with “.”, “!”, or “?”
- Difficult when a “.” do not indicate an EOS: “MR. X”, “3.14”, “Y Corp.”, etc.
- We can detect common abbreviations (“U.S.”), but what if a sentence ends with one? “. . .announced today by the U.S. The ...

توجد نفس المشاكل فى اللغة العربية:

- ”وقدم أ.د. إبراهيم إمام درس عن ...“
- الجمل فى اللغة العربية تتداخل بصورة أكثر تعقيدا

Google n-gram corpus Statistics: http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html#links Size = 24 GB

Number of tokens:	1,024,908,267,229
Number of sentences:	95,119,665,584
Number of unigrams:	13,588,391
Number of bigrams:	314,843,401
Number of trigrams:	977,069,902
Number of fourgrams:	1,313,818,354
Number of fivegrams:	1,176,470,663

Samples of Google n-gram Data

3-gram samples	Freq.
ceramics collectables collectibles	55
ceramics collectables fine	130
ceramics collected by	52
ceramics collectible pottery	50
ceramics collectibles cooking	45
ceramics collection ,	144
ceramics collection .	247
ceramics collection </S>	120
ceramics collection and	43
ceramics collection at	52
ceramics collection is	68
ceramics collection of	76
ceramics collection |	59
ceramics collections ,	66
ceramics collections .	60
ceramics combined with	46
ceramics come from	69
ceramics comes from	660
ceramics community ,	109
ceramics community .	212
ceramics community for	61
ceramics companies .	53
ceramics companies consultants	173
ceramics company !	4432
ceramics company ,	133
ceramics company .	92

4-gram samples	Freq.
serve as the incoming	92
serve as the incubator	99
serve as the independent	794
serve as the index	223
serve as the indication	72
serve as the indicator	120
serve as the indicators	45
serve as the indispensable	111
serve as the indispensible	40
serve as the individual	234
serve as the industrial	52
serve as the industry	607
serve as the info	42
serve as the informal	102
serve as the information	838
serve as the informational	41
serve as the infrastructure	500
serve as the initial	5331
serve as the initiating	125
serve as the initiation	63
serve as the initiator	81
serve as the injector	56
serve as the inlet	41
serve as the inner	87
serve as the input	1323
serve as the inputs	189

Preprocessing: Word Tokenizers

Tokenization is difficult: For example,

“John’s sick”   shall we split “John’s” into one token or two?

If one ! problems in parsing (where’s the verb?)

If two ! what do we do with John’s house?

فى اللغة العربية توجد مشاكل أكثر تعقيدا من ذلك ***Heavy Compounding***

مثلا:

- جملة ”يلعبونها فى الملاعب“ عند حذف السوابق واللواحق يتبقى ”لعب“ وتم حذف الفاعل ”هم“ والمفعول به ”هى“

أيضا إذا كان الكلام يحتوى على تركيبة كيميائية، أو هياكل خاصة بالعلوم:

1,4--xylanase II from Trichoderma reesei
When N-formyl-L-methionyl-L-leucyl-L-phenylalanine (fMLP) was injected. . .
Technetium-99m-CDO-MeB [Bis[1,2-cyclohexanedionedioximato(1-)-O]-[1,2-cyclohexanedione dioximato(2-)-O]methyl-borato(2-)-N,N0,N00,N000,N0000,N00000)-
chlorotechnetium) belongs to a family of compounds. . .

Preprocessing: Morphological Analyzers

Morphological Analyzer

- Reflects changes in case, gender, number, tense, etc.

  give → gives, gave, given
- ***Stemming*** reduce words to a base form
- ***Lemmatization*** reduce words to their lemma (root)

التحليل الصرفى لكلمة: الفِلاَحة

إنسانى	معرف	الجنس	الوزن	الجذر	الساق	اللواحق	السوابق	النوع	الكلمة
✓	✓	مؤنث	فعال	فلح	فلاح	ـة	ال	مصدر	الفِلاَحة

Advantages of Using the Stem as a Word Representative:

- Simple and Fast

Disadvantages of Using the Stem as a Word Representative:

- Can create words that do not exist in the language, e.g., computers → comput
- Often reduces different words to the same stem, e.g., army, arm → arm; stocks, stockings → stock

Preprocessing: Morphological Analyzers (Cont.)

Advantages of Using the Root as a Word Representative:

- The root is an actual word
- Usually provide better accuracy than the stem

Disadvantages of Using the Root as a Word Representative:

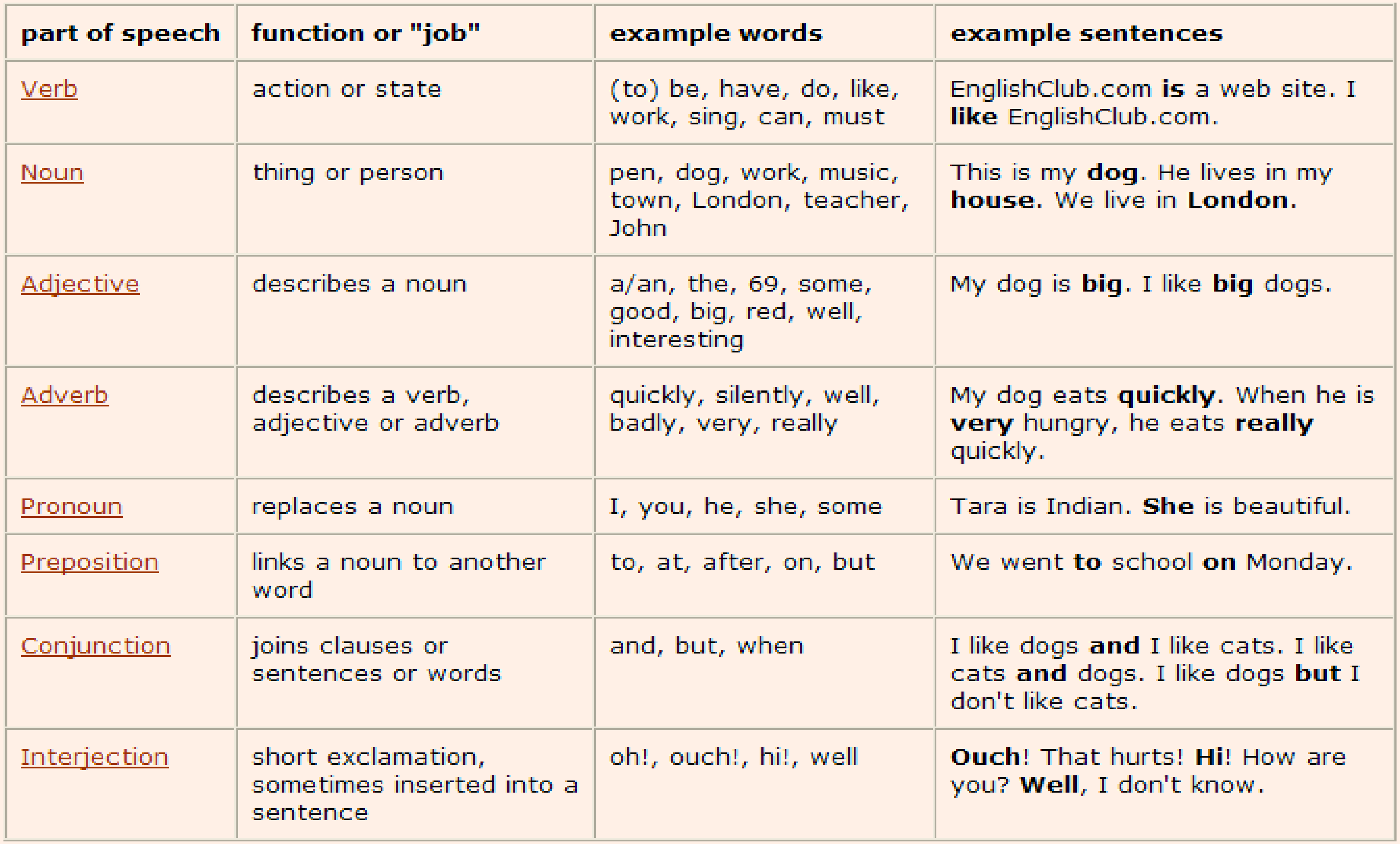
- Significantly complex
- Requires language dependent resources

Get a copy of Porter stemmer (For English) at:

http://www.tartarus.org/~martin/PorterStemmer/

Preprocessing: Part of Speech Tagging (POS)

- A Tagger algorithm assigns a tag for each word statistically
- calculated based on different word order probabilities



part of speech	function or "job"	example words	example sentences
Verb	action or state	(to) be, have, do, like, work, sing, can, must	EnglishClub.com **is** a web site. I **like** EnglishClub.com.
Noun	thing or person	pen, dog, work, music, town, London, teacher, John	This is my **dog**. He lives in my **house**. We live in **London**.
Adjective	describes a noun	a/an, the, 69, some, good, big, red, well, interesting	My dog is **big**. I like **big** dogs.
Adverb	describes a verb, adjective or adverb	quickly, silently, well, badly, very, really	My dog eats **quickly**. When he is **very** hungry, he eats **really** quickly.
Pronoun	replaces a noun	I, you, he, she, some	Tara is Indian. **She** is beautiful.
Preposition	links a noun to another word	to, at, after, on, but	We went **to** school **on** Monday.
Conjunction	joins clauses or sentences or words	and, but, when	I like dogs **and** I like cats. I like cats **and** dogs. I like dogs **but** I don't like cats.
Interjection	short exclamation, sometimes inserted into a sentence	oh!, ouch!, hi!, well	**Ouch**! That hurts! **Hi**! How are you? **Well**, I don't know.

Preprocessing: Part of Speech Tagging (POS)

Verb
work!

Noun	Verb
John	works.

Pronoun	Verb	Noun
He	loves	cats.

Noun	Verb	Verb
John	is	working.

Noun	Verb	Noun	Adverb
Ahmed	speaks	French	well.

Noun	Verb	Adjective	Noun
cats	like	nice	children.

Pronoun	Verb	Preposition	Adjective	Noun	Adverb
She	ran	to	the	station	quickly.

Pronoun	Verb	Adjective	Noun	Conjunction	Pronoun	Verb	Pronoun
She	likes	big	snakes	but	I	hate	them.

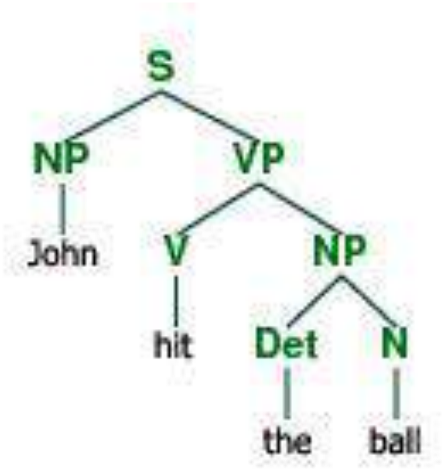
Interjection	Pronoun	Conjunction	Adjective	Noun	Verb	Prep.	Noun	Adverb
Well,	she	and	young	John	walk	to	school	Slowly.

Preprocessing: Syntactic Analysis

- ***Parsing:*** generating a parse tree for the given sentence (needs a grammar, and a lexicon)
- ***Chunking:*** finding syntactic constituents like Noun Phrases (NPs) or Verb Groups (VGs) within a sentence

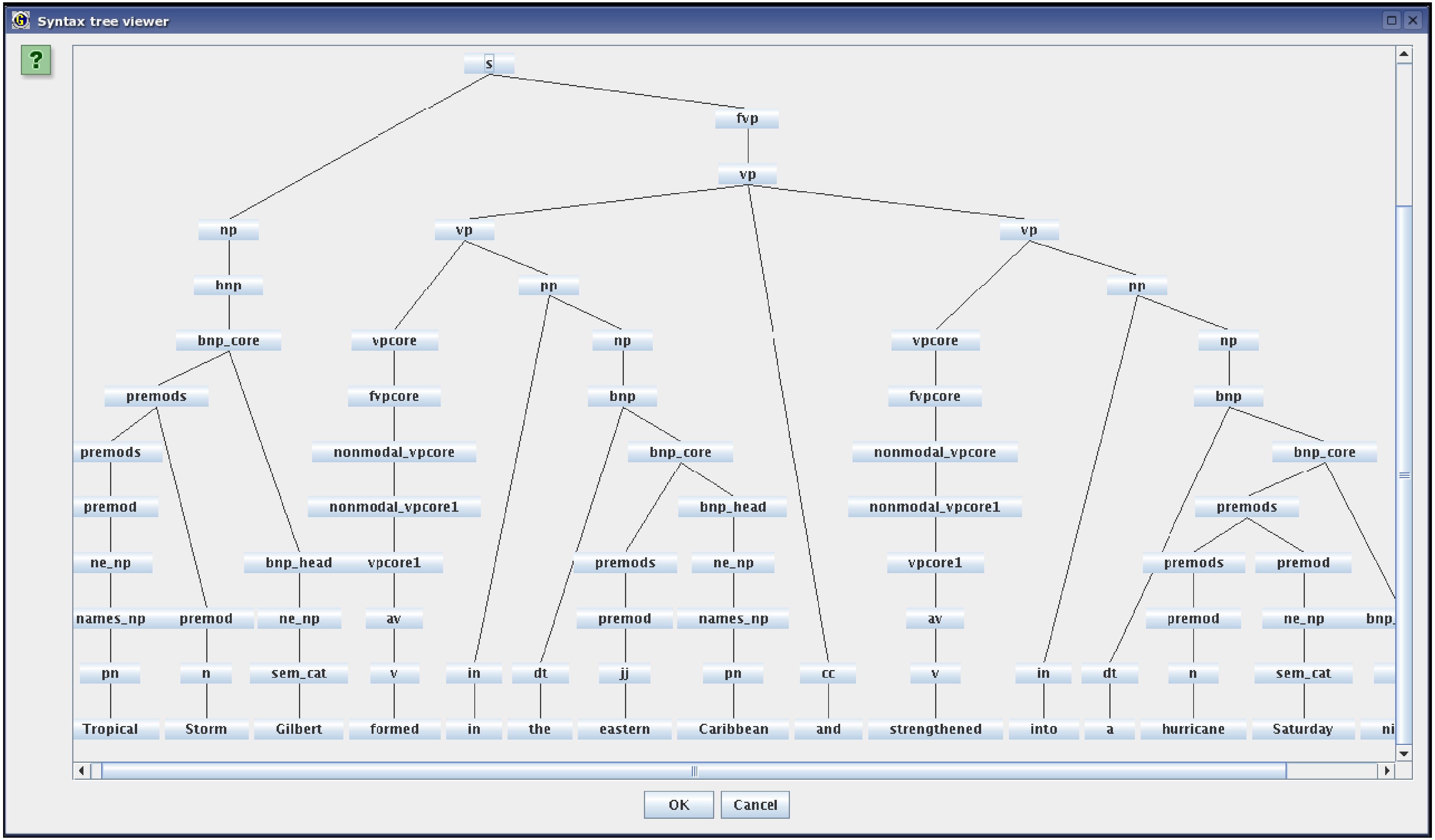
- Parse trees can help in determining relationships such as:
  Who invented X?
  What company created product Y?
  Which organism is this protein coming from?

- Chunks are very useful in finding named entities (NEs), e.g., Persons, Companies, Locations, Patents, Organisms,

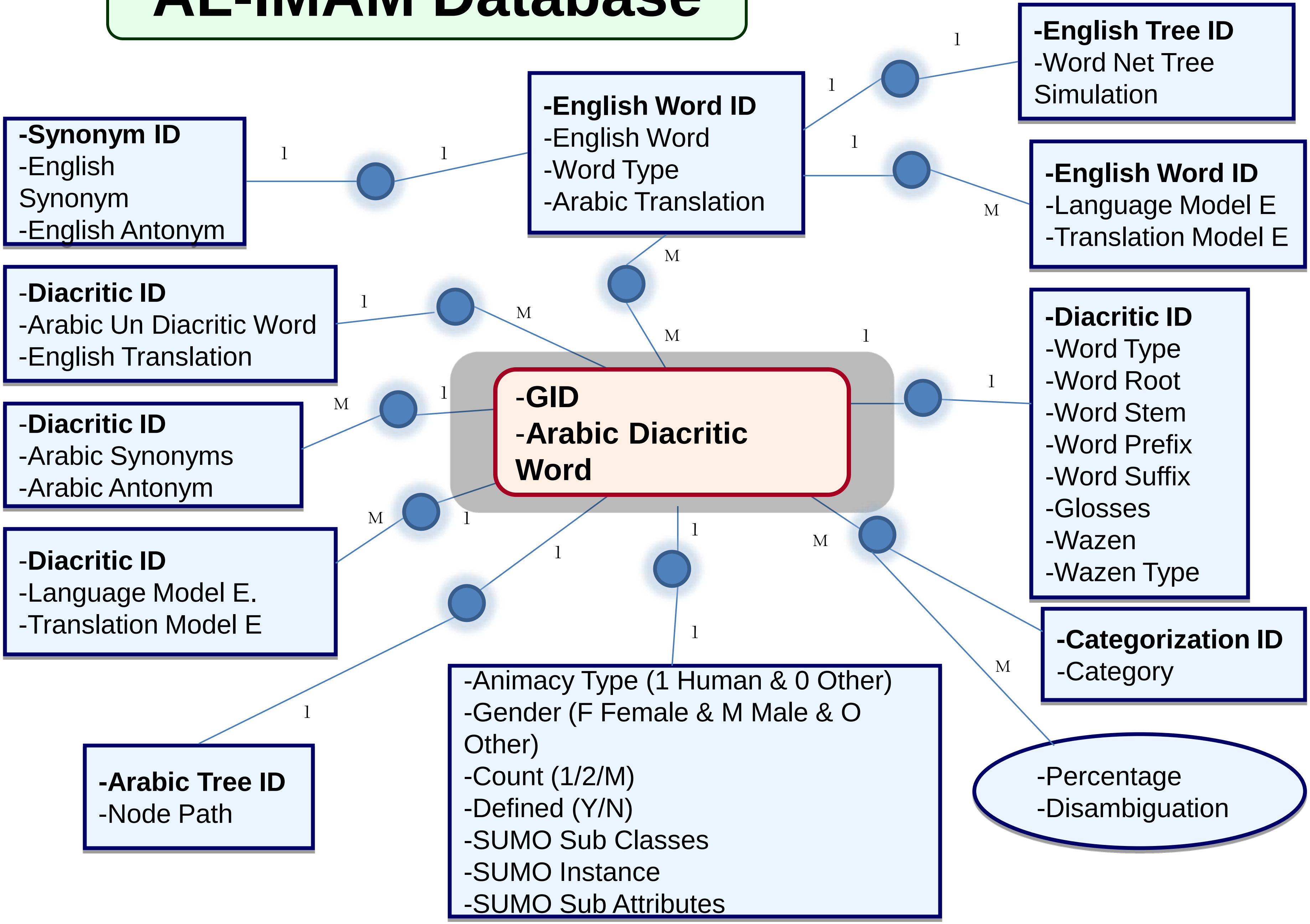


A Parse Tree

Another Example of a Parse Tree



AL-IMAM Database



-GID
-Arabic Diacritic Word

-English Word ID
-English Word
-Word Type
-Arabic Translation

-English Tree ID
-Word Net Tree Simulation

-English Word ID
-Language Model E
-Translation Model E

-Synonym ID
-English Synonym
-English Antonym

-Diacritic ID
-Arabic Un Diacritic Word
-English Translation

-Diacritic ID
-Arabic Synonyms
-Arabic Antonym

-Diacritic ID
-Language Model E.
-Translation Model E

-Arabic Tree ID
-Node Path

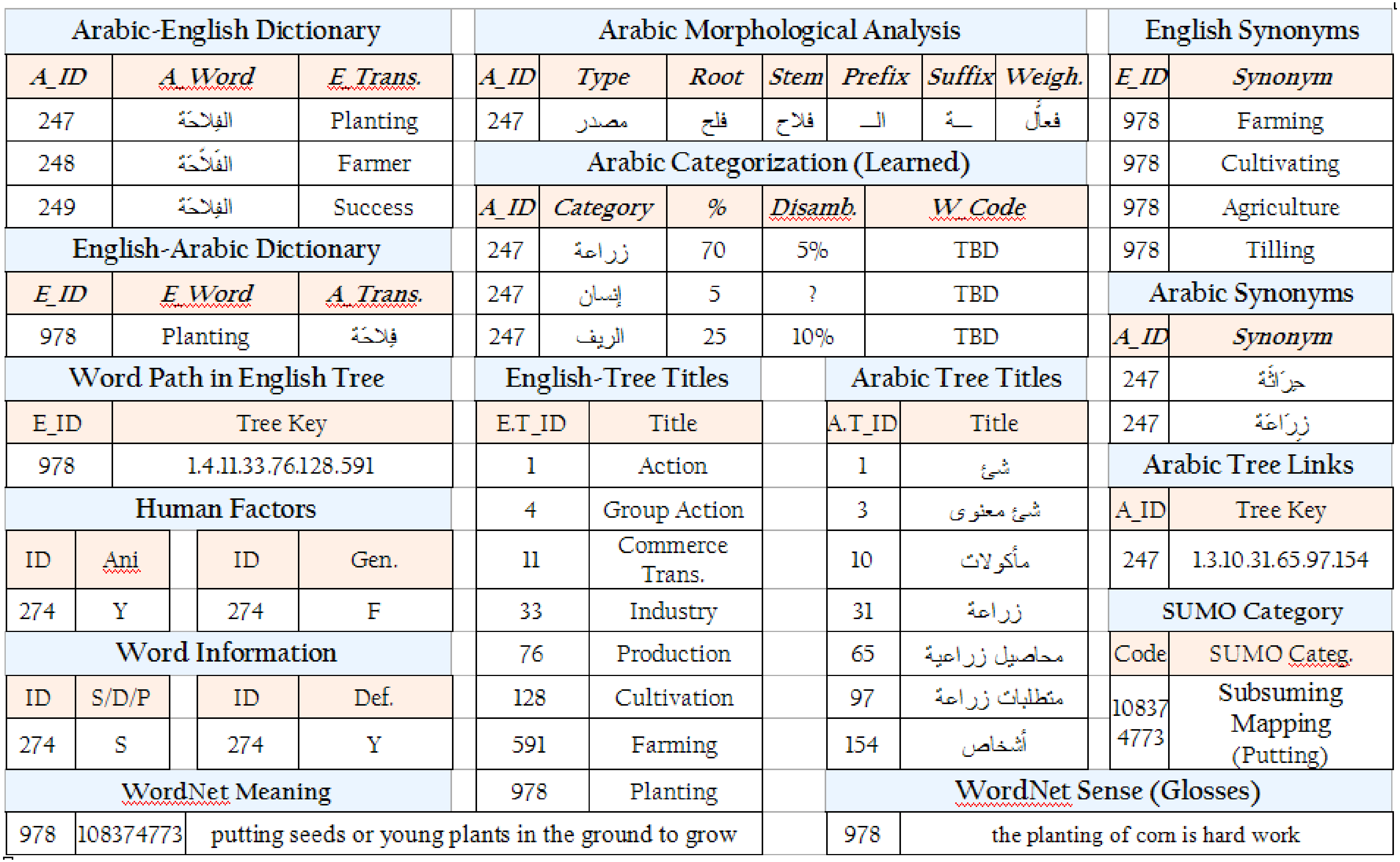
-Animacy Type (1 Human & 0 Other)
-Gender (F Female & M Male & O Other)
-Count (1/2/M)
-Defined (Y/N)
-SUMO Sub Classes
-SUMO Instance
-SUMO Sub Attributes

-Diacritic ID
-Word Type
-Word Root
-Word Stem
-Word Prefix
-Word Suffix
-Glosses
-Wazen
-Wazen Type

-Categorization ID
-Category

-Percentage
-Disambiguation

AL-IMAM Database



Arabic-English Dictionary

A_ID	A_Word	E_Trans.
247	الفِلاحَة	Planting
248	الفَلاّحَة	Farmer
249	الفَلاحَة	Success

English-Arabic Dictionary

E_ID	E_Word	A_Trans.
978	Planting	فِلاحَة

Word Path in English Tree

E_ID	Tree Key
978	1.4.11.33.76.128.591

Human Factors

ID	Ani	ID	Gen.
274	Y	274	F

Word Information

ID	S/D/P	ID	Def.
274	S	274	Y

WordNet Meaning

978	108374773	putting seeds or young plants in the ground to grow

Arabic Morphological Analysis

A_ID	Type	Root	Stem	Prefix	Suffix	Weigh.
247	مصدر	فلح	فلاح	الـــ	ـــة	فِعالَ

Arabic Categorization (Learned)

A_ID	Category	%	Disamb.	W_Code
247	زراعة	70	5%	TBD
247	إنسان	5	?	TBD
247	الريف	25	10%	TBD

English-Tree Titles

E.T_ID	Title
1	Action
4	Group Action
11	Commerce Trans.
33	Industry
76	Production
128	Cultivation
591	Farming
978	Planting

Arabic Tree Titles

A.T_ID	Title
1	شئ
3	شئ معنوى
10	مأكولات
31	زراعة
65	محاصيل زراعية
97	متطلبات زراعة
154	أشخاص

WordNet Sense (Glosses)

978	the planting of corn is hard work

English Synonyms

E_ID	Synonym
978	Farming
978	Cultivating
978	Agriculture
978	Tilling

Arabic Synonyms

A_ID	Synonym
247	حِرَاثَة
247	زِرَاعَة

Arabic Tree Links

A_ID	Tree Key
247	1.3.10.31.65.97.154

SUMO Category

Code	SUMO Categ.
108374773	Subsuming Mapping (Putting)

STATISTICS

Part 6

Introduction

Statistics

- Statistics is a Mathematical Science pertaining to the ***collection***, ***analysis***, ***interpretation*** *or* ***explanation***, *and* ***presentation*** of data

Statistical Terminologies

- Measures of Central Tendency ***(Mean***, Median, Mode)

$$\bar{x} = (1/n)\sum_{i=1}^{n} x_i$$

- ***Population Variance*** measures statistical dispersion of data points from the expected value (mean)

$$Var(X) = E\left[(X - E(X))^2\right]$$
$$= (1/n)\sum_{i=1}^{n}(x_i - \bar{x})^2 = \sigma^2$$

- ***Standard Deviation*** is a measure of the variability or dispersion of a population; Low SD indicates very close data points to the mean; High SD indicates spread out data points

$$sd(X) = \sqrt{\sigma^2}$$

- ***Covariance*** measures how much two variables change together

$$Cov(X,Y) = E\left[(X - E(X))(Y - E(Y))\right]$$

- ***Correlation*** (coefficient) indicates the strength and direction of a *linear* relationship between two random variables

$$Corr(X,Y) = \frac{Cov(X,Y)}{sd(X)*sd(Y)} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

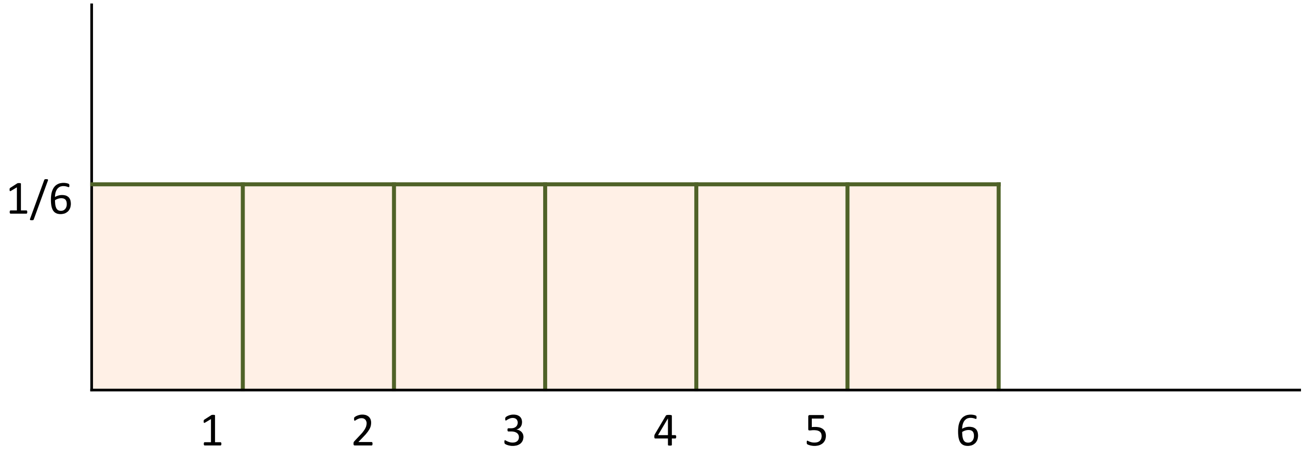
Popular Distributions

Probability Distribution identifies the probability of each value of an unidentified random variable

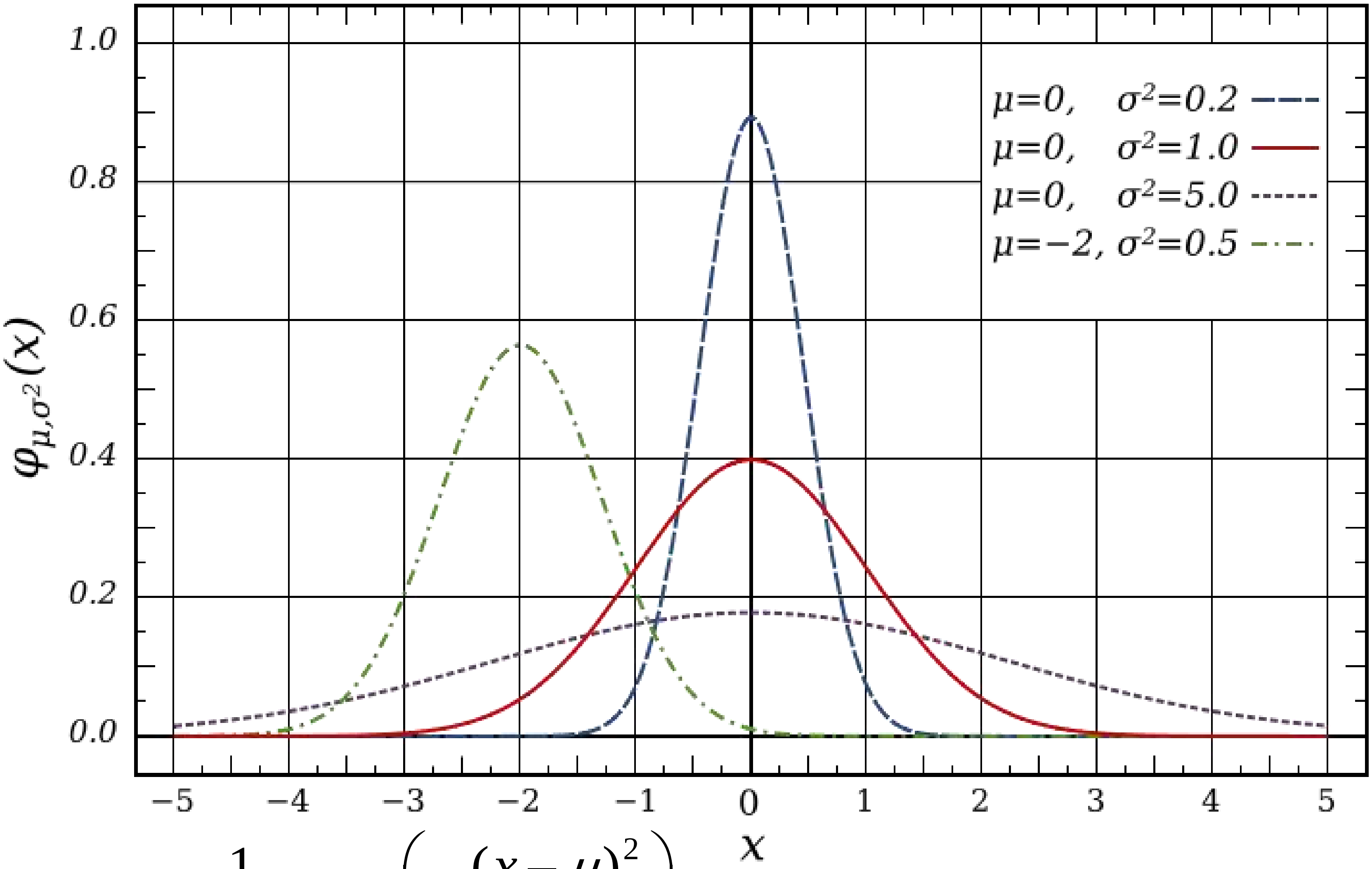
- *Uniform Distribution*
- *Normal (Gaussian) Distribution*
- *Chi-Square Distribution*
- *Exponential Distribution*
- *Poisson Distribution*
- *T Distribution*
- *F Distribution*

The Uniform Distribution

- The probability is equal for all outcomes
- Suppose a fair dice is thrown, the probability of getting any of its 6 faces equal to 1/6
- The area under the line equal to 1

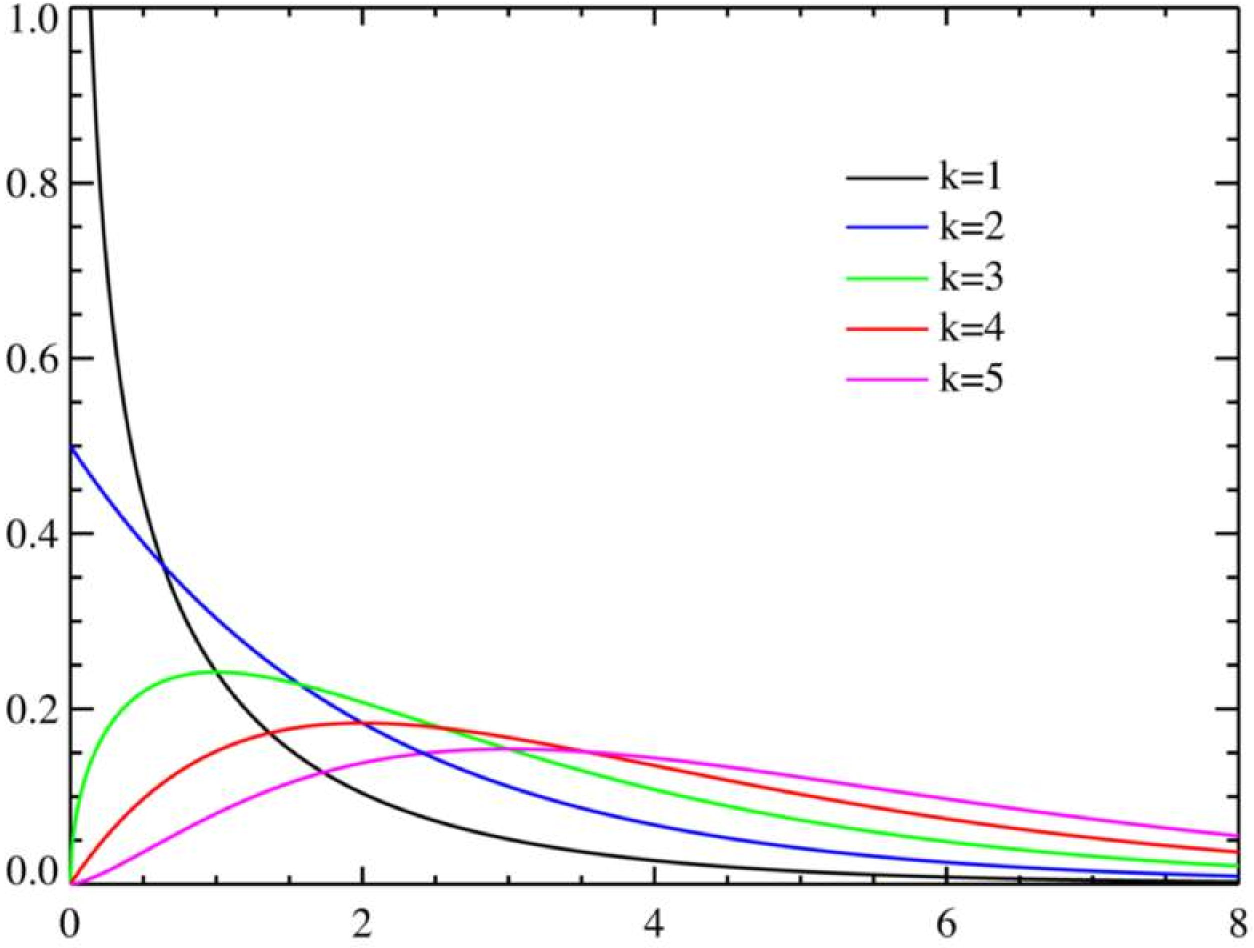


The Normal/Gaussian Distribution



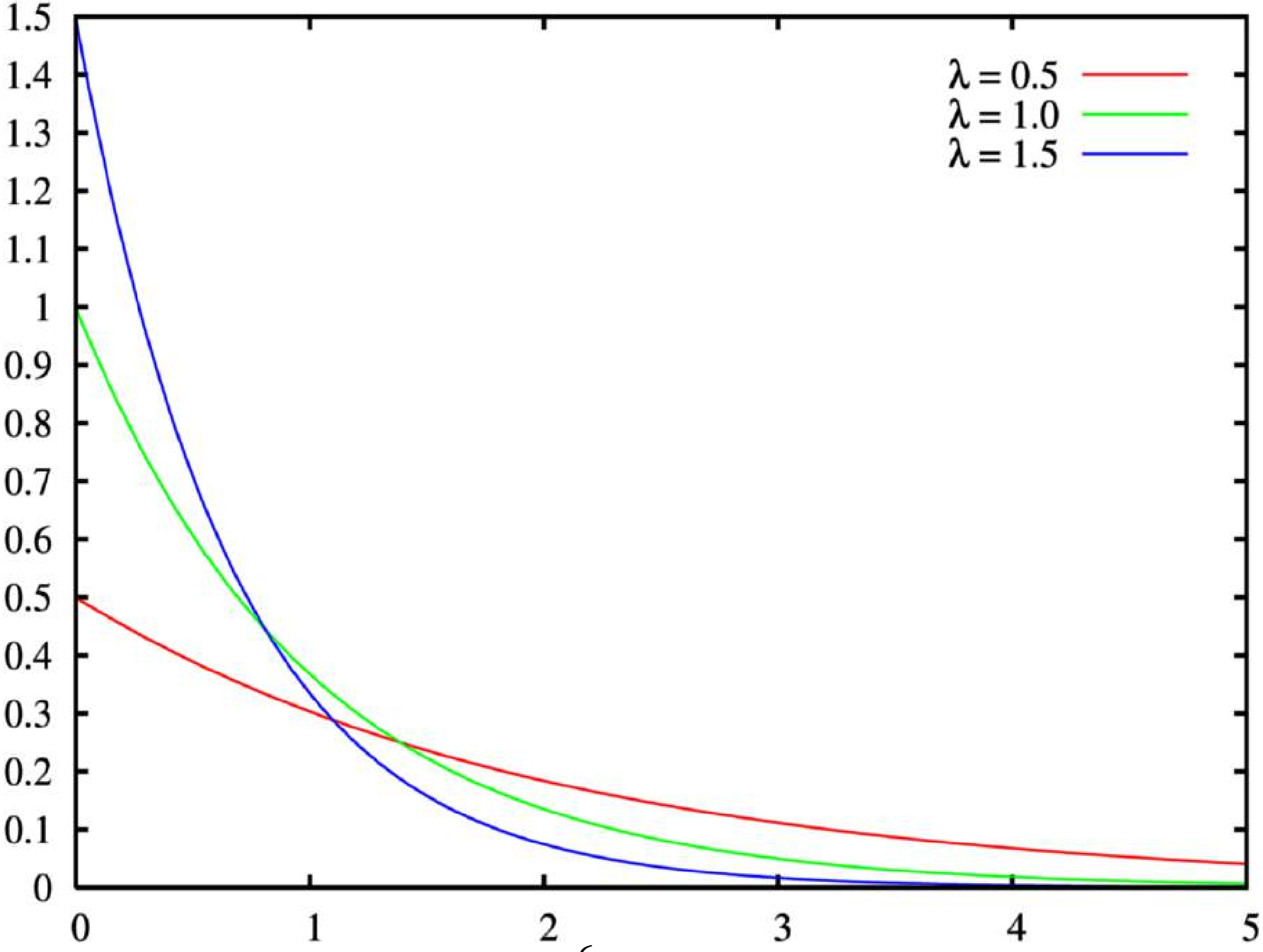
$$P(x)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

The Chi-Square Distribution



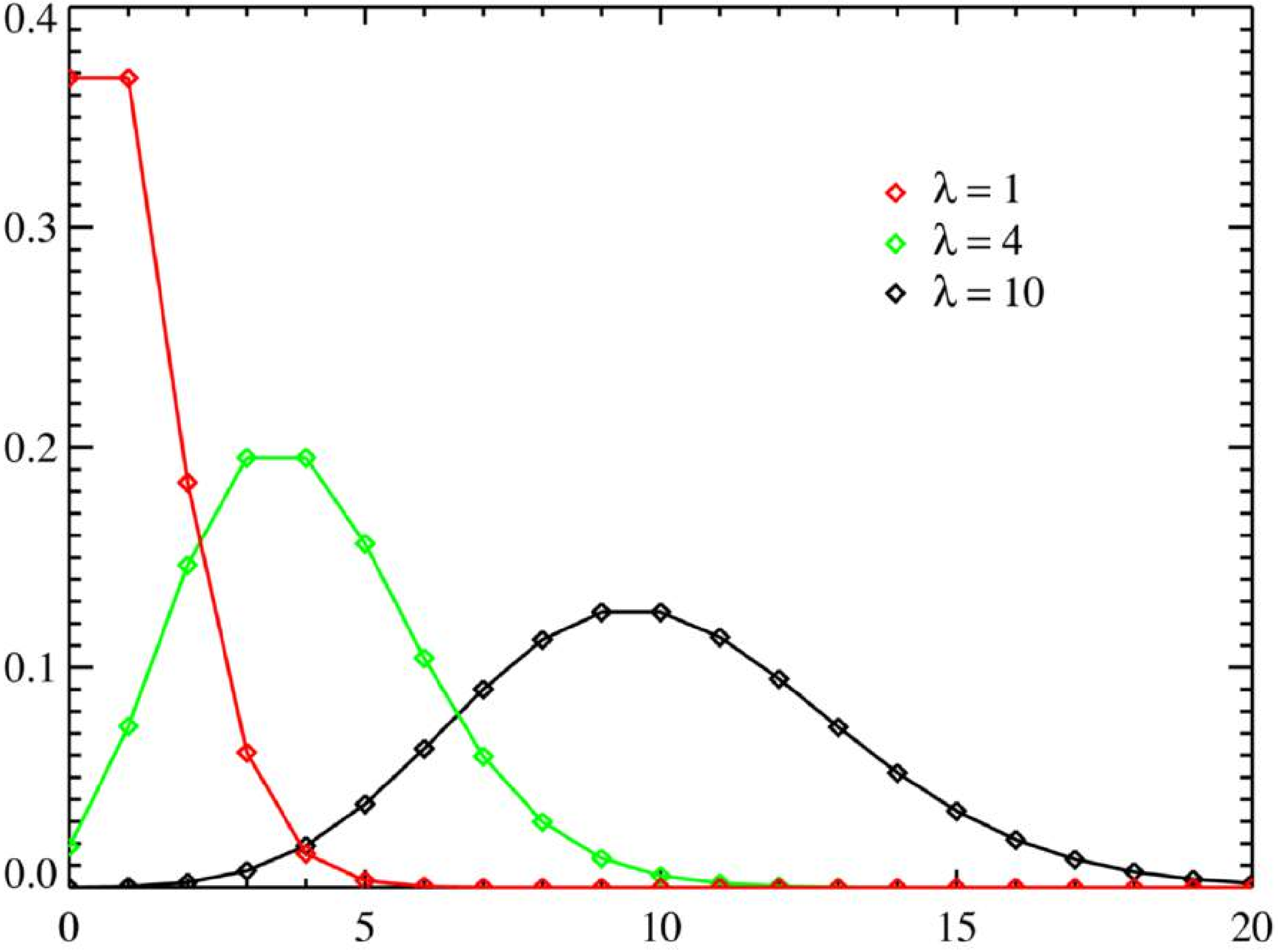
$$f(x;k) = \begin{cases} \dfrac{1}{2^{k/2}\Gamma(k/2)} x^{(k/2)-1} e^{-x/2} & \mathit{for}\ x > 0 \\ 0 & \mathit{for}\ x \le 0 \end{cases}$$

The Exponential Distribution



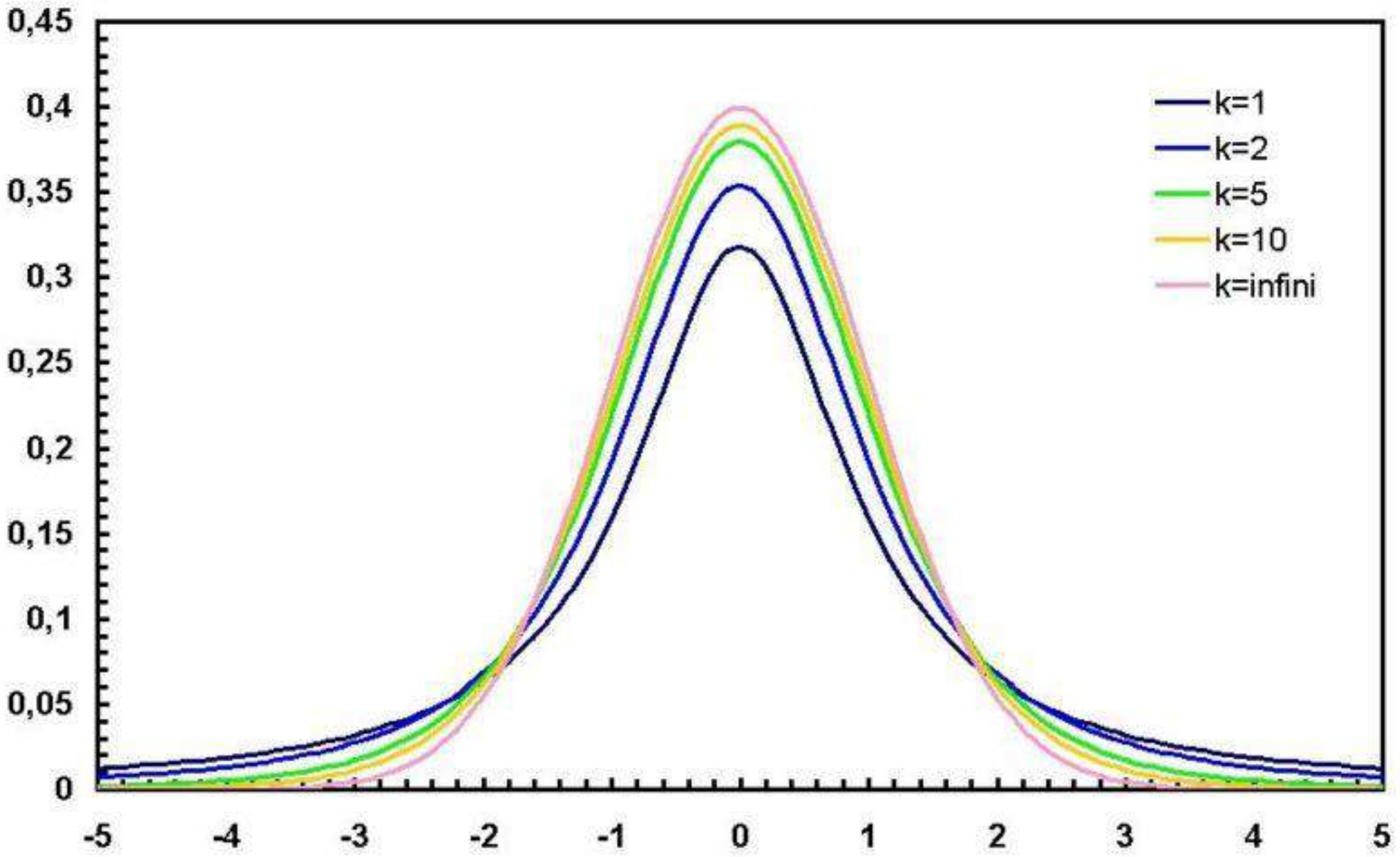
$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & for\ x > 0 \\ 0 & for\ x \le 0 \end{cases}$$

The Poisson Distribution



$$f(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

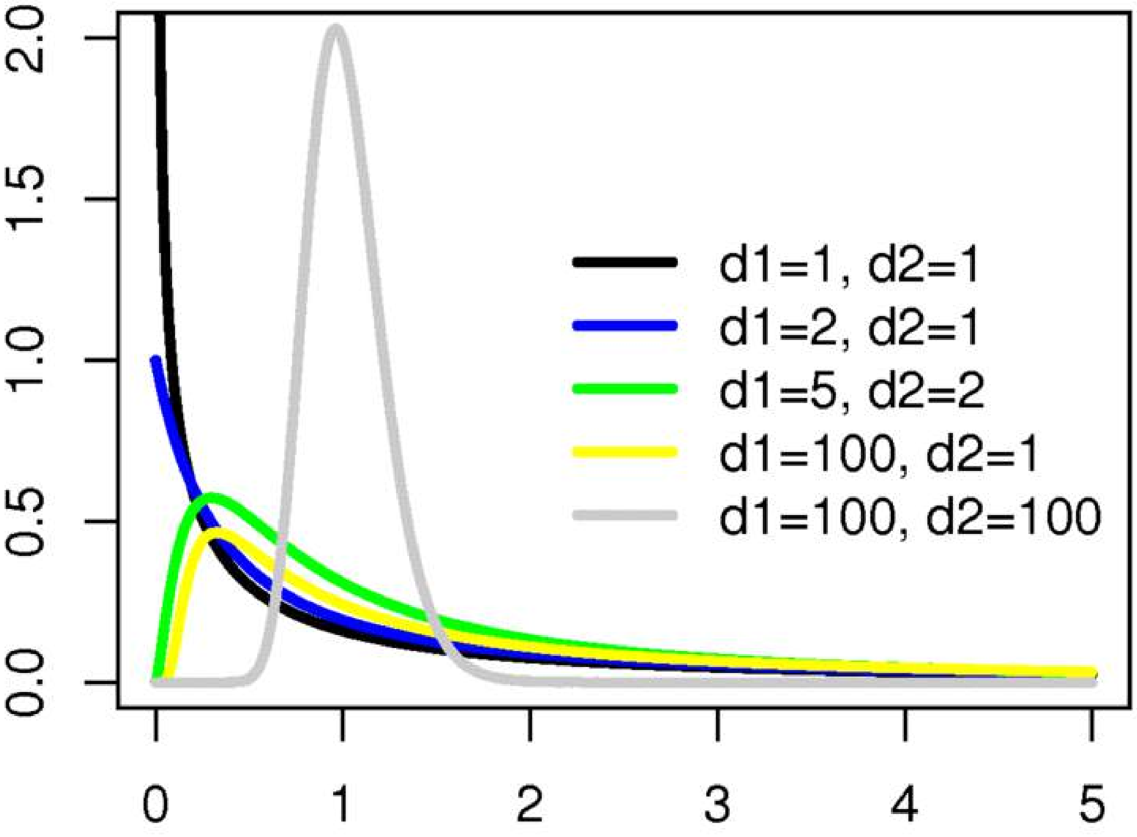
The T Distribution



$$f(t) = \frac{\Gamma(\frac{v+1}{2})}{\sqrt{v\pi}\,\Gamma(\frac{v}{2})}\left(1+\frac{t^2}{v}\right)^{-(v+1)/2}$$

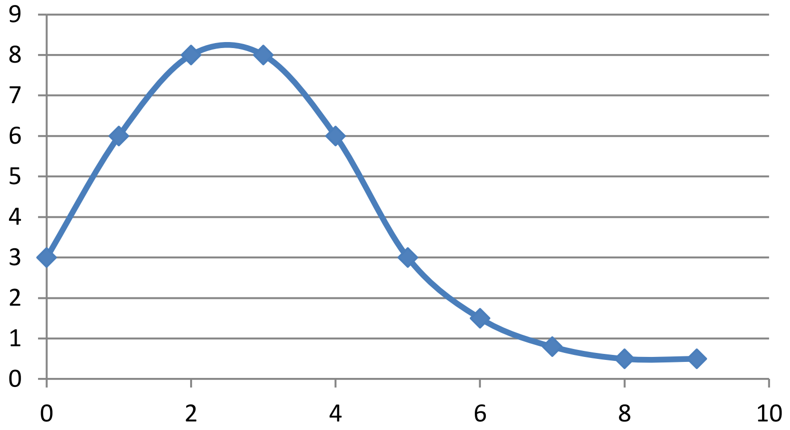
***t*-distribution** arises in the problem of estimating the mean of a normally distributed population when the sample size is small

The F Distribution



$$f(x) = \frac{\sqrt{\frac{(d_1\,x)^{d_1}\,\,d_2^{d_2}}{(d_1\,x+d_2)^{d_1+d_2}}}}{x\,\mathrm{B}\left(\frac{d_1}{2},\frac{d_2}{2}\right)}$$

Fitting Chi-Square



Vector
a

15
14
11
11
6
5
5

$$\max \quad \chi^2 = \sum_{i=1}^{n} \frac{(a_i - E_i)^2}{E_i}$$

$$E_{ij} = (15+14+11+11+6+5+5)/7 = 9.57$$

$$\chi^2 = (1/9.57) * ((15-9.57)^2 + (14-9.57)^2 + (11-9.57)^2 + (11-9.57)^2 + (6-9.57)^2 + (5-9.57)^2 + (5-9.57)^2) = 107.71/9.57 = 11.26$$

Measuring Term-Category Correlation

$$\chi^2(t_k, c_i) = \frac{[P(t_k, c_i)P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i)P(\bar{t}_k, c_i)]^2}{P(t_k)P(\bar{t}_k)P(c_i)P(\bar{c}_i)}$$

$P(t_k, c_i)$ ➔probability document x contains term t and belongs to category c.

$P(\bar{t}_k, c_i)$ ➔probability document x does not contain term t and belongs to category c.

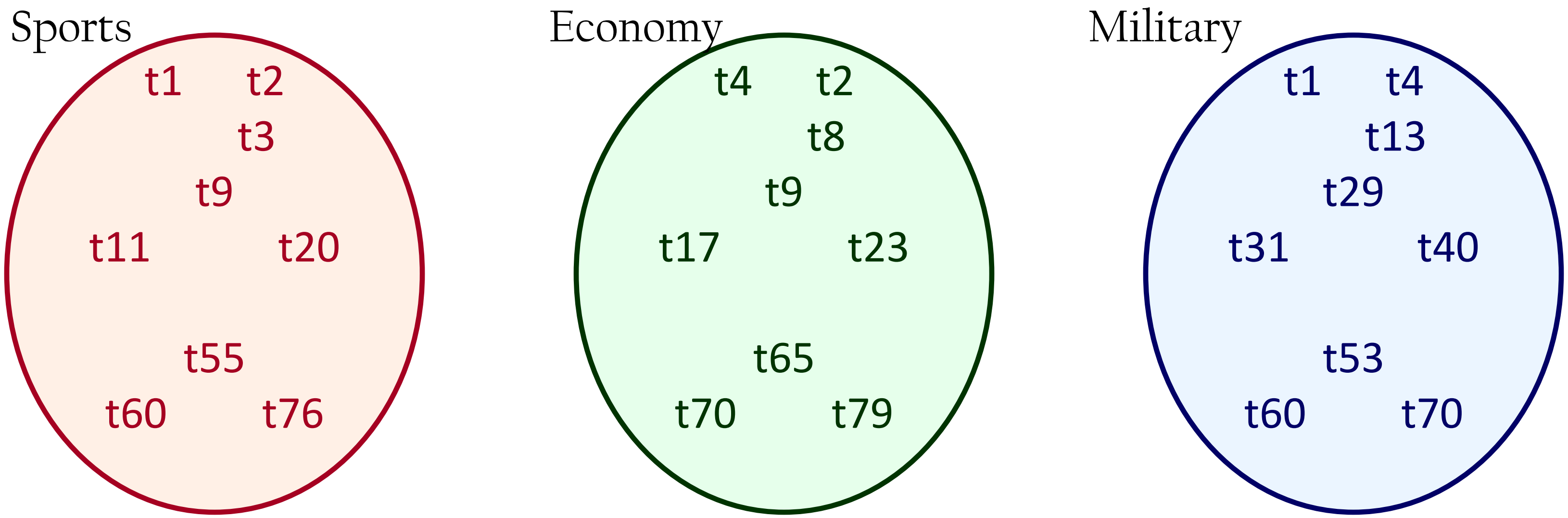
$P(t_k, \bar{c}_i)$ ➔probability document x contains term t and does not belong to category c.

$P(\bar{t}_k, \bar{c}_i)$ ➔probability document x does not contain term t and does not belong to category c.

$P(t)$ ➔probability of term t

$P(c)$ ➔probability of category c

Testing The Membership



Sports

t1 t2

t3

t9

t11 t20

t55

t60 t76

Economy

t4 t2

t8

t9

t17 t23

t65

t70 t79

Military

t1 t4

t13

t29

t31 t40

t53

t60 t70

$$\chi^2(t_k,c_i)=\frac{[P(t_k,c_i)P(\bar{t}_k,\bar{c}_i)-P(t_k,\bar{c}_i)P(\bar{t}_k,c_i)]^2}{P(t_k)P(\bar{t}_k)P(c_i)P(\bar{c}_i)}$$

$$\chi^2(t_1,Sports)=\frac{\left[\frac{1}{9}*\frac{17}{18}-\frac{1}{18}*\frac{8}{9}\right]^2}{\frac{2}{27}*\frac{25}{27}*\frac{9}{27}*\frac{18}{27}}$$

Using Chi-Square for Categorization

Another Example:

Term	**Frequency per Category**				**Total**
	Communication	Phone	Business	Army	
Link	15	6	2	12	35
Wire	10	12	0	8	30
Total	25	18	2	20	**65**

$$\chi^2(link, phone) = \frac{[6/65)*(18/65)-(29/65)*(12/65)]^2}{(35/65)*(30/65)*(18/65)*(47/65)}$$

Using Chi-Square for Multiple sets of Terms

Group 1	Category		Total
	News	Sports	
Term 1	3	2	5
Term 2	0	4	4
Term 3	2	3	5
Total	5	9	14

Group 2	Category		Total
	News	Sports	
Term 5	1	3	4
Term 7	4	6	10
Total	5	9	14

$$\chi^2 = \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{(a_{ij}-E_{ij})^2}{E_{ij}}$$

$$E_{ij} = {(T_{ci} * T_{vj})}/{T}$$

$$\chi^2(Group1) = (3-1.78)^2/1.78 + (2-3.21)^2/3.21 + (0-1.42)^2/1.42$$
$$+ (4-2.57)^2/2.57 + (2-1.78)^2/1.78 + (3-3.21)^2/3.21 = 3.62$$

$$\chi^2(Group\,2) = (1-1.42)^2/1.42 + (3-2.57)^2/2.57 + (4-3.57)^2/3.57$$
$$+ (6-6.43)^2/6.43 =$$

Mingers, J., (1989a). "An Empirical Comparison of selection Measures for Decision-Tree Induction", *Machine Learning*, Vol. 3, No. 3, (pp. 319-342), Kluwer Academic Publishers.

Attribute Selection Criteria: Chi-Square

Example

- T2 is quantized into two intervals 21 (T2<=21) and (T2>21)
- T3 is quantized into two intervals 15 (T3<=15) and (T3>15)

T1	T2	T3	T4	D
1	25	10	A	1
1	30	30	A	0
1	35	25	B	0
1	22	35	B	0
1	19	10	B	1
2	22	30	A	1
2	33	18	B	1
2	14	5	A	1
2	31	15	B	1
3	21	20	A	0
3	15	10	A	0
3	25	20	B	1
3	18	20	B	1
3	20	36	B	1

T2	Decision D		Total
	0	1	
<=21	1	3	4
>21	4	6	10
Total	5	9	14

T1	Decision D		Total
	0	1	
1	3	2	5
2	0	4	4
3	2	3	5
Total	5	9	14

T3	Decision D		Total
	0	1	
<=15	1	4	5
>15	4	5	9
Total	5	9	14

T4	Decision D		Total
	0	1	
A	3	3	6
B	2	6	8
Total	5	9	14

Attribute Selection Criteria: Chi-Square

$$\chi^2(A) = \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{(a_{ij}-E_{ij})^2}{E_{ij}}$$

where A is the attribute to be evaluated against the decision attribute, n is the number of distinct values of A, m is the number of distinct values of the decision attribute, a_{ij} is the correlation frequency of value number i from A and value number j from the decision attribute;

$$E_{ij} = {(T_{ci} * T_{vj})}\big/{T}$$

where T_{ci} is the total number of examples belonging to class ci, T_{vj} is the number of examples containing the value vj of the given attribute

$$\chi^2(T1) = (3-1.78)^2/1.78 + (2-3.21)^2/3.21 + (0-1.42)^2/1.42$$
$$+ (4-2.57)^2/2.57 + (2-1.78)^2/1.78 + (3-3.21)^2/3.21 = 3.62$$

$$\chi^2(T4) = (3-3.9)^2/3.9 + (3-2.1)^2/2.1 + (6-5.1)^2/5.1$$
$$+ (2-2.9)^2/2.9 = 1.1$$

T1	D		Total
	0	1	
1	3	2	5
2	0	4	4
3	2	3	5
Total	5	9	14

T2	D		Total
	0	1	
<=21	1	3	4
>21	4	6	10
Total	5	9	14

T3	D		Total
	0	1	
<=15	1	4	5
>15	4	5	9
Total	5	9	14

T4	D		Total
	0	1	
A	3	3	6
B	2	6	8
Total	5	9	14

Mingers, J., (1989a). “An Empirical Comparison of selection Measures for Decision-Tree Induction”, *Machine Learning*, Vol. 3, No. 3, (pp. 319-342), Kluwer Academic Publishers.

STATISTICS

Part 7

Regression

Linear Regression

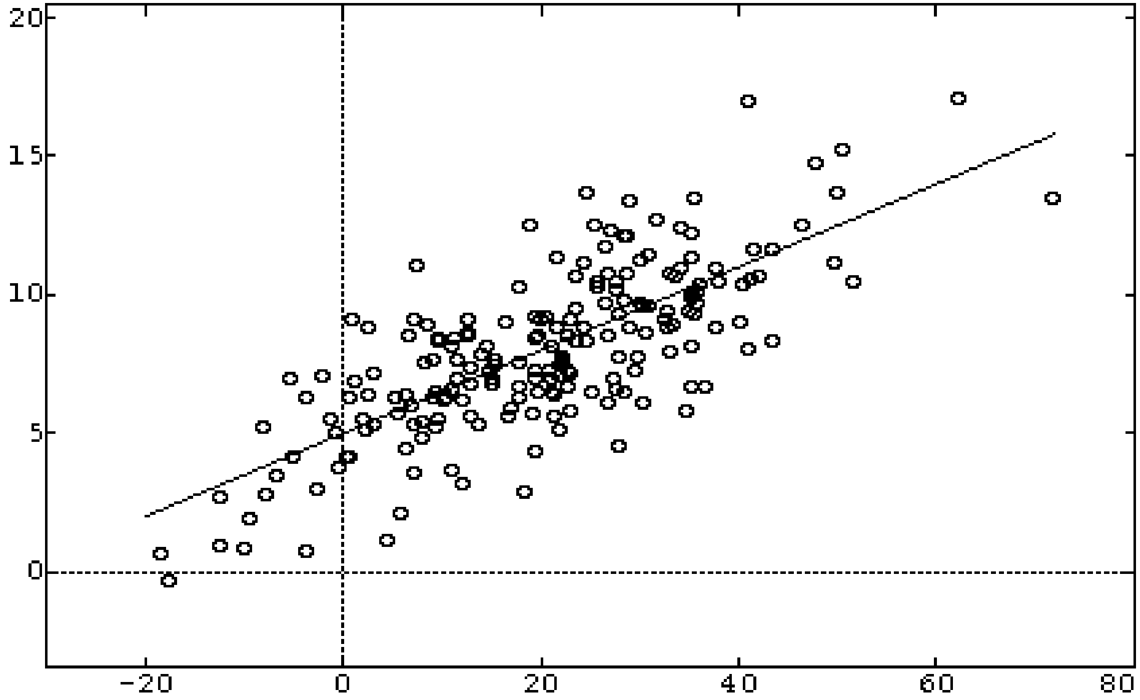
- The linear model states that the dependent variable is *directly proportional* to the value of the independent variable
- Thus if a theory implies that Y increases in direct proportion to an increase in X, it implies a specific mathematical model of behavior

$$y = ax + b$$

In case of two dimensions

$$a = slope = \frac{(y_2 - y_1)}{(x_2 - x_1)}$$

$$b = y_2 - slope * x_2$$



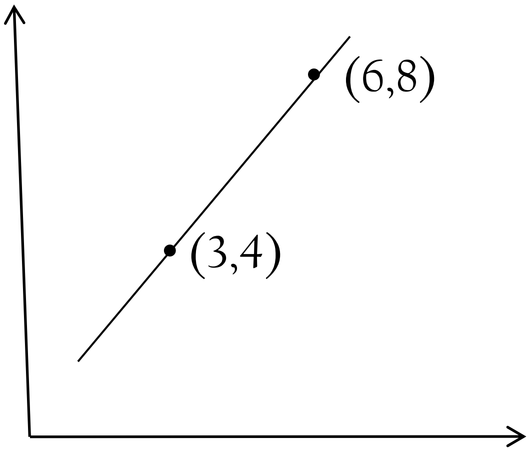
Linear Regression

$$y = ax + b$$

$$8 = 6a + b \quad \& \quad 4 = 3a + b$$

$$\frac{8-b}{6} = a \qquad \& \qquad 4 = 3 * \frac{8-b}{6} + b$$

$$b = 0 \qquad \& \qquad a = \frac{4}{3} = 1.333$$



$$Slope = \frac{8-4}{6-3} = 1.333$$

$$b = 4 - \frac{4}{3} * 3 = 0$$

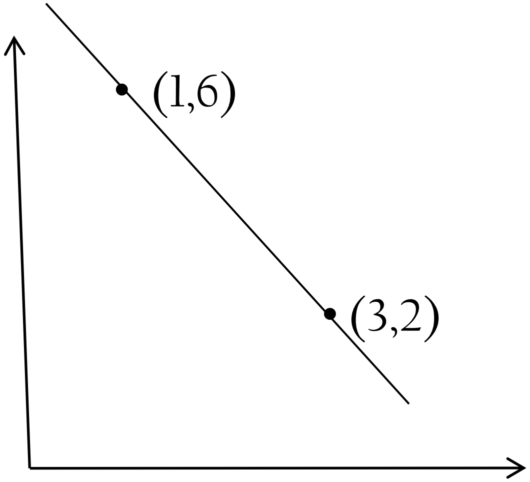
Linear Regression

$$y = ax + b$$

$$6 = a + b \quad \& \quad 2 = 3a + b$$

$$6 - b = a \qquad \& \qquad 2 = 3*(6 - b) + b$$

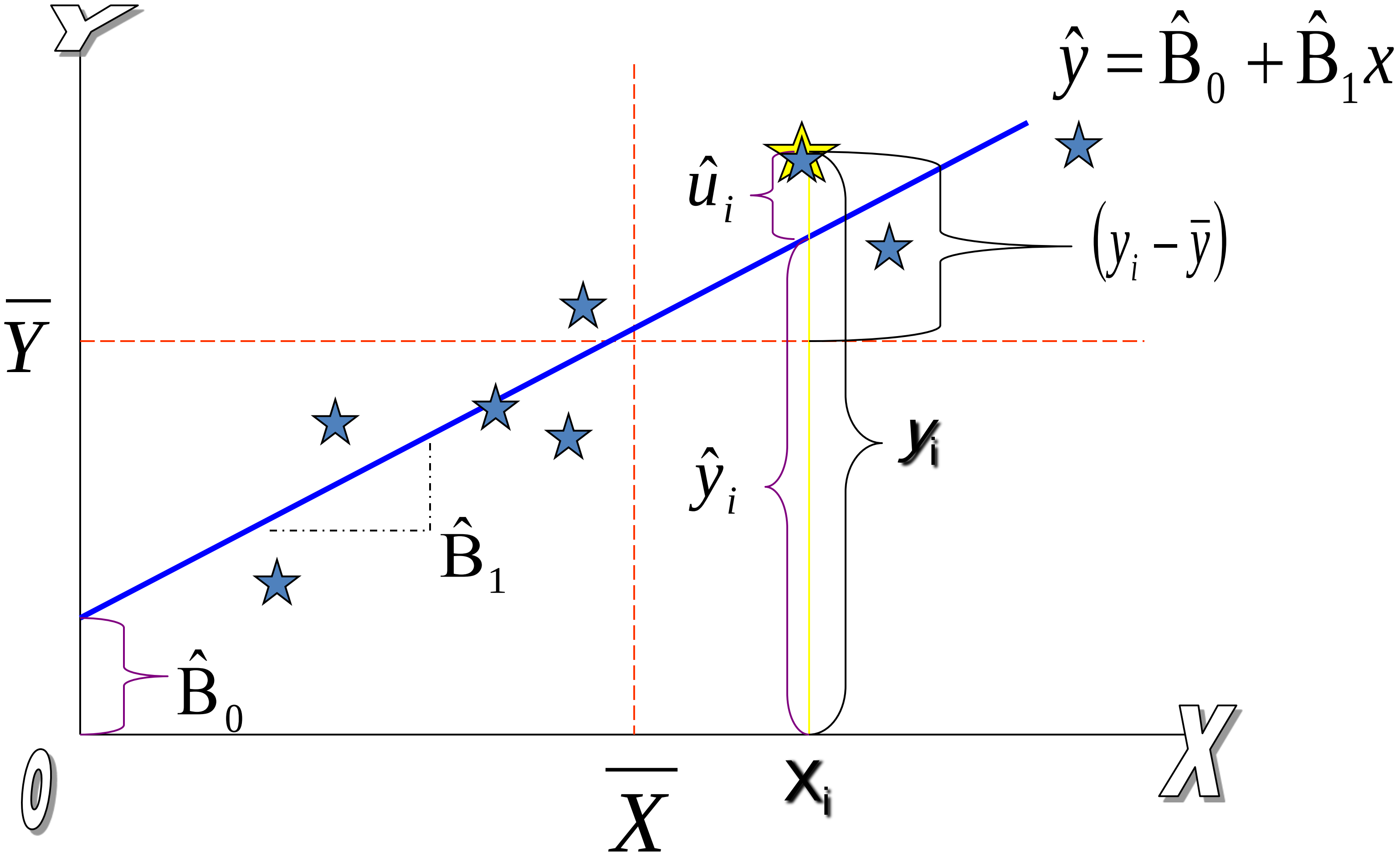
$$b = 8 \qquad \& \qquad a = 6 - 8 = -2$$



$$Slope = \frac{6-2}{1-3} = \frac{4}{-2} = -2$$

$$b = 2 + 2*3 = 8$$

Linear Regression

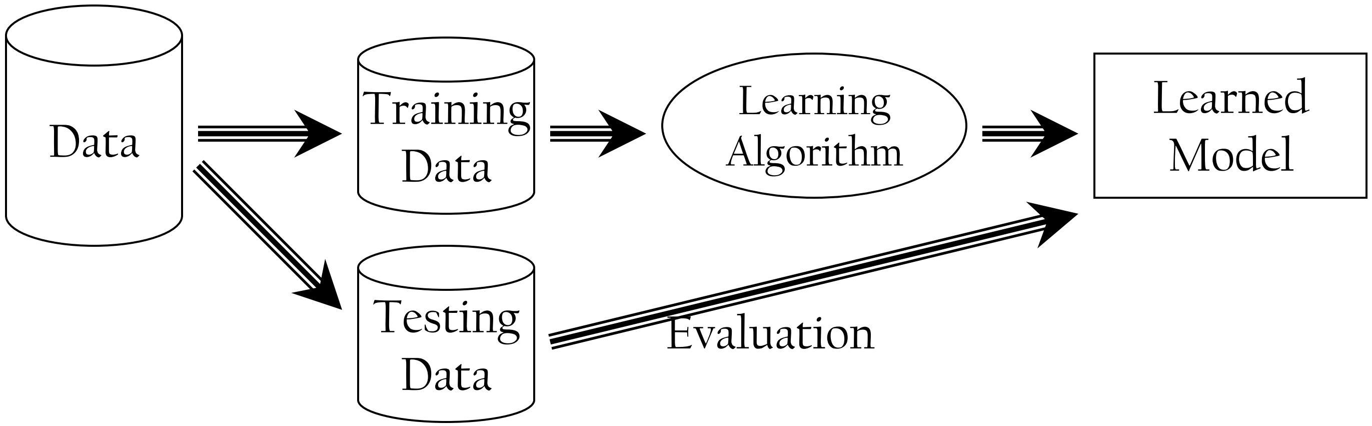


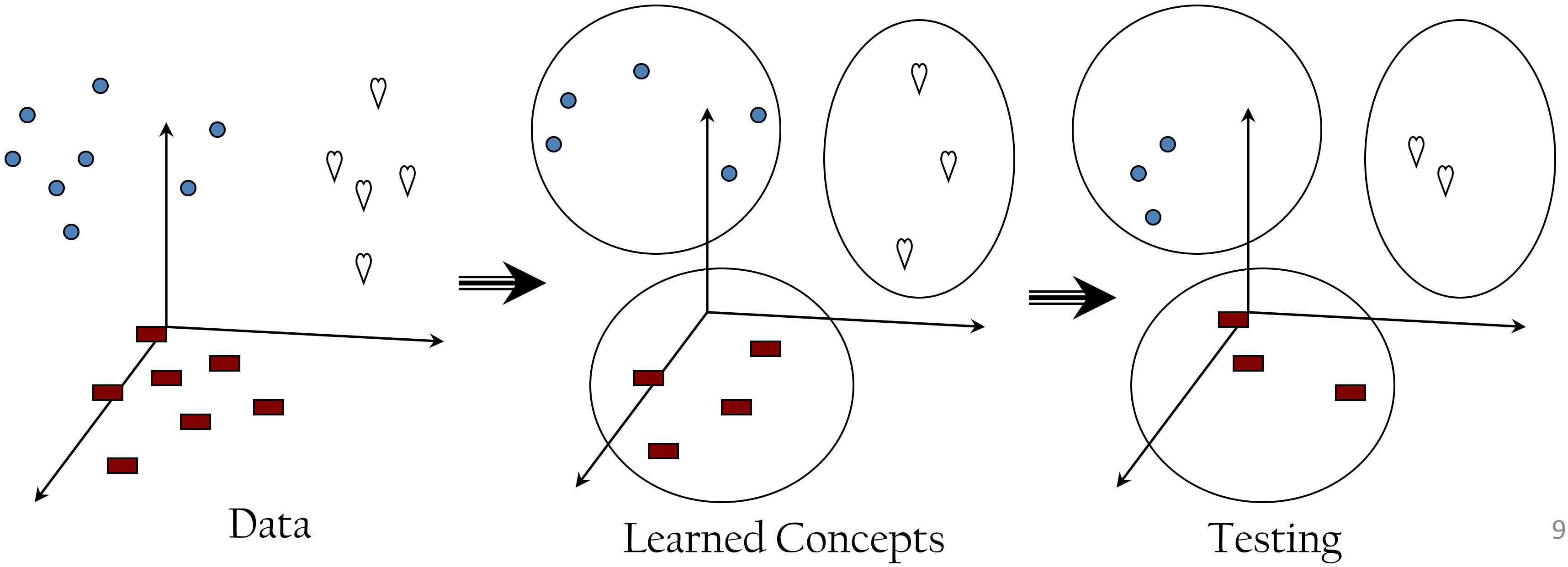
Statistics and Testing

Part 8

Testing Samples & Calculating Accuracy

Training & Testing



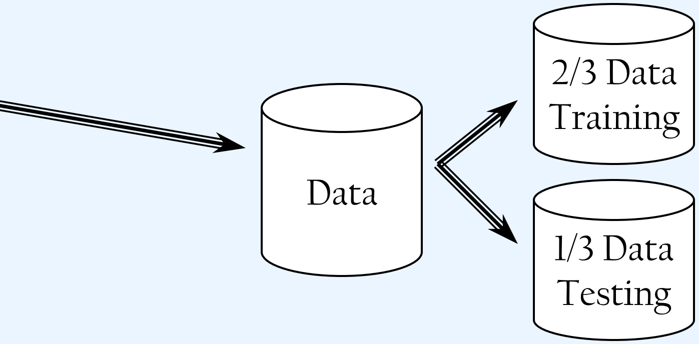


Testing Approaches

- ***Two-Cross-Fold***

  Train on 2/3rd

  Test on 1/3rd

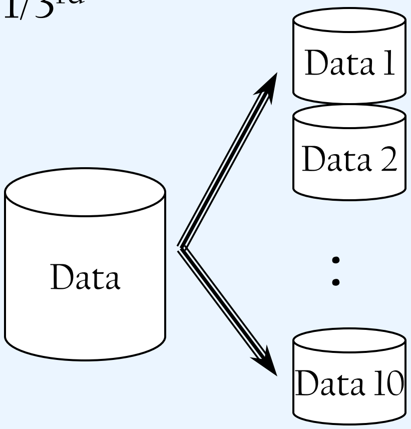


- ***Ten-Cross-Fold***

  Train on 9/10th

  Test on 1/10th

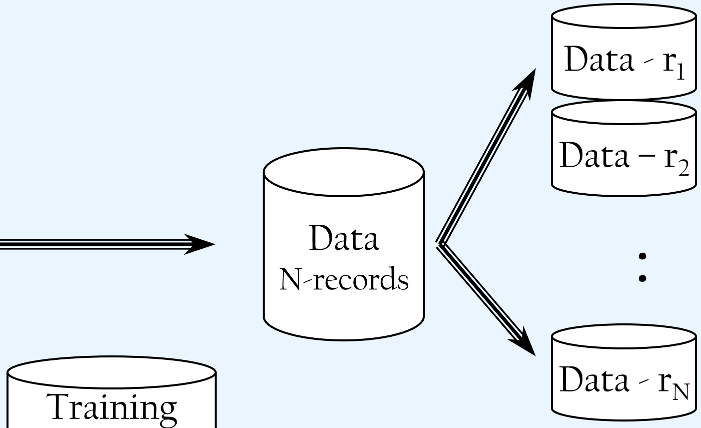
  Repeat 10 times



- ***Hold-One-Out***

  Train on all data but one

  Test on the selected one

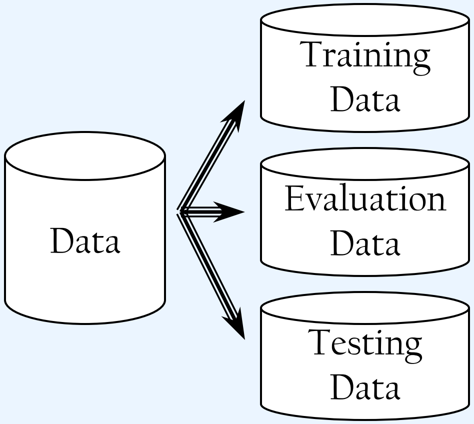


- ***Learning Evaluation vs. Testing***

  Train on Training Data

  Evaluate on Evaluation Data

  Test on Testing Data



Accuracy & Error

Example: Suppose you have a classification model C, and 100 testing records from two classes (P & N). Suppose the following are the classification results:

- Accuracy vs. Error Rate
  - *Accuracy* = (40+45)/100 = 85%
  - *Error Rate* = (10+5)/100 = 15%

		Actual	
		P	N
Obtained	P	TP	FP
	N	FN	TN

- True vs. False Classification
  - *True Positive:* = 88.88%
  - *True Negative:* = 81.82%
  - *False Positive:* = 11.12%
  - *False Negative:* = 18.18%

		Actual	
		P	N
Obtained	P	40	10
	N	5	45

- Flexible Matching
  - *Using Nearest Neighbors (e.g., majority of nearest 3 neighbors)*
  - Using Fuzzy rules (assigning probability for each decision and taking it into consideration when calculating the accuracy)
  - Assigning small weights for the false positive and false negative results (not zero)

- Testing for Multiple Classes ????

Precision, Recall, and F-Measure

Accuracy: is the percentage of correct results

Error: is the percentage of wrong results

Accuracy only reacts to real errors, and doesn’t show how many correct results have been found as such

Precision:

Precision shows the percentage of correct results within an answer:

$$Precision = (tp) / (tp + fp)$$

Recall:

Recall is the percentage of the correct system results over all correct results:

$$Recall = (tp) / (tp + fn)$$

Makhoul, John; Francis Kubala; Richard Schwartz; Ralph Weischedel: Performance measures for information extraction. In: Proceedings of DARPA Broadcast News Workshop, Herndon, VA, February 1999

Precision, Recall, and F-Measure

Precision and Recall can be defined differently for different tasks

For example: In Information Retrieval,

- Recall = |{relevant documents} ∩ {documents retrieved}| /

  / |{relevant documents}|

- Precision = |{relevant documents} ∩ {documents retrieved}| /

  / |{documents retrieved}|

Christopher D. Manning and Hinrich Sch¨utze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.

Precision, Recall, and F-Measure

F-Measure (harmonic mean):

F_β "measures the effectiveness of β times as much importance to recall as precision". The general form of F-Measure:

$$F_\beta = (1+ \beta^2) * (\text{precision} * \text{recall}) / (\beta^2 * \text{precision} + \text{recall})$$

when β=1,

$$F_1 = 2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$$

STATISTICS

Part 9

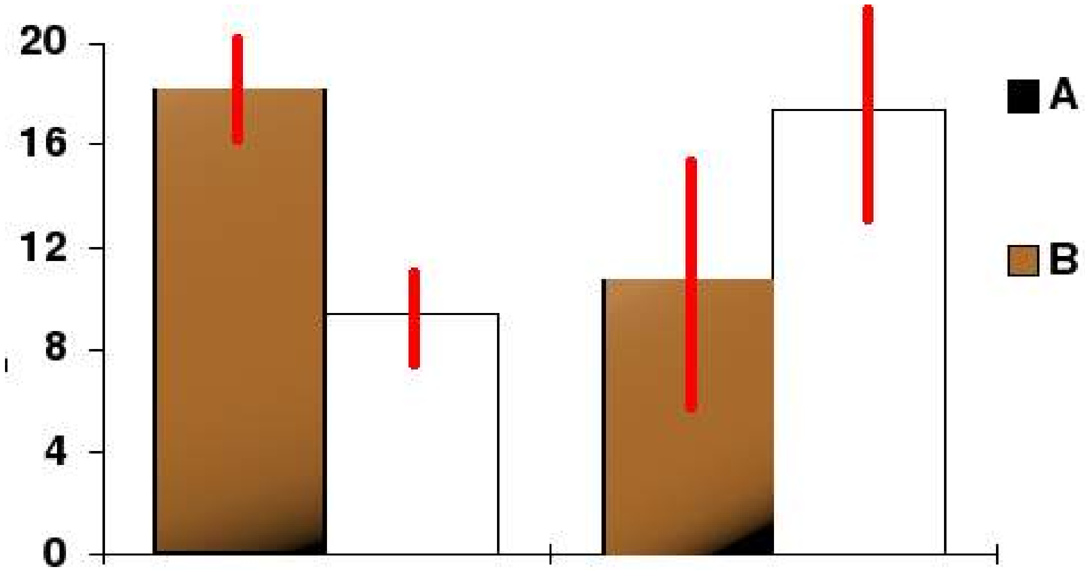
Test of Significance

Test of Significance (1/5)

- The probability that a result is not due to chance; or Is the observed value differs enough from a hypothesized value?
- The hypothesized value is called the null hypothesis
- If this probability is sufficiently low, then the difference between the parameter and the statistic is said to be "statistically significant"
- Just how low is sufficiently low? The choice of 0.05 and 0.01 are most commonly used

- Suppose your algorithm produced error rate of 1.5 and another algorithm produced an error of 2.1 on the same data set; are the two algorithms similar?

Test of Significance (2/5)



- The top ends of the bars indicate observation means
- The red line segments represent the confidence intervals surrounding them
- The difference between the two populations on the left is significant
- However, it is a common misconception to suppose that two parameters whose 95% confidence intervals fail to overlap are significantly different at the 5% level

Test of Significance (3/5)

- The system you are comparing against reported results of 250; the value reported is considered as a random variable X; the distribution of X is assumed as normal distribution with unknown mean and standard deviation σ=2.5; You ran your system 25 times; it reported values (x1, x2, ... , x25); the average of these values is 250.2.

$$\hat{\mu} = \overline{X} = \frac{1}{n}\sum_{i=1}^{25} x_i = 250.2$$

Sample Mean

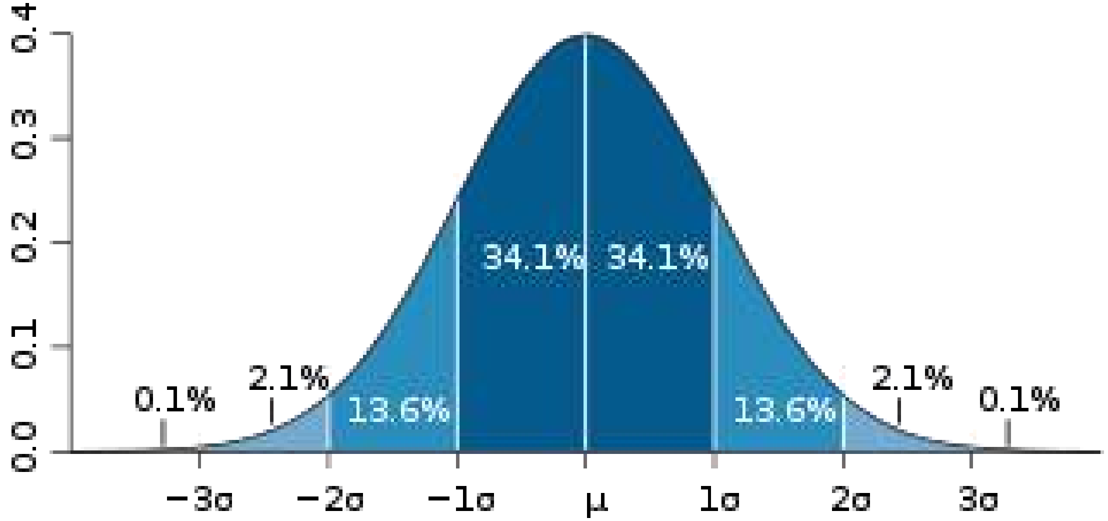
$$\text{Standard Error} = \sigma / \sqrt{n} = 2.5 / \sqrt{25} = 0.5$$

n is the sample size

$$\text{Z} = \frac{\overline{X} - \mu}{\sigma / \sqrt{n}} = \frac{\overline{X} - \mu}{0.5}$$

μ is not known

Test of Significance (4/5)



$$P(-z \le Z \le z) = 1 - \alpha = 0.95$$

$$\Phi(z) = P(Z \le z) = 1 - \frac{\alpha}{2} = 0.975$$

From Tables

$$z = \Phi^{-1}(\Phi(z)) = \Phi^{-1}(0.975) = 1.96 \longleftarrow$$

$$0.95 = 1 - \alpha = P(-z \le Z \le z) = P(-1.96 \le \frac{\bar{X} - \mu}{\sigma / \sqrt{n}} \le 1.96)$$

Test of Significance (5/5)

$$P(-z \le Z \le z) = P(\bar{X} - 1.96\frac{\sigma}{\sqrt{n}} \le \mu \le \bar{X} + 1.96\frac{\sigma}{\sqrt{n}})$$

$$P(-z \le Z \le z) = P(\bar{X} - 1.96 * 0.5 \le \mu \le \bar{X} + 1.96 * 0.5)$$

$$P(-z \le Z \le z) = P(\bar{X} - 0.98 \le \mu \le \bar{X} + 0.98)$$

$$Our\ Interval \quad = (250.2 - 0.98;\ 250.2 + 0.98)$$

$$Our\ Interval \quad = (249.22;\ 251.0)$$

- Any value within this interval is not significant

The Information Theory

Part 9

Introduction
Entropy

The Information Theory

The information conveyed by a message can be measured in bits by its probability

The Information Theory: Given Data

Attributes:
D1, D2, D3, D4

Domain(D1)={1,2,3}

Domain(D2)={1,2}

Domain(D3)={1,2}

Domain(D4)={A,B}

D1	D2	D3	D4	D5
1	2	1	A	1
1	2	2	A	0
1	2	2	B	0
1	2	2	B	0
1	1	1	B	1
2	2	2	A	1
2	2	2	B	1
2	1	1	A	1
2	2	1	B	1
3	1	2	A	0
3	1	1	A	0
3	2	2	B	1
3	1	2	B	1
3	1	2	B	1

Decision Attributes: D5

Domain(D5)={0,1}

Two Decisions: 0, 1

The Information Theory: Given Data

	D1	1		2		3	
D4	D3\D2	1	2	1	2	1	2
A	1		1	1		0	
	2		0		1	0	
B	1	1	1		1	1	
	2		0		1	1	1

D1	D2	D3	D4	D5
1	2	1	A	1
1	2	2	A	0
1	2	1	B	0
1	2	2	B	0
1	1	1	B	1
2	2	2	A	1
2	2	2	B	1
2	1	1	A	1
2	2	1	B	1
3	1	2	A	0
3	1	1	A	0
3	2	2	B	1
3	1	1	B	1
3	1	2	B	1

The Information Theory: Entropy

THE INFORMATION THEORY: information conveyed by a message depends on its probability and can be measured in bits as minus the logarithm (base 2) of that probability

suppose $D_1, \ldots, D_m$ are m attributes and $C_1, \ldots, C_n$ are n decision classes in a given data. Suppose S is any set of cases, and T is the initial set of training cases $S \subset T$. The **frequency of class C_i in the set S** is:

$$freq(C_i, S) = Number\ of\ examples\ in\ S\ belonging\ to\ C_i$$

If |S| is the total number of examples in S, ***the probability that an example selected at random from S belongs to class C_i*** is

$$freq(C_i, S) / |S|$$

The information conveyed by the message that "**a selected example belongs to a given decision class, C_i**", is determined by

$$-\log_2(freq(C_i, S) / |S|) \quad bits$$

The Information Theory: Entropy

The information conveyed by the message "***a selected example belongs to a given decision class, C_i***"

$$-\log_2(freq(C_i, S)/|S|) \quad bits$$

The Entropy: The expected information from a message stating class membership is given by

$$Info(S) = -\sum_{i=1}^{k}(freq(C_i, S)/|S|) * \log_2(freq(C_i, S)/|S|) \quad bits$$

info(S) is known as the ***entropy*** of the set S. When S is the initial set of training examples, *info(S) determines the average amount of information needed to identify the class of an example in S*.

The Information Theory: The Gain Ratio

S

D1	D2	D3	D4	D5
1	2	1	A	1
1	2	2	A	0
1	2	2	B	0
1	2	2	B	0
1	1	1	B	1
2	2	2	A	1
2	2	2	B	1
2	1	1	A	1
2	2	1	B	1
3	1	2	A	0
3	1	1	A	0
3	2	2	B	1
3	1	2	B	1
3	1	2	B	1

Example

$$freq(0,S)=5 \qquad freq(1,S)=9$$

$$freq(0,S)/|S|=5/14 \qquad freq(1,S)/|S|=9/14$$

The Entropy: *the average amount of information needed to identify the class of an example in S*

$$Info(S)=-9/14*\log_2(9/14)-5/14*\log_2(5/14)=0.94bits$$

Using D_1 to Split the data provide 3 subsets of data

$$Info_{D_1}(S_1)=-3/5*\log_2(3/5)-2/5*\log_2(2/5)=0.94$$

$$Info_{D_1}(S_2)=-4/4*\log_2(4/4)=0.94$$

$$Info_{D_1}(S_3)=-2/5*\log_2(2/5)-3/5*\log_2(3/5)=0.94$$

$$Info_{D_1}(S)=(5/14)*Info_{D_1}(S_1)+(4/14)*Info_{D_1}(S_2)+(5/14)*Info_{D_1}(S_3)=0.694$$

The Information Theory: The Gain Ratio

Suppose attribute $\underline{D_i}$ is selected to be the root and it has $\underline{k}$ possible values. The expected information of selecting D to partition the training set S, info_{D_i}(S), can be calculated as follows:

$$Info_{D_i}(S) = \sum_{i=1}^{k} \left({|S_i|}/{|S|} \right) * Info(S_i)$$

S_i is the subset number i of the data; k is the number of values of D_i

The information gained by partitioning the training examples S into subset using the attribute D_1 is given by

$$Gain(D_i) = Info(S) - Info_{D_i}(S)$$

The Information Theory: The Gain Ratio

The attribute to be selected is the attribute with maximum gain value. Quinlan found out that a key attribute will have the maximum gain. This is not good!

$$Split\_Info(S) = -\sum_{i=1}^{k} (|S_i|/|S|) * \log_2(|S_i|/|S|)$$

The gain ratio is given by:

$$Gain\_Ratio(D_i) = Gain(D_i) / Split\_Info(D_i)$$

Quinlan, J.R., (1993). "C4.5: Programs for Machine Learning", Morgan Kaufmann, Los Altos, California.

The Information Theory: The Gain Ratio

Example Cont.

$$Info_{D_1}(S) = (5/14) * Info_{D_1}(S_1) + (4/14) * Info_{D_1}(S_2) + (5/14) * Info_{D_1}(S_3) = 0.694$$

$$Gain(D_1) = 0.94 - 0.694 = 0.246$$

$$Split\_Info(S) = -5/14 * \log_2(5/14) - 4/14 * \log_2(4/14) - 5/14 \log_2(5/14) = 1.577 \quad bits$$

$$Gain\_Ratio(D_1) = 0.246/1.577 = 0.156$$

S

D1	D2	D3	D4	D5
1	2	1	A	1
1	2	2	A	0
1	2	2	B	0
1	2	2	B	0
1	1	1	B	1
2	2	2	A	1
2	2	2	B	1
2	1	1	A	1
2	2	1	B	1
3	1	2	A	0
3	1	1	A	0
3	2	2	B	1
3	1	2	B	1
3	1	2	B	1

Information Gain: Term vs. Category

It measures the classification power of a term

$$IG(t_k, c_i) = \sum_{c \in \{c_i, \bar{c}_i\}} \sum_{t \in \{t_k, \bar{t}_k\}} P(t,c) \log_2 \frac{P(t,c)}{P(t)P(c)}$$

$P(t_k, c_i)$ ➔probability document x contains term t and belongs to category c.

$P(\bar{t}_k, c_i)$ ➔probability document x does not contain term t and belongs to category c.

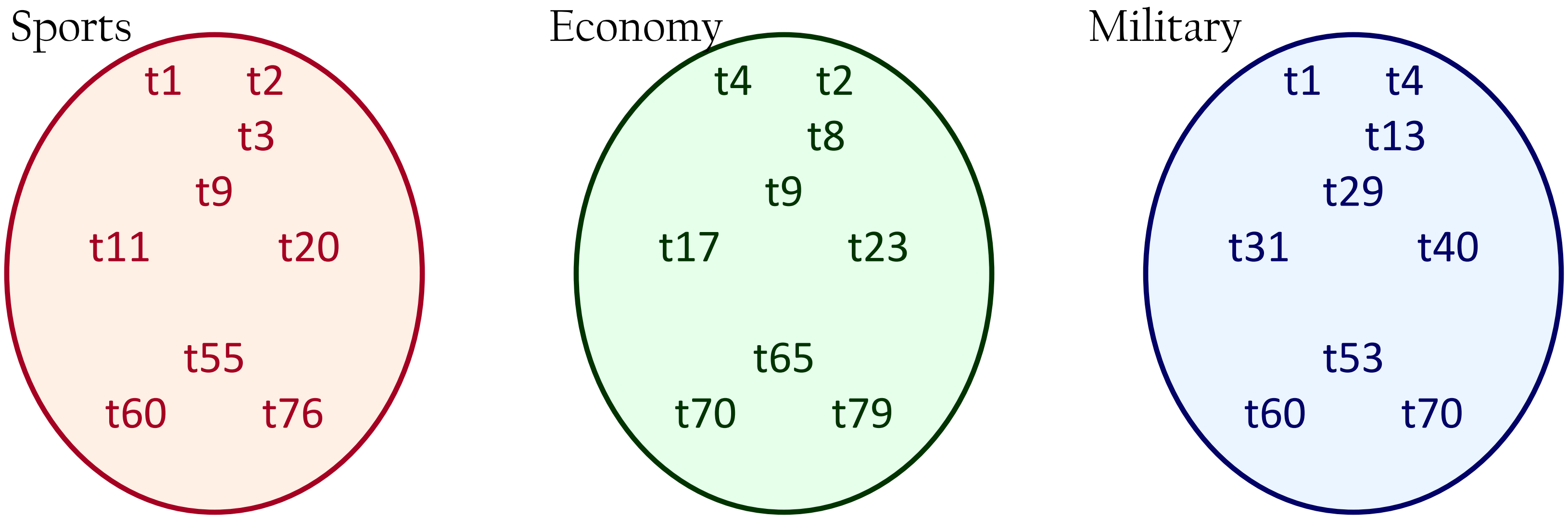
$P(t_k, \bar{c}_i)$ ➔probability document x contains term t and does not belong to category c.

$P(\bar{t}_k, \bar{c}_i)$ ➔probability document x does not contain term t and does not belong to category c.

$P(t)$ ➔probability of term t.

$P(c)$ ➔probability of category c.

Testing The Membership



Sports

t1 t2 t3 t9 t11 t20 t55 t60 t76

Economy

t4 t2 t8 t9 t17 t23 t65 t70 t79

Military

t1 t4 t13 t29 t31 t40 t53 t60 t70

$$IG(t_k, c_i) = \sum_{c\in\{c_i,\bar{c}_i\}} \sum_{t\in\{t_k,\bar{t}_k\}} P(t,c)\log_2 \frac{P(t,c)}{P(t)P(c)}$$

$$IG(t_1, sport) = \frac{1}{9}*\log_2 \frac{1/9}{(2/27)*(9/27)} + \frac{8}{9}*\log_2 \frac{8/9}{(25/27)*(9/27)}$$
$$+\frac{1}{18}*\log_2 \frac{1/18}{(2/27)*(18/27)} + \frac{17}{27}*\log_2 \frac{17/27}{(25/27)*(18/27)}$$

The Gain Ratio

$$GR(t_k, c_i) = \frac{\displaystyle\sum_{c \in \{c_i, \bar{c}_i\}} \sum_{t \in \{t_k, \bar{t}_k\}} P(t,c) \log_2 \frac{P(t,c)}{P(t)P(c)}}{\displaystyle -\sum_{c \in \{c_i, \bar{c}_i\}} P(c) \log_2 P(c)}$$

$P(t_k, c_i)$ ➔probability document x contains term t and belongs to category c.

$P(\bar{t}_k, c_i)$ ➔probability document x does not contain term t and belongs to category c.

$P(t_k, \bar{c}_i)$ ➔probability document x contains term t and does not belong to category c.

$P(\bar{t}_k, \bar{c}_i)$ ➔probability document x does not contain term t and does not belong to category c.

$P(t)$ ➔probability of term t.

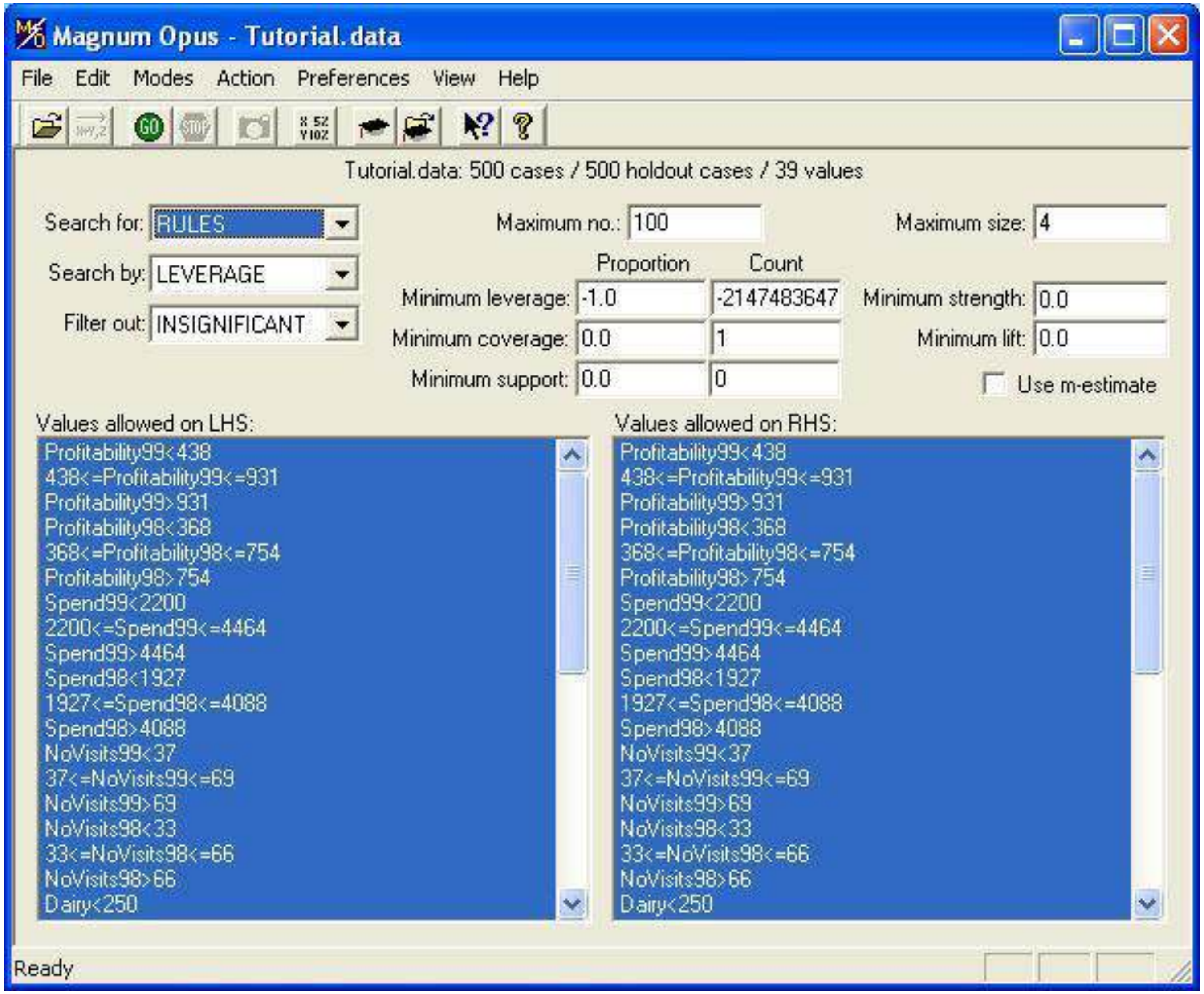
$P(c)$ ➔probability of category c.

STATISTICAL ASSOCIATIONS

Part II

Association Rules
http://giwebb.com/

The Magnum Opus System



Tutorial.data: 500 cases / 500 holdout cases / 39 values

Search for: RULES

Search by: LEVERAGE

Filter out: INSIGNIFICANT

Maximum no.: 100

Maximum size: 4

	Proportion	Count
Minimum leverage:	-1.0	-2147483647
Minimum coverage:	0.0	1
Minimum support:	0.0	0

Minimum strength: 0.0

Minimum lift: 0.0

□ Use m-estimate

Values allowed on LHS:

Profitability99<438
438<=Profitability99<=931
Profitability99>931
Profitability98<368
368<=Profitability98<=754
Profitability98>754
Spend99<2200
2200<=Spend99<=4464
Spend99>4464
Spend98<1927
1927<=Spend98<=4088
Spend98>4088
NoVisits99<37
37<=NoVisits99<=69
NoVisits99>69
NoVisits98<33
33<=NoVisits98<=66
NoVisits98>66
Dairy<250

Values allowed on RHS:

Profitability99<438
438<=Profitability99<=931
Profitability99>931
Profitability98<368
368<=Profitability98<=754
Profitability98>754
Spend99<2200
2200<=Spend99<=4464
Spend99>4464
Spend98<1927
1927<=Spend98<=4088
Spend98>4088
NoVisits99<37
37<=NoVisits99<=69
NoVisits99>69
NoVisits98<33
33<=NoVisits98<=66
NoVisits98>66
Dairy<250

Ready

Attributes and their values for the Tutorial database

- Profitability99: numeric 3
- Profitability98: numeric 3
- Spend99: numeric 3
- Spend98: numeric 3
- NoVisits99: numeric 3
- NoVisits98: numeric 3
- Dairy: numeric 3
- Deli: numeric 3
- Bakery: numeric 3
- Grocery: numeric 3
- SocioEconomicGroup: categorical
- Promotion1: t, f
- Promotion2: t, f

The Magnum Opus System: Example

bananas
plums, lettuce, tomatoes
celery, confectionery
confectionery
apples, carrots, tomatoes, potatoes
potatoes
confectionery
carrots
confectionery
apples, oranges, lettuce, tomatoes
peaches, oranges, celery, potatoes, confectionery
beans
oranges, lettuce, carrots, tomatoes
apples, bananas, plums, carrots, tomatoes, onions, confectionery
apples, potatoes
lettuce, peas, beans
carrots, tomatoes
grapes, plums, lettuce, beans, potatoes, onions
confectionery
confectionery
carrots, peas, potatoes, onions, confectionery
tomatoes
confectionery
carrots, potatoes
peaches, apples, bananas
lettuce, beans, tomatoes, potatoes, confectionery
grapes, lettuce, tomatoes, confectionery
oranges

oranges, lettuce, confectionery
tomatoes
lettuce, carrots, tomatoes, confectionery
celery, potatoes, confectionery
oranges, carrots, beans, potatoes
peaches, oranges, bananas
lettuce, carrots, tomatoes, potatoes, onions
onions
peaches, apples, lettuce, peas, potatoes, onions
oranges, carrots, confectionery
bananas
lettuce, carrots, tomatoes, potatoes
carrots, confectionery
oranges, plums
peaches, oranges, lettuce, peas
lettuce, carrots, beans, tomatoes
plums, lettuce, peas, tomatoes, potatoes
carrots, tomatoes
bananas, lettuce, onions, confectionery
oranges, tomatoes
oranges, potatoes
confectionery
oranges, plums, potatoes
bananas, lettuce, carrots, tomatoes, potatoes
potatoes
lettuce, tomatoes, onions
lettuce, onions
apples, oranges, beans
corn

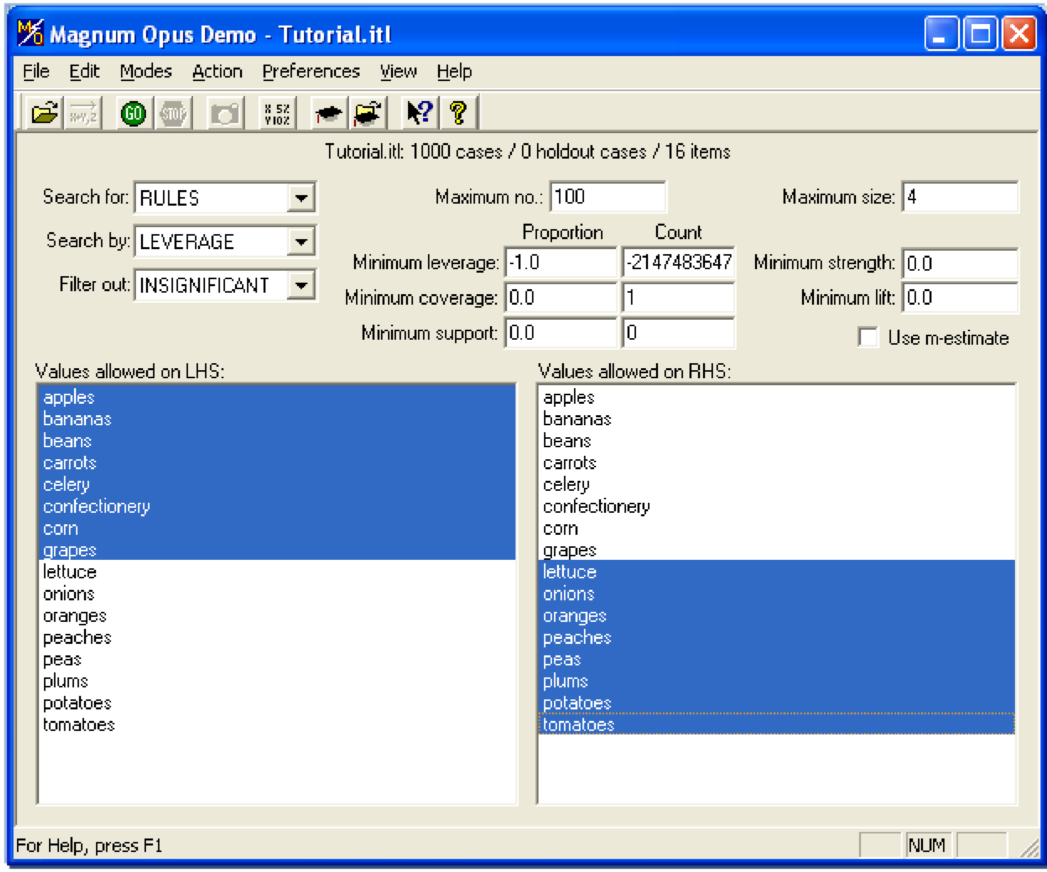
The Magnum Opus System

carrots -> tomatoes
[Coverage=0.175 (175); Support=0.085 (85); Strength=0.486; Lift=1.85; Leverage=0.0390 (39.0); p=1.83E-012]

bananas -> peaches
[Coverage=0.127 (127); Support=0.040 (40); Strength=0.315; Lift=2.42; Leverage=0.0235 (23.5); p=2.74E-009]

carrots -> potatoes
[Coverage=0.175 (175); Support=0.068 (68); Strength=0.389; Lift=1.37; Leverage=0.0185 (18.5); p=0.000575]

apples -> peaches
[Coverage=0.221 (221); Support=0.044 (44); Strength=0.199; Lift=1.53; Leverage=0.0153 (15.3); p=0.000635]


Magnum Opus Demo - Tutorial.itl
File Edit Modes Action Preferences View Help
GO
STOP

Tutorial.itl: 1000 cases / 0 holdout cases / 16 items

Search for: RULES
Search by: LEVERAGE
Filter out: INSIGNIFICANT

Maximum no.: 100
Maximum size: 4

	Proportion	Count
Minimum leverage:	-1.0	-2147483647
Minimum coverage:	0.0	1
Minimum support:	0.0	0

Minimum strength: 0.0
Minimum lift: 0.0
□ Use m-estimate

Values allowed on LHS:	Values allowed on RHS:
apples	apples
bananas	bananas
beans	beans
carrots	carrots
celery	celery
confectionery	confectionery
corn	corn
grapes	grapes
lettuce	lettuce
onions	onions
oranges	oranges
peaches	peaches
peas	peas
plums	plums
potatoes	potatoes
tomatoes	tomatoes

For Help, press F1
NUM

bananas & apples -> peaches
[Coverage=0.029 (29); Support=0.017 (17); Strength=0.586; Lift=4.51; Leverage=0.0132 (13.2); p=0.000540]

apples -> lettuce
[Coverage=0.221 (221); Support=0.058 (58); Strength=0.262; Lift=1.21; Leverage=0.0100 (10.0); p=0.0404]

carrots & beans -> potatoes
[Coverage=0.010 (10); Support=0.007 (7); Strength=0.700; Lift=2.47; Leverage=0.0042 (4.2); p=0.0420]

The Magnum Opus System: Example

```
ID001, bananas
ID002, plums
ID002, lettuce
ID002, tomatoes
ID003, celery
ID003, confectionery
ID004, confectionery
ID005, apples
ID005, carrots
ID005, tomatoes
ID005, potatoes
ID006, potatoes
ID007, confectionery
ID008, carrots
ID009, confectionery
ID00a, apples
ID00a, oranges
ID00a, lettuce
ID00a, tomatoes
ID00b, peaches
ID00b, oranges
ID00b, celery
ID00b, potatoes
ID00b, confectionery
ID00c, beans
ID00d, oranges
ID00d, lettuce
ID00d, carrots
ID00d, tomatoes
```

```
ID00e, apples
ID00e, bananas
ID00e, plums
ID00e, carrots
ID00e, tomatoes
ID00e, onions
ID00e, confectionery
ID00f, apples
ID00f, potatoes
ID010, lettuce
ID010, peas
ID010, beans
ID011, carrots
ID011, tomatoes
ID012, grapes
ID012, plums
ID012, lettuce
ID012, beans
ID012, potatoes
ID012, onions
ID013, confectionery
ID014, confectionery
ID015, carrots
ID015, peas
ID015, potatoes
ID015, onions
ID015, confectionery
ID016, tomatoes
ID017, confectionery
```

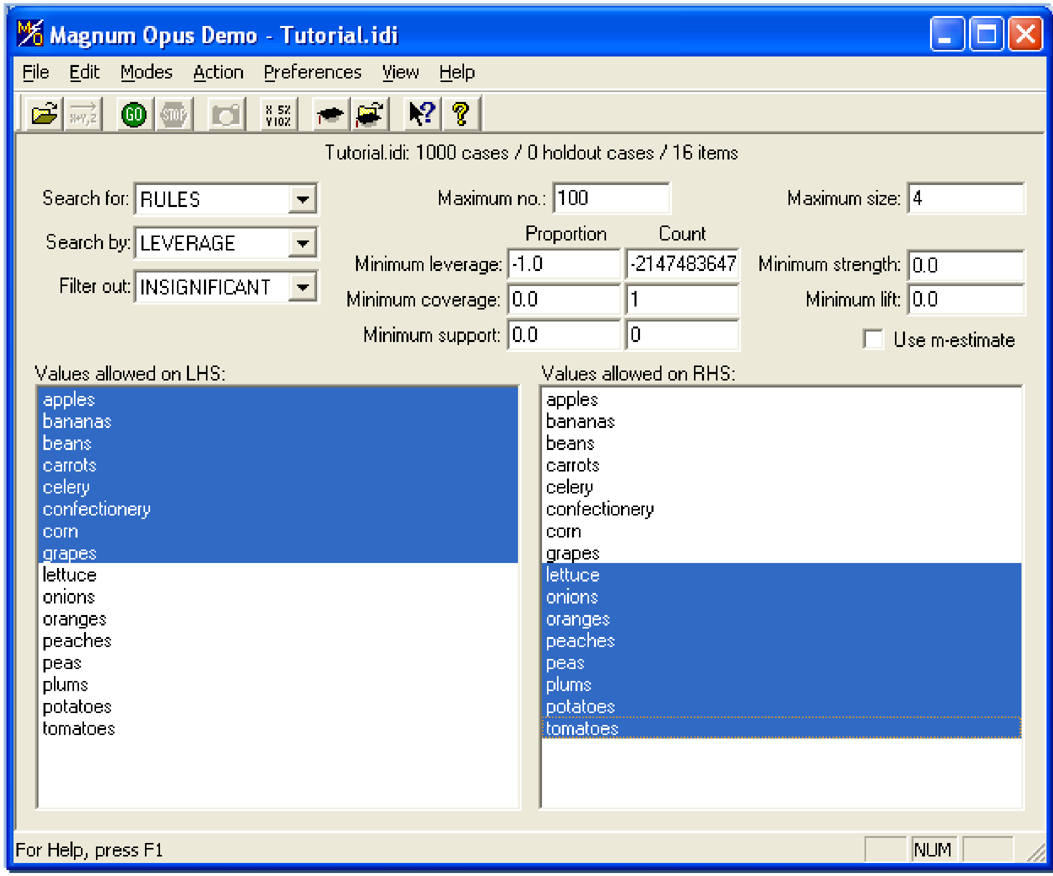
The Magnum Opus System

carrots -> tomatoes
[Coverage=0.175 (175); Support=0.085 (85); Strength=0.486; Lift=1.85; Leverage=0.0390 (39.0); p=1.83E-012]

bananas -> peaches
[Coverage=0.127 (127); Support=0.040 (40); Strength=0.315; Lift=2.42; Leverage=0.0235 (23.5); p=2.74E-009]

carrots -> potatoes
[Coverage=0.175 (175); Support=0.068 (68); Strength=0.389; Lift=1.37; Leverage=0.0185 (18.5); p=0.000575]

apples -> peaches
[Coverage=0.221 (221); Support=0.044 (44); Strength=0.199; Lift=1.53; Leverage=0.0153 (15.3); p=0.000635]



Tutorial.idi: 1000 cases / 0 holdout cases / 16 items

Search for: RULES

Search by: LEVERAGE

Filter out: INSIGNIFICANT

Maximum no.: 100

Maximum size: 4

	Proportion	Count
Minimum leverage:	-1.0	-2147483647
Minimum coverage:	0.0	1
Minimum support:	0.0	0

Minimum strength: 0.0

Minimum lift: 0.0

□ Use m-estimate

Values allowed on LHS:	Values allowed on RHS:
[✓ apples]	apples
[✓ bananas]	bananas
[✓ beans]	beans
[✓ carrots]	carrots
[✓ celery]	celery
[✓ confectionery]	confectionery
[✓ corn]	corn
[✓ grapes]	grapes
lettuce	[✓ lettuce]
onions	[✓ onions]
oranges	[✓ oranges]
peaches	[✓ peaches]
peas	[✓ peas]
plums	[✓ plums]
potatoes	[✓ potatoes]
tomatoes	[✓ tomatoes]

bananas & apples -> peaches
[Coverage=0.029 (29); Support=0.017 (17); Strength=0.586; Lift=4.51; Leverage=0.0132 (13.2); p=0.000540]

apples -> lettuce
[Coverage=0.221 (221); Support=0.058 (58); Strength=0.262; Lift=1.21; Leverage=0.0100 (10.0); p=0.0404]

carrots & beans -> potatoes
[Coverage=0.010 (10); Support=0.007 (7); Strength=0.700; Lift=2.47; Leverage=0.0042 (4.2); p=0.0420]

The Magnum Opus System: Example

```
829, 709, 5250, 6560, 70, 82, 1074, 390, 878, 1995, C, f, f
141, 118, 722, 928, 19, 16, 15, 155, 139, 404, C, f, f
1044, 783, 3591, 4026, 63, 61, 81, 218, 232, 2908, D2, f, t
78, 63, 331, 336, 7, 8, 54, 68, 63, 167, D1, t, f
511, 419, 2142, 1947, 34, 33, 59, 106, 239, 1477, C, f, f
987, 1402, 4032, 5376, 56, 64, 891, 681, 995, 1411, C, f, f
313, 286, 1137, 1008, 22, 18, 153, 63, 146, 762, D1, t, f
1800, 859, 7350, 3159, 75, 81, 441, 2315, 1433, 1837, D1, f, f
226, 126, 1034, 612, 11, 6, 351, 377, 259, 196, C, f, f
58, 28, 343, 140, 24, 14, 24, 18, 35, 248, A, t, f
1136, 597, 4602, 3068, 59, 59, 554, 870, 949, 2623, D1, f, f
376, 274, 1980, 1675, 22, 25, 356, 261, 344, 792, C, f, f
223, 172, 1656, 1400, 18, 14, 355, 430, 323, 579, C, f, f
1808, 976, 7600, 7396, 80, 86, 501, 718, 852, 5928, C, f, f
114, 180, 462, 1008, 14, 16, 4, 28, 27, 364, D2, f, f
1169, 1125, 4356, 3723, 45, 51, 359, 427, 134, 2107, D1, t, f
226, 235, 1230, 1575, 15, 15, 414, 284, 267, 418, D1, f, f
493, 189, 2408, 1035, 28, 23, 318, 503, 344, 1083, D1, f, f
915, 842, 4260, 5487, 71, 59, 1265, 796, 1148, 1917, C, f, t
1263, 739, 6136, 4277, 52, 47, 903, 1060, 589, 2208, B, f, f
668, 429, 4992, 5841, 78, 59, 988, 955, 593, 1697, B, f, f
259, 187, 1069, 930, 12, 10, 329, 182, 76, 481, B, t, f
1021, 778, 4118, 3127, 58, 53, 432, 467, 432, 2388, D1, f, f
751, 425, 3159, 1896, 27, 24, 262, 147, 542, 1516, C, f, f
1397, 929, 6210, 5162, 54, 58, 1630, 2329, 1676, 1552, C, f, t
336, 526, 1620, 3534, 60, 57, 211, 272, 183, 939, B, f, f
38, 52, 182, 518, 14, 14, 16, 17, 9, 131, C, f, t
578, 869, 1960, 3555, 70, 79, 219, 185, 212, 1274, D2, f, t
```

Profitability99: numeric 3
Profitability98: numeric 3
Spend99: numeric 3
Spend98: numeric 3
NoVisits99: numeric 3
NoVisits98: numeric 3
Dairy: numeric 3
Deli: numeric 3
Bakery: numeric 3
Grocery: numeric 3
SocioEconomicGroup: categorical
Promotion1: t, f
Promotion2: t, f

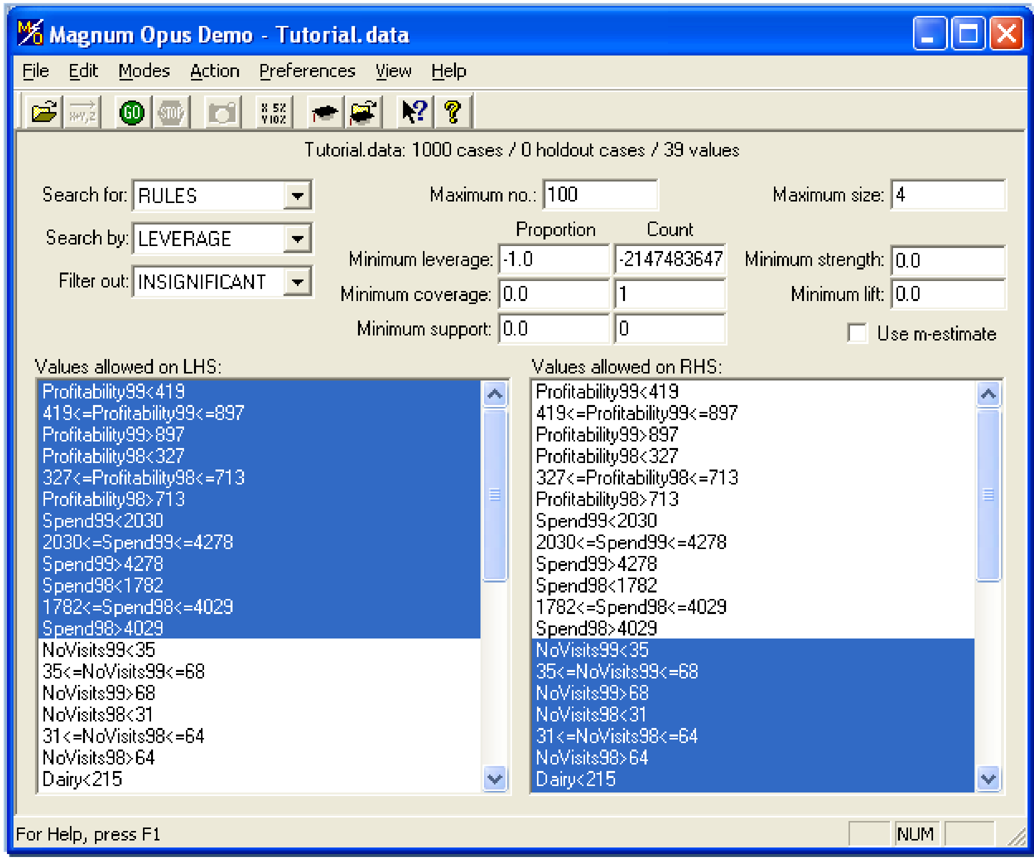
The Magnum Opus System

Spend98<1782 -> NoVisits98<31
[Coverage=0.331 (331); Support=0.277 (277);
Strength=0.837; Lift=2.57; Leverage=0.1694 (169.4);
p=1.64E-136]

Spend99<2030 -> Grocery<873
[Coverage=0.333 (333); Support=0.278 (278);
Strength=0.835; Lift=2.51; Leverage=0.1671 (167.1);
p=1.13E-130]

Profitability99<419 -> Grocery<873
[Coverage=0.333 (333); Support=0.277 (277);
Strength=0.832; Lift=2.50; Leverage=0.1661 (166.1);
p=6.14E-129]

Profitability99<419 & Spend99<2030 -> Grocery<873
[Coverage=0.302 (302); Support=0.265 (265);
Strength=0.877; Lift=2.64; Leverage=0.1644 (164.4);
p=2.52E-008]


Magnum Opus Demo - Tutorial.data
File Edit Modes Action Preferences View Help
GO
STOP

Tutorial.data: 1000 cases / 0 holdout cases / 39 values

Search for: RULES
Search by: LEVERAGE
Filter out: INSIGNIFICANT

Maximum no.: 100
Maximum size: 4

	Proportion	Count
Minimum leverage:	-1.0	-2147483647
Minimum coverage:	0.0	1
Minimum support:	0.0	0

Minimum strength: 0.0
Minimum lift: 0.0
□ Use m-estimate

Values allowed on LHS:
Profitability99<419
419<=Profitability99<=897
Profitability99>897
Profitability98<327
327<=Profitability98<=713
Profitability98>713
Spend99<2030
2030<=Spend99<=4278
Spend99>4278
Spend98<1782
1782<=Spend98<=4029
Spend98>4029
NoVisits99<35
35<=NoVisits99<=68
NoVisits99>68
NoVisits98<31
31<=NoVisits98<=64
NoVisits98>64
Dairy<215

Values allowed on RHS:
Profitability99<419
419<=Profitability99<=897
Profitability99>897
Profitability98<327
327<=Profitability98<=713
Profitability98>713
Spend99<2030
2030<=Spend99<=4278
Spend99>4278
Spend98<1782
1782<=Spend98<=4029
Spend98>4029
NoVisits99<35
35<=NoVisits99<=68
NoVisits99>68
NoVisits98<31
31<=NoVisits98<=64
NoVisits98>64
Dairy<215

For Help, press F1 NUM

Spend99<2030 -> NoVisits99<35
[Coverage=0.333 (333); Support=0.272 (272); Strength=0.817; Lift=2.48; Leverage=0.1624 (162.4); p=2.42E-123]

Spend98<1782 -> NoVisits99<35
[Coverage=0.331 (331); Support=0.271 (271); Strength=0.819; Lift=2.49; Leverage=0.1621 (162.1); p=4.58E-123]

Spend99<2030 & Spend98<1782 -> NoVisits99<35
[Coverage=0.259 (259); Support=0.246 (246); Strength=0.950; Lift=2.89; Leverage=0.1608 (160.8); p=7.04E-027]

Statistical Association

Magnum Opus

DEMO

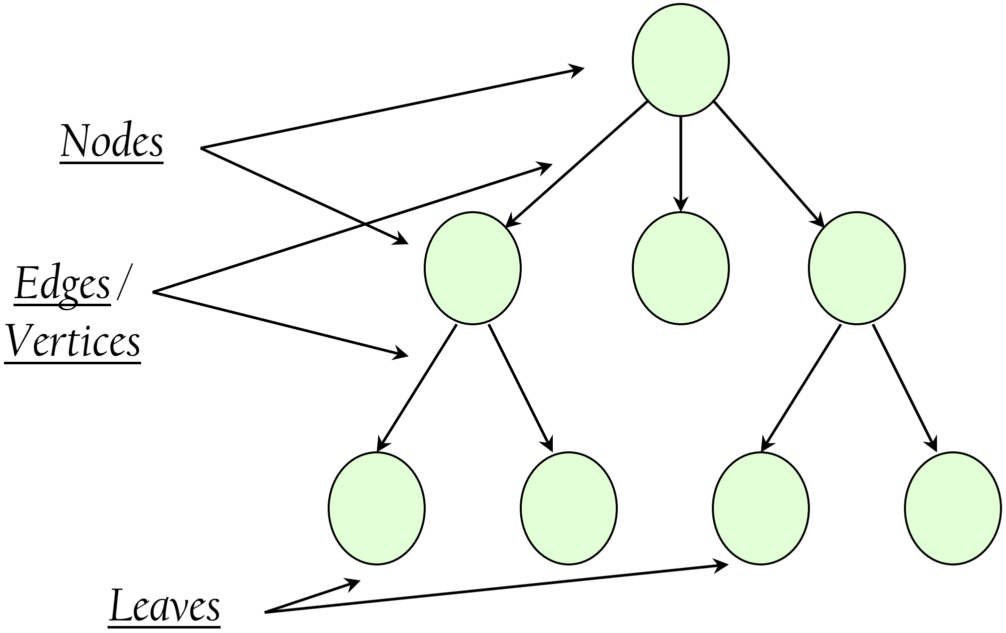
DECISION TREES

Part 12

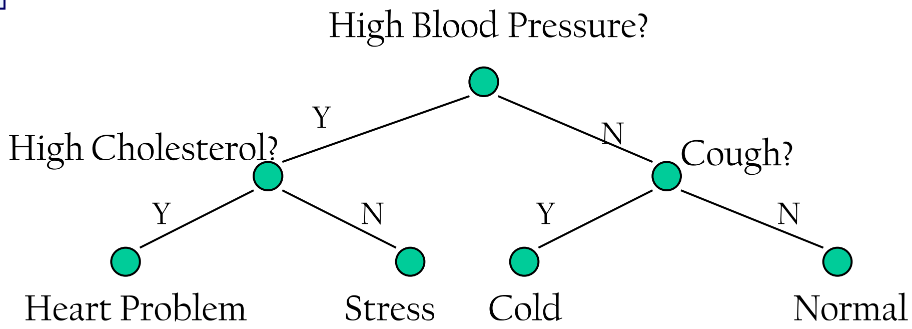
Using Statistical &
Information Theory
http://rulequest.com/

Learning Decision Trees

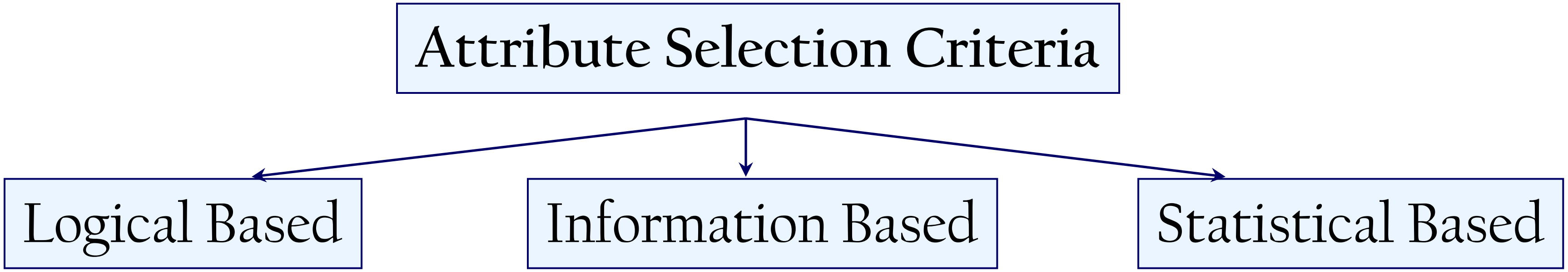
- A *Tree* is a Directed Acyclic Graph *(DAG)* + each node has one parent at most
- A *Decision Tree* is a tree where nodes associated with attributes, edges associated with attribute values, and leaves associated with decisions



Example:



Learning Decision Trees



Attribute Selection Criteria

Logical Based

Information Based

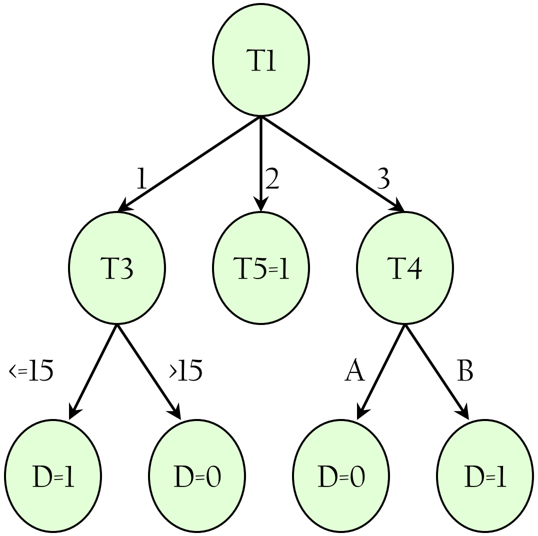
Statistical Based

Information Theory

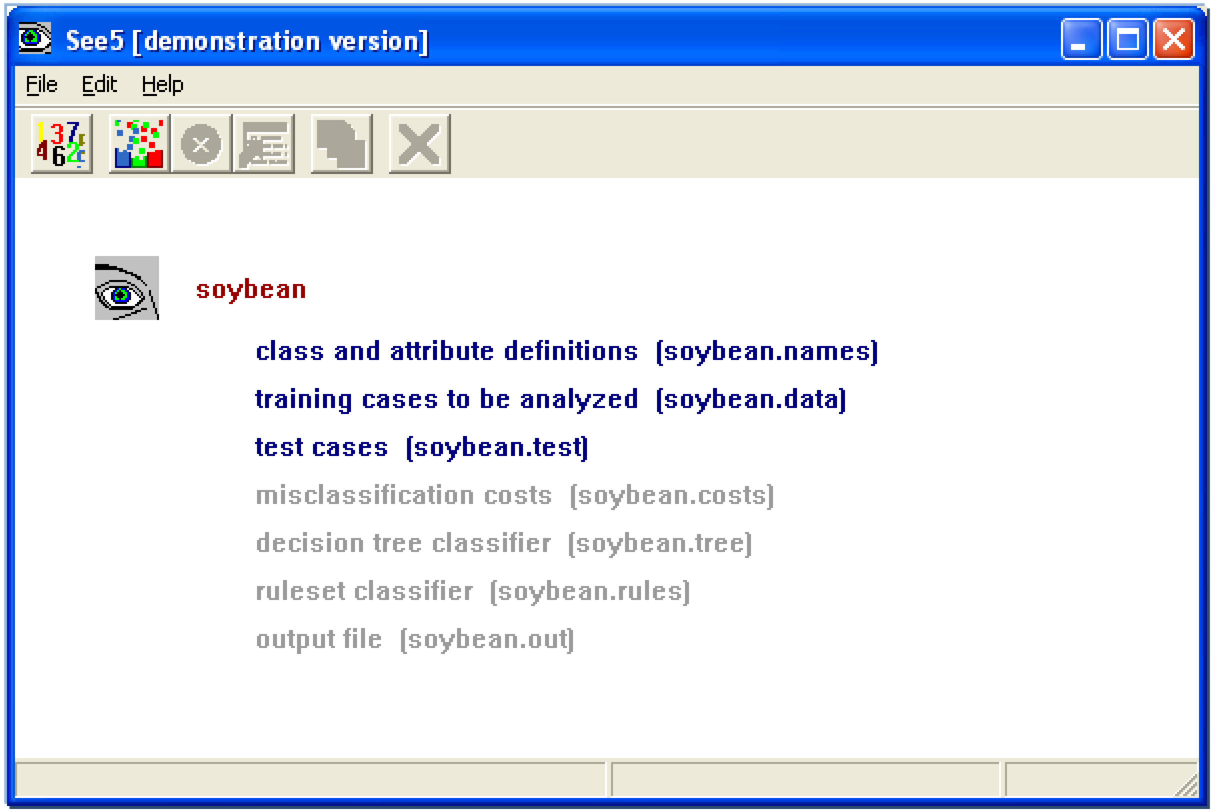
Example

- ●T2 is quantized into two intervals at 21 (T2<=21) and (T2>21)
- ●T3 is quantized into two intervals at 15 (T3<=15) and (T3>15)

T1	T2	T3	T4	D
1	25	10	A	1
1	30	30	A	0
1	35	25	B	0
1	22	35	B	0
1	19	10	B	1
2	22	30	A	1
2	33	18	B	1
2	14	5	A	1
2	31	15	B	1
3	21	20	A	0
3	15	10	A	0
3	25	20	B	1
3	18	20	B	1
3	20	36	B	1



C5



soybean

class and attribute definitions [soybean.names]

training cases to be analyzed [soybean.data]

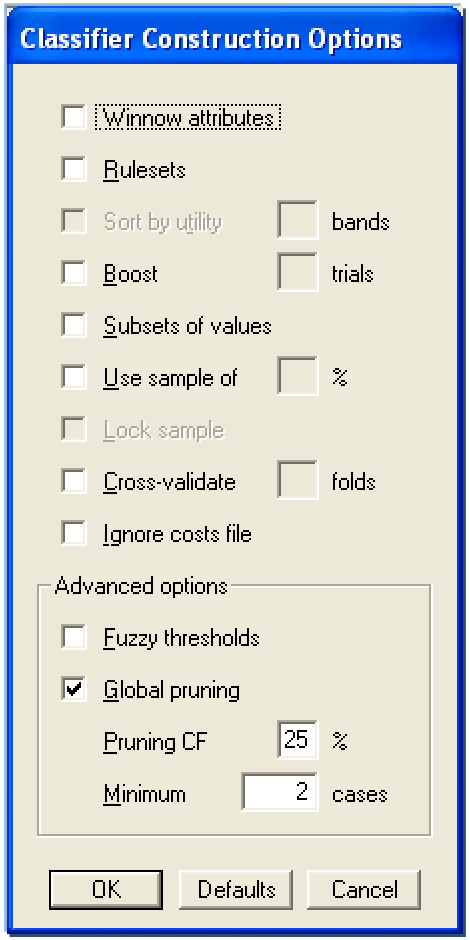
test cases [soybean.test]

misclassification costs [soybean.costs]

decision tree classifier [soybean.tree]

ruleset classifier [soybean.rules]

output file [soybean.out]



□ Winnow attributes

□ Rulesets

□ Sort by utility   □ bands

□ Boost   □ trials

□ Subsets of values

□ Use sample of   □ %

□ Lock sample

□ Cross-validate   □ folds

□ Ignore costs file

Advanced options

□ Fuzzy thresholds

☑ Global pruning

Pruning CF 25 %

Minimum 2 cases

Decision Trees

C5

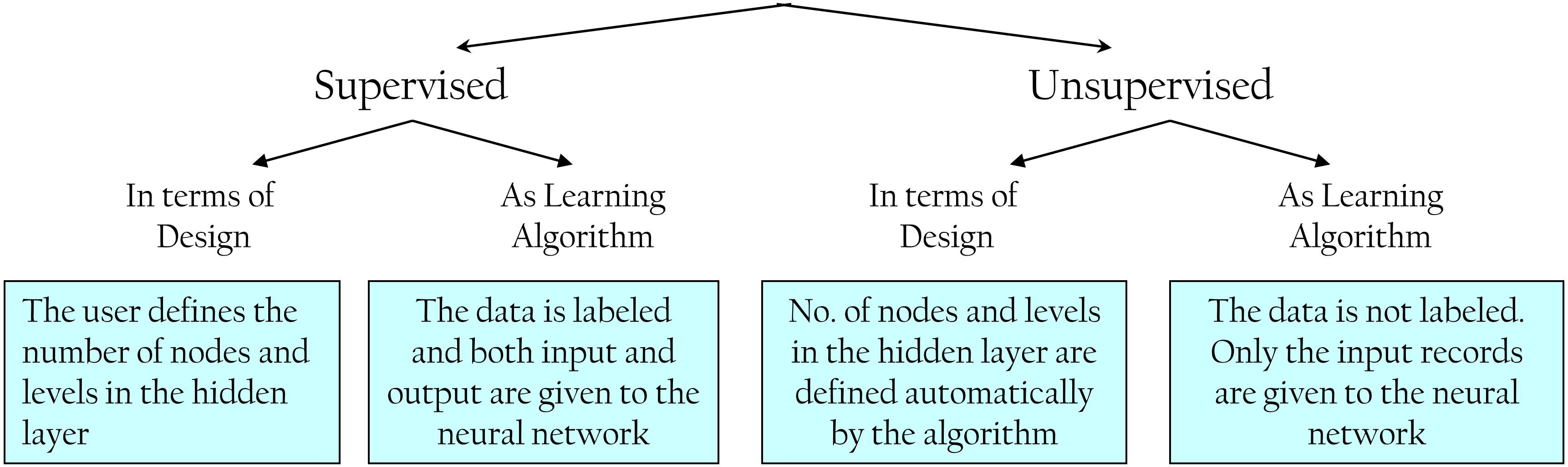
DEMO

NEURAL NETWORKS

Part 13

How It Works?

Learning Neural Networks



Supervised

In terms of Design

The user defines the number of nodes and levels in the hidden layer

As Learning Algorithm

The data is labeled and both input and output are given to the neural network

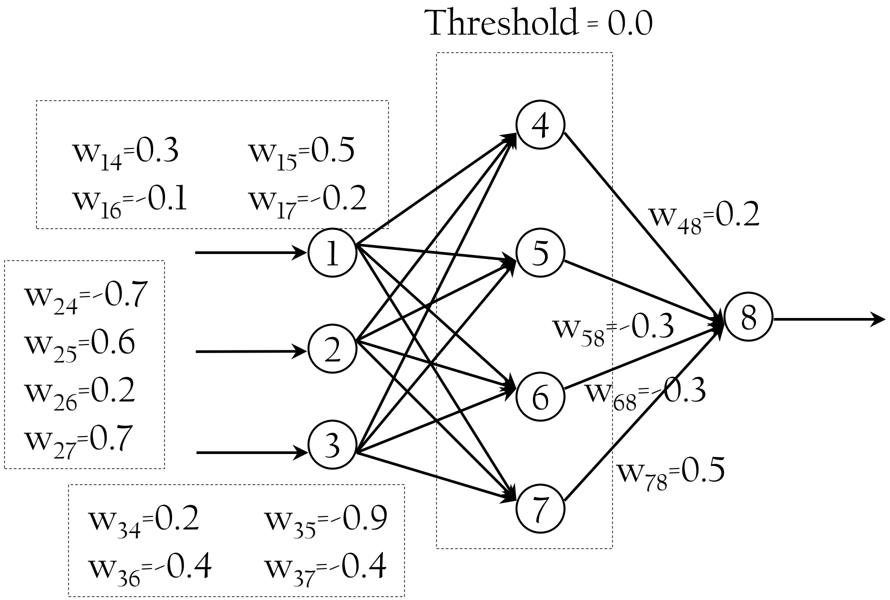
Unsupervised

In terms of Design

No. of nodes and levels in the hidden layer are defined automatically by the algorithm

As Learning Algorithm

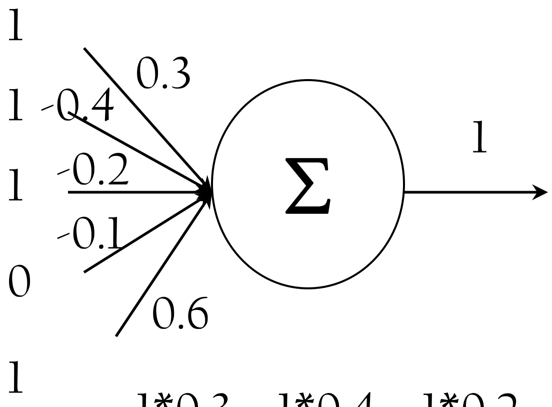
The data is not labeled. Only the input records are given to the neural network



Test Data

A	B	C	Decision
0	0	0	
0	0	1	
0	1	0	
0	1	1	1
1	0	0	
1	0	1	
1	1	0	
1	1	1	

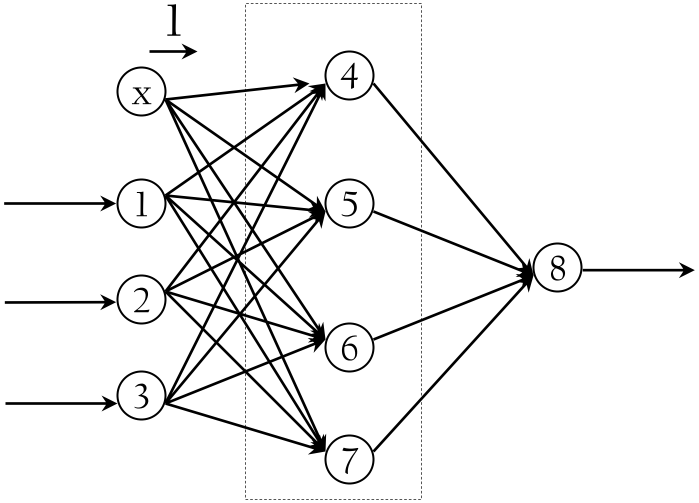
Learning Neural Networks



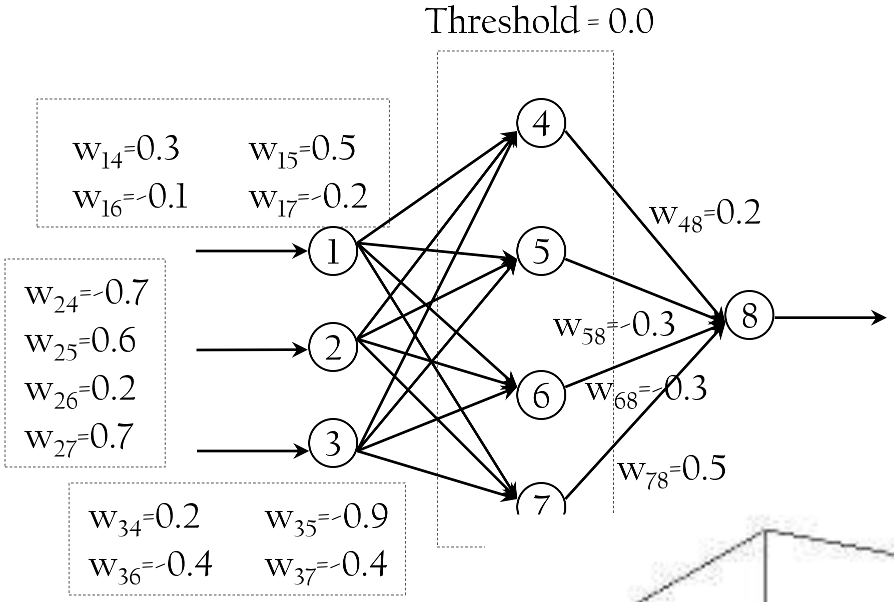
The Sigmoid Function

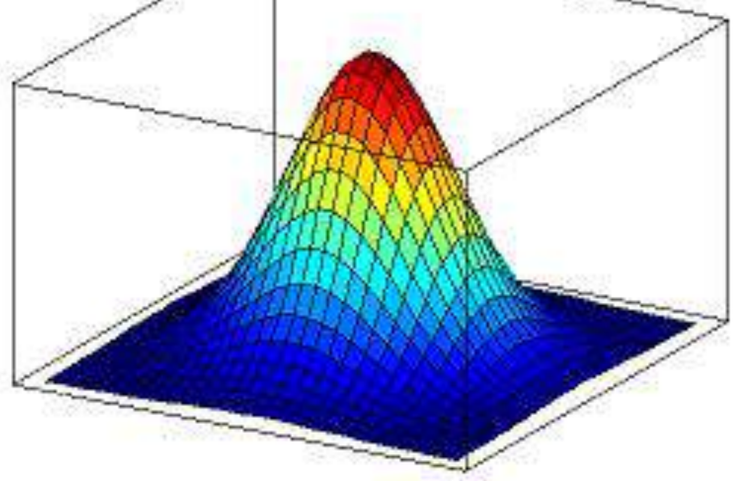
=1*0.3 – 1*0.4 – 1*0.2 – 0*0.1 + 1*0.6 = 0.3 > 0.0

To avoid setting the threshold:



Learning Neural Networks





Test Data

A	B	C	Decision
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

MACHINE TRANSLATION

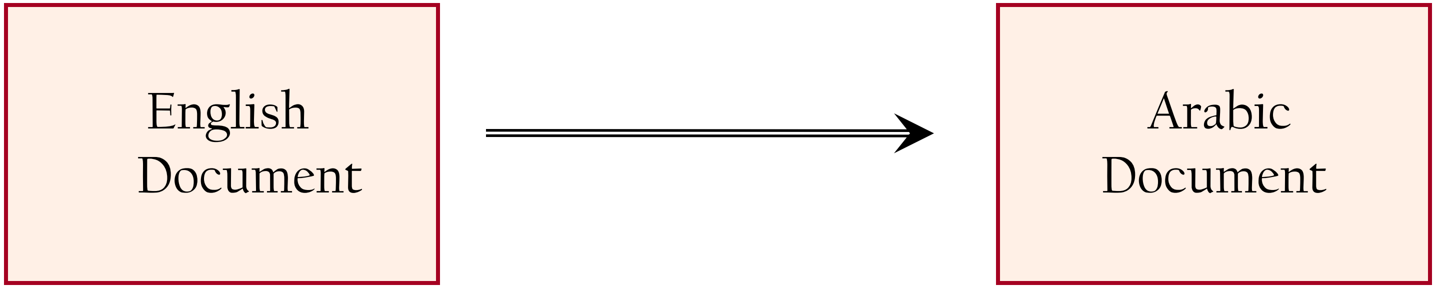
Part 14

Statistical Machine Translation

Statistical Machine Translation

- For each English sentence “e”, we need the Arabic sentence “a” which maximize $P(a|e)$

  $P(a|e)=P(a)*P(e|a)/P(e)$



English Document

Arabic Document

Language Model

- A statistical **language model** assigns a probability to a sequence of *m* words by means of a probability distribution
- Record every sentence that anyone ever says in Arabic; Suppose you record a database of one billion utterances; If the sentence "كيف حالك؟" appears 76,413 times in that database, then we say P(كيف حالك؟) = 76,413/1,000,000,000 = 0.000076413
- One big problem is that many perfectly good sentences will be assigned a P(a) of zero

Arabic Sentence	Probability
كيف حالك	0.000076413
الولد سعيد	0.000066392

N-Grams

- An n-word substring is called an n-gram
- If n=2, we say bigram. If n=3, we say trigram
- Let P(y | x) be the probability that word y follows word x

P(y | x) = number-of-occurrences(“xy”) / number-of-occurrences(“x”)

P(z | x y) = number-of-occurrences(“xyz”) / number-of-occurrences(“xy”)

➔ P(ذهب الولد إلى المدرسة) = P(ذهب | start-of-sentence) *
P(الولد | ذهب) * P(إلى | الولد) * P(المدرسة | إلى) *
P(end-of-sentence | المدرسة)

➔ P(ذهب الولد إلى المدرسة) = P(ذهب | start-of-sentence) *
P(الولد | start-of-sentence ,ذهب) * P(إلى | ذهب, الولد) *
P(المدرسة | الولد, إلى) * P(end-of-sentence | إلى، المدرسة) *
P(end-of-sentence | end-of-sentence ,المدرسة)

N-Grams Language Model

$$P(w_1,\ldots,w_m)=\prod_{i=1}^{m}P(w_i\mid w_1,\ldots,w_{i-1})\approx\prod_{i=1}^{m}P(w_i\mid w_{i-(n-1)},\ldots,w_{i-1})$$

$$P(w_i\mid w_{i-(n-1)},\ldots,w_{i-1})=\frac{count(w_{i-(n-1)},\ldots,w_i)}{count(w_{i-(n-1)},\ldots,w_{i-1})}$$

Example:

In a bigram (n=2) language model, the approximation looks like

$$P(I,saw,the,red,house)\approx P(I)P(saw\mid I)P(the\mid saw)P(red\mid the)P(house\mid red)$$

In a trigram (n=3) language model, the approximation looks like

$$P(I,saw,the,red,house)\approx P(I)P(saw\mid I)P(the\mid I,saw)P(red\mid saw,the)P(house\mid the,red)$$

Translation Model

- P(e | a), the probability of an English string "e" given an Arabic string "a"; This is called a translation model
- P(e | a) will be a module in overall English-to-Arabic machine translation system; When we see an actual English string e, we want to reason backwards ... What Arabic string a is likely to be expressed, and likely to subsequently translate to e? We're looking for the a that maximizes P(a) * P(e | a)

Arabic Sentence	English Sentence	P(a|e)
ذهب الولد إلى المدرسة	The boy went to School	0.0034
إنخفاض البورصة اليوم	Today, the stock market went down	0.00021
:	:	

- Example, BuckWalter

Translation Model

- For each word a_i in an Arabic sentence (i = 1 … l), we choose a fertility ϕ_i. The choice of fertility depends on the Arabic word in question. It is not dependent on the other Arabic words in the Arabic sentence, or on their fertilities
- For each word a_i, we generate ϕ_i English words. The choice of English word depends on the Arabic word that generates it. It is not dependent on the Arabic context around the Arabic word. It is not dependent on other English words that have been generated from this or any other Arabic word
- All those English words are permuted. Each English word is assigned an absolute target "position slot." For example, one word may be assigned position 3, and another word may be assigned position 2 -- the latter word would then precede the former in the final English sentence. The choice of position for a English word is dependent solely on the absolute position of the Arabic word that generates it

REFERENCES

- W. Weaver (1955). Translation (1949). In: *Machine Translation of Languages*, MIT Press, Cambridge, MA.
- P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer (1993). The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, **19(2)**, 263-311.
- S. Vogel, H. Ney and C. Tillmann. 1996. HMM-based Word Alignment in StatisticalTranslation. In COLING '96: The 16th International Conference on Computational Linguistics, pp. 836-841, Copenhagen, Denmark.
- F. Och and H. Ney. (2003). A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics, 29(1):19-51
- P. Koehn, F.J. Och, and D. Marcu (2003). Statistical phrase based translation. In *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT/NAACL)*.
- D. Chiang (2005). A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*.
- F. Och and H. Ney. (2003). A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics, 29(1):19-51
- P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, E. Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. ACL 2007, Demonstration Session, Prague, Czech Republic
- Q. Gao, S. Vogel, "Parallel Implementations of Word Alignment Tool", Software Engineering, Testing, and Quality Assurance for Natural Language Processing, pp. 49-57, June, 2008
- W. J. Hutchens and H. Somers. (1992). An Introduction to Machine Translation, 18.3:322. ISBN 0-12-36280-X

REFERENCES

- W. The Sage Dictionary of Statistics, pg. 76, Duncan Cramer, Dennis Howitt, 2004, ISBN 076194138X
- E.L. Lehmann and Joseph P. Romano (2005). *Testing Statistical Hypotheses* (3E ed.). New York, NY: Springer. ISBN 0387988645
- D.R. Cox and D.V.Hinkley (1974). *Theoretical Statistics*. ISBN 0412124293.
- Fisher, Sir Ronald A. (1956) [1935]. "Mathematics of a Lady Tasting Tea". in James Roy Newman. *The World of Mathematics, volume 3.* http://books.google.com/books?id=oKZwtLQTmNAC&pg=PA1512&dq=%22mathematics+of+a+lady+tasting+tea%22&sig=8-NQlCLzrh-oV0wjfwa0EgspSNU
- R.A. Fisher, the Life of a Scientist, Box, 1978, p134
- Mccloskey, Deirdre (2008). *The Cult of Statistical Significance*. Ann Arbor: University of Michigan Press. ISBN 0472050079
- *What If There Were No Significance Tests?*, Harlow, Mulaik & Steiger, 1997, ISBN 978-0-8058-2634-0
- Rosnow, R.L. & Rosenthal, R. (1989). Statistical procedures and the justification of knowledge in psychological science. American Psychologist, 44, 1276-1284
- Loftus, G.R. 1991. On the tyranny of hypothesis testing in the social sciences. Contemporary Psychology 36: 102-105
- Cohen, J. 1990. Things I have learned (so far). American Psychologist 45: 1304-1312. ^ Introductory Statistics, Fifth Edition, 1999, pg. 521, Neil A. Weiss, ISBN 0-201-59877-9
- Ioannidis JP (July 2005). "Contradicted and initially stronger effects in highly cited clinical research". *JAMA* **294** (2): 218–28.

Tutorial on Statistics, Probability and Information Theory for Language Engineers

Prof. Ibrahim F. Imam

Full Professor and Assistant Dean,
College of Computing and Information Technology
Arab Academy for Science, Technology & Maritime Transport, Cairo

Adjunct Professor, Computer Science Department,
College of Engineering, Virginia Tech. University, VA, USA

Email: ifi05@yahoo.com Phone: 012-2242929

Contents of the Tutorial

1- Main Presentation in PDF Slides

2- Presentation on Statistics in Excel in PDF Slides

3- Statistical Machine Translation File “SMT.rtf”

4- Three Files on How to Apply Statistics in Excel

5- Two Machine Learning Demo Programs C5 & Opus

6-

OUTLINE

<u>BASIC MATHEMATICS</u>

<u>Part 0</u>

Basic Concepts

BASIC MATHEMATICS

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n$$

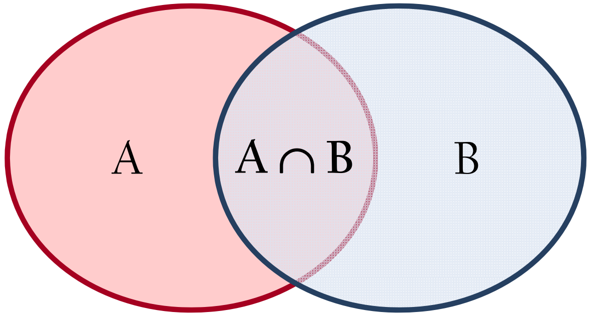
$$\prod_{i=1}^{n} i = 1 * 2 * \ldots * n$$

$$\sum_{i=1}^{n} ki = k \sum_{i=1}^{n} i$$

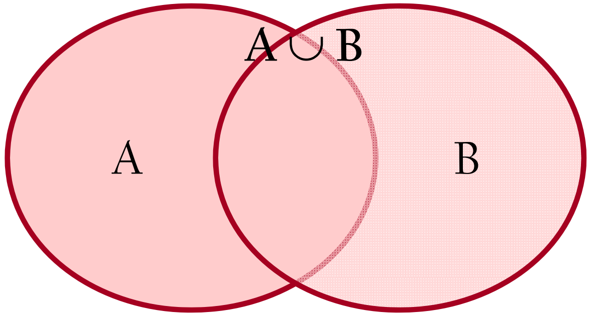
$$\prod_{i=1}^{n} ki = k \prod_{i=1}^{n} i$$

Introduction to Set Theory

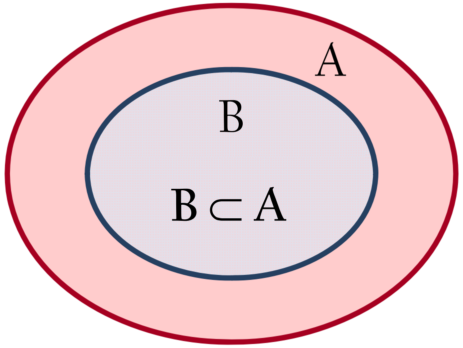
- A set is a collection of distinct items (Example: A = {1, 2, 3, 4, 5})



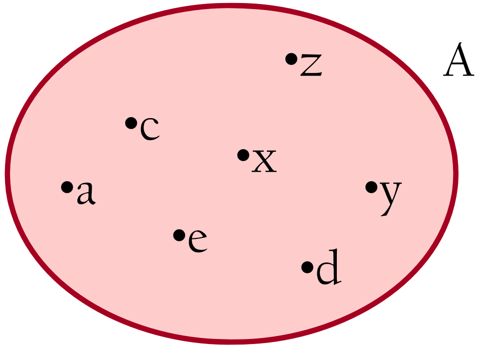
Intersection



Union



Sub-set & Super-set



$x \in A$; $a \in A$; $d \in A$; ...

Introduction to Set Theory

- $A = \{a, c, e, d, x, y, z\}$ $B = \{b, c, d, y, m, n\}$ $C = \{c, d\}$

$A \cap B = \{c, d, y\}$

Intersection

$A \cup B = \{a, b, c, d, e, m, n, x, y, z\}$

Union

$A \not\subset B$ $C \subset B$ $C \subset A$

Sub-set & Super-set

$x \in A;$ $x \notin B;$ $x \notin C$

Belong Relationship

Φ/ϕ is the empty set $\cap \cup \subset \not\subset \in \notin \neg \wedge \vee$

Introduction to Set Theory

- $A \cap (B \cap C) = (A \cap B) \cap C$ & $A \cup (B \cup C) = (A \cup B) \cup C$
- $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
- $\neg(\neg A) = A$
- $\neg(A \cap B) = \neg A \cup \neg B$

Introduction to Propositional Logic

- It is also called the Zero Order Logic
- A sentence X can be either true or false (1 or 0)

X
0
1

Y
0
1

X	Y	$X \wedge Y$
0	0	0
0	1	0
1	0	0
1	1	1

X	Y	$X \vee Y$
0	0	0
0	1	1
1	0	1
1	1	1

X	Y	$X \rightarrow Y$
0	0	1
0	1	1
1	0	0
1	1	1

X	Y	X XOR Y
0	0	1
0	1	0
1	0	0
1	1	1

$X \rightarrow Y = \neg X \vee Y$
$\neg(X \wedge Y) = \neg X \vee \neg Y$
$X \wedge X = X$ & $X \vee X = X$
$X \vee (Y \wedge Z) = (X \vee Y) \wedge (X \vee Z)$
$\neg(\neg X) = X$

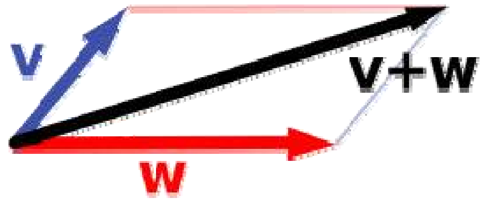
Introduction to Vectors

Part 1

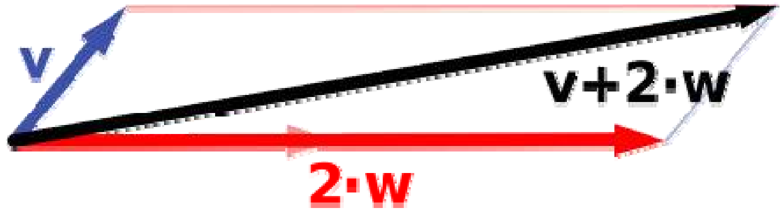
Representing Documents As Vectors

Introduction to Vectors

Adding two vectors
$(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2)$

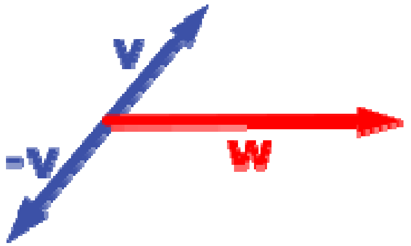


Multiplying a vector by a constant and adding it to another vector
$(x_1, y_1) + (2.x_2,2. y_2) = (x_1 + 2x_2, y_1 + 2y_2)$



Multiplying a vector by -1
$-(x_1, y_1) = (-x_1, -y_1)$

Multiplying a vector by a constant
$2 . (x_2, y_2) = (2x_2, 2y_2)$



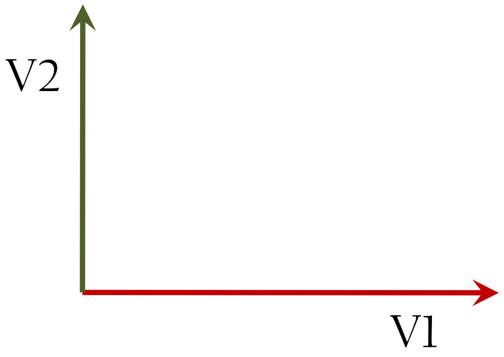
Introduction to Vectors

Multiplying two orthogonal vectors equal to zero.

Examples:

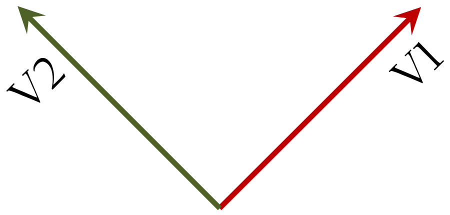
V1 =(5, 0) & V2=(0, 4)

V1 . V2 = 0



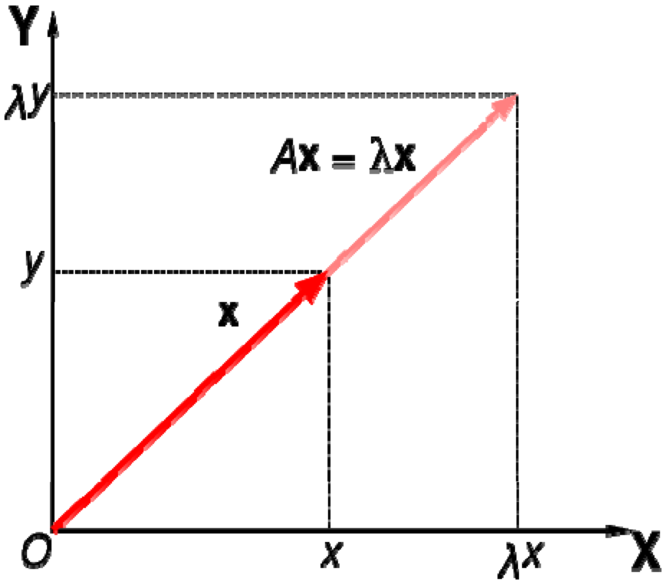
V1 =(5, 4) & V2=(-4, 5)

V1 . V2 = 0



Eigen Values & Eigen Vectors

- An eigenvector of a matrix *A* is a nonzero vector *x*, where *A.x* is similar to applying a linear transformation λ to *x* which, may change in length, but not direction
- *A* acts to stretch the vector *x*, not change its direction, so *x* is an eigenvector of *A*



$$Ax - \lambda Ix = 0$$

$$(A - \lambda I)x = 0$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \lambda\begin{bmatrix} x \\ y \end{bmatrix}$$

$$if\ \ there\ exist\ an\ inverse\ (A - \lambda I)^{-1},\ then\ x = 0$$

$$we\ need\ \det(A - \lambda I) = 0\ to\ avoid\ the\ trevial\ solution\ x = 0$$

$$\det(A - \lambda I) = 0$$

Example on Eigen Values & Eigen Vectors

- Suppose ***A*** is 2x2 matrix

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$\det \begin{bmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{bmatrix} = (2-\lambda)^2 - 1 = 0$$

$$\lambda = 1 \quad or \quad \lambda = 3$$

$$for\ \lambda = 3, \quad \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 3 \begin{bmatrix} x \\ y \end{bmatrix}$$

$$for\ \lambda = 1, \quad \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 1 \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} 2x+y \\ x+2y \end{bmatrix} = \begin{bmatrix} 3x \\ 3y \end{bmatrix}$$

$$2x+y=3x$$

$$\boxed{x=y}$$

$$\begin{bmatrix} 2x+y \\ x+2y \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

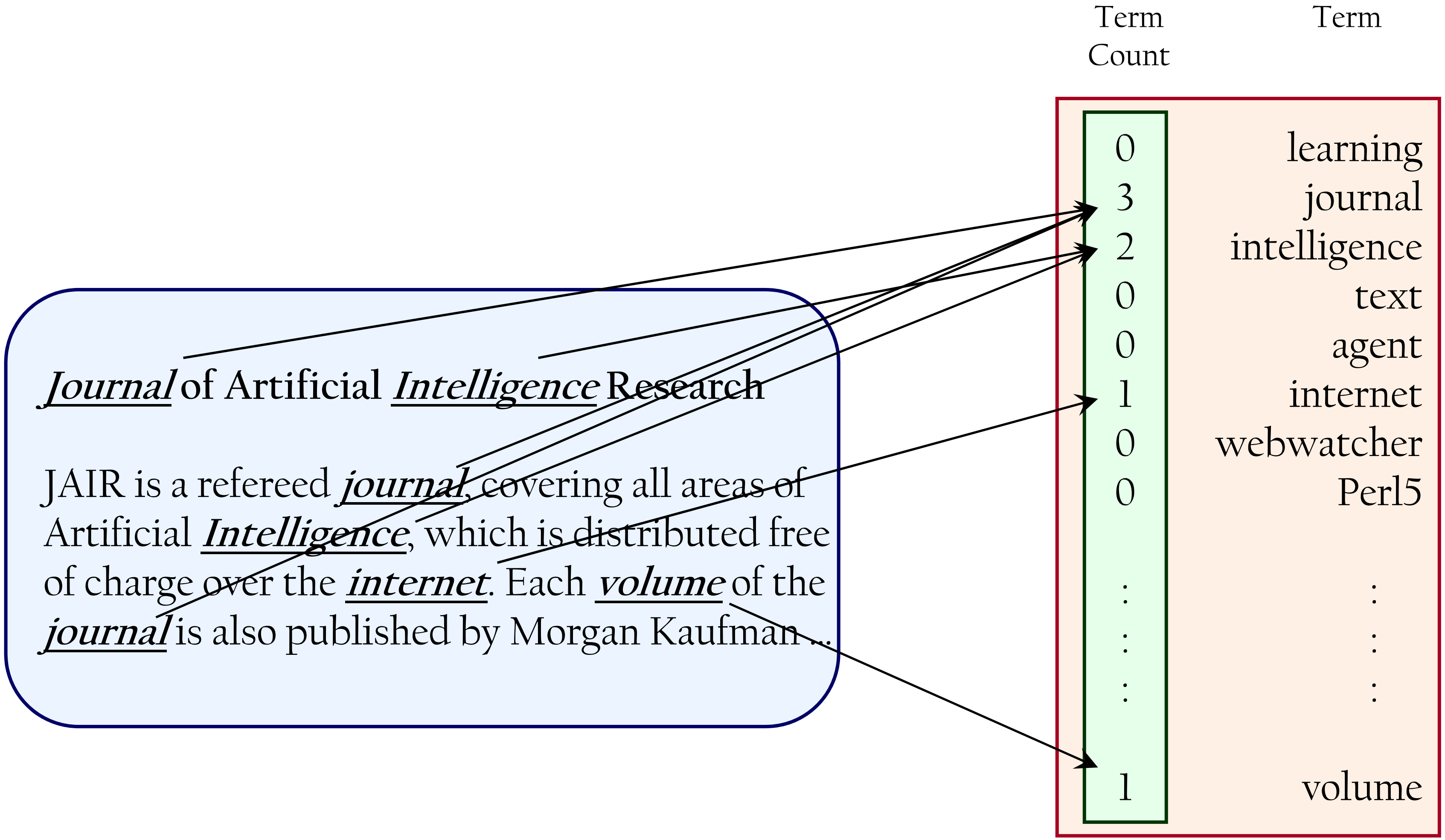
$$2x+y=x$$

$$\boxed{x=-y}$$

The eigenvectors are:

$$\boxed{\begin{bmatrix} 1 \\ 1 \end{bmatrix}} \qquad \boxed{\begin{bmatrix} 1 \\ -1 \end{bmatrix}}$$

Representing Documents as Vectors



Journal of Artificial ***Intelligence*** Research

JAIR is a refereed ***journal***, covering all areas of Artificial ***Intelligence***, which is distributed free of charge over the ***internet***. Each ***volume*** of the ***journal*** is also published by Morgan Kaufman ...

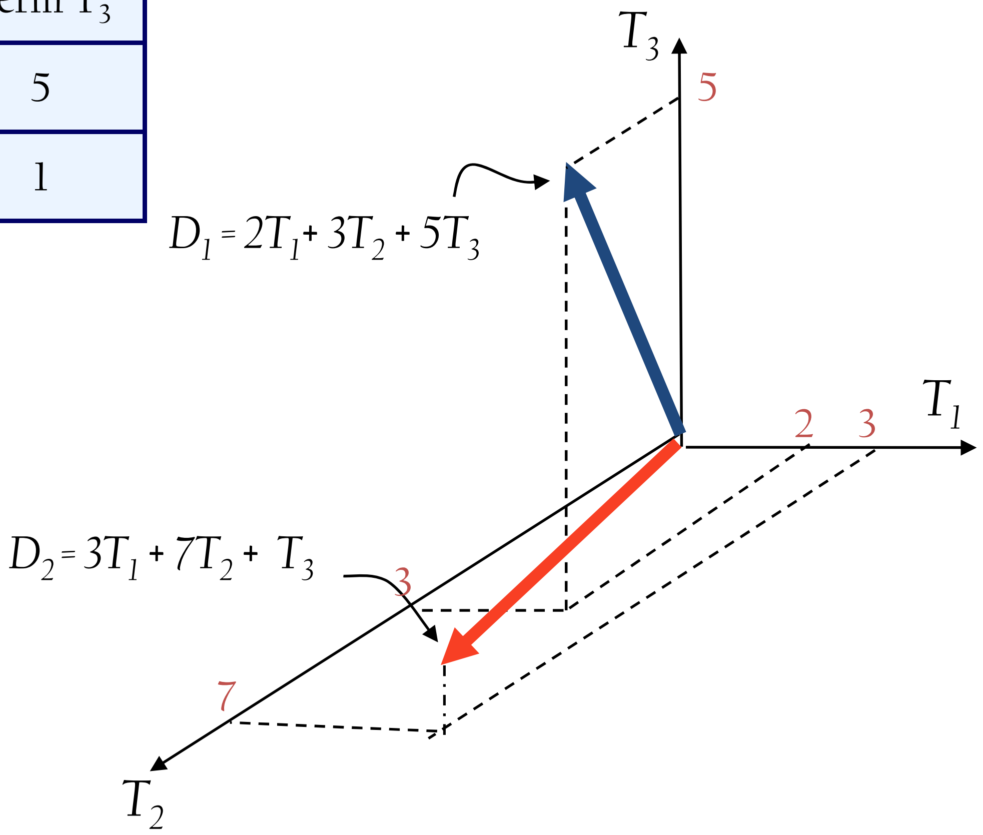
Term Count	Term
0	learning
3	journal
2	intelligence
0	text
0	agent
1	internet
0	webwatcher
0	Perl5
:	:
:	:
:	:
1	volume

Documents as Vectors

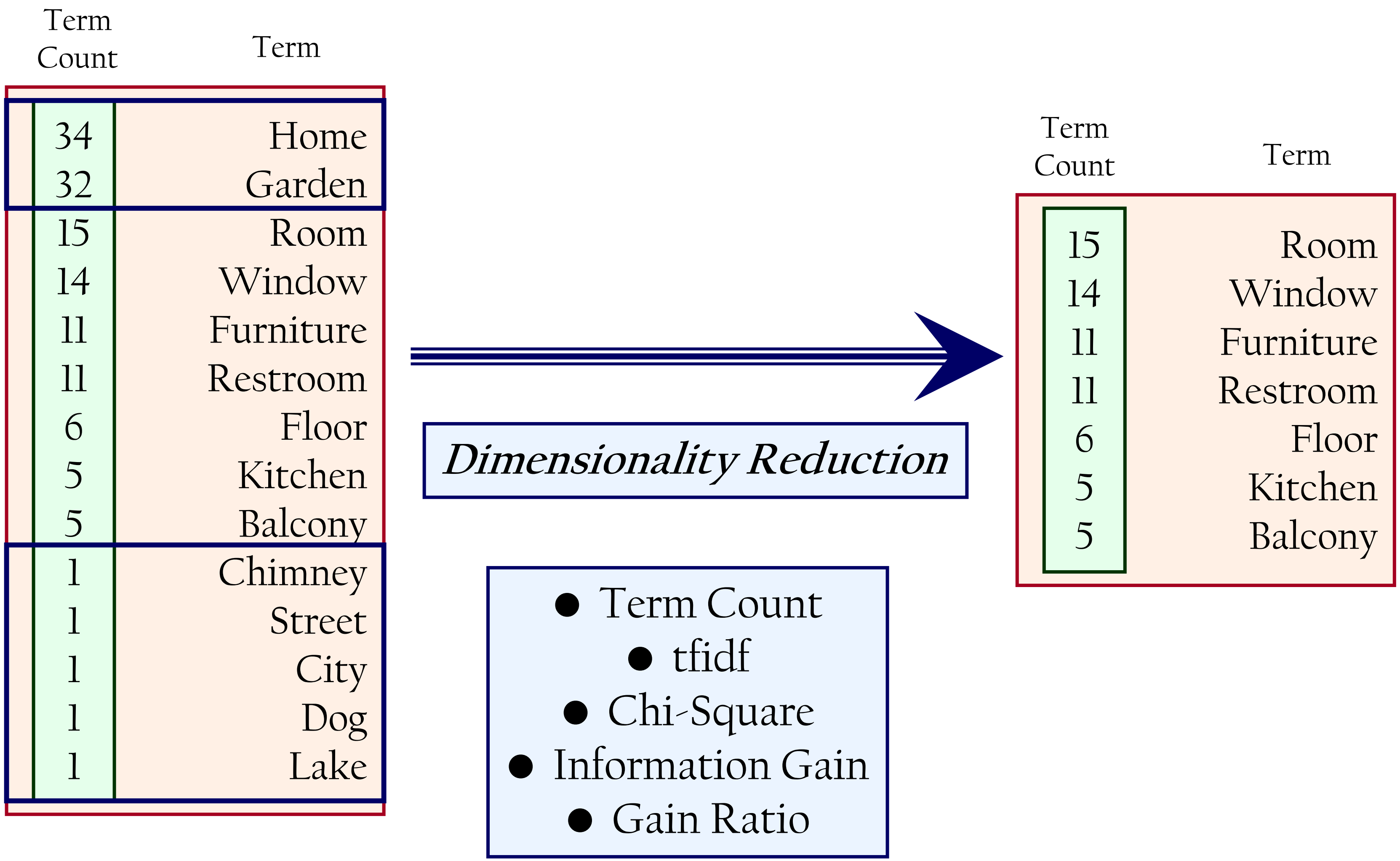
Suppose we have two documents containing three nouns only

	Term T_1	Term T_2	Term T_3
Document D_1	2	3	5
Document D_2	3	7	1

$$D_1 \quad \begin{pmatrix} 2 \\ 3 \\ 5 \end{pmatrix} \qquad D_2 \quad \begin{pmatrix} 3 \\ 7 \\ 1 \end{pmatrix}$$



Dimensionality Reduction



Term Count	Term
34	Home
32	Garden
15	Room
14	Window
11	Furniture
11	Restroom
6	Floor
5	Kitchen
5	Balcony
1	Chimney
1	Street
1	City
1	Dog
1	Lake

Dimensionality Reduction

- Term Count
- tfidf
- Chi-Square
- Information Gain
- Gain Ratio

Term Count	Term
15	Room
14	Window
11	Furniture
11	Restroom
6	Floor
5	Kitchen
5	Balcony

PROBABILITY

Part 2

-Introduction
-Terminology

What Is Probability?

- A priori probability ***P(e)***: The chance that e happens
- Conditional probability ***P(f / e)***: The chance of f given e
- Joint probability ***P(e, f)***: The chance of e and f both happening; If e and f are independent, then P(e, f) = P(e) * P(f); If e and f are dependent then P(e, f) = P(e) * P(f | e)

  For example, if e stands for "the first roll of the die comes up 5" and f stands for "the second roll of the die comes up 3," then P(e,f) = P(e) * P(f) = 1/6 * 1/6 = 1/36.

$$\sum_{e} P(e) = 1 \qquad \sum_{e} P(e \mid f) = 1$$

BASIC Probabilities

$$P(A \cup B) = \begin{cases} P(A) + P(B) & A\,\&\,B \text{ are not dependant} \\ P(A) + P(B) - P(A,B) & A\,\&\,B \text{ are dependant} \end{cases}$$

- For example, when drawing a single card at random from a regular deck of cards, the chance of getting a heart or a face card (J,Q,K) (or one that is both) is

$$\frac{13}{52} + \frac{12}{52} - \frac{3}{52} = \frac{22}{52}$$

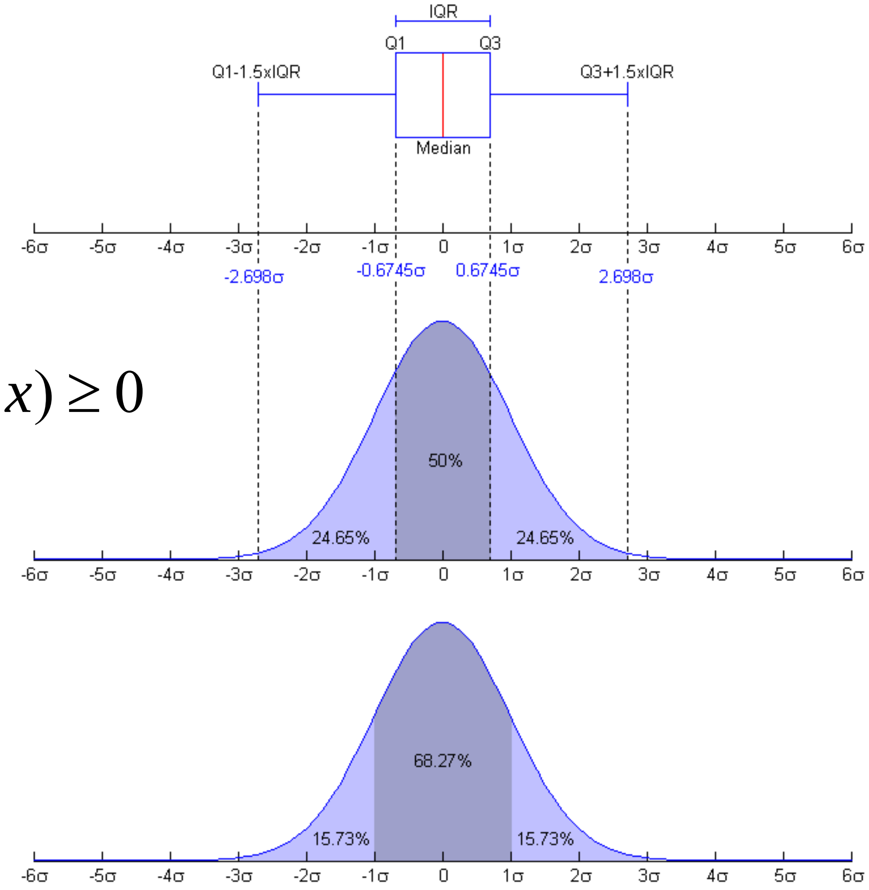
A	$P(A) \in [0,1]$
not A	$P(A') = 1 - P(A)$
A or B	$P(A \cup B) = P(A) + P(B) - P(A \cap B)$ $= P(A) + P(B) \quad \text{if A and B are mutually exclusive}$
A and B	$P(A \cap B) = P(A\|B)P(B)$ $= P(A)P(B) \quad \text{if A and B are independent}$
A given B	$P(A \mid B) = \dfrac{P(A \cap B)}{P(B)}$

Probability Density Function PDF

- Probability density function (pdf) is a function that represents a probability distribution in terms of integrals

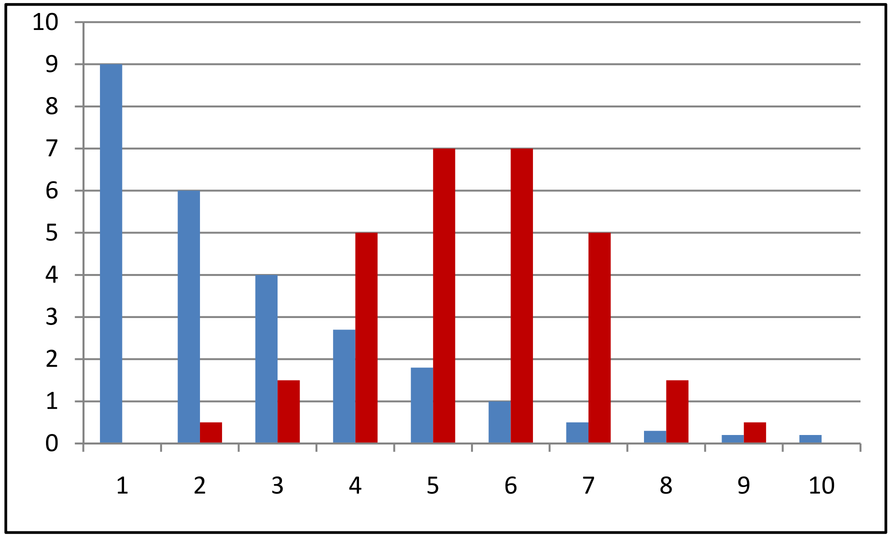
$$\int_a^b f(x)\,dx$$

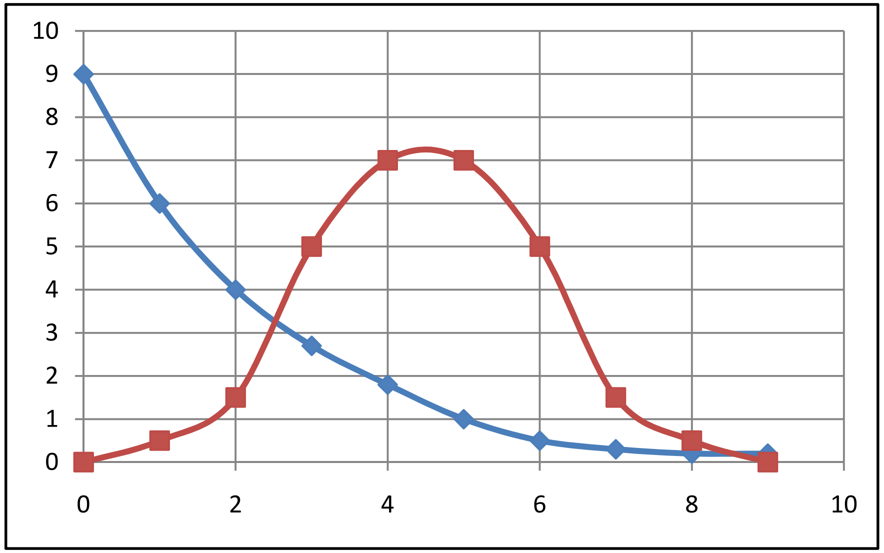
$$\int_{-\infty}^{\infty} f(x)dx = 1 \qquad \& \qquad f(x) \geq 0$$



Probability Density Function PDF

- The Summation is used with Discrete Data





Conditional & Bayesian Probability

- **Conditional probability** is the probability of some event *A*, given the occurrence of some other event *B*
- Conditional probability is written *P*(*A*|*B*), and is read "the probability of *A*, given *B*"

$$P(A \mid B) = \frac{P(A,B)}{P(B)}$$

- Bayesian probability, the probability of a hypothesis given the data (the *posterior*), is proportional to the product of the likelihood times the prior probability (often just called the *prior*)
- The likelihood brings in the effect of the data, while the prior specifies the belief in the hypothesis before the data was observed

$$P(A \mid B) = \frac{P(A)P(B \mid A)}{P(B)}$$

STATISTICS

Part 3

Introduction

Statistics

- Statistics is a Mathematical Science pertaining to the ***collection***, ***analysis***, ***interpretation** or* ***explanation**, and **presentation*** of data

Statistical Terminologies

- Measures of Central Tendency ***(Mean***, Median, Mode)

$$\bar{x} = (1/n)\sum_{i=1}^{n} x_i$$

- ***Population Variance*** measures statistical dispersion of data points from the expected value (mean)

$$Var(X) = E\left[(X - E(X))^2\right]$$

$$= (1/n)\sum_{i=1}^{n}(x_i - \bar{x})^2 = \sigma^2$$

- ***Standard Deviation*** is a measure of the variability or dispersion of a population; Low SD indicates very close data points to the mean; High SD indicates spread out data points

$$sd(X) = \sqrt{\sigma^2}$$

- ***Covariance*** measures how much two variables change together

$$Cov(X,Y) = E\left[(X - E(X))(Y - E(Y))\right]$$

- ***Correlation*** (coefficient) indicates the strength and direction of a *linear* relationship between two random variables

$$Corr(X,Y) = \frac{Cov(X,Y)}{sd(X) * sd(Y)} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

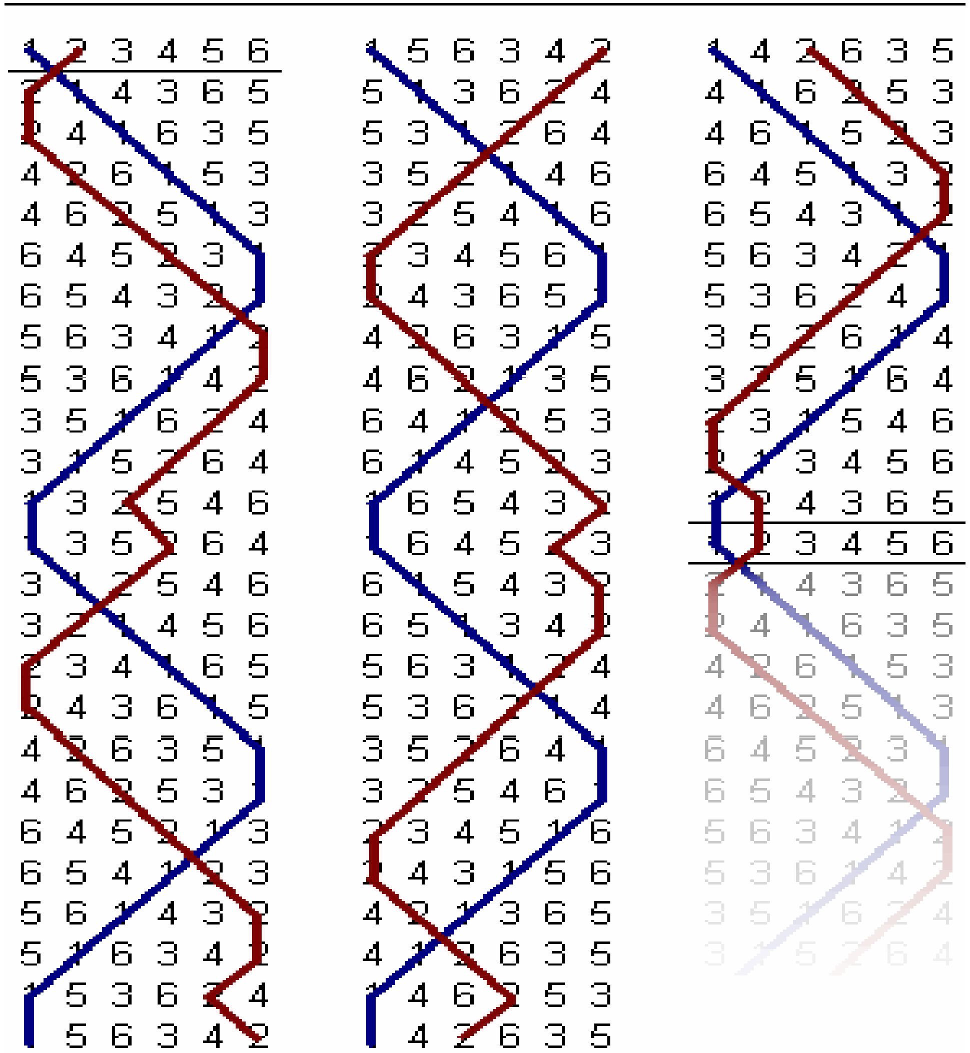
STATISTICS

Part 4

Permutations & Computations

Introduction to Permutations & Computations

Plain Bob Minor



```
1 2 3 4 5 6    1 5 6 3 4 2    1 4 2 6 3 5
2 1 4 3 6 5    5 1 3 6 2 4    4 1 6 2 5 3
2 4 1 6 3 5    5 3 1 2 6 4    4 6 1 5 2 3
4 2 6 1 5 3    3 5 2 1 4 6    6 4 5 1 3 2
4 6 2 5 1 3    3 2 5 4 1 6    6 5 4 3 1 2
6 4 5 2 3 1    2 3 4 5 6 1    5 6 3 4 2 1
6 5 4 3 2 1    2 4 3 6 5 1    5 3 6 2 4 1
5 6 3 4 1 2    4 2 6 3 1 5    3 5 2 6 1 4
5 3 6 1 4 2    4 6 2 1 3 5    3 2 5 1 6 4
3 5 1 6 2 4    6 4 1 2 5 3    2 3 1 5 4 6
3 1 5 2 6 4    6 1 4 5 2 3    2 1 3 4 5 6
1 3 2 5 4 6    1 6 5 4 3 2    1 2 4 3 6 5
1 3 5 2 6 4    1 6 4 5 2 3    1 2 3 4 5 6
3 1 2 5 4 6    6 1 5 4 3 2    2 1 4 3 6 5
3 2 1 4 5 6    6 5 1 3 4 2    2 4 1 6 3 5
2 3 4 1 6 5    5 6 3 1 2 4    4 2 6 1 5 3
2 4 3 6 1 5    5 3 6 2 1 4    4 6 2 5 1 3
4 2 6 3 5 1    3 5 2 6 4 1    6 4 5 2 3 1
4 6 2 5 3 1    3 2 5 4 6 1    6 5 4 3 2 1
6 4 5 2 1 3    2 3 4 5 1 6    5 6 3 4 1 2
6 5 4 1 2 3    2 4 3 1 5 6    5 3 6 1 4 2
5 6 1 4 3 2    4 2 1 3 6 5    3 5 1 6 2 4
5 1 6 3 4 2    4 1 2 6 3 5    3 1 5 2 6 4
1 5 3 6 2 4    1 4 6 2 5 3
1 5 6 3 4 2    1 4 2 6 3 5
```

Permutations

- Suppose an ordered set of *n* different objects
- For ***ordered*** selection of *r* objects from a set of *n* (n≥*r*) different objects, the number of permutations of *r* from *n*, *i.e.* the number of different possible ordered selections, is usually denoted by $P_{r.}^{n}$

$$P_r^n = \frac{n!}{(n-r)!}$$

لدينا ثلاثة أرقام ا، ب، ج. يتم إختيار أول رقم وضربه
فى 10، ويتم ضرب الرقم الثانى فى 100، ويتم
ضرف الرقم الثالث فى 1000، ثم يتم جمع الثلاثة
أرقام الجديدة. كم رقم يمكن إستنتاجه من هذه الأرقام
الثلاثة.

مثال: 1، 2، 3 (3210، 3120، 2130، ...)
الحل: ؟

$P_0^n = 1$	$P_1^n = n$	$P_n^n = n!$

Permutations

Example:

r	g	b	y

Suppose we have 4 elements and need to select 3 elements in order; there are 24 different combinations

$$P_3^4 = \frac{4!}{(4-3)!} = \frac{4!}{1!} = 4*3*2 = 24$$

r	g	b	r	b	g	g	r	b	g	b	r
b	g	r	b	r	g	r	g	y	r	y	g
g	r	y	g	y	r	y	r	g	y	g	r
r	b	y	r	y	b	b	r	y	b	y	r
y	r	b	y	b	r	g	b	y	g	y	b
b	g	y	b	y	g	y	g	b	y	b	g

Permutations

- Suppose a set {A, B, C}, we have 6 (=3!) permutations of {*A*, *B*, *C*} are *ABC*, *ACB*, *BAC*, *BCA*, *CAB and CBA*
- Suppose a set {A, B, C, D}, there are 24 = P^4_3 = (4 × 3 × 2) permutations of 3 letters from {*A*, *B*, *C*, *D*}
- If the *n* objects are not all different, and there are n_r objects of type 1, n_2 objects of type 2, ..., n_k objects of type *k*, where $n_1+n_2+\ldots+n_k=n$, then the number of different ordered arrangements is

$$\frac{n!}{n_1!n_2!n_3!\ldots n_k!}$$

a	a	a	b	b	b	c	c	c	c	d	d	d	d

$$\frac{14!}{3!*3!*4!*4!}$$

Computations

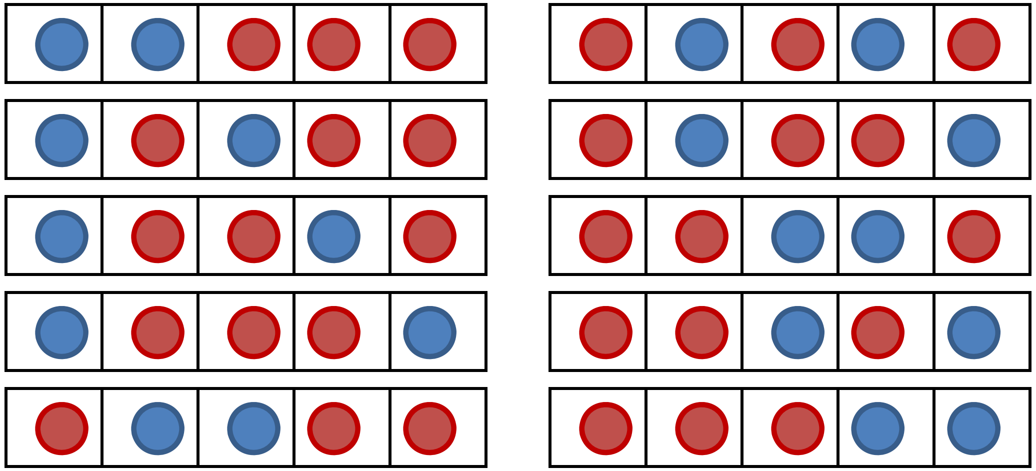
The number of ways of picking k ***unordered*** outcomes from n possibilities. Also known as the binomial coefficient or choice number and read “n choose k,“

$$C_k^n = \binom{n}{k} = \frac{n!}{k!(n-k)!}$$

لدينا ثلاثة كرات حمراء و كرتان زرقاء. كم طريقة يمكن بها ترتيب الخمس كرات.

مثال: (ح،ح،ح،ز،ز)، (ح،ح،ز،ح،ز)

الحل:



Computations

For example: suppose we have the set {1, 2, 3, 4}, we need to calculate the number of combinations of selecting two elements out of the set

$$C_2^4 = \binom{4}{2} = \frac{4!}{2!*2!} = 6$$

namely {1,2}, {1,3}, {1,4}, {2,3}, {2,4}, and {3,4}.

Suppose we have 4 places and filled only 2 of them. The combination to fill the other two cells with the other two numbers equal to 1. Muir (1960) uses the nonstandard notations

$$\overline{C}_k^n = \binom{n-k}{k} \qquad \overline{C}_2^4 = \binom{2}{2} = \frac{2!}{2!*0!} = 1$$

$C_0^n = 1$	$C_1^n = n$	$C_n^n = 1$

STATISTICS

Part 5

Popular Distributions

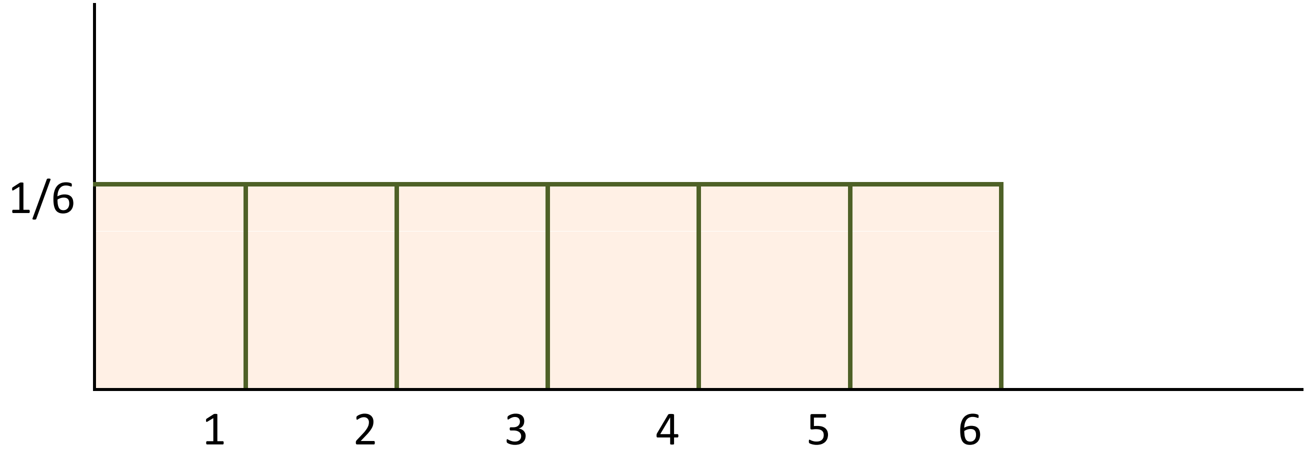
Popular Distributions

Probability Distribution identifies the probability of each value of an unidentified random variable

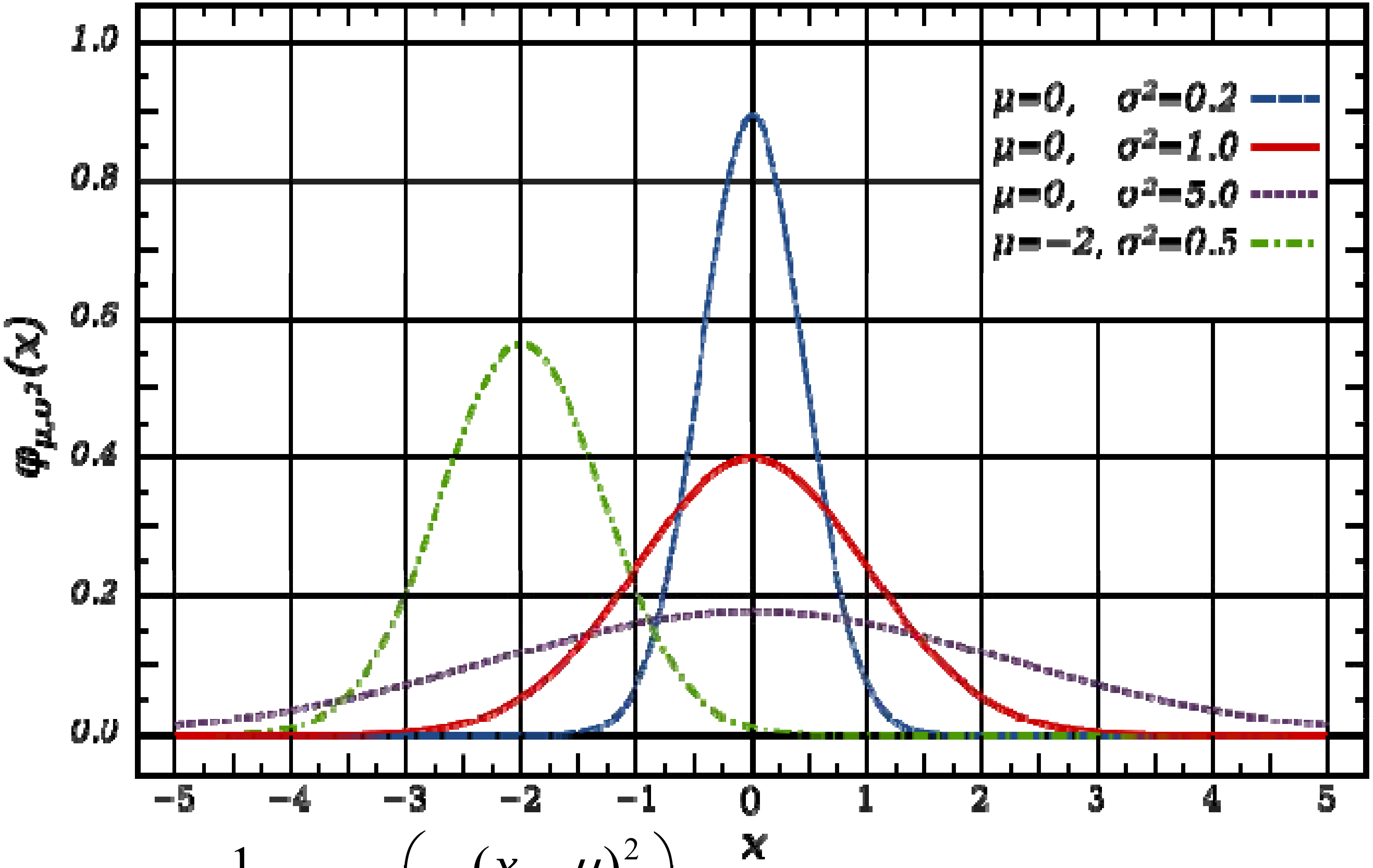
- *Uniform Distribution*
- *Normal (Gaussian) Distribution*
- *Chi-Square Distribution*
- *Exponential Distribution*
- *Poisson Distribution*
- *T Distribution*
- *F Distribution*

The Uniform Distribution

- The probability is equal for all outcomes
- Suppose a fair dice is thrown, the probability of getting any of its 6 faces equal to 1/6
- The area under the line equal to 1

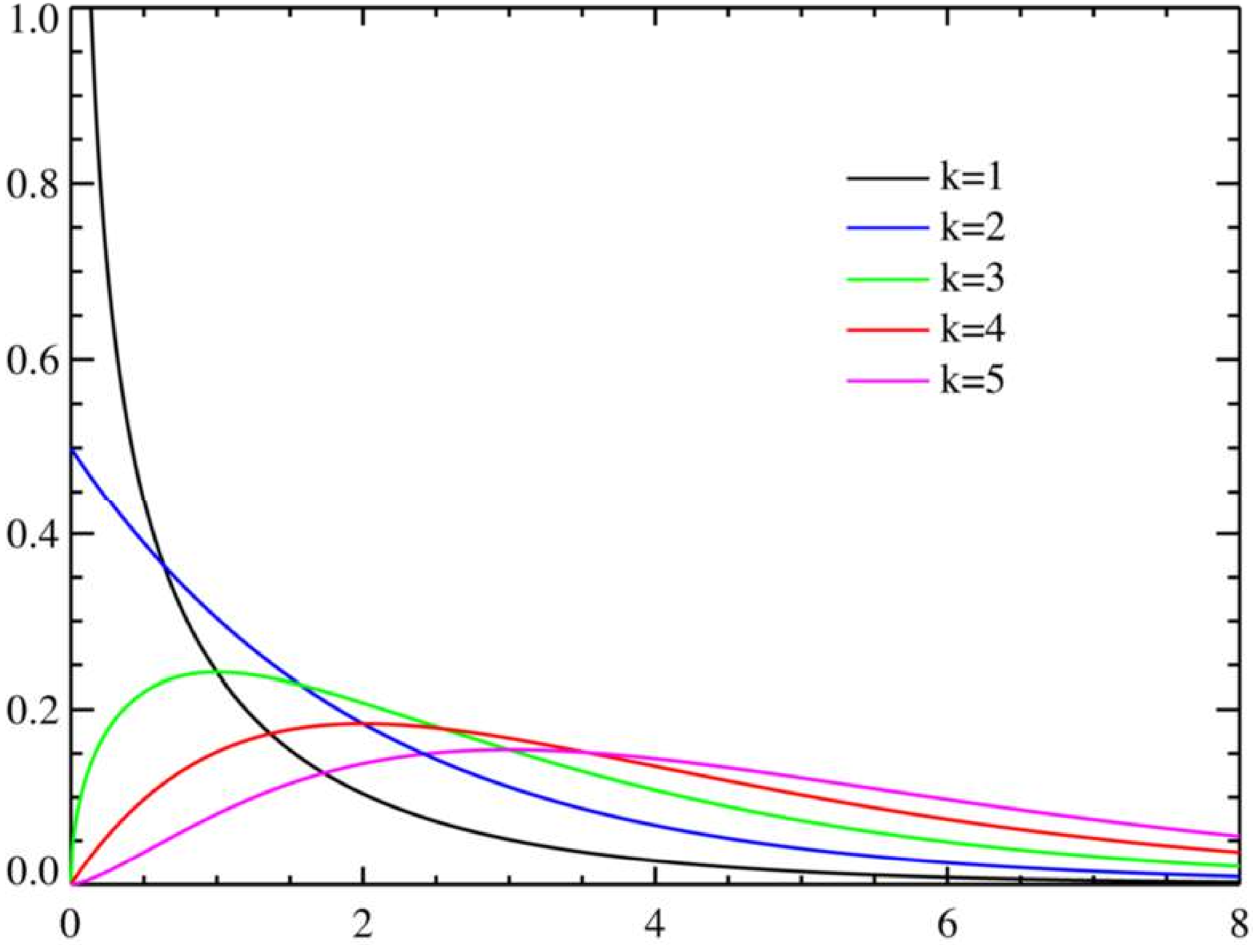


The Normal/Gaussian Distribution



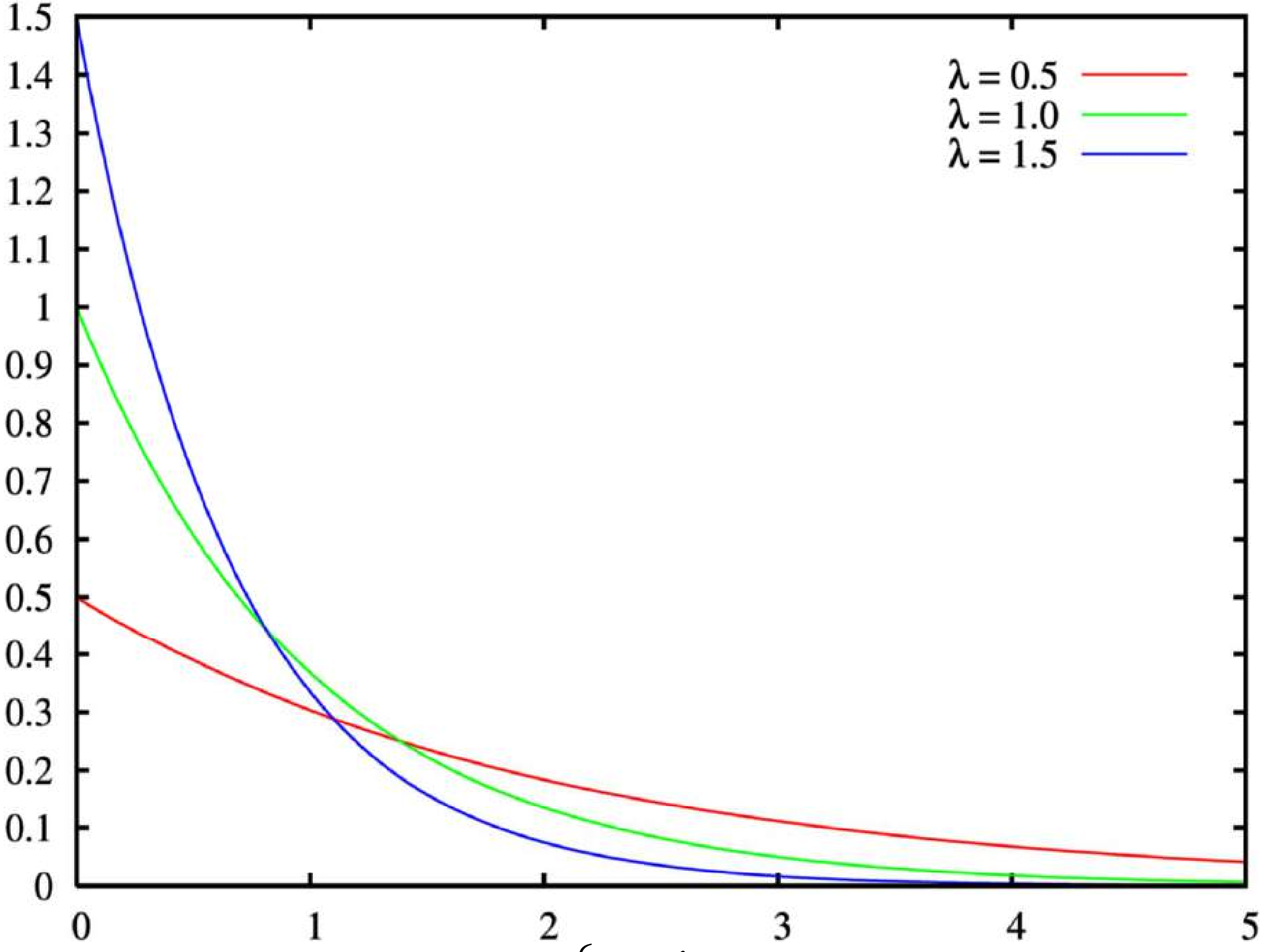
$$P(x)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

The Chi-Square Distribution



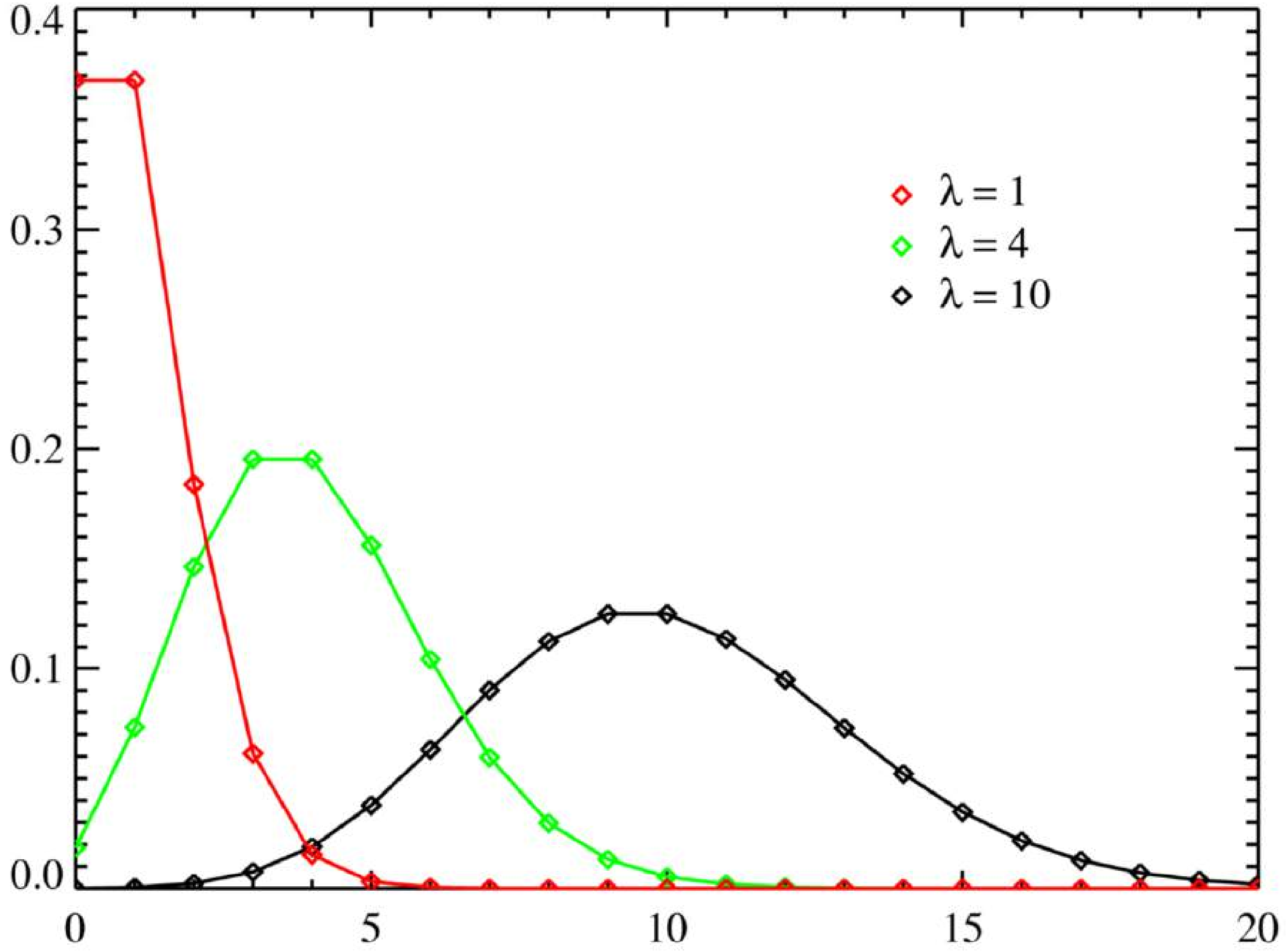
$$f(x;k)=\begin{cases}\dfrac{1}{2^{k/2}\Gamma(k/2)}x^{(k/2)-1}e^{-x/2} & for\ x>0\\ 0 & for\ x\le 0\end{cases}$$

The Exponential Distribution



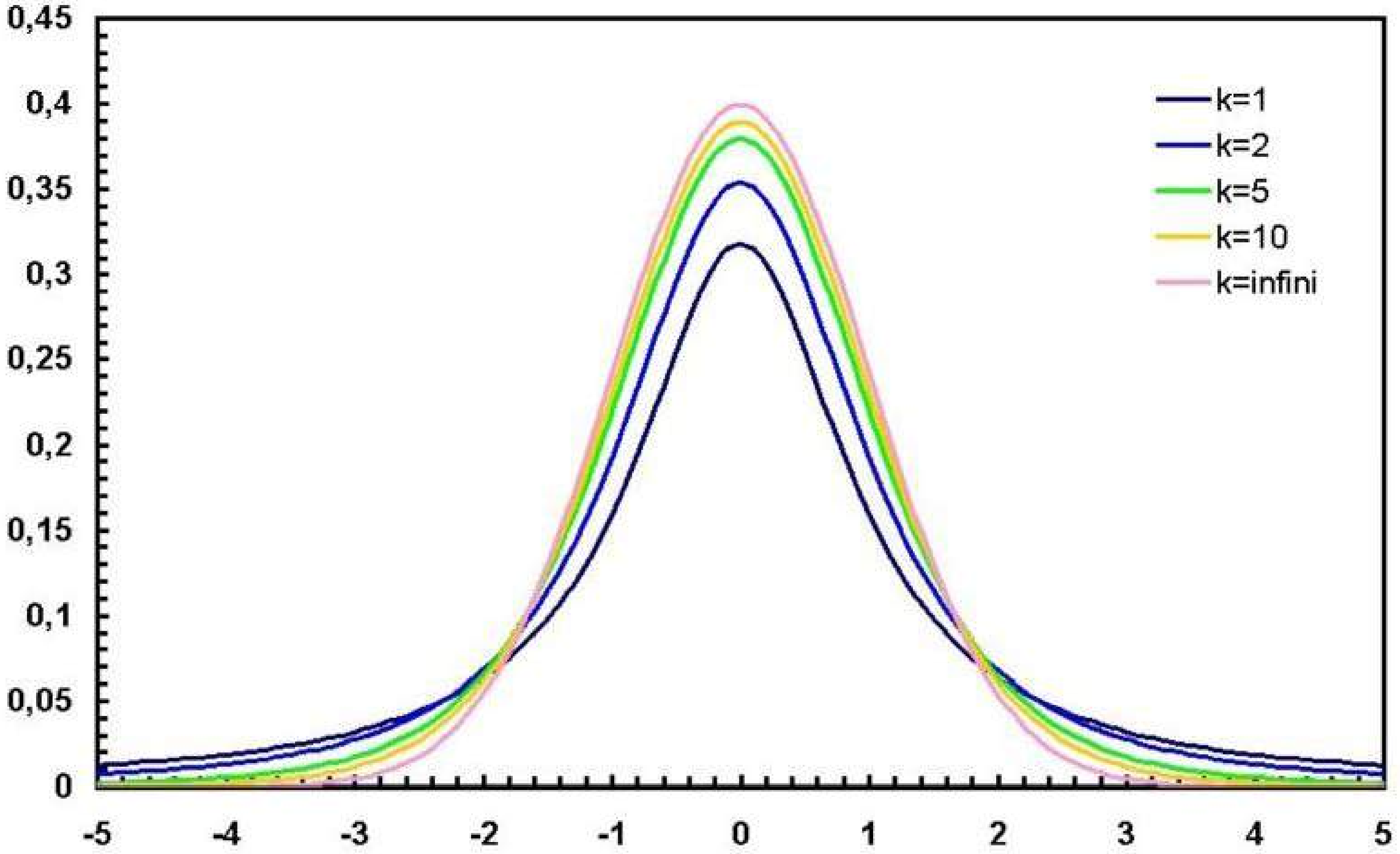
$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & for\ x > 0 \\ 0 & for\ x \leq 0 \end{cases}$$

The Poisson Distribution



$$f(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

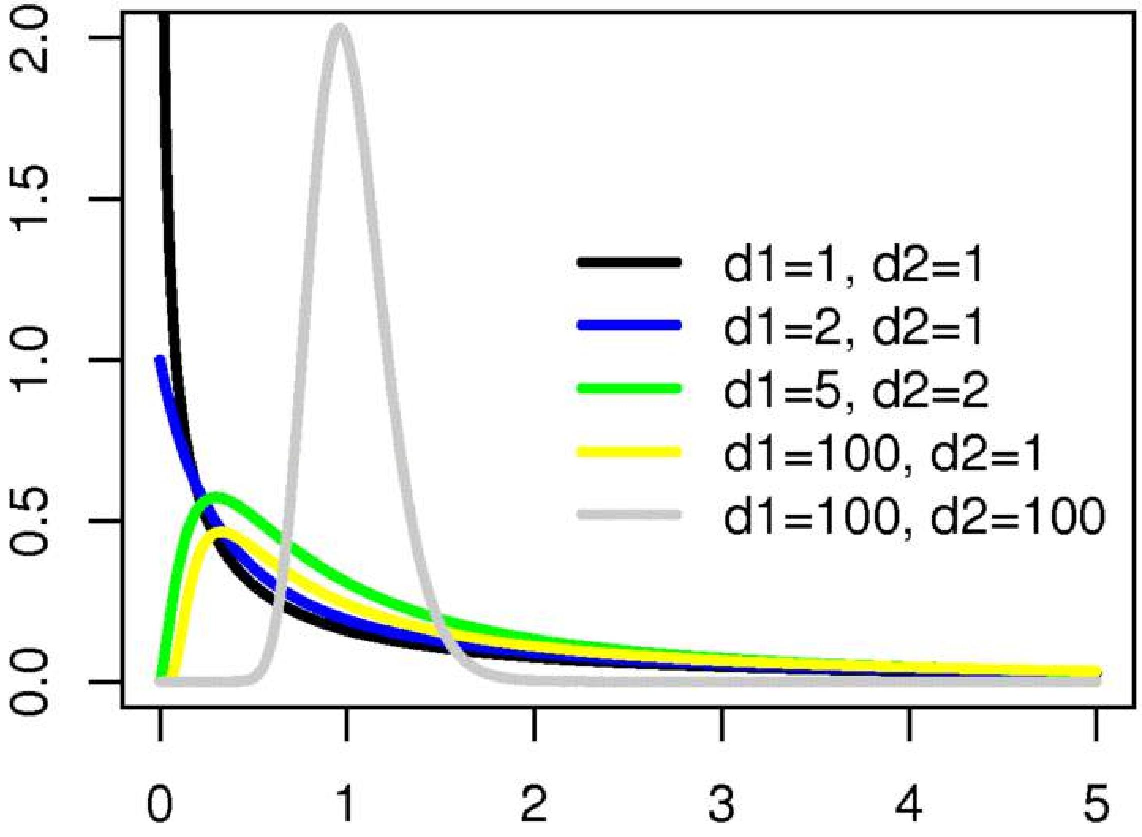
The T Distribution



$$f(t) = \frac{\Gamma(\frac{v+1}{2})}{\sqrt{v\pi}\,\Gamma(\frac{v}{2})}\left(1+\frac{t^2}{v}\right)^{-(v+1)/2}$$

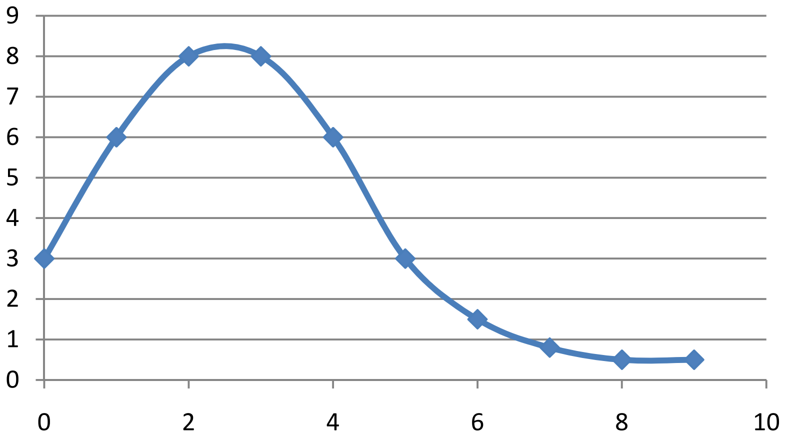
***t*-distribution** arises in the problem of estimating the mean of a normally distributed population when the sample size is small

The F Distribution



$$f(x) = \frac{\sqrt{\frac{(d_1\,x)^{d_1}\;d_2^{d_2}}{(d_1\,x+d_2)^{d_1+d_2}}}}{x\,\mathrm{B}\left(\frac{d_1}{2},\,\frac{d_2}{2}\right)}$$

Fitting Chi-Square



Vector
a

15
14
11
11
6
5
5

$$\max \quad \chi^2 = \sum_{i=1}^{n} \frac{(a_i - E_i)^2}{E_i}$$

$$E_{ij} = (15+14+11+11+6+5+5)/7 = 9.57$$

$$\chi^2 = (1/9.57) * ((15-9.57)^2 + (14-9.57)^2 + (11-9.57)^2 + (11-9.57)^2 + (6-9.57)^2 + (5-9.57)^2 + (5-9.57)^2) = 107.71/9.57 = 11.26$$

Measuring Term-Category Correlation

$$\chi^2(t_k, c_i) = \frac{[P(t_k, c_i)P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i)P(\bar{t}_k, c_i)]^2}{P(t_k)P(\bar{t}_k)P(c_i)P(\bar{c}_i)}$$

$P(t_k, c_i)$ ➔probability document x contains term t and belongs to category c.

$P(\bar{t}_k, c_i)$ ➔probability document x does not contain term t and belongs to category c.

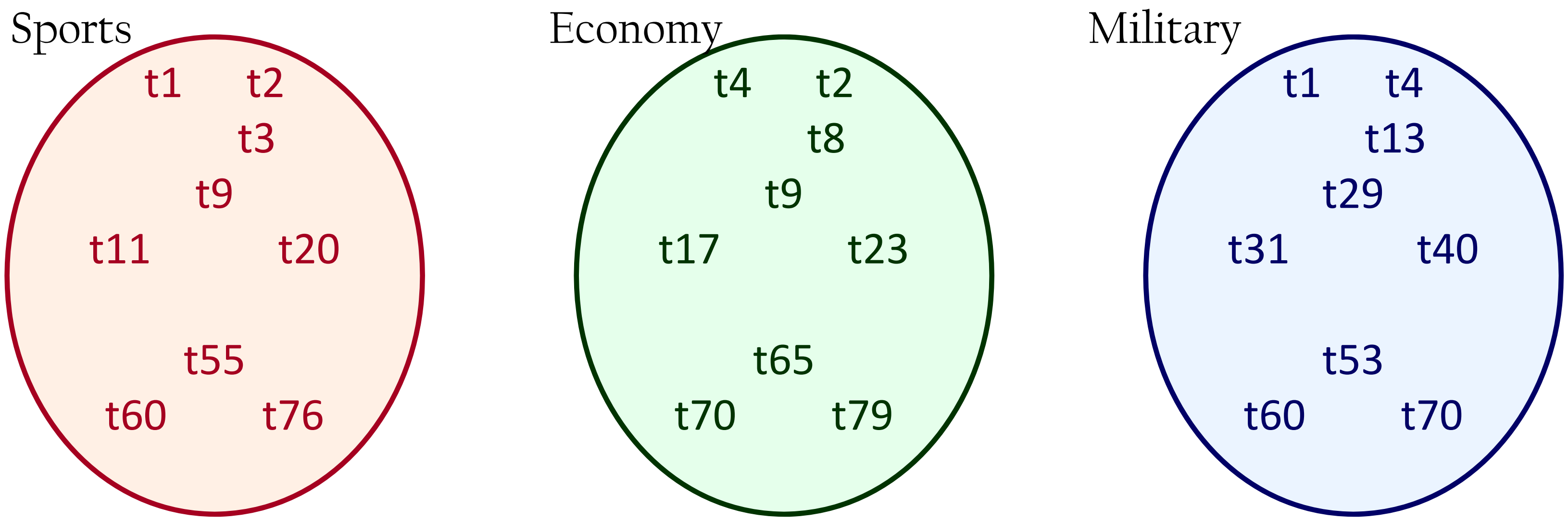
$P(t_k, \bar{c}_i)$ ➔probability document x contains term t and does not belong to category c.

$P(\bar{t}_k, \bar{c}_i)$ ➔probability document x does not contain term t and does not belong to category c.

$P(t)$ ➔probability of term t

$P(c)$ ➔probability of category c

Testing The Membership



$$\chi^2(t_k,c_i) = \frac{[P(t_k,c_i)P(\bar{t}_k,\bar{c}_i) - P(t_k,\bar{c}_i)P(\bar{t}_k,c_i)]^2}{P(t_k)P(\bar{t}_k)P(c_i)P(\bar{c}_i)}$$

$$\chi^2(t_1, Sports\ ) = \frac{\left[\frac{1}{9}*\frac{14}{16} - \frac{1}{16}*\frac{8}{9}\right]^2}{\frac{2}{27}*\frac{25}{27}*\frac{9}{27}*\frac{18}{27}}$$

Using Chi-Square for Categorization

Another Example:

Term	Frequency per Category				Total
	Communication	Phone	Business	Army	
Link	15	6	2	12	35
Wire	10	12	0	8	30
Total	25	18	2	20	**65**

$$\chi^2(link\,,\,phone\,) = \frac{[6/65)*(18/65)-(29/65)*(12/65)]^2}{(35/65)*(30/65)*(18/65)*(47/65)}$$

Using Chi-Square for Multiple sets of Terms

Group 1	Category		Total
	0	1	
Term 1	3	2	5
Term 2	0	4	4
Term 3	2	3	5
Total	5	9	14

Group 2	Category		Total
	0	1	
Term 5	1	3	4
Term 7	4	6	10
Total	5	9	14

$$\chi^2 = \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{(a_{ij}-E_{ij})^2}{E_{ij}}$$

$$E_{ij} = {(T_{ci} * T_{vj})}\big/{T}$$

$$\chi^2(Group\,1) = (3-1.78)^2/1.78 + (2-3.21)^2/3.21 + (0-1.42)^2/1.42$$
$$+ (4-2.57)^2/2.57 + (2-1.78)^2/1.78 + (3-3.21)^2/3.21 = 3.62$$

$$\chi^2(Group\;2) = (1-1.42)^2/1.42 + (3-2.57)^2/2.57 + (4-3.57)^2/3.57$$
$$+ (6-6.43)^2/6.43 =$$

Mingers, J., (1989a). "An Empirical Comparison of selection Measures for Decision-Tree Induction", *Machine Learning*, Vol. 3, No. 3, (pp. 319-342), Kluwer Academic Publishers.

Attribute Selection Criteria: Chi-Square

Example

- T2 is quantized into two intervals 21 (T2<=21) and (T2>21)
- T3 is quantized into two intervals 15 (T3<=15) and (T3>15)

T1	T2	T3	T4	D
1	25	10	A	1
1	30	30	A	0
1	35	25	B	0
1	22	35	B	0
1	19	10	B	1
2	22	30	A	1
2	33	18	B	1
2	14	5	A	1
2	31	15	B	1
3	21	20	A	0
3	15	10	A	0
3	25	20	B	1
3	18	20	B	1
3	20	36	B	1

T2	Decision D		Total
	0	1	
<=21	1	3	4
>21	4	6	10
Total	5	9	14

T1	Decision D		Total
	0	1	
1	3	2	5
2	0	4	4
3	2	3	5
Total	5	9	14

T3	Decision D		Total
	0	1	
<=15	1	4	5
>15	4	5	9
Total	5	9	14

T4	Decision D		Total
	0	1	
A	3	3	6
B	2	6	8
Total	5	9	14

Attribute Selection Criteria: Chi-Square

$$\chi^2(A) = \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{(a_{ij}-E_{ij})^2}{E_{ij}}$$

where A is the attribute to be evaluated against the decision attribute, n is the number of distinct values of A, m is the number of distinct values of the decision attribute, a_{ij} is the correlation frequency of value number i from A and value number j from the decision attribute;

$$E_{ij} = {(T_{ci} * T_{vj})}/{T}$$

where T_{ci} is the total number of examples belonging to class ci, T_{vj} is the number of examples containing the value vj of the given attribute

$$\chi^2(X1) = (3-1.78)^2/1.78 + (2-3.21)^2/3.21 + (0-1.42)^2/1.42$$
$$+ (4-2.57)^2/2.57 + (2-1.78)^2/1.78 + (3-3.21)^2/3.21 = 3.62$$

$$\chi^2(X4) = (3-3.9)^2/3.9 + (3-2.1)^2/2.1 + (6-5.1)^2/5.1$$
$$+ (2-2.9)^2/2.9 = 1.1$$

D1	Decision D5		Total
	0	1	
1	3	2	5
2	0	4	4
3	2	3	5
Total	5	9	14

D2	Decision D5		Total
	0	1	
<=21	1	3	4
>21	4	6	10
Total	5	9	14

D3	Decision D5		Total
	0	1	
<=15	1	4	5
>15	4	5	9
Total	5	9	14

D4	Decision D5		Total
	0	1	
A	3	3	6
B	2	6	8
Total	5	9	14

Mingers, J., (1989a). "An Empirical Comparison of selection Measures for Decision-Tree Induction", *Machine Learning*, Vol. 3, No. 3, (pp. 319-342), Kluwer Academic Publishers.

STATISTICS

Part 6

Regression

Linear Regression

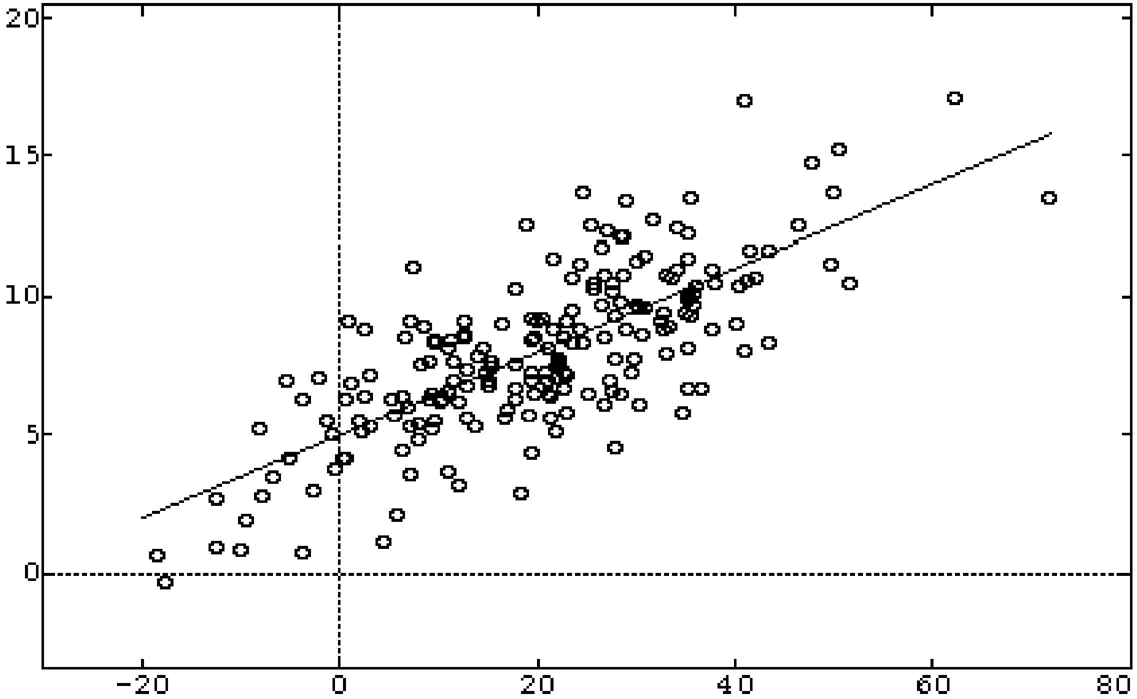
- The linear model states that the dependent variable is *directly proportional* to the value of the independent variable
- Thus if a theory implies that Y increases in direct proportion to an increase in X, it implies a specific mathematical model of behavior

$$y = ax + b$$

In case of two dimensions

$$a = slope = \frac{(y_2 - y_1)}{(x_2 - x_1)}$$

$$b = y_2 - slope * x_2$$



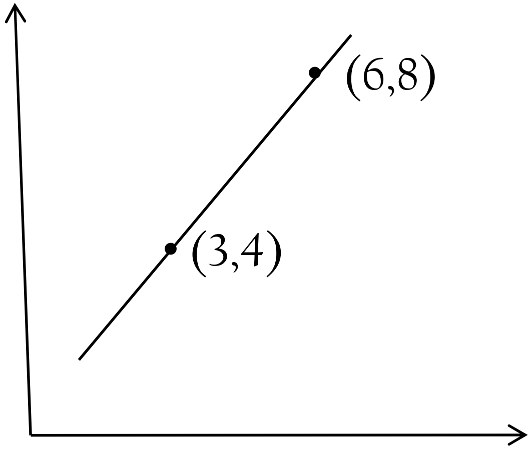
Linear Regression

$$y = ax + b$$

$$8 = 6a + b \quad \& \quad 4 = 3a + b$$

$$\frac{8-b}{6} = a \qquad \& \qquad 4 = 3 * \frac{8-b}{6} + b$$

$$b = 0 \qquad \& \qquad a = \frac{4}{3} = 1.333$$



$$Slope = \frac{8-4}{6-3} = 1.333$$

$$b = 4 - \frac{4}{3} * 3 = 0$$

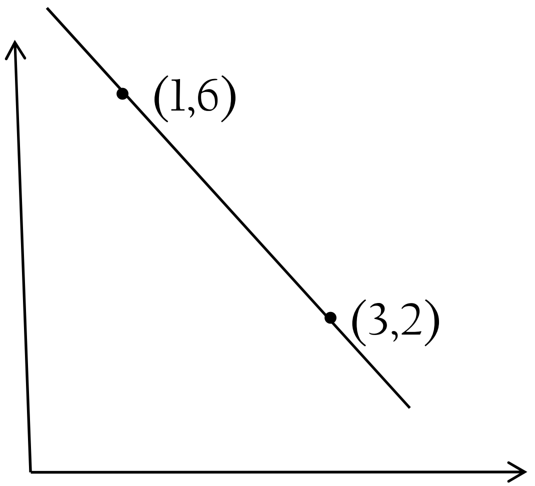
Linear Regression

$$y = ax + b$$

$$6 = a + b \quad \& \quad 2 = 3a + b$$

$$6 - b = a \qquad \& \qquad 2 = 3*(6 - b) + b$$

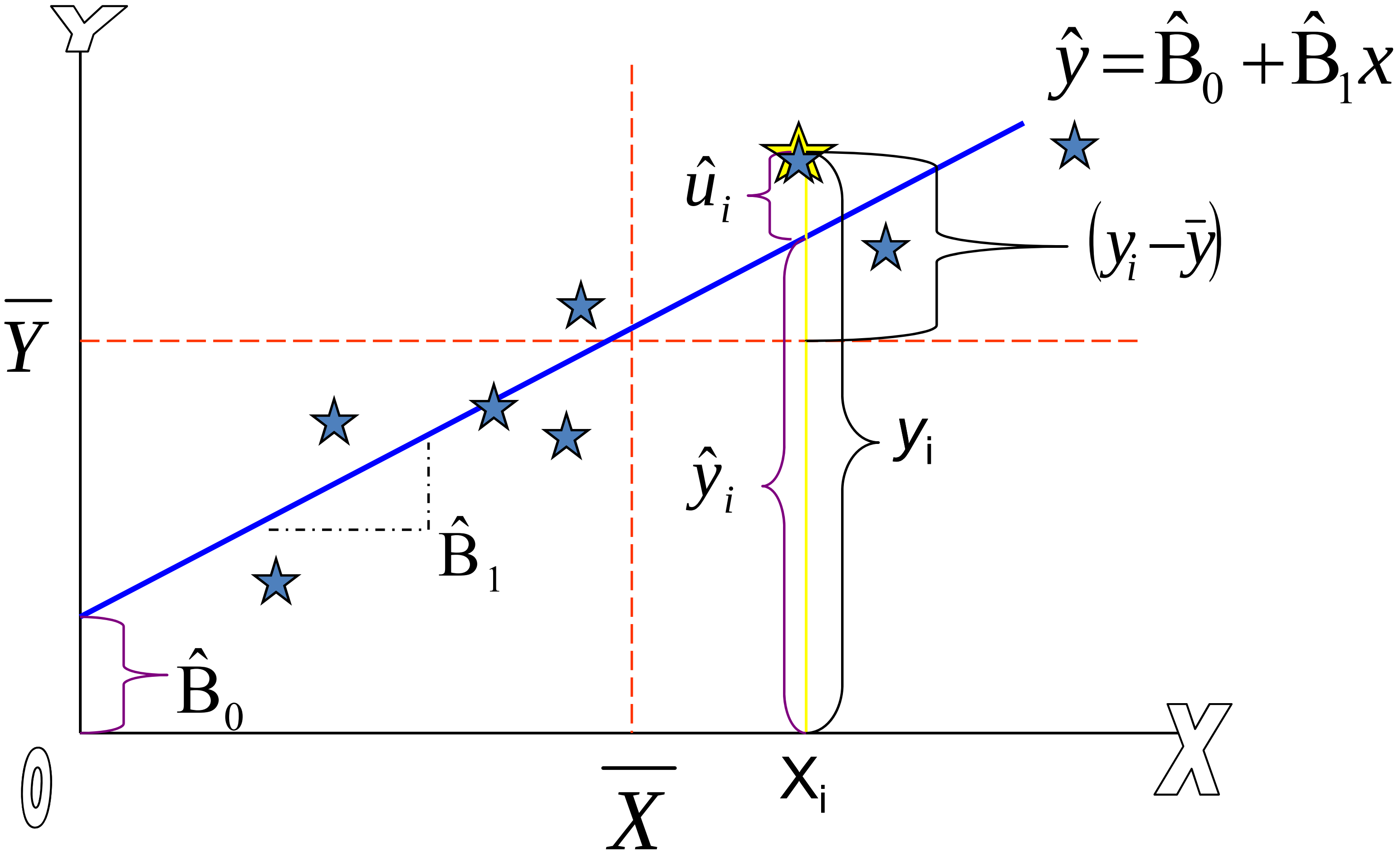
$$b = 8 \qquad \& \qquad a = 6 - 8 = -2$$



$$Slope = \frac{6-2}{1-3} = \frac{4}{-2} = -2$$

$$b = 2 + 2*3 = 8$$

Linear Regression

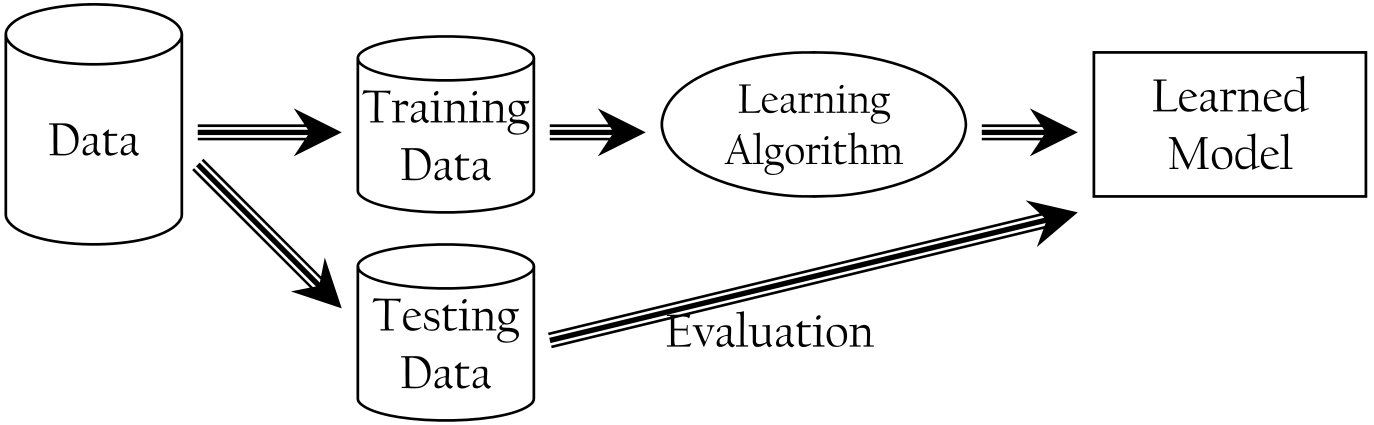


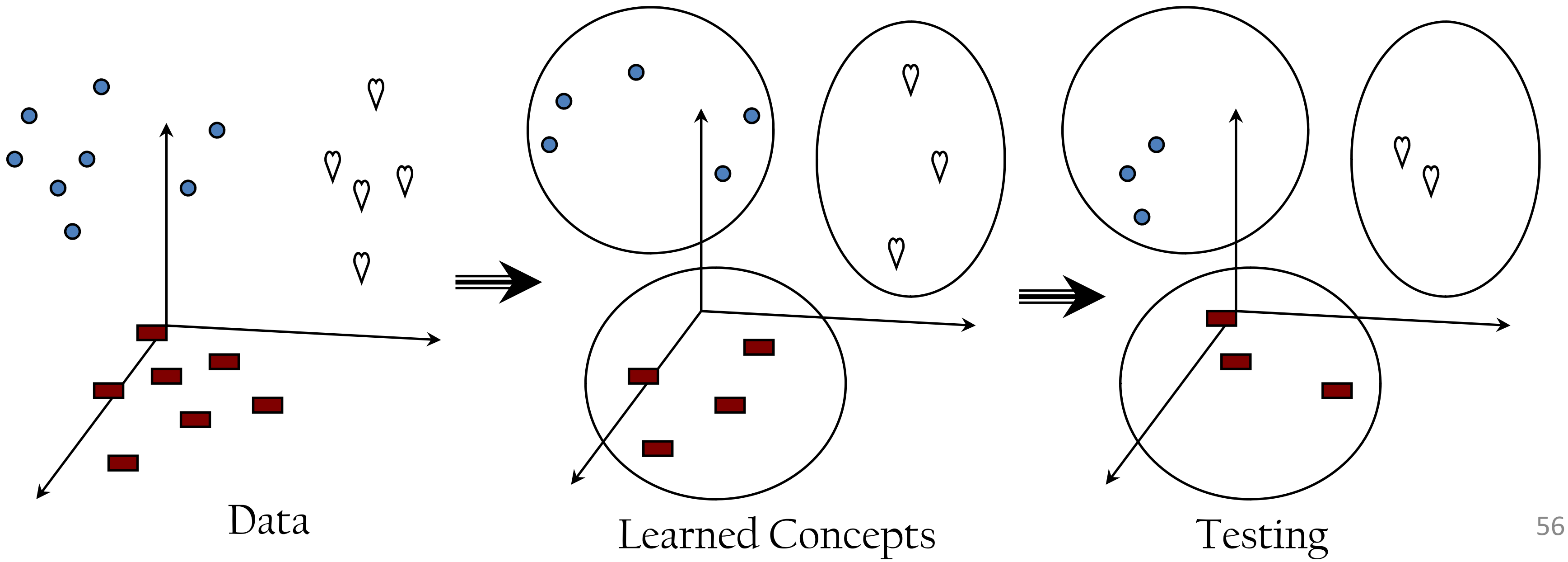
Statistics and Testing

Part 7

Testing Samples & Calculating Accuracy

Training & Testing



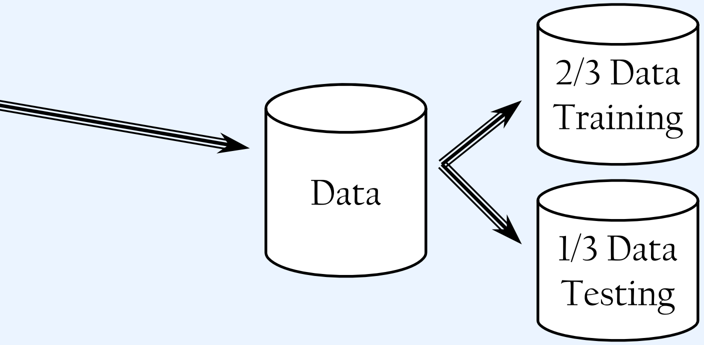


Testing Approaches

- ***Two-Cross-Fold***

  Train on 2/3rd

  Test on 1/3rd

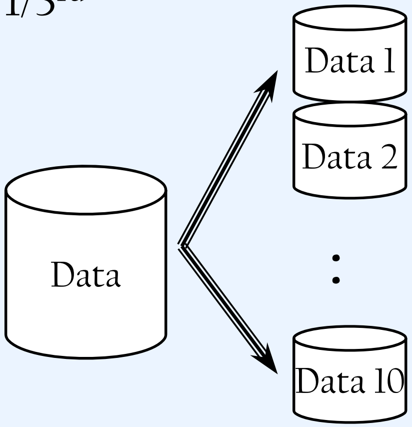


- ***Ten-Cross-Fold***

  Train on 9/10th

  Test on 1/10th

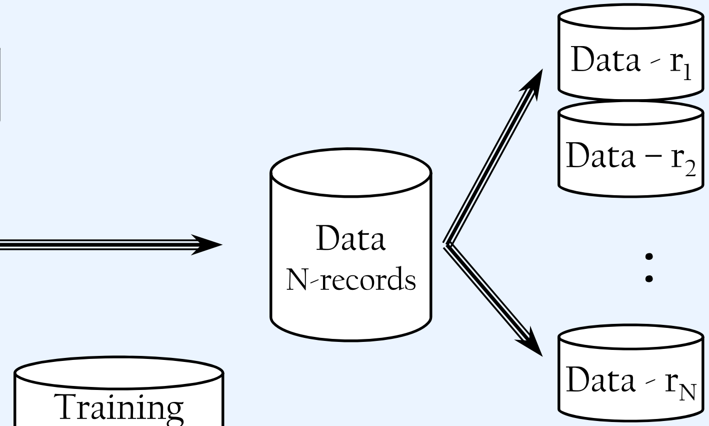
  Repeat 10 times



- ***Hold-One-Out***

  Train on all data but one

  Test on the selected one

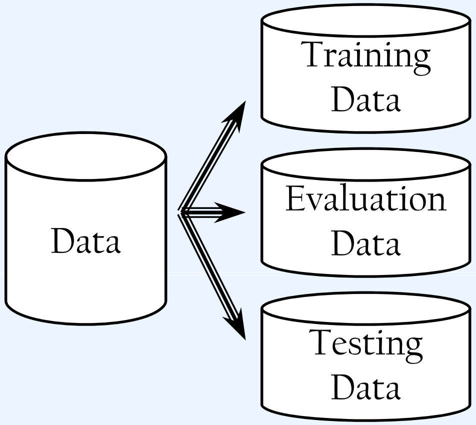


- ***Learning Evaluation vs. Testing***

  Train on Training Data

  Evaluate on Evaluation Data

  Test on Testing Data



Accuracy & Error

Example: Suppose you have a classification model C, and 100 testing records from two classes (P & N). Suppose the following are the classification results:

- Accuracy vs. Error Rate
  - *Accuracy* = (40+45)/100 = 85%
  - *Error Rate* = (10+5)/100 = 15%

		Actual	
		P	N
Obtained	P	TP	FP
	N	FN	TN

- True vs. False Classification
  - *True Positive:* = 88.88%
  - *True Negative:* = 81.82%
  - *False Positive:* = 11.12%
  - *False Negative:* = 18.18%

		Actual	
		P	N
Obtained	P	40	10
	N	5	45

- Flexible Matching
  - *Using Nearest Neighbors (e.g., majority of nearest 3 neighbors)*
  - Using Fuzzy rules (assigning probability for each decision and taking it into consideration when calculating the accuracy)
  - Assigning small weights for the false positive and false negative results (not zero)
- Testing for Multiple Classes ????

Precision, Recall, and F-Measure

Accuracy: is the percentage of correct results

Error: is the percentage of wrong results

Accuracy only reacts to real errors, and doesn't show how many correct results have been found as such

Precision:

Precision shows the percentage of correct results within an answer:

$$Precision = (tp) / (tp + fp)$$

Recall:

Recall is the percentage of the correct system results over all correct results:

$$Recall = (tp) / (tp + fn)$$

Makhoul, John; Francis Kubala; Richard Schwartz; Ralph Weischedel: Performance measures for information extraction. In: Proceedings of DARPA Broadcast News Workshop, Herndon, VA, February 1999

Precision, Recall, and F-Measure

Precision and Recall can be defined differently for different tasks

For example: In Information Retrieval,

- Recall = |{relevant documents} ∩ {documents retrieved}| /

  / |{relevant documents}|

- Precision = |{relevant documents} ∩ {documents retrieved}| /

  / |{documents retrieved}|

Christopher D. Manning and Hinrich Sch¨utze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.

Precision, Recall, and F-Measure

F-Measure (harmonic mean):

F_β "measures the effectiveness of β times as much importance to recall as precision". The general form of F-Measure:

$$F_\beta = (1+ \beta^2) * (\text{precision} * \text{recall}) / (\beta^2 * \text{precision} + \text{recall})$$

when $\beta=1$,

$$F_1 = 2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$$

STATISTICS

Part 8

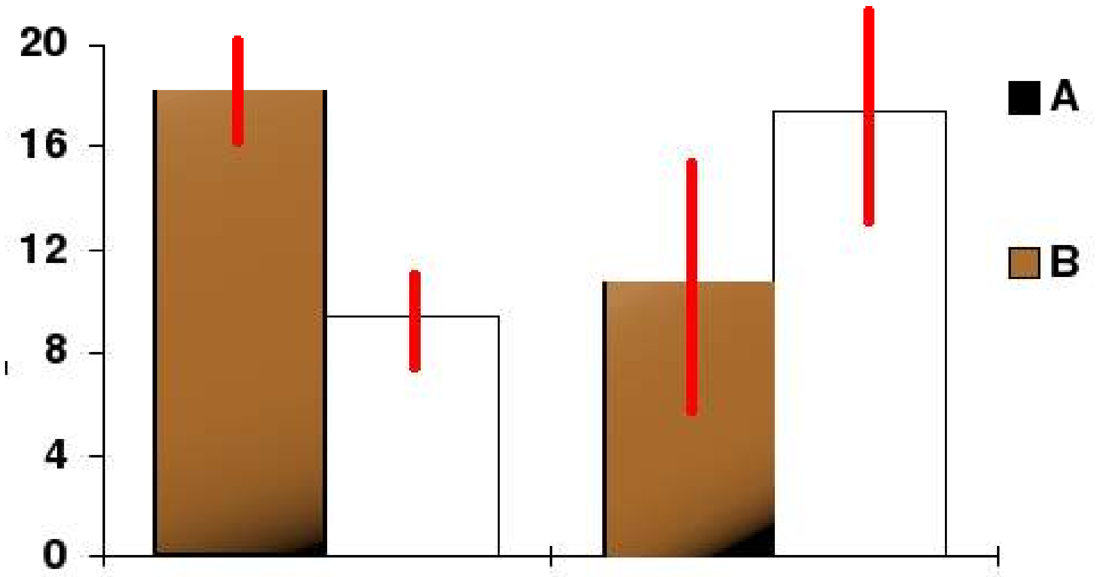
Test of Significance

Test of Significance (1/5)

- The probability that a result is not due to chance; or Is the observed value differs enough from a hypothesized value?
- The hypothesized value is called the null hypothesis
- If this probability is sufficiently low, then the difference between the parameter and the statistic is said to be "statistically significant"
- Just how low is sufficiently low? The choice of 0.05 and 0.01 are most commonly used

- Suppose your algorithm produced error rate of 1.5 and another algorithm produced an error of 2.1 on the same data set; are the two algorithms similar?

Test of Significance (2/5)



- The top ends of the bars indicate observation means
- The red line segments represent the confidence intervals surrounding them
- The difference between the two populations on the left is significant
- However, it is a common misconception to suppose that two parameters whose 95% confidence intervals fail to overlap are significantly different at the 5% level

Test of Significance (3/5)

- The system you are comparing against reported results of 250; the value reported is considered as a random variable X; the distribution of X is assumed as normal distribution with unknown mean and standard deviation σ=2.5; You ran your system 25 times; it reported values (x1, x2, … , x25); the average of these values is 250.2.

$$\hat{\mu} = \overline{X} = \frac{1}{n}\sum_{i=1}^{25} x_i = 250.2$$

Sample Mean

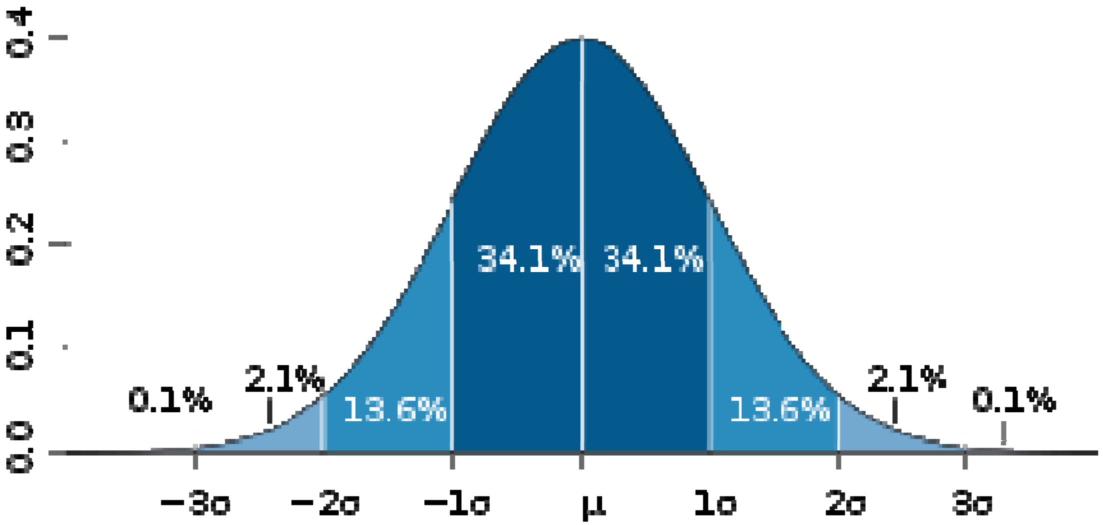
$$\text{Standard Error} = \sigma / \sqrt{n} = 2.5 / \sqrt{25} = 0.5$$

n is the sample size

$$\mathrm{Z} = \frac{\overline{X} - \mu}{\sigma / \sqrt{n}} = \frac{\overline{X} - \mu}{0.5}$$

μ is not known

Test of Significance (4/5)



$$P(-z \le Z \le z) = 1 - \alpha = 0.95$$

$$\Phi(z) = P(Z \le z) = 1 - \frac{\alpha}{2} = 0.975$$

From Tables

$$z = \Phi^{-1}(\Phi(z)) = \Phi^{-1}(0.975) = 1.96$$

$$0.95 = 1 - \alpha = P(-z \le Z \le z) = P(-1.96 \le \frac{\overline{X} - \mu}{\sigma / \sqrt{n}} \le 1.96)$$

Test of Significance (5/5)

$$P(-z \le Z \le z) = P(\overline{X} - 1.96\frac{\sigma}{\sqrt{n}} \le \mu \le \overline{X} + 1.96\frac{\sigma}{\sqrt{n}})$$

$$P(-z \le Z \le z) = P(\overline{X} - 1.96 * 0.5 \le \mu \le \overline{X} + 1.96 * 0.5)$$

$$P(-z \le Z \le z) = P(\overline{X} - 0.98 \le \mu \le \overline{X} + 0.98)$$

$$Our\ Interval \quad = (250.2 - 0.98; 250.2 + 0.98)$$

$$Our\ Interval \quad = (249.22; 251.0)$$

- Any value within this interval is not significant

The Information Theory

Part 9

Introduction
Entropy

The Information Theory

The information conveyed by a message can be measured in bits by its probability

The Information Theory: Given Data

Attributes:
D1, D2, D3, D4

Domain(D1)={1,2,3}

Domain(D2)={1,2}

Domain(D3)={1,2}

Domain(D4)={A,B}

D1	D2	D3	D4	D5
1	2	1	A	1
1	2	2	A	0
1	2	2	B	0
1	2	2	B	0
1	1	1	B	1
2	2	2	A	1
2	2	2	B	1
2	1	1	A	1
2	2	1	B	1
3	1	2	A	0
3	1	1	A	0
3	2	2	B	1
3	1	2	B	1
3	1	2	B	1

Decision Attributes: D5

Domain(D5)={0,1}

Two Decisions: 0, 1

The Information Theory: Given Data

	D1	**1**		**2**		**3**	
D4	D3\D2	1	2	1	2	1	2
A	1		1	1		0	
	2		0		1	0	
B	1	1	1		1	1	
	2		0		1	1	1

D1	D2	D3	D4	**D5**
1	2	1	A	**1**
1	2	2	A	**0**
1	2	1	B	**0**
1	2	2	B	**0**
1	1	1	B	**1**
2	2	2	A	**1**
2	2	2	B	**1**
2	1	1	A	**1**
2	2	1	B	**1**
3	1	2	A	**0**
3	1	1	A	**0**
3	2	2	B	**1**
3	1	1	B	**1**
3	1	2	B	**1**

The Information Theory: Entropy

THE INFORMATION THEORY: information conveyed by a message depends on its probability and can be measured in bits as minus the logarithm (base 2) of that probability

suppose $D_1, \ldots, D_m$ are m attributes and $C_1, \ldots, C_n$ are n decision classes in a given data. Suppose S is any set of cases, and T is the initial set of training cases $S \subset T$. The **frequency of class C_i in the set S** is:

$$freq(C_i, S) = Number\ of\ \ examples\ in\ S\ belonging\ to\ C_i$$

If |S| is the total number of examples in S, ***the probability that an example selected at random from S belongs to class C_i*** is

$$freq(C_i, S) / |S|$$

The information conveyed by the message that "**a selected example belongs to a given decision class, C_i**", is determined by

$$-\log_2(freq(C_i, S) / |S|) \quad bits$$

The Information Theory: Entropy

The information conveyed by the message "***a selected example belongs to a given decision class, C_i***"

$$-\log_2(freq(C_i, S)/|S|) \quad bits$$

The Entropy: The expected information from a message stating class membership is given by

$$Info(S) = -\sum_{i=1}^{k}(freq(C_i, S)/|S|)*\log_2(freq(C_i, S)/|S|) \quad bits$$

info(S) is known as the ***entropy*** of the set S. When S is the initial set of training examples, *info(S) determines the average amount of information needed to identify the class of an example in S*.

The Information Theory: The Gain Ratio

Example

$$freq(0,S)=5 \qquad freq(1,S)=9$$

$$freq(0,S)/|S|=5/14 \qquad freq(1,S)/|S|=9/14$$

The Entropy: the average amount of information needed to identify the class of an example in S

$$Info(S)=-9/14*\log_2(9/14)-5/14*\log_2(5/14)=0.94bits$$

Using D_1 to Split the data provide 3 subsets of data

$$Info_{D_1}(S_1)=-3/5*\log_2(3/5)-2/5*\log_2(2/5)=0.94$$

$$Info_{D_1}(S_2)=-4/4*\log_2(4/4)=0.94$$

$$Info_{D_1}(S_3)=-2/5*\log_2(2/5)-3/5*\log_2(3/5)=0.94$$

$$Info_{D_1}(S)=({}^{5}\!/_{14})*Info_{D_1}(S_1)+({}^{4}\!/_{14})*Info_{D_1}(S_2)+({}^{5}\!/_{14})*Info_{D_1}(S_3)=0.694$$

S

D1	D2	D3	D4	D5
1	2	1	A	1
1	2	2	A	0
1	2	2	B	0
1	2	2	B	0
1	1	1	B	1
2	2	2	A	1
2	2	2	B	1
2	1	1	A	1
2	2	1	B	1
3	1	2	A	0
3	1	1	A	0
3	2	2	B	1
3	1	2	B	1
3	1	2	B	1

The Information Theory: The Gain Ratio

Suppose attribute $\underline{D_i}$ is selected to be the root and it has $\underline{k}$ possible values. The expected information of selecting D to partition the training set S, info$_{D_i}$(S), can be calculated as follows:

$$Info_{D_i}(S) = \sum_{i=1}^{k} \left({|S_i|}/{|S|} \right) * Info(S_i)$$

S_i is the subset number i of the data; k is the number of values of D_i

The information gained by partitioning the training examples S into subset using the attribute D_1 is given by

$$Gain(X_i) = Info(S) - Info_{D_i}(S)$$

The Information Theory: The Gain Ratio

The attribute to be selected is the attribute with maximum gain value. Quinlan found out that a key attribute will have the maximum gain. This is not good!

$$Split\_Info(S) = -\sum_{i=1}^{k} (|S_i|/|S|) * \log_2(|S_i|/|S|)$$

The gain ratio is given by:

$$Gain\_Ratio(D_i) = Gain(D_i) / Split\_Info(D_i)$$

Quinlan, J.R., (1993). “C4.5: Programs for Machine Learning”, Morgan Kaufmann, Los Altos, California.

The Information Theory: The Gain Ratio

Example Cont.

$$Info_{D_1}(S) = (5/14) * Info_{D_1}(S_1) + (4/14) * Info_{D_1}(S_2) + (5/14) * Info_{D_1}(S_3) = 0.694$$

$$Gain(D_1) = 0.94 - 0.694 = 0.246$$

$$Split\_Info(S) = -5/14 * \log_2(5/14) - 4/14 * \log_2(4/14) - 5/14 \log_2(5/14) = 1.577 \quad bits$$

$$Gain\_Ratio(D_1) = 0.246 / 1.577 = 0.156$$

S

D1	D2	D3	D4	D5
1	2	1	A	1
1	2	2	A	0
1	2	2	B	0
1	2	2	B	0
1	1	1	B	1
2	2	2	A	1
2	2	2	B	1
2	1	1	A	1
2	2	1	B	1
3	1	2	A	0
3	1	1	A	0
3	2	2	B	1
3	1	2	B	1
3	1	2	B	1

Information Gain: Term vs. Category

It measures the classification power of a term

$$IG(t_k, c_i) = \sum_{c \in \{c_i, \bar{c}_i\}} \sum_{t \in \{t_k, \bar{t}_k\}} P(t, c) \log_2 \frac{P(t, c)}{P(t) P(c)}$$

$P(t_k, c_i)$ ➔probability document x contains term t and belongs to category c.

$P(\bar{t}_k, c_i)$ ➔probability document x does not contain term t and belongs to category c.

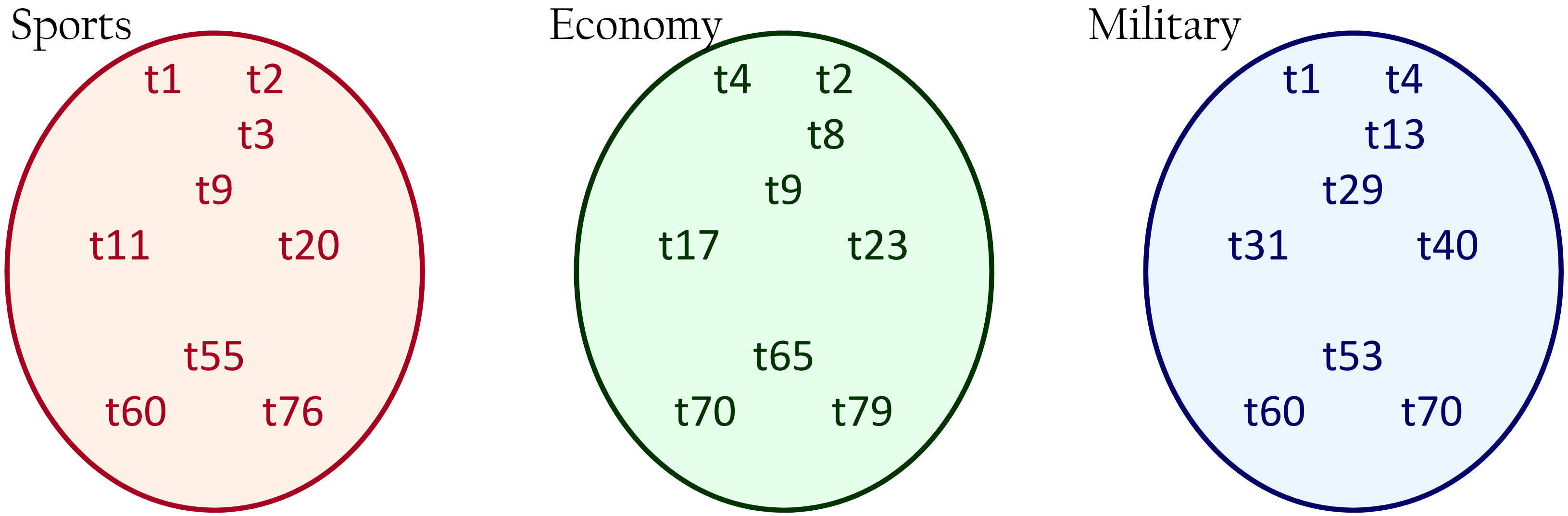
$P(t_k, \bar{c}_i)$ ➔probability document x contains term t and does not belong to category c.

$P(\bar{t}_k, \bar{c}_i)$ ➔probability document x does not contain term t and does not belong to category c.

$P(t)$ ➔probability of term t.

$P(c)$ ➔probability of category c.

Testing The Membership



$$IG(t_k, c_i) = \sum_{c \in \{c_i, \bar{c}_i\}} \sum_{t \in \{t_k, \bar{t}_k\}} P(t,c) \log_2 \frac{P(t,c)}{P(t)P(c)}$$

$$IG(t_1, sport) = \frac{1}{9} * \log_2 \frac{1/9}{(2/27)*(9/27)} + \frac{8}{9} * \log_2 \frac{8/9}{(25/27)*(9/27)}$$

$$+ \frac{1}{18} * \log_2 \frac{1/18}{(2/27)*(18/27)} + \frac{17}{27} * \log_2 \frac{17/27}{(25/27)*(18/27)}$$

The Gain Ratio

$$GR(t_k, c_i) = \frac{\sum_{c \in \{c_i, \overline{c}_i\}} \sum_{t \in \{t_k, \overline{t}_k\}} P(t,c) \log_2 \frac{P(t,c)}{P(t)P(c)}}{-\sum_{c \in \{c_i, \overline{c}_i\}} P(c) \log_2 P(c)}$$

$P(t_k, c_i)$ ➔probability document x contains term t and belongs to category c.

$P(\overline{t}_k, c_i)$ ➔probability document x does not contain term t and belongs to category c.

$P(t_k, \overline{c}_i)$ ➔probability document x contains term t and does not belong to category c.

$P(\overline{t}_k, \overline{c}_i)$ ➔probability document x does not contain term t and does not belong to category c.

$P(t)$ ➔probability of term t.

$P(c)$ ➔probability of category c.

Basics for Language Engineers

Part 10

Evaluating Documents

Term Frequency & Inverse Document Frequency

Usually a combination of the term frequency and the inverse document frequency

$$TFIDF = w_{ik} = tf_{ik} \times idf_{ik}$$

$$tf_{ik} = 1 + \log_2(tr_{ik}) \qquad \textit{and zero when} \log = 0$$

$$idf_{ik} = \log_2\left(\frac{N}{n_{ik}}\right) \qquad \textit{and zero when} \log = 0$$

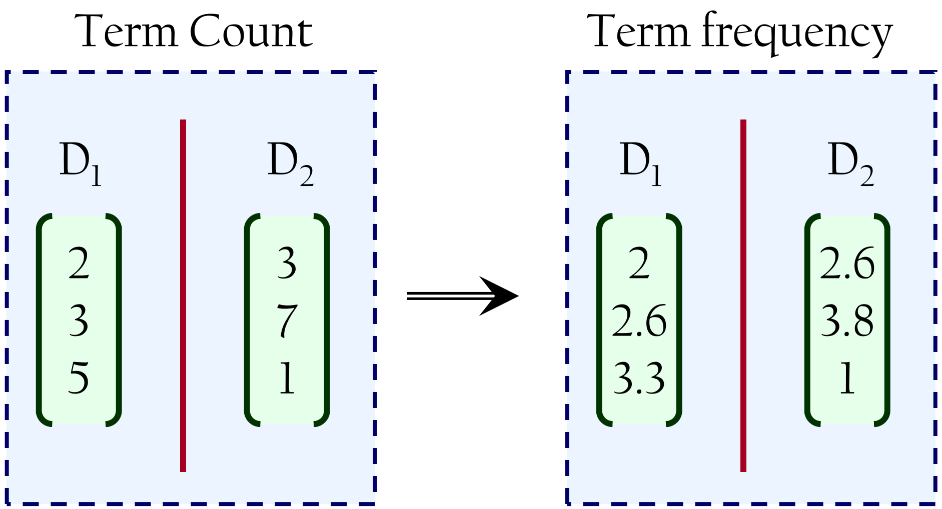
tf_{ik} is the term frequency of term *i* in document *k*, tr_{ik} is the count of term *i* in document *k*, idf_{ik} is the inverse document frequency of term *i* in document *k*, N is the total number of documents in the collection, n_{ik} is the number of occurrence of term *i* in document *k*, w_{ik} is the weight of term *i* in document *k*. Logarithm has been used to reduces the difference between the weight of high and low frequency terms. Logarithm of base 2 is used when vectors are full of binary TFIDF weights 0 and 1. Logarithm of base 10 is used when vectors are full of TFIDF weights except binary ones. TFIDF weights values are not normalized.

The Magical Recipe

$$tf_{ik} = 1 + \log_2(tr_{ik}) \qquad and \;\; zero \;\; when \;\; \log = 0$$

$$idf_{ik} = \log_2(\frac{N}{n_{ik}}) \qquad and \;\; zero \;\; when \;\; \log = 0$$

$$\log_2 x = \log_{10} x / \log_{10} 2$$



STATISTICAL ASSOCIATIONS

Part II

Association Rules

Learning Term-Association

T1	T2	T3	T4	T5	T6	T7	
1	1	1	1	1	1	1	D1
2	1	2	1	1	1	2	D2
1	2	3	1	1	1	3	D3
2	2	1	2	1	2	4	D4
1	1	2	2	1	1	5	D5
2	1	3	2	1	2	6	D6
1	2	1	3	2	2	7	D7
2	2	2	3	2	2	8	D8
1	1	3	3	2	2	9	D9
2	1	1	1	2	1	1	D10
1	2	2	1	2	2	2	D11
2	2	3	1	2	1	3	D12
1	1	1	2	3	1	4	D13
2	1	2	2	3	1	5	D14
1	2	3	2	3	1	6	D15
2	2	1	3	3	1	7	D16
1	1	2	3	3	2	8	D17
2	1	3	3	3	1	9	D18

D1	D2	D3	D4	D5	D6	D7	
1	1	1	1	1	1	1	T1
2	1	2	1	1	1	2	T2
1	2	3	1	1	1	3	T3
2	2	1	2	1	2	4	T4
1	1	2	2	1	1	5	T5
2	1	3	2	1	2	6	T6
1	2	1	3	2	2	7	T7
2	2	2	3	2	2	8	T8
1	1	3	3	2	2	9	T9
2	1	1	1	2	1	1	T10
1	2	2	1	2	2	2	T11
2	2	3	1	2	1	3	T12
1	1	1	2	3	1	4	T13
2	1	2	2	3	1	5	T14
1	2	3	2	3	1	6	T15
2	2	1	3	3	1	7	T16
1	1	2	3	3	2	8	T17
2	1	3	3	3	1	9	T18

Learning Term-Association

AR Syntax:

(condition 1) (condition 2) … (condition n)   strength of association

Suppose we quantized the term weights

Drive two association rules with two Conditions and frequency greater than 0.25.

(T1 = 1) (T6 = 1)	5/18
(T1 = 2) (T2 = 1)	5/18

Question:
Drive association rules with two conditions and frequency greater than 0.38.

T1	T2	T3	T4	T5	T6	T7	T8
1	1	1	1	1	1	1	1
2	1	2	1	1	1	2	2
1	2	3	1	1	1	3	3
2	2	1	2	1	2	4	4
1	1	2	2	1	1	5	5
2	1	3	2	1	2	6	6
1	2	1	3	2	2	7	1
2	2	2	3	2	2	8	2
1	1	3	3	2	2	9	3
2	1	1	1	2	1	1	4
1	2	2	1	2	2	2	5
2	2	3	1	2	1	3	6
1	1	1	2	3	1	4	1
2	1	2	2	3	1	5	2
1	2	3	2	3	1	6	3
2	2	1	3	3	1	7	4
1	1	2	3	3	2	8	5
2	1	3	3	3	1	9	6

Learning Term-Association

The strength of an association rule can be measure by:

- ● Leverage
- ● Coverage
- ● Support
- ● Strength
- ● Lift

1. Calculating LEVERAGE for the rule:

(T1 = 2) (T2 = 1)

- ● Number of records = 16
- ● Records having (T1 = 2) = 8
- ● Records having (T2 = 1) = 9
- ● Records having (T1 = 2) (T2 = 1) = 4
- ● % of the cover (T1 = 2) (T2 = 1) = 4/16
- ● Records expected to be covered by (T1 = 2) (T2 = 1) if they were independent = (8 * 9) / 16 = 4.5
- ● Leverage Count = 4.5 – 4 = 0.5
- ● Leverage Proportion = 0.5 / 16 = 1/32

T1	T2	T3	T4	T5
1	1	1	1	1
2	1	2	1	1
1	2	3	1	1
2	2	1	2	1
1	1	2	2	1
2	1	3	2	1
1	2	1	3	2
2	2	2	3	2
1	1	3	3	2
2	1	1	1	2
1	2	2	1	2
2	2	3	1	2
1	1	1	2	3
2	1	2	2	3
1	2	3	2	3
2	1	1	3	3

Learning Term-Association

2. Calculating COVERAGE for the rule:

(T1 = 2) (T2 = 1)

- The coverage count for all conditions but the last one (T2=1) = 8
- The coverage proportional = 8/16 = 1/2

3. Calculating SUPPORT for the rule:

(T1 = 2) (T2 = 1)

- The support count for all conditions = 4
- The support proportional = 4/16 = 1/4

4. Calculating STRENGTH for the rule:

(T1 = 2) (T2 = 1)

- The strength count for all conditions but the last one (T2=1) = 8
- The last condition covers 4 out of those 8
- The strength proportional = 4/8 = 1/2

T1	T2	T3	T4	T5
1	1	1	1	1
2	1	2	1	1
1	2	3	1	1
2	2	1	2	1
1	1	2	2	1
2	1	3	2	1
1	2	1	3	2
2	2	2	3	2
1	1	3	3	2
2	1	1	1	2
1	2	2	1	2
2	2	3	1	2
1	1	1	2	3
2	1	2	2	3
1	2	3	2	3
2	1	1	3	3

Learning Term-Association

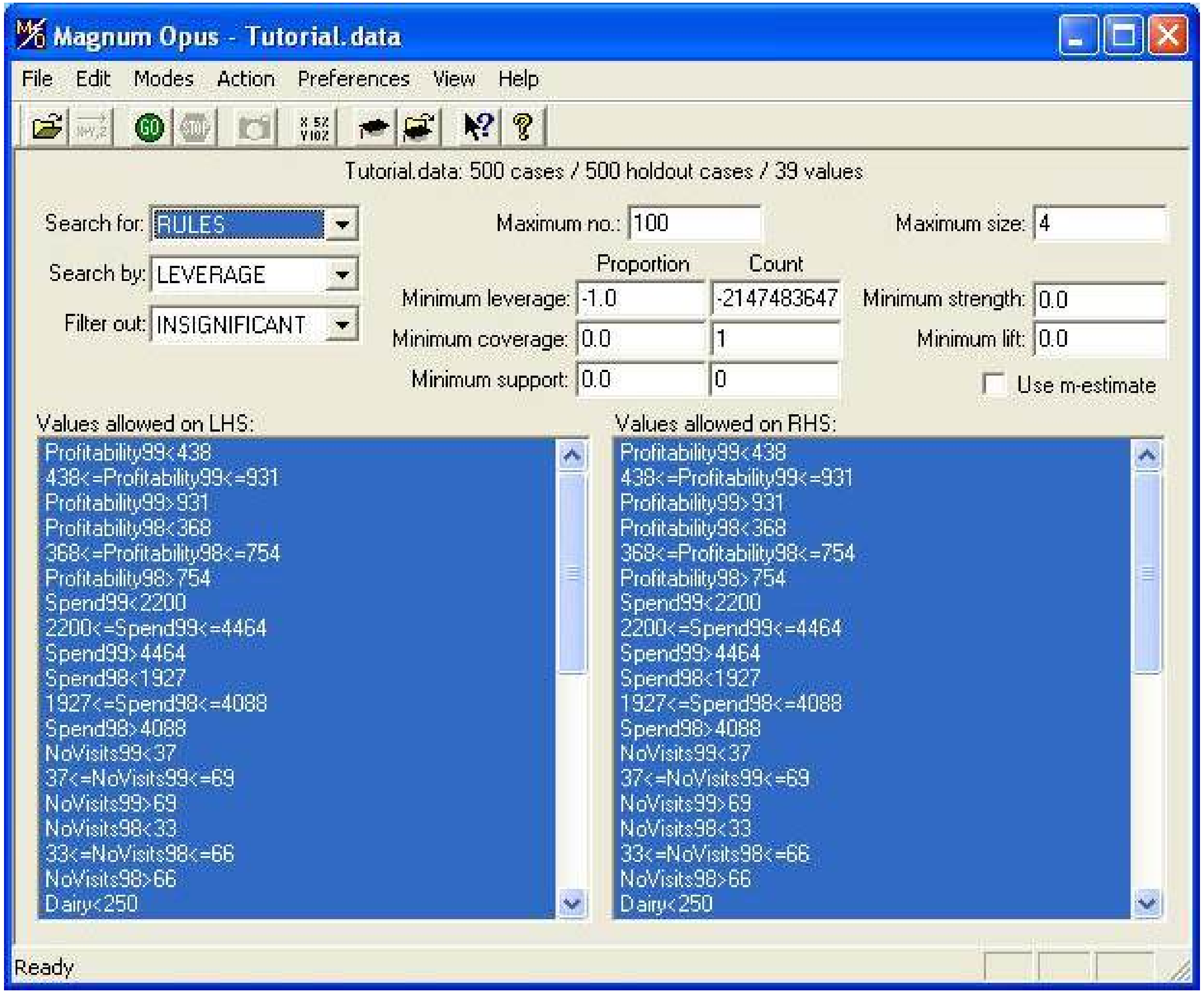
5. Calculating LIFT for the rule:

(T1 = 2) (T2 = 1)

- Total number of examples = 16
- Records covered by all conditions but the last condition (T2=1) = 8
- Records covered by the last condition = 8
- Records covered by all conditions = 4
- Strength = 4 / 8 = 1/2
- Cover proportion of all conditions but the last one (T2=1) = 8 / 16 = 1/2
- LIFT = strength / (cover proportion of all condition but the last) = (1/2) / (1/2) = 1

T1	T2	T3	T4	T5
1	1	1	1	1
2	1	2	1	1
1	2	3	1	1
2	2	1	2	1
1	1	2	2	1
2	1	3	2	1
1	2	1	3	2
2	2	2	3	2
1	1	3	3	2
2	1	1	1	2
1	2	2	1	2
2	2	3	1	2
1	1	1	2	3
2	1	2	2	3
1	2	3	2	3
2	1	1	3	3

The Magnum Opus System



Magnum Opus - Tutorial.data
File Edit Modes Action Preferences View Help

Tutorial.data: 500 cases / 500 holdout cases / 39 values

Search for: RULES

Search by: LEVERAGE

Filter out: INSIGNIFICANT

Maximum no.: 100

Maximum size: 4

	Proportion	Count
Minimum leverage:	-1.0	-2147483647
Minimum coverage:	0.0	1
Minimum support:	0.0	0

Minimum strength: 0.0

Minimum lift: 0.0

□ Use m-estimate

Values allowed on LHS:

Profitability99<438
438<=Profitability99<=931
Profitability99>931
Profitability98<368
368<=Profitability98<=754
Profitability98>754
Spend99<2200
2200<=Spend99<=4464
Spend99>4464
Spend98<1927
1927<=Spend98<=4088
Spend98>4088
NoVisits99<37
37<=NoVisits99<=69
NoVisits99>69
NoVisits98<33
33<=NoVisits98<=66
NoVisits98>66
Dairy<250

Values allowed on RHS:

Profitability99<438
438<=Profitability99<=931
Profitability99>931
Profitability98<368
368<=Profitability98<=754
Profitability98>754
Spend99<2200
2200<=Spend99<=4464
Spend99>4464
Spend98<1927
1927<=Spend98<=4088
Spend98>4088
NoVisits99<37
37<=NoVisits99<=69
NoVisits99>69
NoVisits98<33
33<=NoVisits98<=66
NoVisits98>66
Dairy<250

Ready

Attributes and their values for the Tutorial database

- Profitability99: numeric 3
- Profitability98: numeric 3
- Spend99: numeric 3
- Spend98: numeric 3
- NoVisits99: numeric 3
- NoVisits98: numeric 3
- Dairy: numeric 3
- Deli: numeric 3
- Bakery: numeric 3
- Grocery: numeric 3
- SocioEconomicGroup: categorical
- Promotion1: t, f
- Promotion2: t, f

Statistical Association

Magnum Opus

DEMO

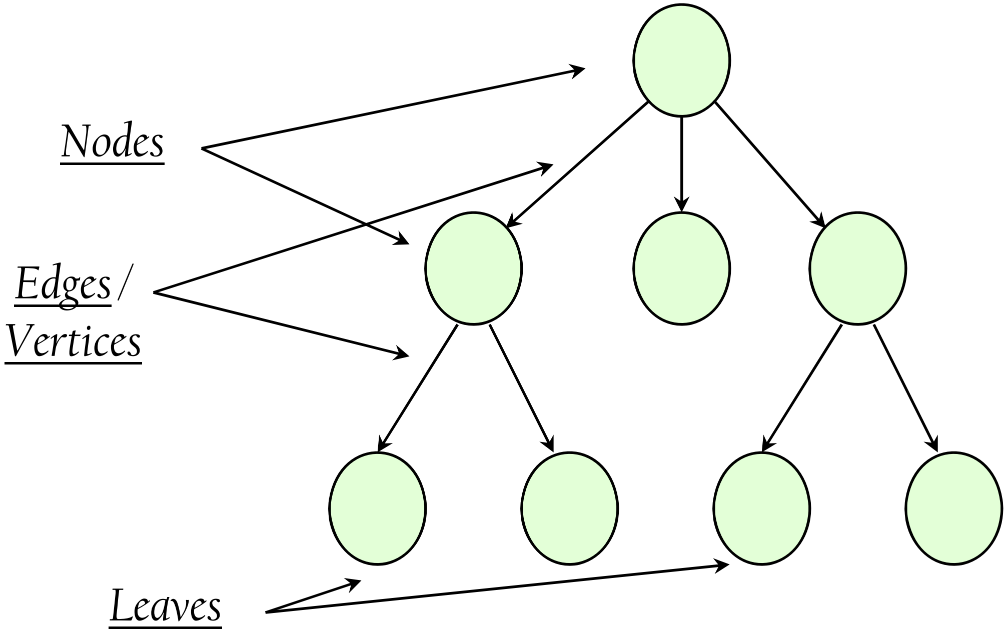
DECISION TREES

Part 12

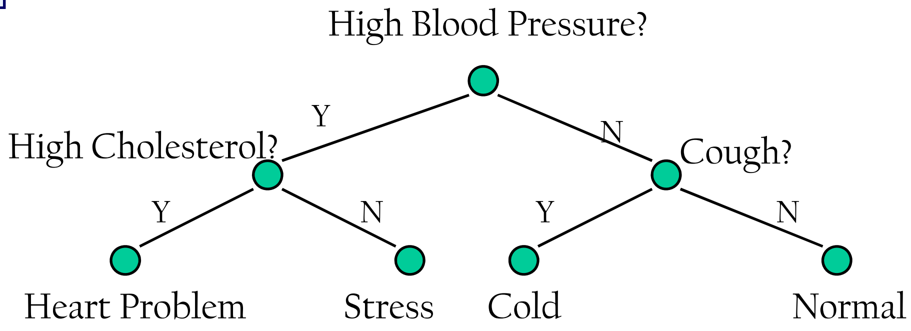
Using Statistical & Information Theory

Learning Decision Trees

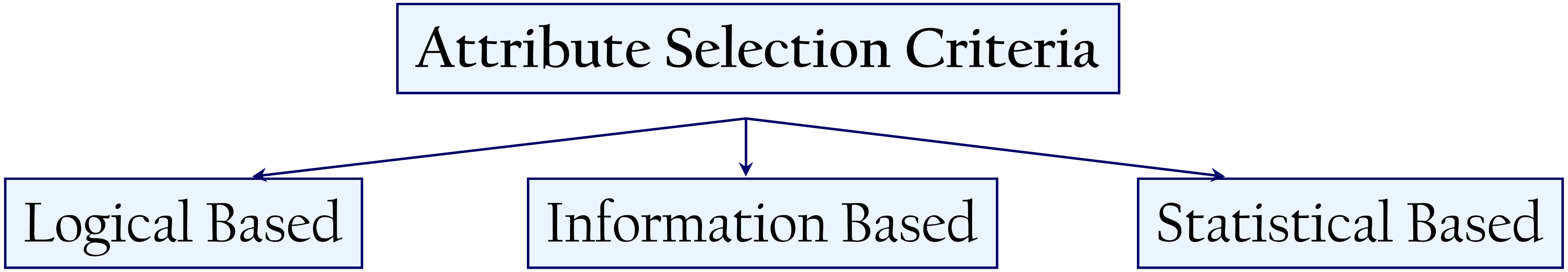
- A *Tree* is a Directed Acyclic Graph *(DAG)* + each node has one parent at most
- A *Decision Tree* is a tree where nodes associated with attributes, edges associated with attribute values, and leaves associated with decisions



Example:



Learning Decision Trees



Attribute Selection Criteria

Logical Based

Information Based

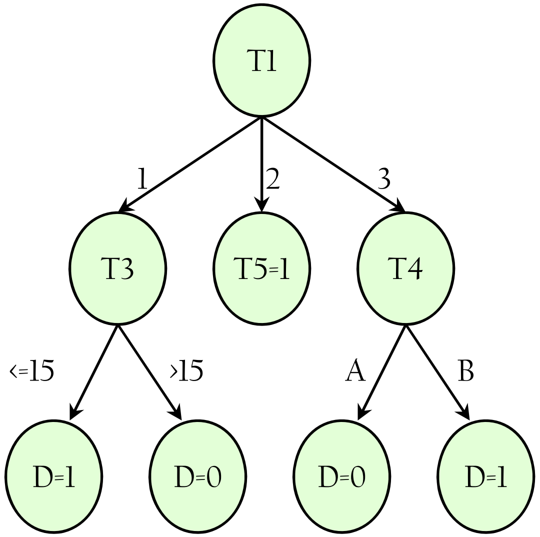
Statistical Based

Information Theory

Example

- T2 is quantized into two intervals at 21 (T2<=21) and (T2>21)
- T3 is quantized into two intervals at 15 (T3<=15) and (T3>15)

T1	T2	T3	T4	D
1	25	10	A	1
1	30	30	A	0
1	35	25	B	0
1	22	35	B	0
1	19	10	B	1
2	22	30	A	1
2	33	18	B	1
2	14	5	A	1
2	31	15	B	1
3	21	20	A	0
3	15	10	A	0
3	25	20	B	1
3	18	20	B	1
3	20	36	B	1



Decision Trees

C5

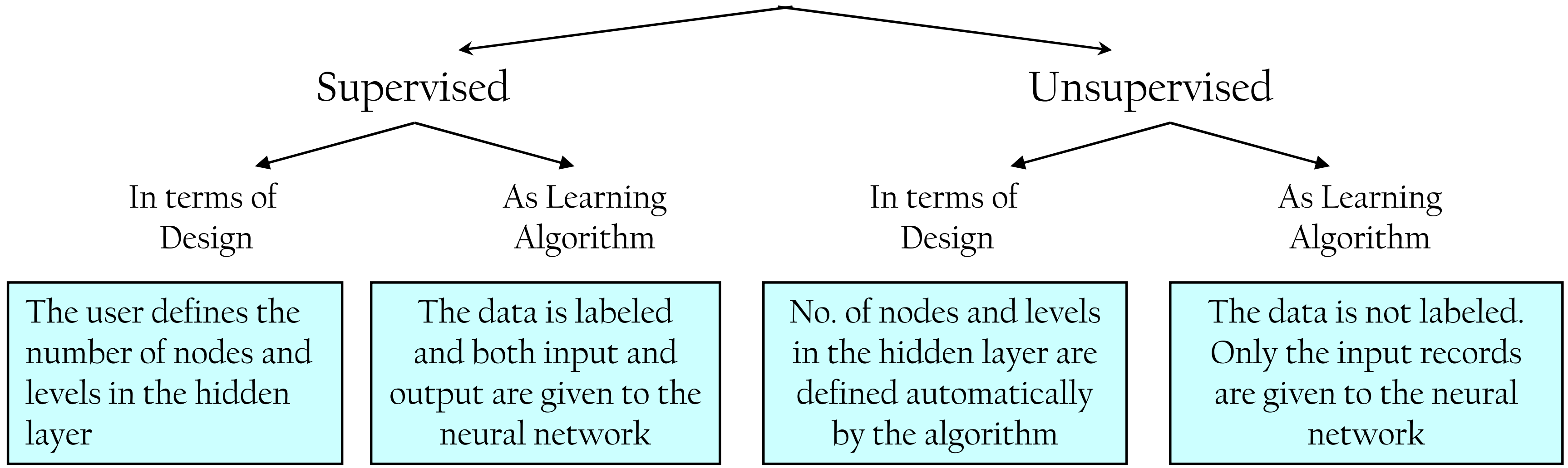
DEMO

NEURAL NETWORKS

Part 13

How It Works?

Learning Neural Networks



Supervised

Unsupervised

In terms of Design

As Learning Algorithm

In terms of Design

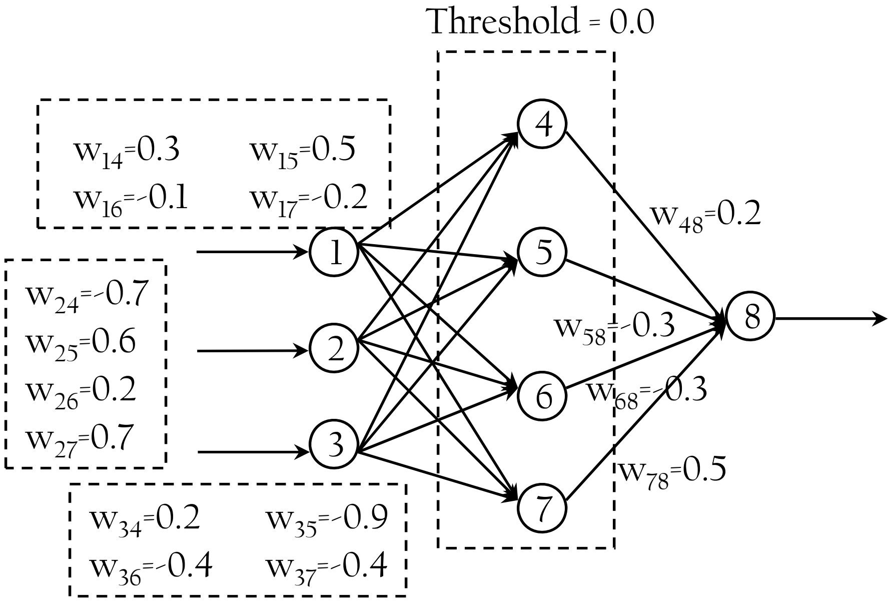
As Learning Algorithm

The user defines the number of nodes and levels in the hidden layer

The data is labeled and both input and output are given to the neural network

No. of nodes and levels in the hidden layer are defined automatically by the algorithm

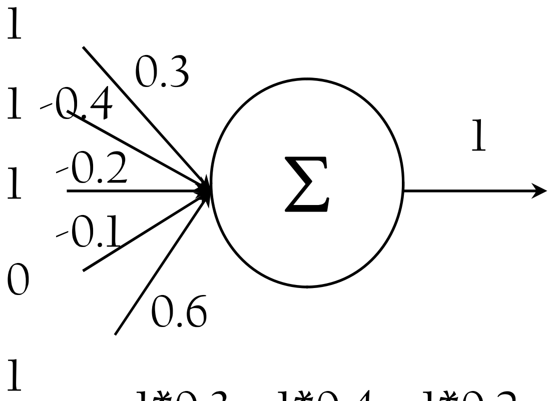
The data is not labeled. Only the input records are given to the neural network



Test Data

A	B	C	Decision
0	0	0	
0	0	1	
0	1	0	
0	1	1	1
1	0	0	
1	0	1	
1	1	0	
1	1	1	

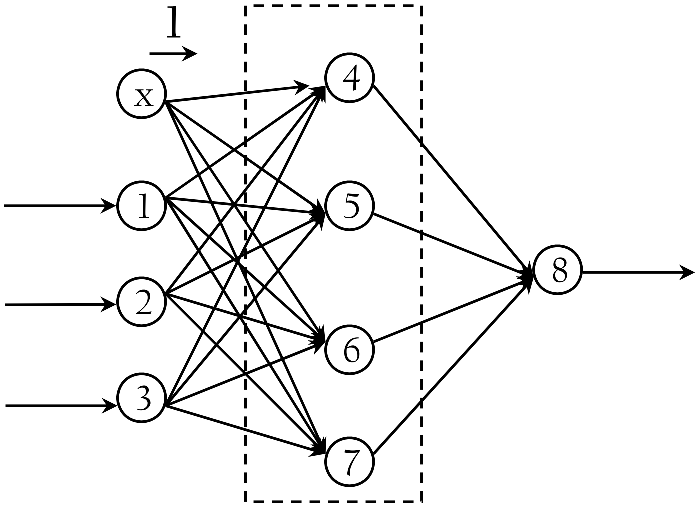
Learning Neural Networks



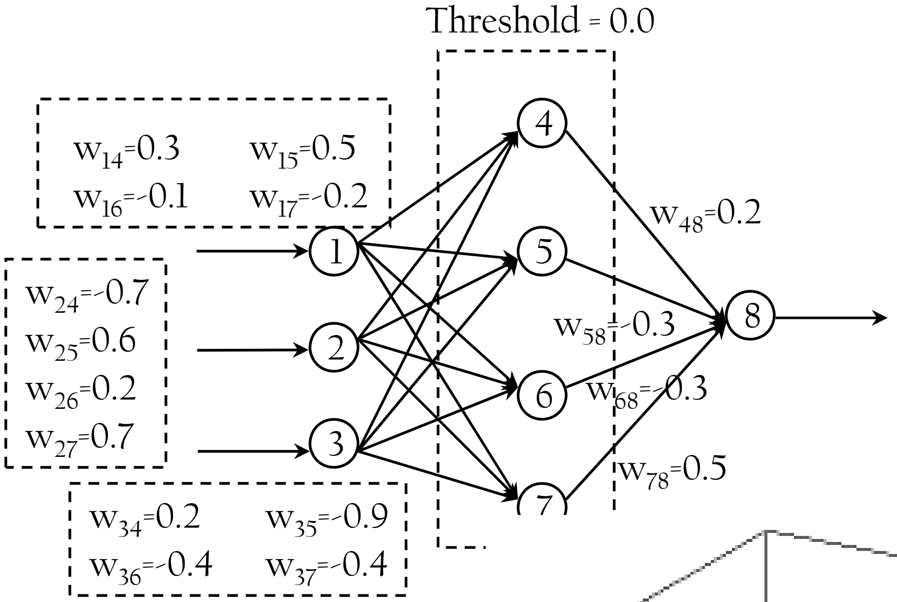
The Sigmoid Function

=1*0.3 – 1*0.4 – 1*0.2 – 0*0.1 + 1*0.6 = 0.3 > 0.0

To avoid setting the threshold:

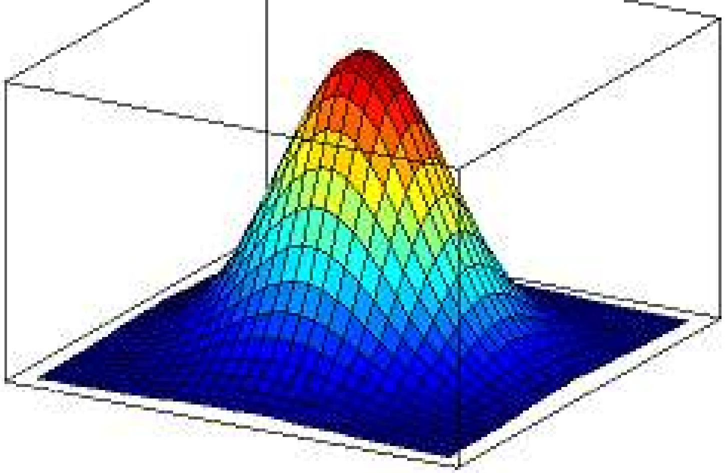


Learning Neural Networks



Test Data

A	B	C	Decision
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



MACHINE TRANSLATION

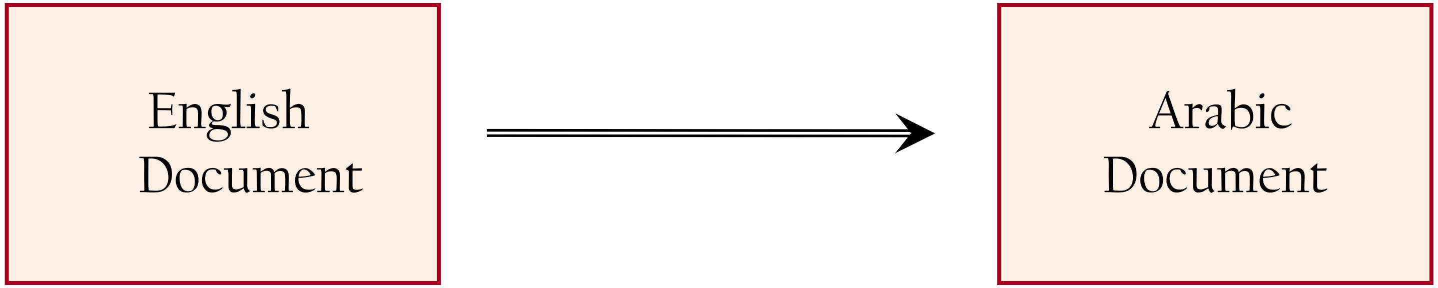
Part 14

Statistical Machine Translation

Statistical Machine Translation

- For each English sentence “e”, we need the Arabic sentence “a” which maximize $P(a|e)$

  $P(a|e)=P(a)*P(e|a)/P(e)$



English Document

Arabic Document

Language Model

- A statistical **language model** assigns a probability to a sequence of *m* words by means of a probability distribution
- Record every sentence that anyone ever says in Arabic; Suppose you record a database of one billion utterances; If the sentence “كيف حالك؟” appears 76,413 times in that database, then we say P(كيف حالك؟) = 76,413/1,000,000,000 = 0.000076413
- One big problem is that many perfectly good sentences will be assigned a P(e) of zero

Arabic Sentence	Probability
كيف حالك	0.000076413
الولد سعيد	0.000066392

N-Grams

- An n-word substring is called an n-gram
- If n=2, we say bigram. If n=3, we say trigram
- Let P(y | x) be the probability that word y follows word x

  P(y | x) = number-of-occurrences(“xy”) / number-of-occurrences(“x”)

  P(z | x y) = number-of-occurrences(“xyz”) / number-of-occurrences(“xy”)

➔ P(ذهب الولد إلى المدرسة) = P(ذهب | start-of-sentence) *
P(الولد | ذهب) * P(إلى | الولد) * P(المدرسة | إلى) *
P(end-of-sentence | المدرسة)

➔ P(ذهب الولد إلى المدرسة) = P(ذهب | start-of-sentence) *
P(الولد | start-of-sentence ,ذهب) * P(إلى | الولد ,ذهب) *
P(المدرسة | إلى ,الولد) * P(end-of-sentence | المدرسة ،إلى) *
P(end-of-sentence | end-of-sentence ,المدرسة)

N-Grams Language Model

$$P(w_1,\dots,w_m)=\prod_{i=1}^{m}P(w_i\mid w_1,\dots,w_{i-1})\approx\prod_{i=1}^{m}P(w_i\mid w_{i-(n-1)},\dots,w_{i-1})$$

$$P(w_i\mid w_{i-(n-1)},\dots,w_{i-1})=\frac{count(w_{i-(n-1)},\dots,w_i)}{count(w_{i-(n-1)},\dots,w_{i-1})}$$

Example:

In a bigram (n=2) language model, the approximation looks like

$$P(I,saw,the,red,house)\approx P(I)P(saw\mid I)P(the\mid saw)P(red\mid the)P(house\mid red)$$

In a trigram (n=3) language model, the approximation looks like

$$P(I,saw,the,red,house)\approx P(I)P(saw\mid I)P(the\mid I,saw)P(red\mid saw,the)P(house\mid the,red)$$

Translation Model

- P(a | e), the probability of an Arabic string “a” given an English string “e”. This is called a translation model
- P(a | e) will be a module in overall English-to-Arabic machine translation system; When we see an actual English string e, we want to reason backwards ... What Arabic string a is (1) likely to be uttered, and (2) likely to subsequently translate to e? We're looking for the a that maximizes P(a) * P(e | a)

Arabic Sentence	English Sentence	P(a|e)
ذهب الولد إلى المدرسة	The boy went to School	0.0034
إنخفاض البورصة اليوم	Today, the stock market went down	0.00021
:	:	

Translation Model

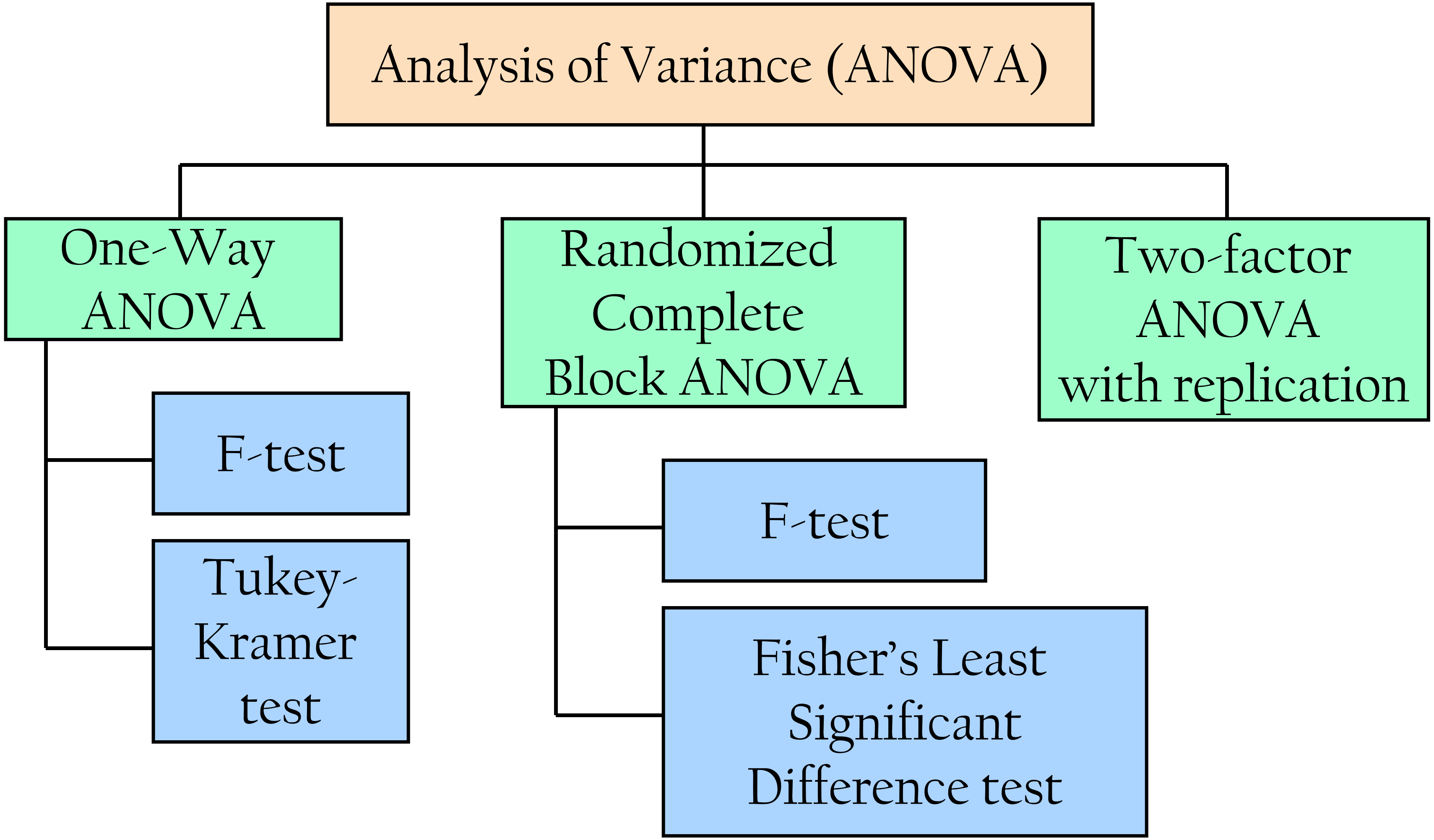
- For each word a_i in an Arabic sentence ($i = 1 \dots l$), we choose a fertility ϕ_i. The choice of fertility depends on the Arabic word in question. It is not dependent on the other Arabic words in the Arabic sentence, or on their fertilities
- For each word a_i, we generate ϕ_i English words. The choice of English word depends on the Arabic word that generates it. It is not dependent on the Arabic context around the Arabic word. It is not dependent on other English words that have been generated from this or any other Arabic word
- All those English words are permuted. Each English word is assigned an absolute target "position slot." For example, one word may be assigned position 3, and another word may be assigned position 2 -- the latter word would then precede the former in the final English sentence. The choice of position for a English word is dependent solely on the absolute position of the Arabic word that generates it

STATISTICS

Part 15

Analysis of Variance
ANOVA

Analysis of Variance ANOVA



Analysis of Variance (ANOVA)

One-Way ANOVA

F-test

Tukey-Kramer test

Randomized Complete Block ANOVA

F-test

Fisher's Least Significant Difference test

Two-factor ANOVA with replication

ONE WAY ANOVA

- Evaluate the difference among the means of three or more populations
- Assumptions

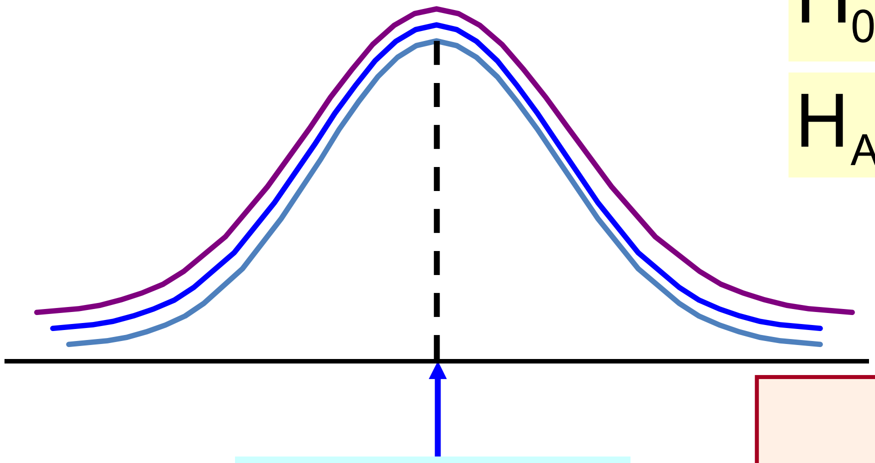
  Populations are normally distributed

  Populations have equal variances

  Samples are randomly and independently drawn

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

$$H_A : \text{Not all } \mu_i \text{ are the same}$$



$$\mu_1 = \mu_2 = \mu_3$$

All Means are the same:

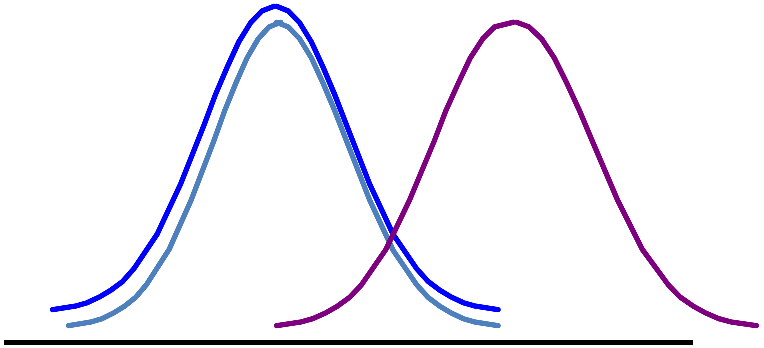
The Null Hypothesis is True

ONE WAY ANOVA

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

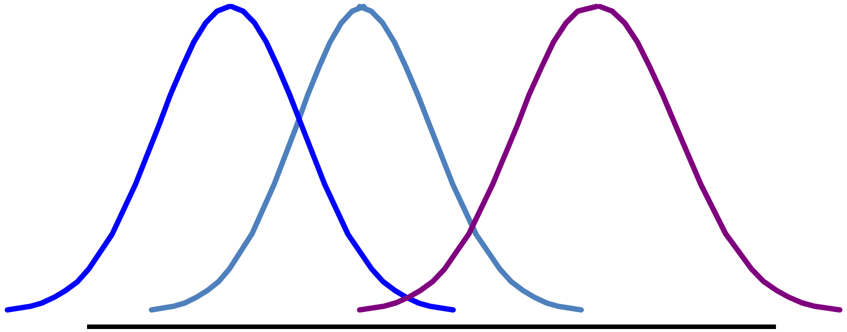
$$H_A : \text{Not all } \mu_i \text{ are the same}$$

At least one mean is different:
The Null Hypothesis is NOT true
(Treatment Effect is present)



$$\mu_1 = \mu_2 \neq \mu_3$$

or



$$\mu_1 \neq \mu_2 \neq \mu_3$$

Partitioning the Variations

SST = SSB + SSW

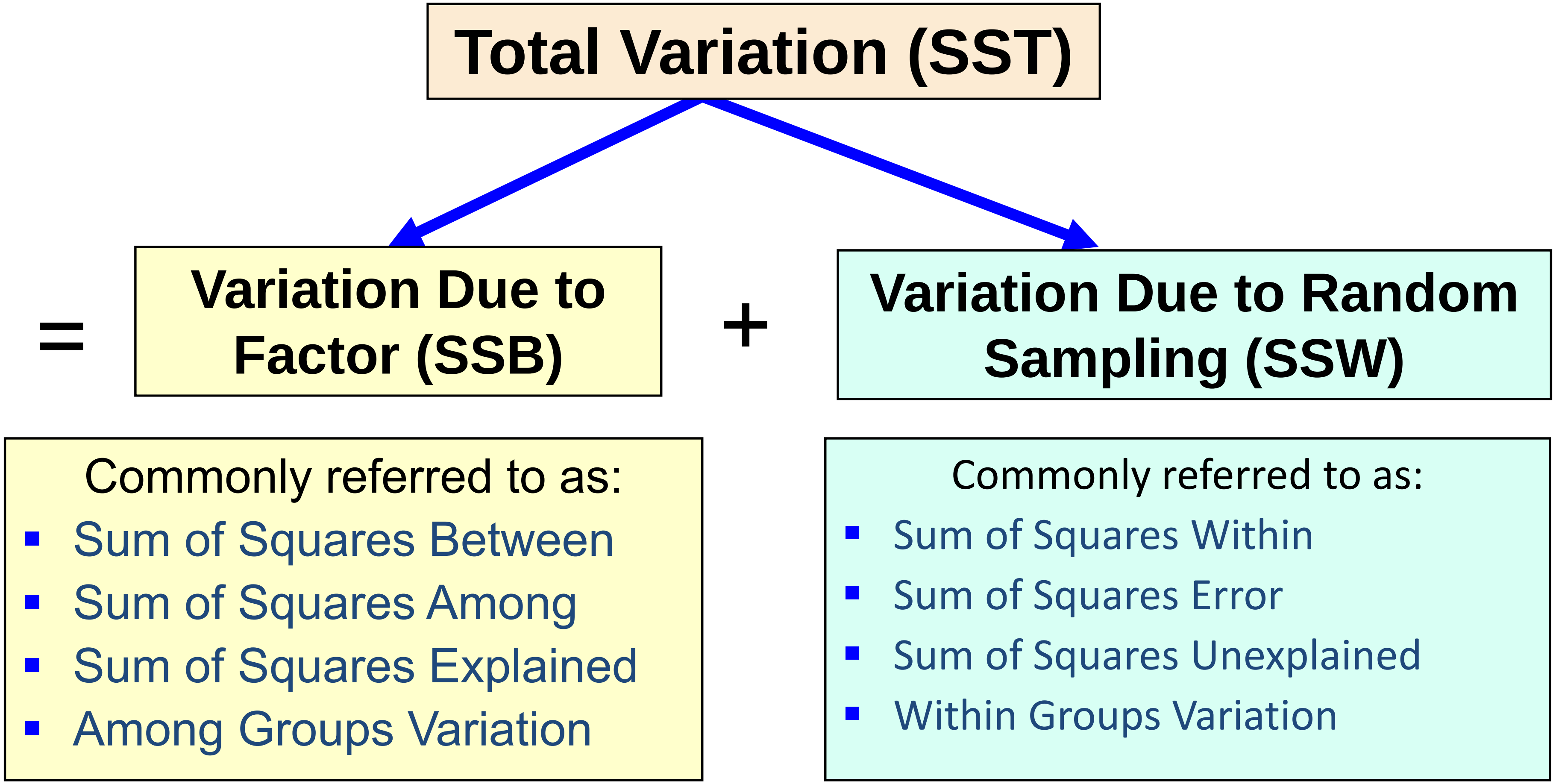
SST = Total Sum of Squares
SSB = Sum of Squares Between
SSW = Sum of Squares Within

Total Variation = the aggregate dispersion of the individual data values across the various factor levels (SST)

Between-Sample Variation = dispersion among the factor sample means (SSB)

Within-Sample Variation = dispersion that exists among the data values within a particular factor level (SSW)

Partition of Total Variation



Total Variation (SST)

= **Variation Due to Factor (SSB)** + **Variation Due to Random Sampling (SSW)**

Commonly referred to as:

- Sum of Squares Between
- Sum of Squares Among
- Sum of Squares Explained
- Among Groups Variation

Commonly referred to as:

- Sum of Squares Within
- Sum of Squares Error
- Sum of Squares Unexplained
- Within Groups Variation

Total Sum of Squares

$$\boxed{SST} = SSB + SSW$$

$$\boxed{SST = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (x_{ij} - \bar{\bar{x}})^2}$$

Where:

SST = Total sum of squares

k = number of populations (levels or treatments)

n_i = sample size from population i

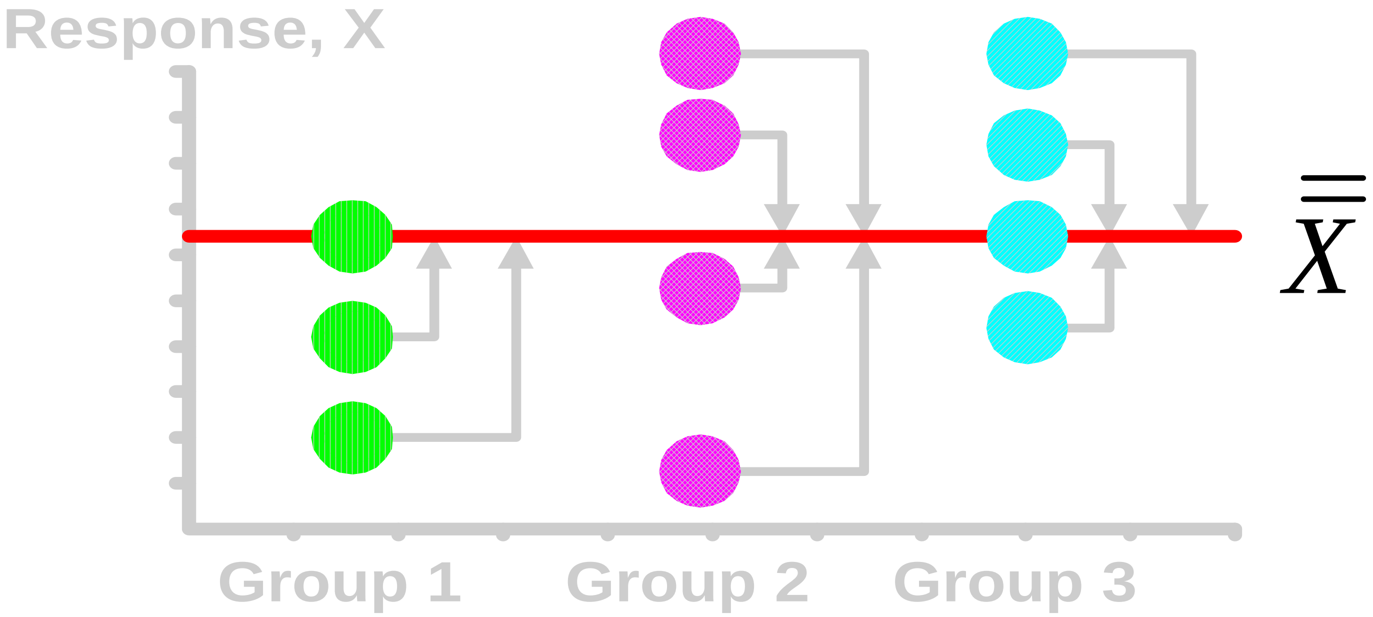
x_{ij} = jth measurement from population i

$\bar{\bar{x}}$ = grand mean (mean of all data values)

Total Variation

(continued)

$$\boxed{SST = (x_{11} - \bar{\bar{x}})^2 + (x_{12} - \bar{\bar{x}})^2 + \ldots + (x_{kn_k} - \bar{\bar{x}})^2}$$



Sum of Squares Between

$$\mathrm{SST} = \boxed{\mathrm{SSB}} + \mathrm{SSW}$$

$$\boxed{\mathrm{SSB} = \sum_{i=1}^{k} n_i(\bar{x}_i - \bar{\bar{x}})^2}$$

Where:

SSB = Sum of squares between

k = number of populations

n_i = sample size from population i

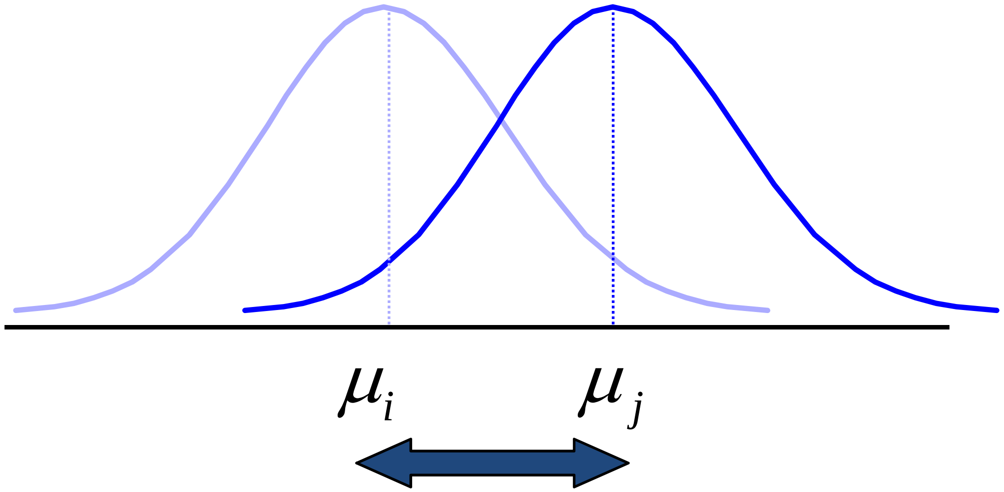
$\bar{x}_i$ = sample mean from population i

$\bar{\bar{x}}$ = grand mean (mean of all data values)

Between-Group Variation

$$\mathrm{SSB} = \sum_{i=1}^{k} n_i (\bar{x}_i - \bar{\bar{x}})^2$$

Variation Due to
Differences Among Groups



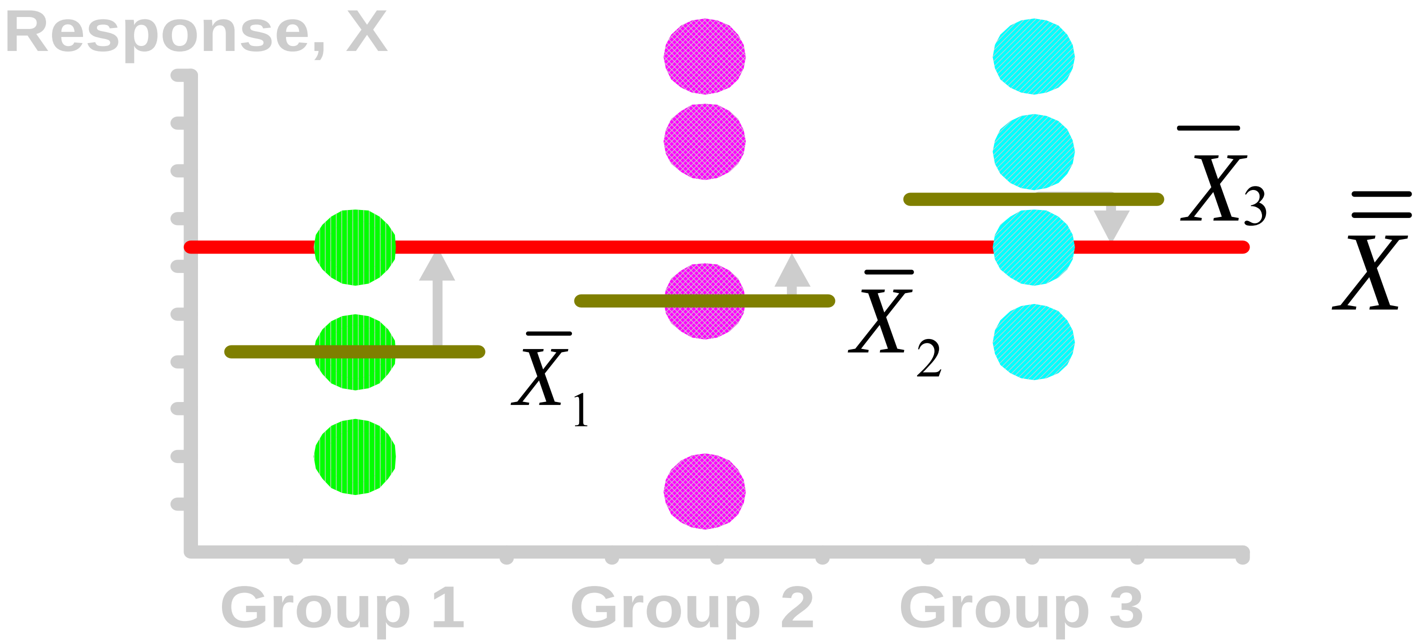
$$\mathrm{MSB} = \frac{\mathrm{SSB}}{k-1}$$

Mean Square Between =
SSB/degrees of freedom

Between-Group Variation

(continued)

$$\boxed{SSB = n_1(\bar{x}_1 - \bar{\bar{x}})^2 + n_2(\bar{x}_2 - \bar{\bar{x}})^2 + \ldots + n_k(\bar{x}_k - \bar{\bar{x}})^2}$$



Sum of Squares Within

$$SST = SSB + \boxed{SSW}$$

$$\boxed{SSW = \sum_{i=1}^{k} \sum_{j=1}^{n_j} (x_{ij} - \bar{x}_i)^2}$$

Where:

SSW = Sum of squares within

k = number of populations

n_i = sample size from population i

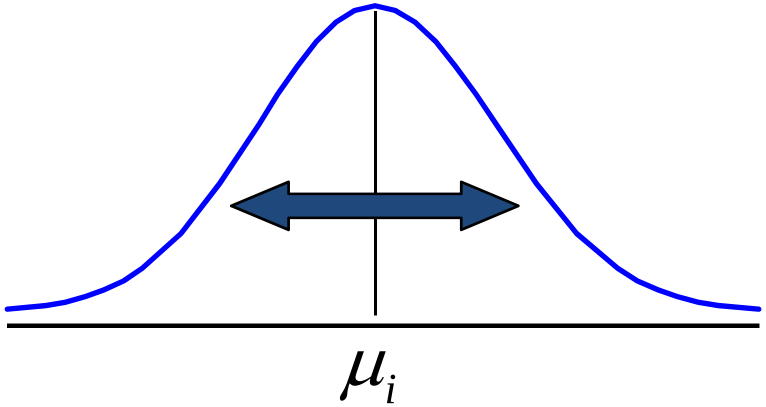
$\bar{x}_i$ = sample mean from population i

x_{ij} = j^{th} measurement from population i

Within-Group Variation

$$SSW = \sum_{i=1}^{k} \sum_{j=1}^{n_j} (x_{ij} - \bar{x}_i)^2$$

Summing the variation within each group and then adding over all groups



$$MSW = \frac{SSW}{N - k}$$

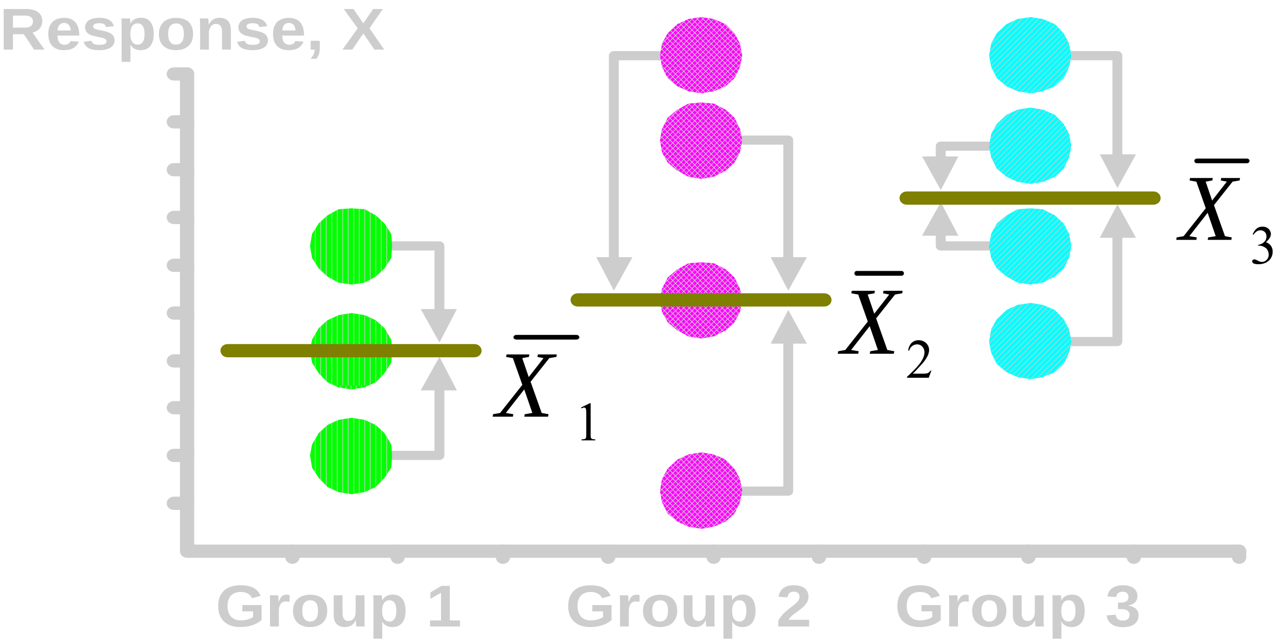
Mean Square Within =

SSW/degrees of freedom

Within-Group Variation

(continued)

$$\boxed{\text{SSW} = (x_{11} - \bar{x}_1)^2 + (x_{12} - \bar{x}_2)^2 + \ldots + (x_{kn_k} - \bar{x}_k)^2}$$

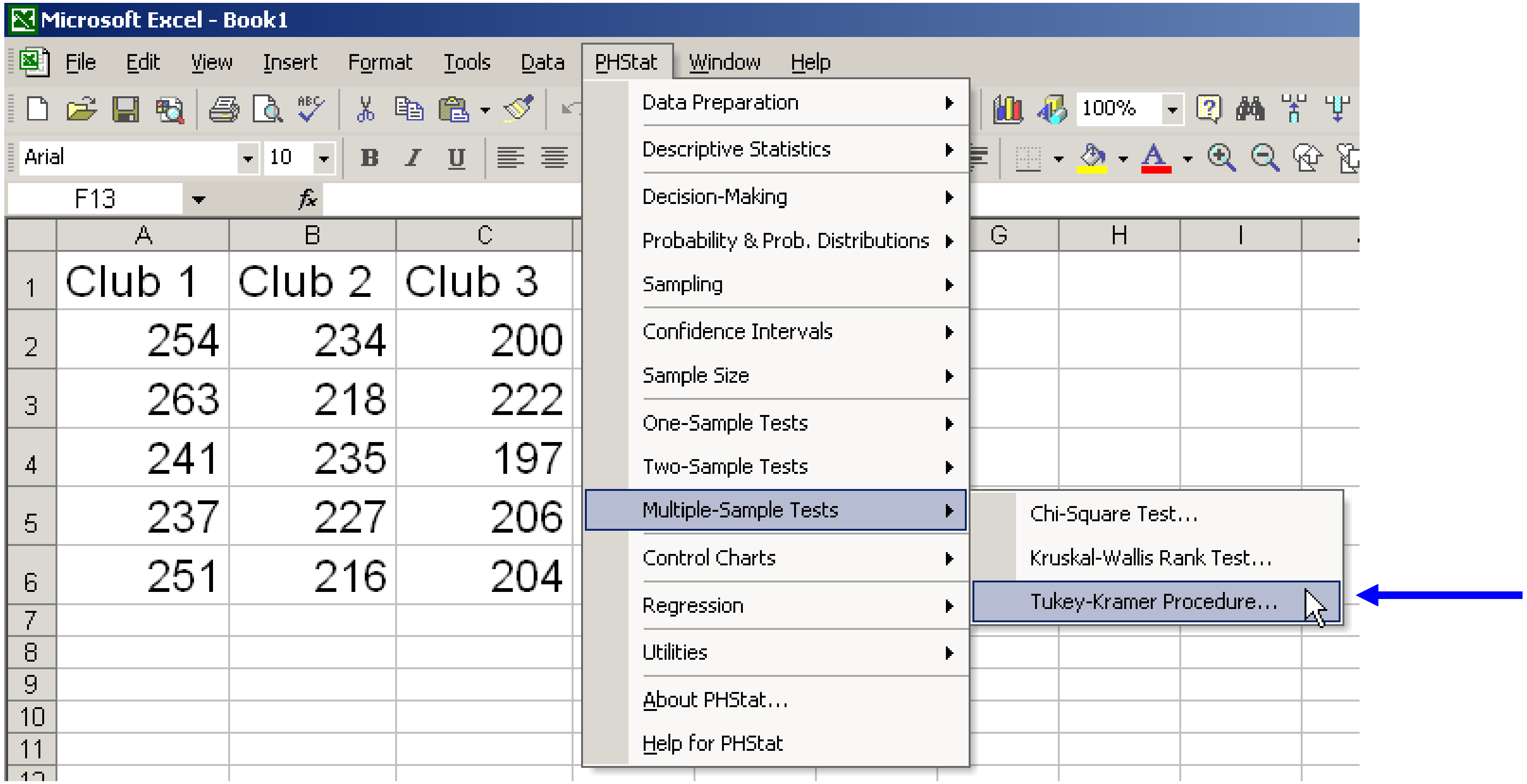


One-Way ANOVA Table

Source of Variation	SS	df	MS	F ratio
Between Samples	SSB	k - 1	$MSB = \frac{SSB}{k - 1}$	$F = \frac{MSB}{MSW}$
Within Samples	SSW	N - k	$MSW = \frac{SSW}{N - k}$	
Total	SST = SSB+SSW	N - 1		

k = number of populations
N = sum of the sample sizes from all populations
df = degrees of freedom

Tukey-Kramer in PHStat



Club 1	Club 2	Club 3
254	234	200
263	218	222
241	235	197
237	227	206
251	216	204

Probability

Part 16

Bayesian Networks

Bayesian Networks (Watch Me!)

Conclusion

1- Basic Concepts

2- Introduction to Vectors

3- Probability

4- Statistics

5- Regression

6- Statistics & Testing

7- Test of Significance

8- Information Theory

9- Basics for Language Engineers

10- Statistical Association

11- Statistical Machine Translation

12- Analysis of Variance

13- Bayesian Networks

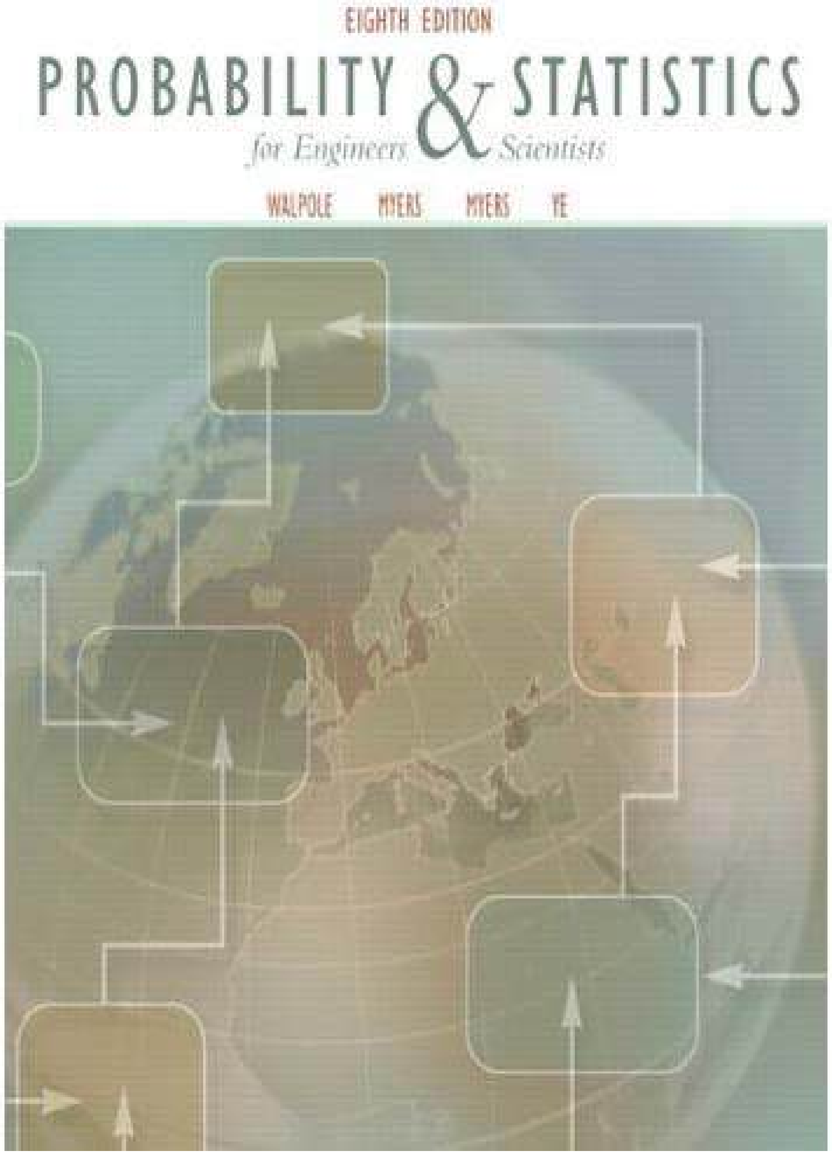
REFERENCES

- W. Weaver (1955). Translation (1949). In: *Machine Translation of Languages*, MIT Press, Cambridge, MA.
- P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer (1993). The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, **19(2)**, 263-311.
- S. Vogel, H. Ney and C. Tillmann. 1996. HMM-based Word Alignment in StatisticalTranslation. In COLING '96: The 16th International Conference on Computational Linguistics, pp. 836-841, Copenhagen, Denmark.
- F. Och and H. Ney. (2003). A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics, 29(1):19-51
- P. Koehn, F.J. Och, and D. Marcu (2003). Statistical phrase based translation. In *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT/NAACL)*.
- D. Chiang (2005). A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*.
- F. Och and H. Ney. (2003). A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics, 29(1):19-51
- P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, E. Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. ACL 2007, Demonstration Session, Prague, Czech Republic
- Q. Gao, S. Vogel, "Parallel Implementations of Word Alignment Tool", Software Engineering, Testing, and Quality Assurance for Natural Language Processing, pp. 49-57, June, 2008
- W. J. Hutchens and H. Somers. (1992). An Introduction to Machine Translation, 18.3:322. ISBN 0-12-36280-X

REFERENCES

- W. The Sage Dictionary of Statistics, pg. 76, Duncan Cramer, Dennis Howitt, 2004, ISBN 076194138X
- E.L. Lehmann and Joseph P. Romano (2005). *Testing Statistical Hypotheses* (3E ed.). New York, NY: Springer. ISBN 0387988645
- D.R. Cox and D.V.Hinkley (1974). *Theoretical Statistics*. ISBN 0412124293.
- Fisher, Sir Ronald A. (1956) [1935]. "Mathematics of a Lady Tasting Tea". in James Roy Newman. *The World of Mathematics, volume 3.* http://books.google.com/books?id=oKZwtLQTmNAC&pg=PA1512&dq=%22mathematics+of+a+lady+tasting+tea%22&sig=8-NQlCLzrh-oV0wjfwa0EgspSNU
- R.A. Fisher, the Life of a Scientist, Box, 1978, p134
- Mccloskey, Deirdre (2008). *The Cult of Statistical Significance*. Ann Arbor: University of Michigan Press. ISBN 0472050079
- *What If There Were No Significance Tests?*, Harlow, Mulaik & Steiger, 1997, ISBN 978-0-8058-2634-0
- Rosnow, R.L. & Rosenthal, R. (1989). Statistical procedures and the justification of knowledge in psychological science. American Psychologist, 44, 1276-1284
- Loftus, G.R. 1991. On the tyranny of hypothesis testing in the social sciences. Contemporary Psychology 36: 102-105
- Cohen, J. 1990. Things I have learned (so far). American Psychologist 45: 1304-1312. ^ Introductory Statistics, Fifth Edition, 1999, pg. 521, Neil A. Weiss, ISBN 0-201-59877-9
- Ioannidis JP (July 2005). "Contradicted and initially stronger effects in highly cited clinical research". *JAMA* **294** (2): 218–28.

REFERENCES



EIGHTH EDITION

PROBABILITY & STATISTICS

for Engineers & Scientists

WALPOLE MYERS MYERS YE