從零開始的資料探勘: 從資料到知識
Data Mining from Scratch: from Data to Knowledge

姜俊宇
Jyun-Yu Jiang

2016 台灣資料科學年會
Data Science in Taiwan Conference 2016

July 14, 2016

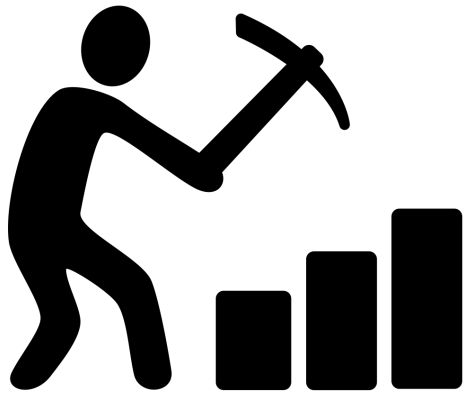
Outline

Outline

Outline

What is Data Mining?

- Extract useful information from data
- Transform information into understandable structure
- The analysis step in knowledge discovery in databases (KDD)



Credit: Matt Brooks @ Noun Project

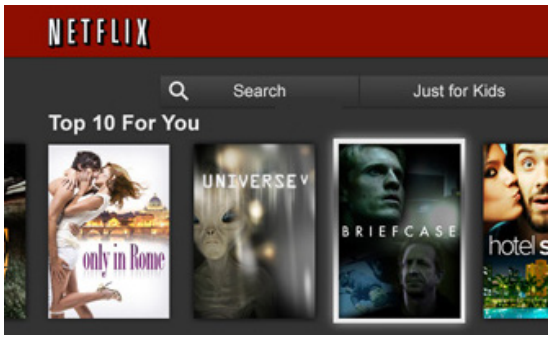
i.e., data mining is to mine human-understandable knowledge from data.

Data mining leads out various applications in real world.

Search Engines

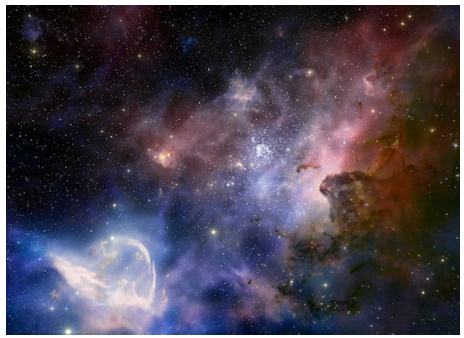
Spam Filtering

Advertising



Recommender Systems

Facebook Newsfeed



... and more!!

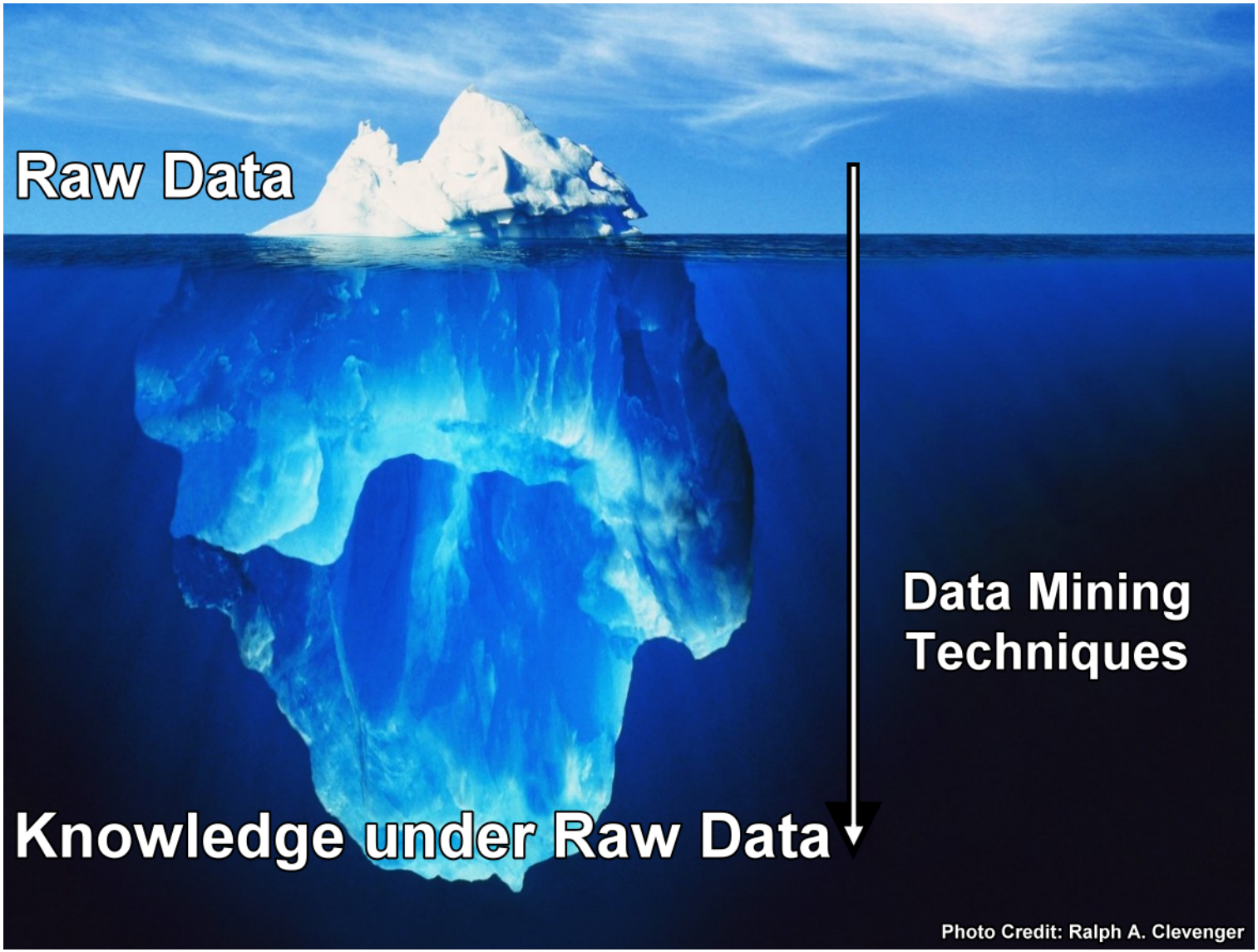
However, raw data in real-world are so messy.

- Real-world data are usually not organized.
- Most of data are logs (outcomes) of services or sensors.
- Problems might be too difficult to be directly solved.

Clicks	Impressions	User ID	Ad ID	Query ID	Depth	Position
0	3	1434...4125	9027238	23808	1	1
0	1	2010.....927	22141749	38772	2	2
0	2	7903.....889	20869452	2010	1	1
2	3	2994.....615	20061280	23803	1	1
0	1	2010.....927	22141715	38772	3	3
0	1	6767.....692	21099585	8834	2	2
0	1	1529...1541	21811282	36841	3	3

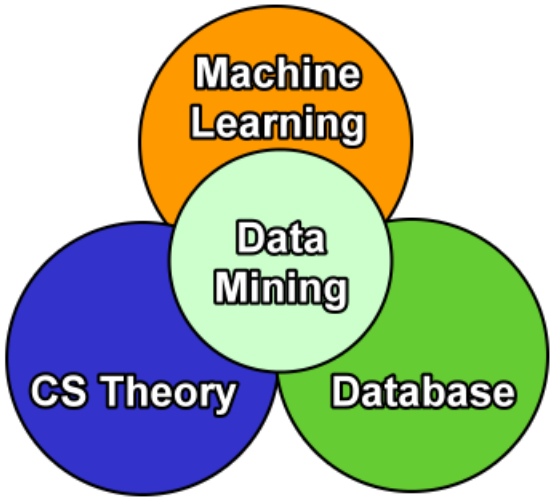
Table: Real data captured from an online advertising dataset (KDDCUP '12).

Data mining techniques are required to discover knowledge.



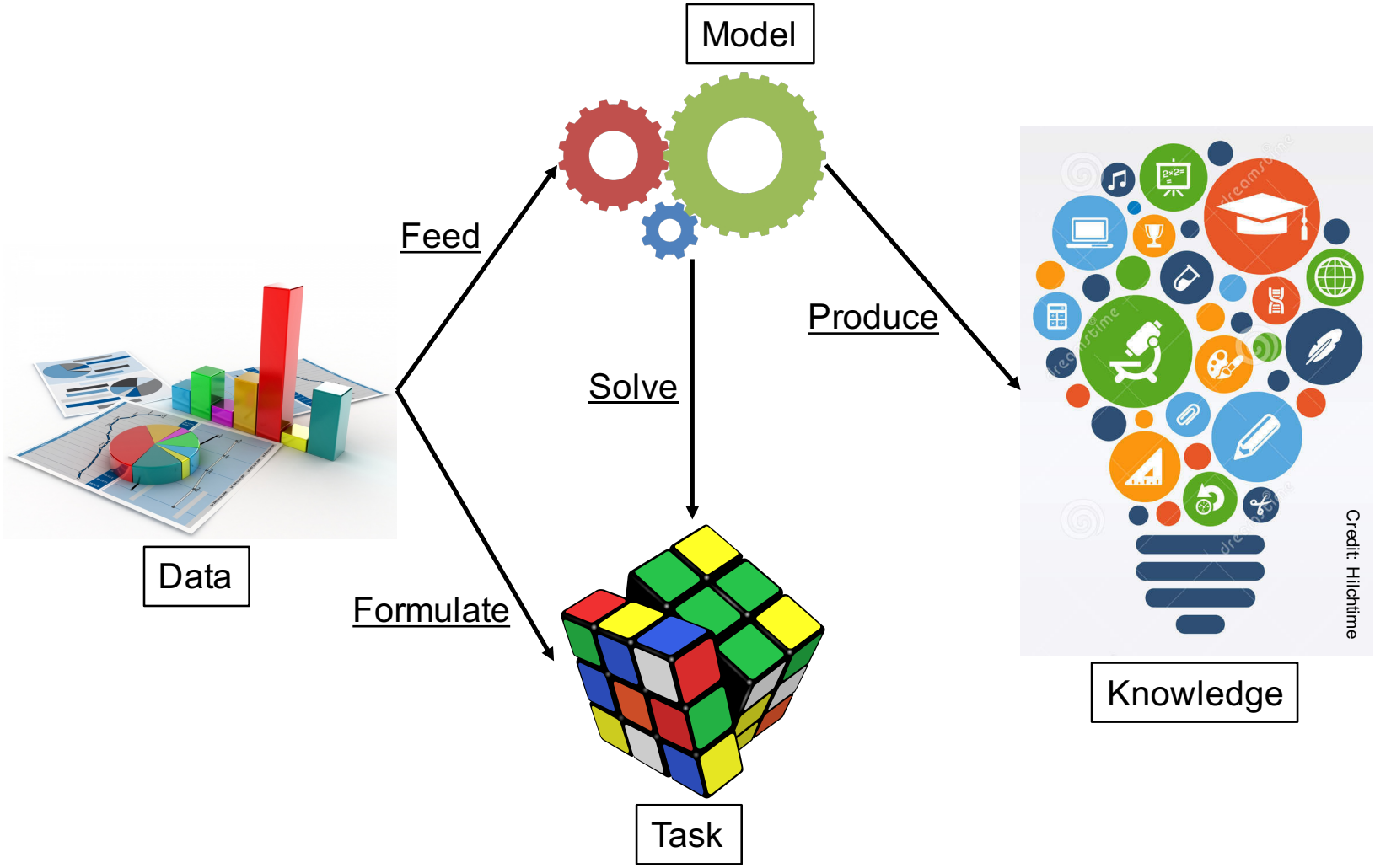
More about data mining...

- It is an area overlapping with
  - Database – (Large-scale) Data
  - Machine Learning – (Learning) Model
  - CS Theory – Algorithms



Outline

Roadmap: from Data to Task to Knowledge



Models in Data Mining

- Data mining techniques build models to solve problems.

Model Properties [Cios et al., Data Mining 2007]

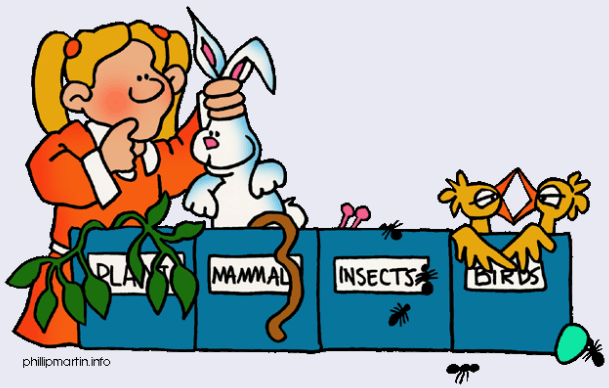
- Valid – hold on new data with some certainty.
- Novel – non-obvious to the system.
- Useful – should be possible to act on the item.
- Understandable – humans should be able to interpret the pattern.

Tasks in Data Mining

- Problems should be well formulated before being solved.
- Two categories of tasks [Fayyad et al., 1996]

Predictive Tasks

- Predict unknown values
- e.g., animal classification



Credit: Phillip Martin

Descriptive Tasks

- Find patterns to describe data
- e.g., social group clustering

More Fundamental Tasks in Detail

- Problems can be further decomposed into some fundamental tasks.

Predictive Tasks

- Classification
- Regression
- Ranking
- ...

Descriptive Tasks

- Clustering
- Summarization
- Association Rule Learning
- ...

Classification

- Generalize known structure to apply to new data.
- Learn a classifier (model) to classify new data.
- Example:
  - Given the current social network, predict whether two nodes will be linked in the future.
  - Classes: "linked" and "not linked."

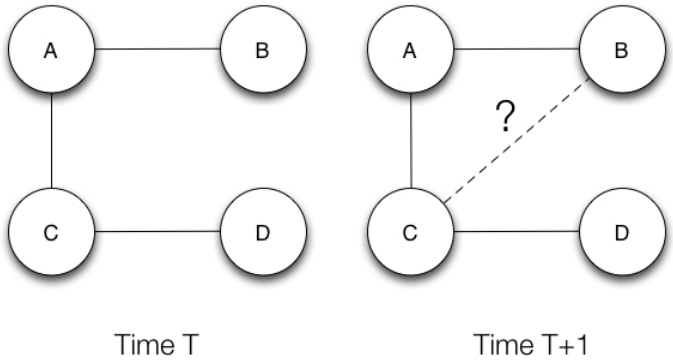


Figure: Link Prediction [Hsieh et al., WWW 2013]

Regression

- Find a function which model the data with least error.
- The output might be a numerical value.
- Example:
  - Predict the click-through rate (CTR) in search engine advertising.

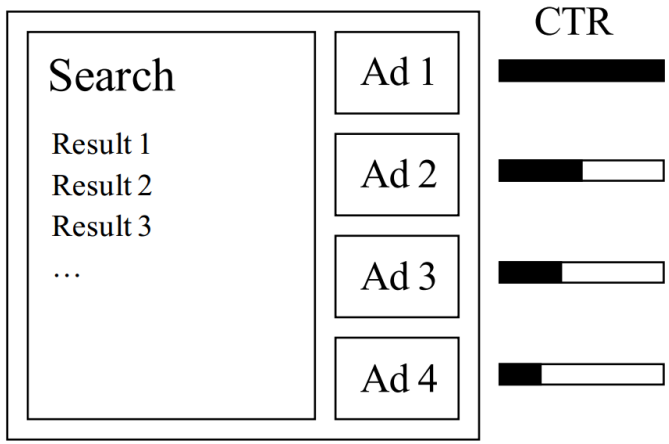


Figure: Predict CTR of Ads. [Richardson et al., WWW 2007]

Ranking

- Produce a permutation to items in a new list
- Items ranked in higher positions should be more important
- Example:
  - Rank webpages in a search engine
  - Webpages in higher positions are more relevant.

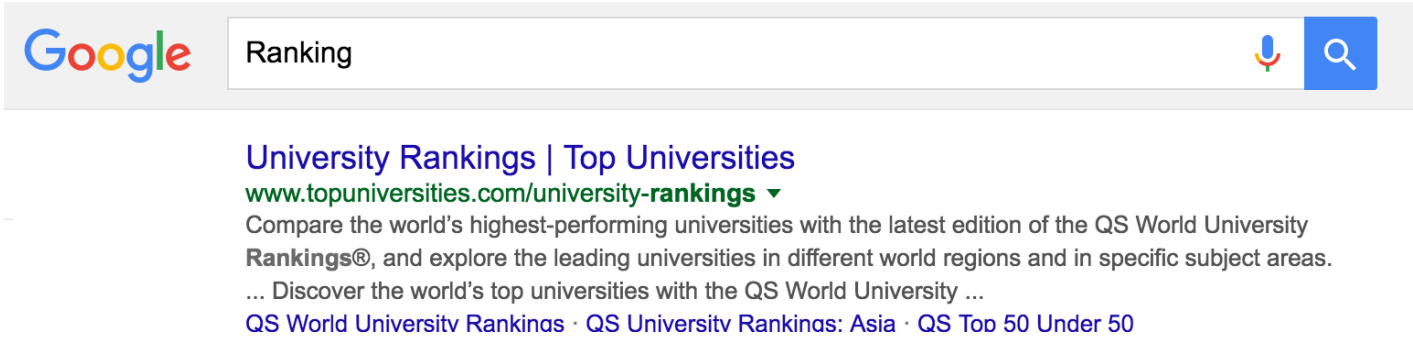


Figure: Google Search

Clustering

- Discover groups and structures (clusters) in the data.
- Learn clusters without using known structures in the data.
- Example:
  - Identify point-of-interests (special locations) to geo-located photos.
  - Photos in close locations (same cluster) represent the same POI.

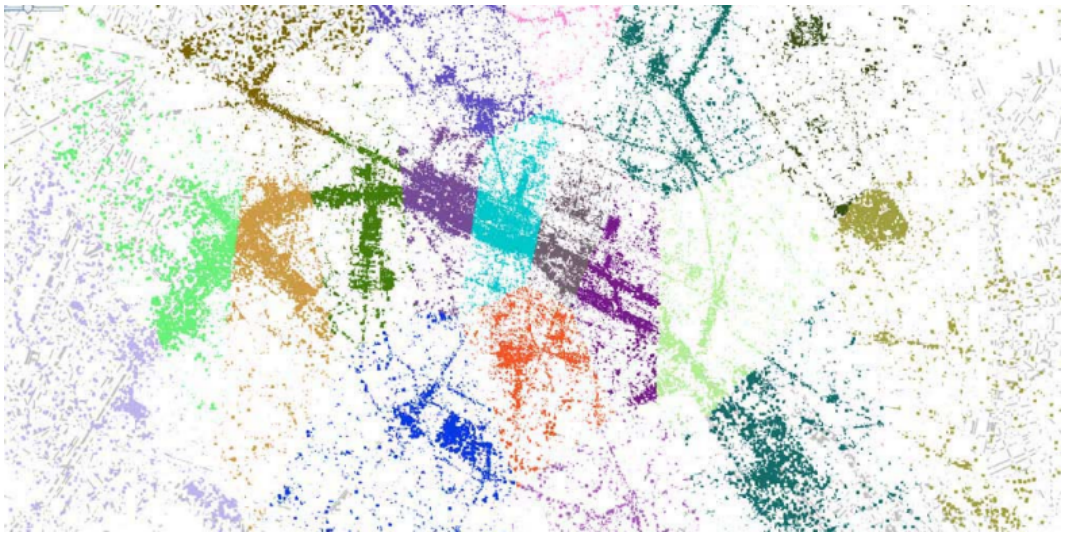


Figure: Identify POIs to Photos with Clustering [Yang et al., SIGIR 2011]

Summarization

- Provide a more compact representation of the data.
- Text – Document Summarization
- General Information – Information Visualization
- Example:
  - Summarize a webpage into a snippet.
  - The snippets might be short but compact.

Tom Bosley: Biography from Answers.com
Tom Bosley won a Tony Award in 1958 for his lead role as New York mayor Fiorello LaGuardia in the Broadway musical Fiorello!. But **Tom Bosley** is better remembered for his role on the long-running TV show Happy Days (1974-1984).
www.answers.com/topic/**tom-bosley**

(a) Baseline Snippet

Tom Bosley: Biography from Answers.com
Bosley died at 4:00 a.m. of heart failure on October 19, 2010, at a hospital near his home in Palm Springs, California. His agent, Sheryl Abrams, said **Bosley** had been battling lung cancer.
www.answers.com/topic/**tom-bosley**

(b) Temporal Snippet

Figure: Summarize a webpage into a snippet [Svore et al., SIGIR 2012]

Association Rule Learning

- Discover relationships between variables.
- The relationships might help other applications or analyses.
- Example 1: [Brin et al., SIGMOD 1997]
  - Frequent itemset in market basket data.
  - The tale of beer and diaper
- Example 2: [Chow et al., KDD 2008]
  - Detect privacy leaks on the internet.
  - Private identity and public information.

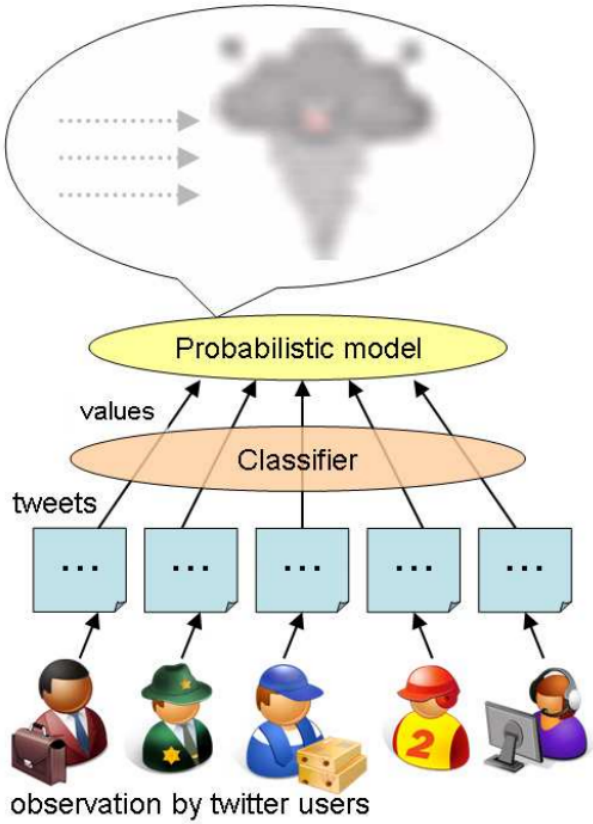
Photo Credit: Take Two

Combination of Multiple Tasks

- Decompose a problem into multiple tasks
- Example: Event detection with Twitter
  - Classification + Regression
  - Is a tweet talking the event?
  - Estimate the event probability

Benefits

- Divide and conquer the problem!
- Smaller tasks might be solved previously.



[Sakaki et al., WWW 2010]

Outline

From Machine Learning View

- ML algorithms can be well utilized to learn models.
- Information can be encoded as feature vectors.
  - i.e., training data while building models
- Feature selection and model choosing are important.

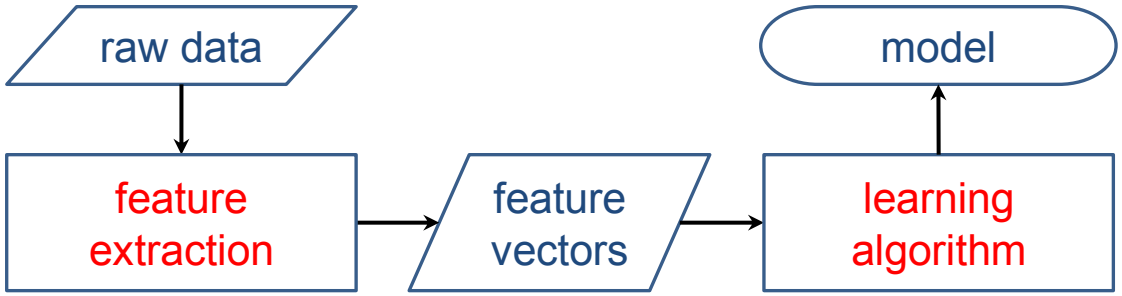


Figure: The illustration of utilizing ML algorithms to data mining.

Types of Machine Learning Models

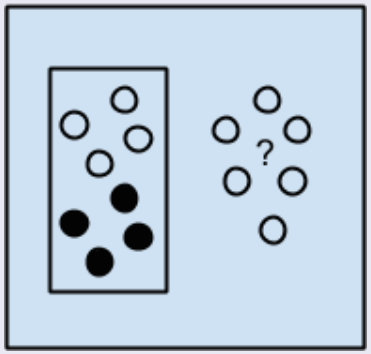
- With different data, different algorithms are suitable for use.
- It depends on data itself and application scenarios.

Supervised Learning v.s. Unsupervised Learning

Do the training data have the corresponding target labels?

Supervised Learning

- Learning with labeled data

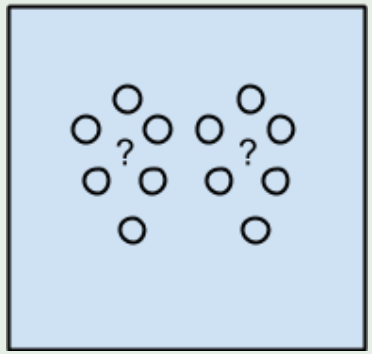


Credit: Machine Learning Masterery

- e.g., classification

Unsupervised Learning

- Learning without labeled data

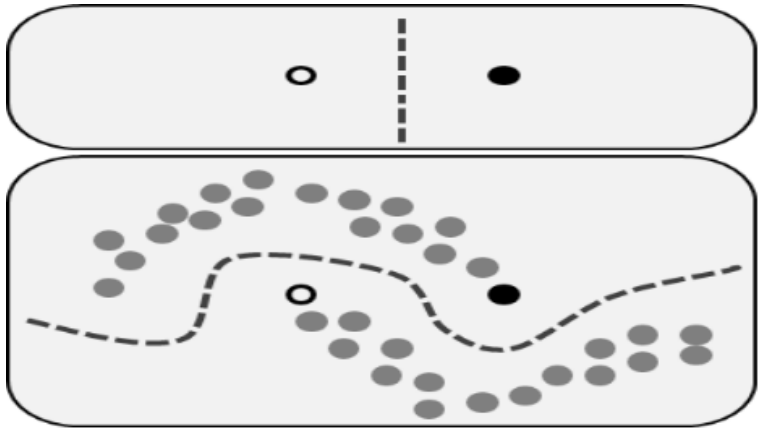


Credit: Machine Learning Masterery

- e.g., clustering

Semi-supervised Learning

- Learn with both labeled and unlabeled data
- Main idea – similar data might have similar labels.
- Example – Targeted Event Detection [Hua et al., KDD 2013]
  - Automatic labeling unlabeled data
  - Propagate labels to similar unlabeled tweets.

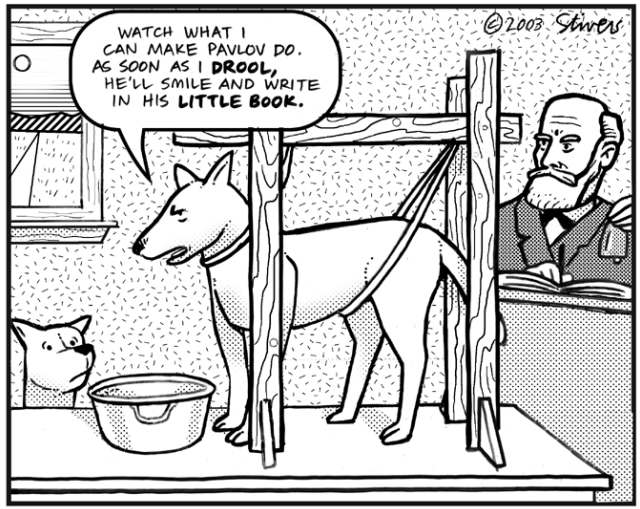


Credit: Wikipedia

Reinforcement Learning

No explicit labels, but implicit observations from environments

- Reward or punish with the observations
- Pavlov's conditioning experiment (1902)
- A different but natural way for learning



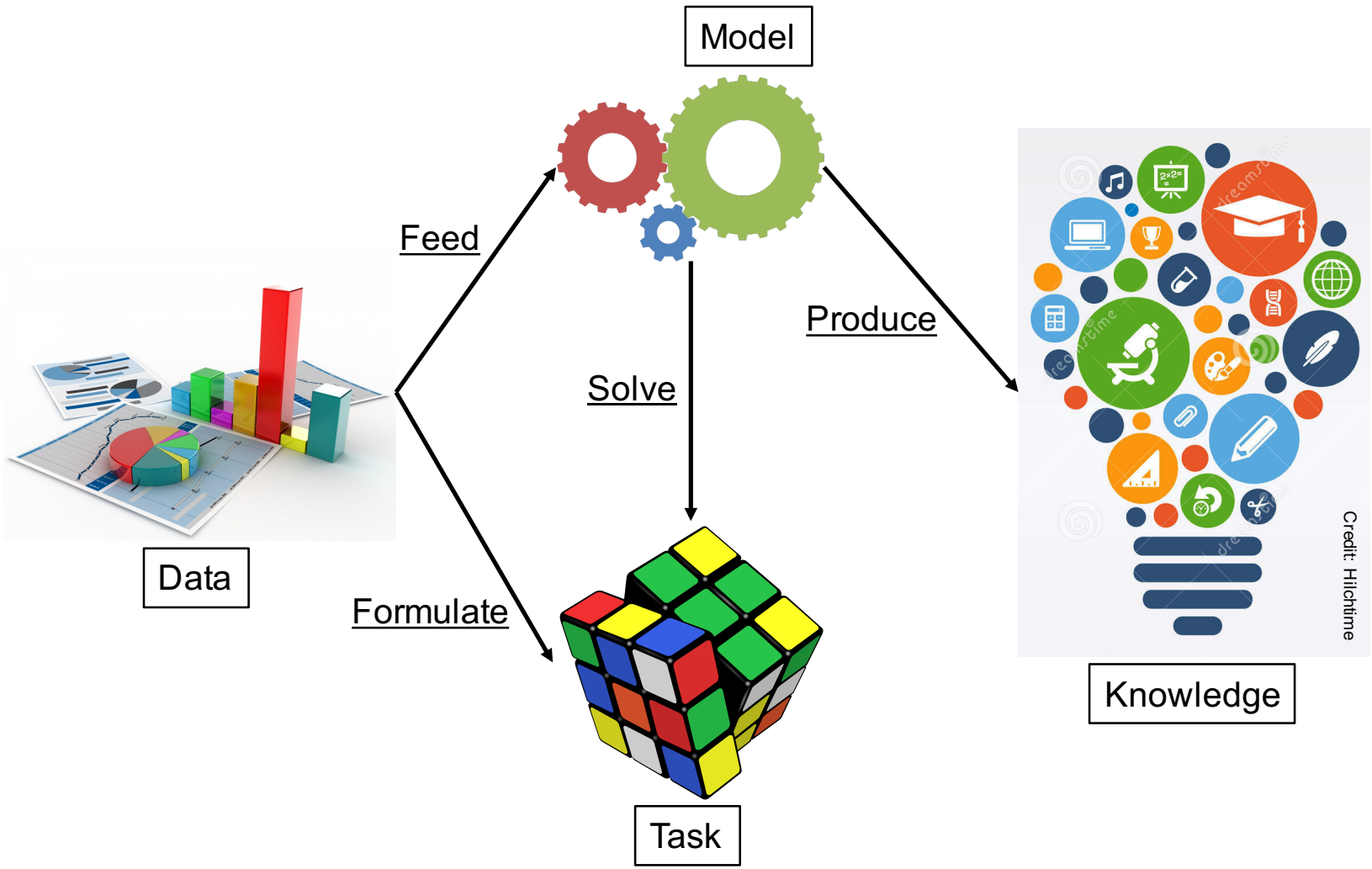
Credit: Mark Stivers

Reinforcement: learn with implicit feedback from environments (often sequentially)

- Example – Google AlphaGO [Silver et al., Nature 2016]
  - No label for each step
  - Implicit feedback of win and lose

Outline

Innovation: from Data to Task to Knowledge



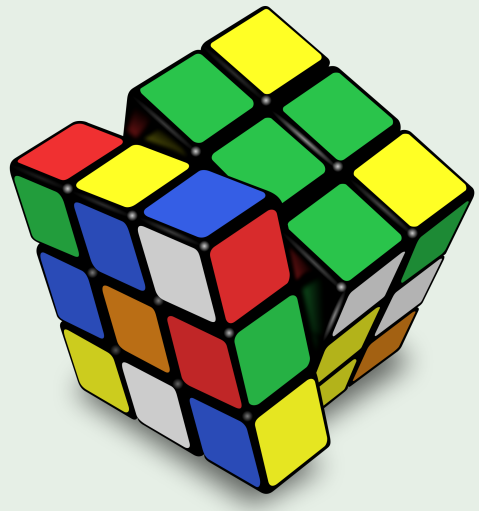
Two Key Foundations

Data

- Source of information

Task

- Problem to be solved



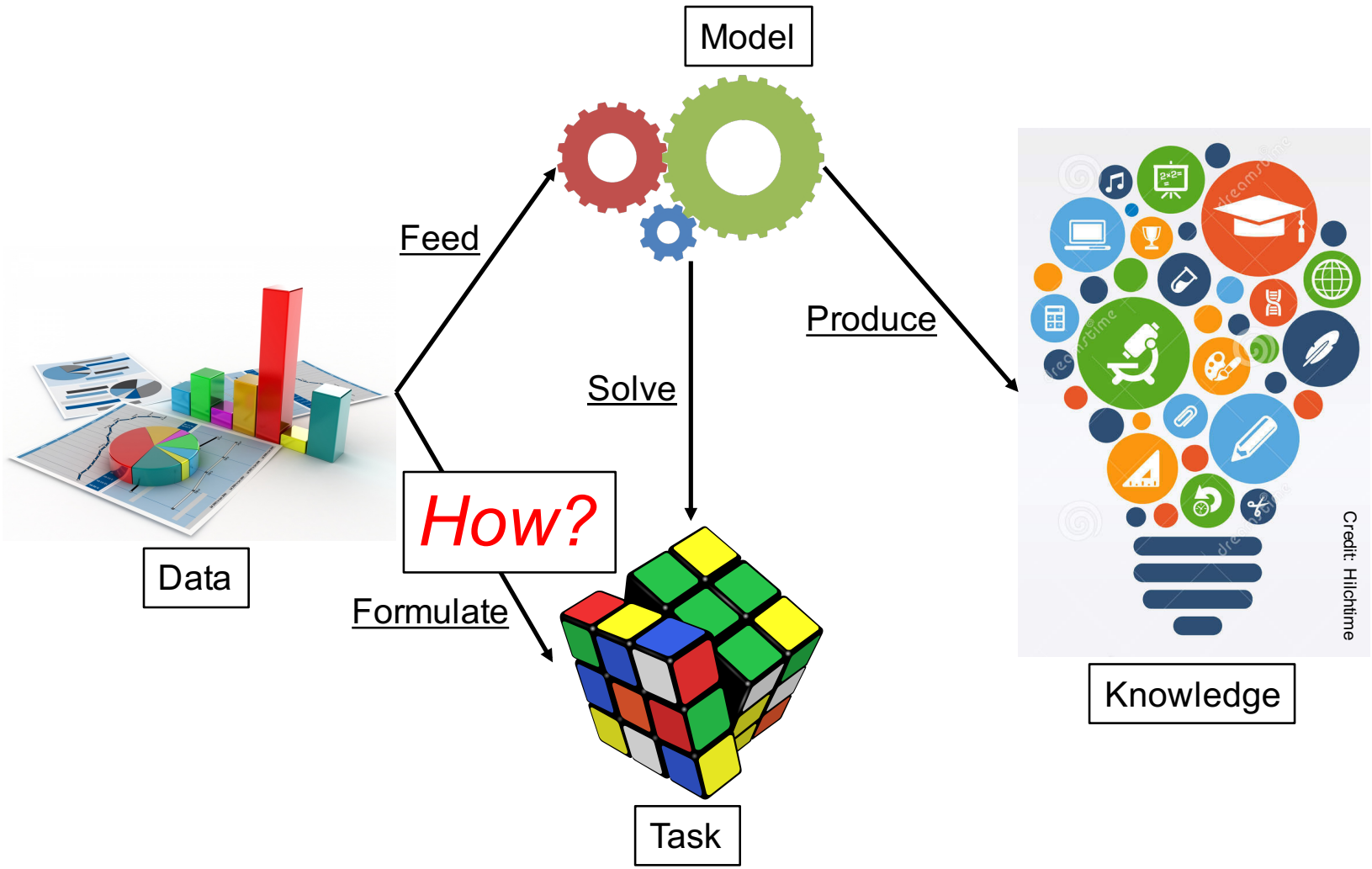
Then the model can solve the task with data and produce knowledge!

Where are data? Everywhere!!

- Social services (Facebook, Twitter, ...)
- Network (social networks, road networks, ...)
- Sensor (time-series, ...)
- Image (photos, fMRI, ...)
- Text (news, documents, ...)
- Web (forums, websites, ...)
- Public data (population, ubike logs, ...)
- Commercial data (transactions, customers, ...)
- ... and more!!

Credit: memegenerator.net

How to innovate a data mining application?



Data-driven Approach

Innovate the task from specific data

What can we do with this data?

For air quality data, we can...

- infer current quality [KDD '14]
- predict future quality [KDD '15]
- good station selection [KDD '15]
- ...

For social event data, we can...

- find potential guests [ICWSM '16]
- recommend events [ICDE '15]
- rank user influence [MDM '15]
- ...

Problem-driven Approach

Collect relevant data for a specific task

What data are helpful for solving this task?

Music Recommendation

- listening logs [Computer '09]
- music tags [WSDM '10]
- social network [WSDM '11]
- ...

Traffic Estimation

- meteorology [SIGSPATIAL '15]
- past traffic [TIST '13]
- social network [IJCAI '13]
- ...

Outline

Tools for Data Mining

Sometimes you don't need to reinvent the wheels!

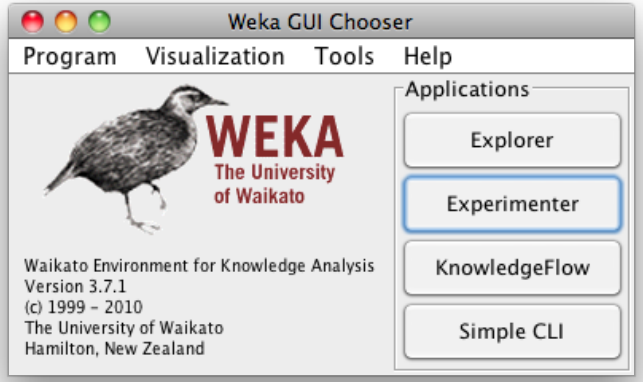
- Graphical User Interface (GUI)
  - Weka ···
- Command Line Interface (CLI)
  - LIBSVM & LIBLINEAR
  - RankLib
  - Mahout ···
- Callable Library
  - scikit-learn
  - XGBoost ···

Almost all methods introduced today can be covered by these toolkits.

Credit: Bill Proud

Weka

- Toolkit by University of Wakaito
- Implemented in Java
- `www.cs.waikato.ac.nz/ml/weka/`



Functions

- classification
- regression
- clustering
- association rule mining

Advantages

- various models
- user-friendly GUI
- easy visualization

Disadvantages

- slower speed
- parameter tuning
- large data in GUI
- confusing format

scikit-learn

- A Python machine learning library
- Python interface
- Core implemented by C (i.e., NumPy)
- `http://scikit-learn.org/`

Functions

- classification
- regression
- clustering
- dimension reduction

Advantages

- faster speed
- higher flexibility
- various models
- parameter tuning

Disadvantages

- parameter tuning
- self data-handling
- No GUI
- need to code

  [it might be good? :)]

LIBSVM & LIBLINEAR

- Toolkit by National Taiwan University
- Library for Support Vector Machine
- Support large-scale data (LIBLINEAR)
- Implemented in pure C
- `www.csie.ntu.edu.tw/~cjlin/libsvm/`

Functions

- classification
- regression

Advantages

- high efficiency
- pure CLI
- parameter tuning
- fixed data format

Disadvantages

- no GUI
- only SVM

RankLib

- Toolkit in Lemur project of CMU and UMass
- Library for ranking models
- Implemented in Java
- `www.lemurproject.org`

Functions

- Ranking

Advantages

- various models
- pure CLI
- parameter tuning
- fixed data format

Disadvantages

- no GUI
- only for ranking

Short Summary

In the lecture 1, you have learned ...

- The roadmap and tasks of data mining
- How machine learning be helpful
- How to innovate a project with data & tasks
- Some useful toolkits

Next: How to find the clues in the data?

Tea break and communication time!

Outline

Outline

Real-world Data is Dirty and Messy!

Credit: memegenerator.net

Product Review: Do I like iPhone?

"I bought an iPhone a few days ago. It is such a nice phone. The touch screen is really cool. The voice quality is clear too. It is much better than my old Blackberry, which was a terrible phone and so difficult to type with its tiny keys. However, my mother was mad with me as I did not tell her before I bought the phone. She also thought the phone was too expensive, ..." — An example from [Feldman, IJCAI '13]

Taxi Trajectory: Where is the destination?

ECML/PKDD Taxi Trajectory Prediction Competition

Labradoodle or Fried Chicken?

Deep Learning Training Set (`http://imgur.com/a/K4RWn`)

Sheepdog or Mop?

Deep Learning Training Set (`http://imgur.com/a/K4RWn`)

崩 ╰(〒皿〒)╯ 潰

Wait... Why can we recognize dogs and fried chicken?

Dogs...

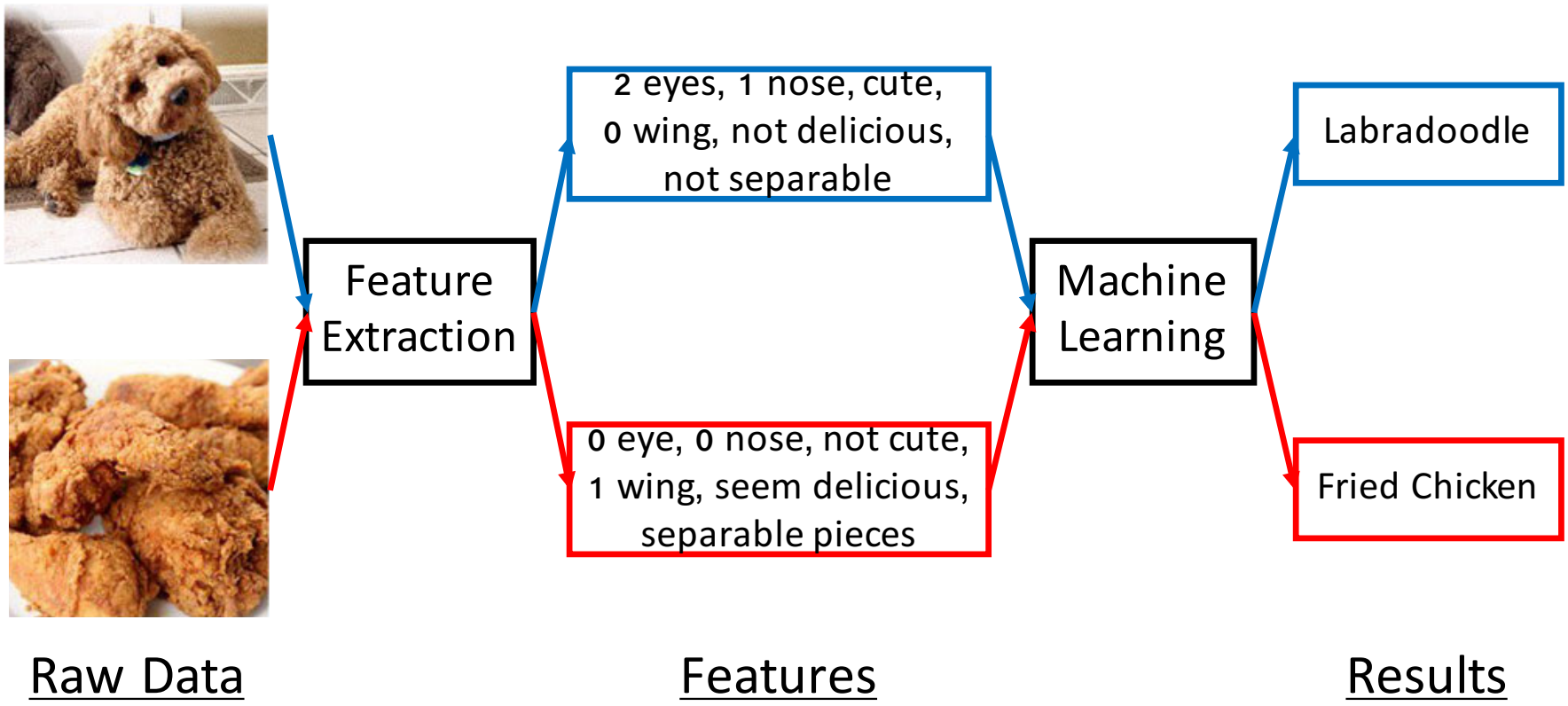
- have eyes, but fried chicken do not.
- have noses, but fried chicken do not.
- are cute, but fried chicken are not.

Fried chicken...

- seem delicious, but dogs do not.
- are separable pieces, but dogs are not.
- may have wings, but dogs have no wing.

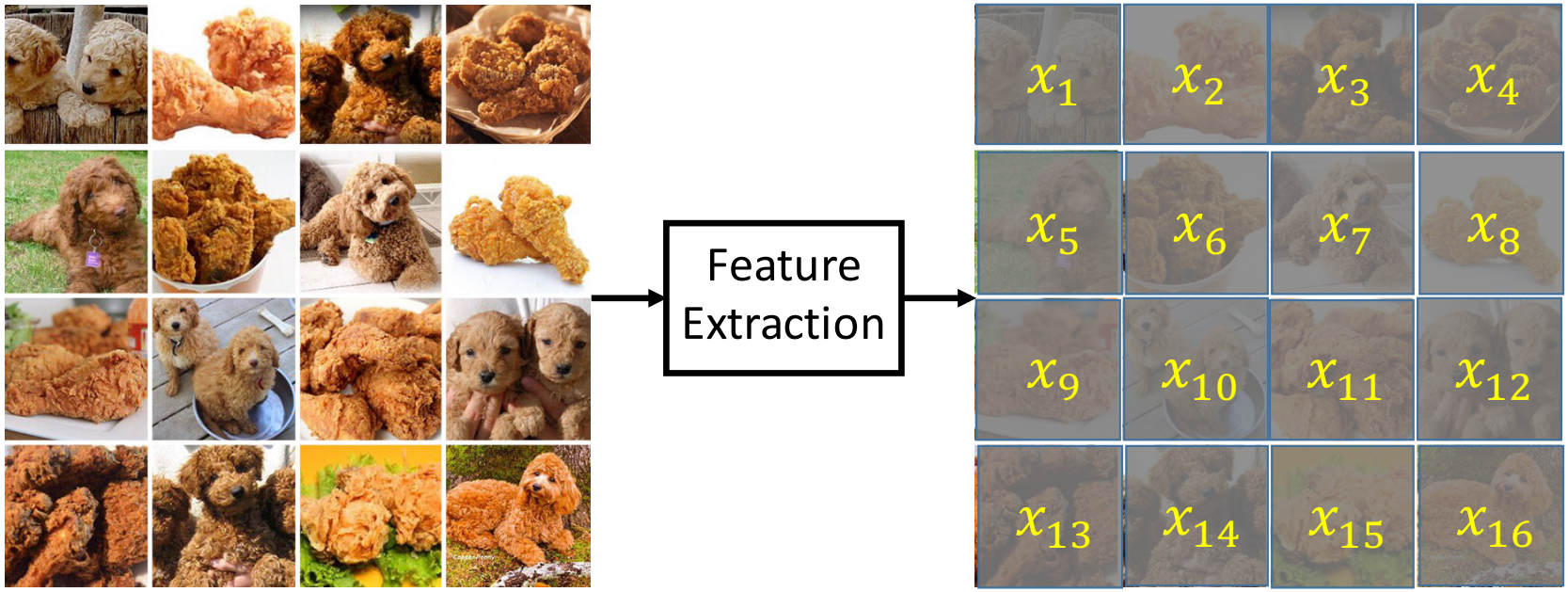
We observe important properties for making decisions!

Features: Representation of Properties



General Representation

- Raw data may not be comparable with each other.
  - meaningless and too rough information
  - different-size images and various-length documents
- Features are same-length vectors with meaningful information.



Outline

How to extract features?

Different data need different features.

They may be including but not limited to

- Categorical features
- Statistical features
- Text features
- Image features
- Signal features

Categorical Features

- Some information is categorical and not numerical.
  - blood type (A, B, AB, O)
  - current weather (sunny, raining, cloudy, ...)
  - country of birth (Taiwan, Japan, US, ...)
- Extend a n-categorical feature into n binary features

Blood Type

	A	B	AB	O
Type B =	0	1	0	0
Type O =	0	0	0	1

Country of Birth

	TW	JP	US	Other
Japan =	0	1	0	0
Other =	0	0	0	0/1

Statistical Features

- Encode numerous numerical values into a feature
  - e.g., revenues of citizens, traffic in a week
- Apply statistical measures to represent data characteristics.

Conventional Measures

- Min
- Max
- Mean value
- Median value
- Mode value
- Standard deviation

Mean vs. Median

Outliers affect statistical results a lot!

- Features for salaries of citizens
  - Data: 22000, 22000, 33000, 1000000

Mean

$$\frac{22k + 22k + 33k + 1000k}{4} = 269250$$

Median

$$\frac{22000 + 33000}{2} = 27500$$

Which is more representative for these salaries?

Statistical Features for Grouped Data

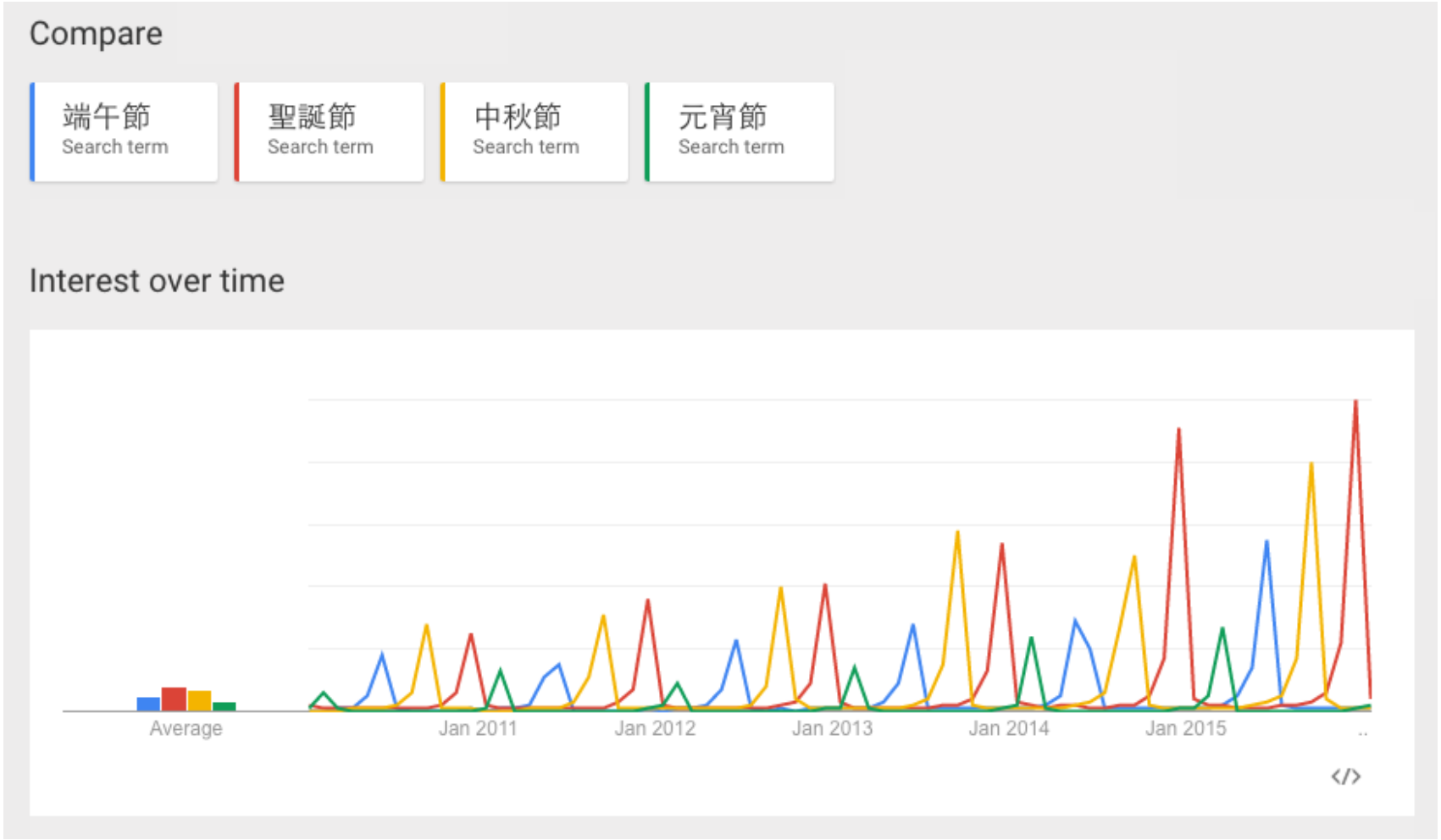
The single statistical measure may not be able to well represent data.

- Especially for non-uniform data
  - Data: 1, 1, 1000000, 1000000
- Separate data into n groups
  - n statistical features can be induced

Conventional Grouping Criteria

- Time (year, month, hour, ...)
- Region (country, state, ...)
- Demographics (age, gender, ...)
- ...

Case Study: Time-dependent Web Search



Google Trend from 2010-01 to 2016-01

Text Features: Text mining is difficult

Credit: Course by Alexander Gray

High Variety and Uncertainty

Text documents can be written in different languages, lengths, topics, ...

王建民沒登板 皇家客場輸費城人

（中央社記者曹宇帆洛杉磯3日專電）美國職棒大聯盟堪薩斯市皇家隊今天作客費城迎戰費城人，先發投手范杜拉投不滿3局就因腳踝受傷退場，之後中繼投手也壓不住費城人的攻勢，終場皇家2比7敗給費城人。

范杜拉（Yordano Ventura）1局下就被費城人捕手魯普（Cameron Rupp）重擊，3分全壘打使皇家陷入落後苦戰，3局下又被左外野手艾希（Cody Asche）追加陽春全壘打，皇家0比4落後。

Barack Obama @BarackObama · 7月1日
Chicago raising its minimum wage is another reminder: Congress needs to keep up and #RaiseTheWage for hard-working Americans nationwide.

飯田里穂 @rippialoha · 2小時
2ndシングル『片想い接近』もいよいよ発売まで

あと2日..！

Upper: Yahoo! News; Bottom: Twitter tweets

Alphabetic or Non-alphabetic: Text Segmentation

- Separated characters may be meaningless.
  - "科學" and "學科" have the same character set.

Text Segmentation or Tokenization

"資料科學" should be segmented into two words of "資料" and "科學"

Alphabetic Languages

- e.g., English and French
- much easier
- spaces and punctuations
- "data science"
  - "data" + "science"

Non-alphabetic Languages

- e.g., Chinese and Japanese
- more difficult
- "全台大停電"
  - "全台" + "大" + "停電"
  - "全" + "台大" + "停電"

Grammar Manners: Stemming and Lemmatization

- Grammar rules in alphabetic languages make data messy.
- Different words may have same or similar meanings.
  - "image", "imaging", "imaged", "imagination" and "images"
  - "bring" and "brought"; "fly" and "flies"
  - "be", "am", "is" and "are"

Stemming

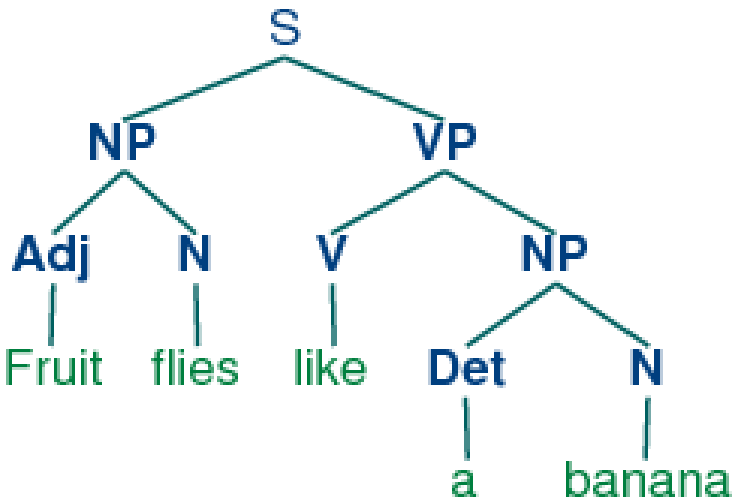
- remove inflectional endings
- "cats" → "cat"
- "image", "imaging" → "imag"

Lemmatization

- apply predefined lemmas
- "fly", "flies", "flied" → "fly"
- "am", "is", "are" → "be"

Grammar Manners: Part-of-Speech (POS) Tagging

- A word in different POS (詞性) can represent various meanings.
  - exploit: (N) 功績 (V) 利用
  - 落漆: (Adj) 遜掉了 (V) 牆壁油漆脫落
- Sometimes words need to be tagged for their POS.



An example from NLTK

NLTK: Toolkit for English Natural Language Processing

- Natural Language Toolkit
- Implemented in Python
- Python interface
- Much complete toolkit for NLP
- `http://www.nltk.org/`

Functions

- segmentation
- POS tagging
- dependency parsing
- ...

Advantages

- convenient interface
- complete functions
- lots of documents

Disadvantages

- low efficiency
- only for English

jieba (結巴): Toolkit for Chinese segmentation and tagging

- Open-sourced Chinese NLP toolkit
- Implementations in various platforms
- Mainly for segmentation and POS tagging
- `https://github.com/fxsjy/jieba`

The logo of go-implementation for jieba

Functions
- segmentation
- POS tagging
- simple analysis

Advantages
- various platforms
- for Chinese
- open source

Disadvantages
- less functions

Bag-of-Words (BoW): A simplifying representation

A bag of known words for representation

Example

- D1 = "I love data. You love data, too."
- D2 = "I also love machine learning."
- 8 BoW features from the corpus
  - I, love, data, you, too, also, machine, learning
- F1 = [1, 2, 2, 1, 1, 0, 0, 0]
- F2 = [1, 1, 0, 0, 0, 1, 1, 1]

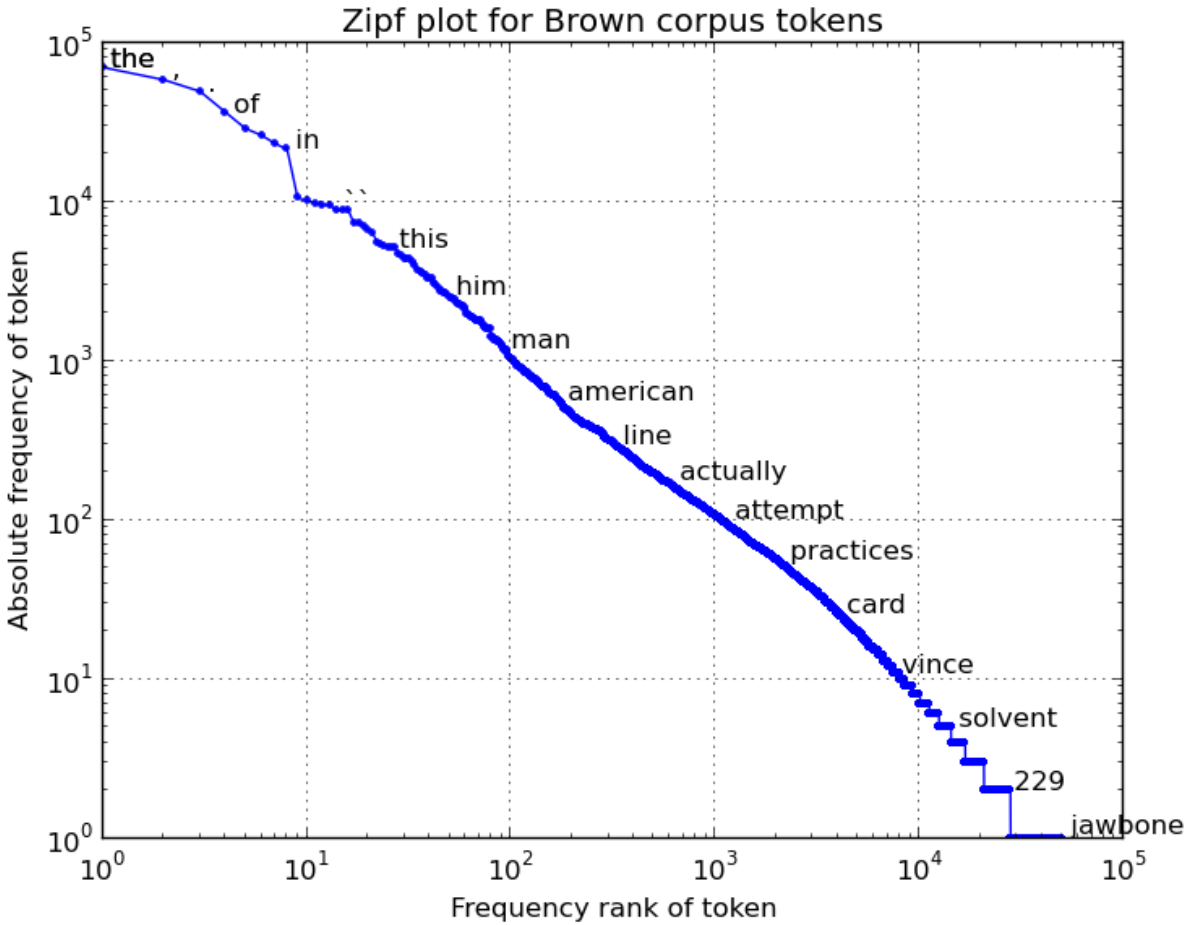
Problem

Greater term frequency, more representative?

An installation art in CMU

Zipf's Law [Zipf, 1949]

Words with high term frequencies may be just common terms!

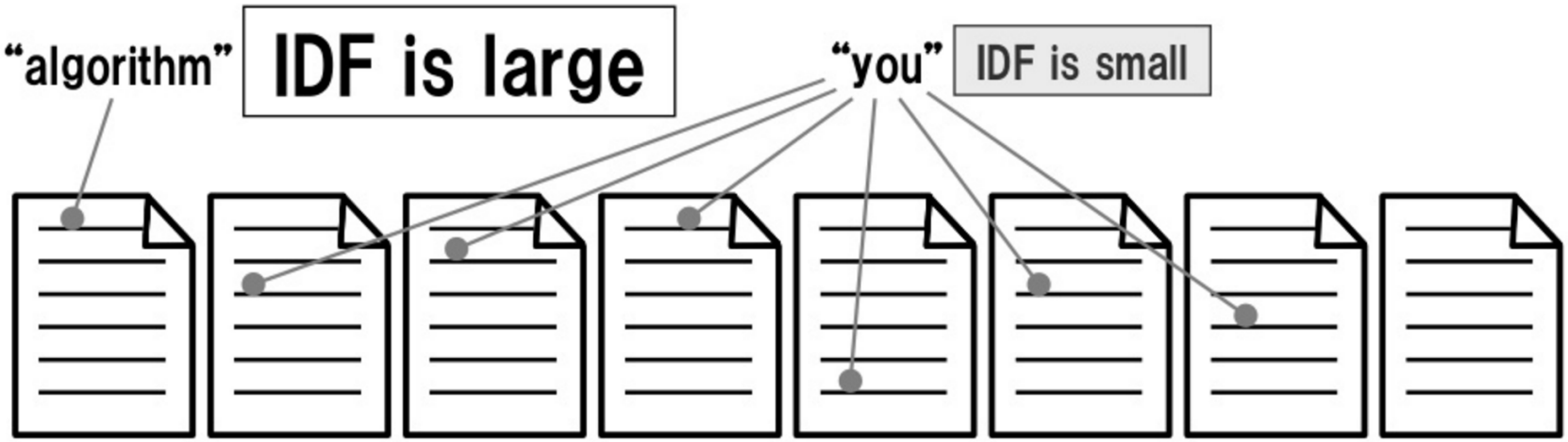


TF-IDF: Importance Estimation for a Word in a Document

Term Frequency (TF) $\times$ Inverse Document Frequency (IDF)

$$IDF(w) = \log \frac{|D|}{df(w)}$$

$|D|$: Number of all documents
$df(w)$: Number of documents with w



[Shirakawa et al., WWW 2015]

Meaningless Words: Stopwords

extremely frequent but meaningless words

English

a, the, and, to, be, at, ...

Chinese

的, 了, 嗎, 吧, 一下, 但是, ...

Problem: Is IDF still helpful?

Continuous Words: *n*-gram Features

Some words are meaningful only when they are observed together

Example

- "Micheal" and "Jordan"
  - "Micheal Jordan" has a special meaning.

Types

- Character-level *n*-gram
  - Obtain patterns in a word
  - prefix, suffix and root
- Word-level *n*-gram
  - Obtain information of word phrase

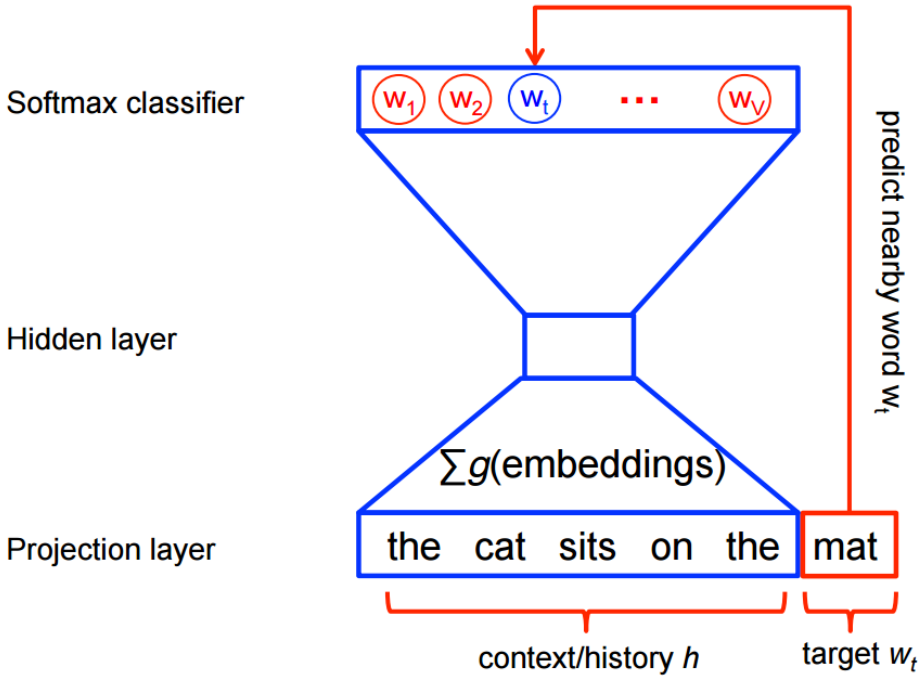
Usually combine with bag of ~~words~~ *n*-grams

Word2Vec: Deep Feature Extraction [Mikolov et al., NIPS 2013]

Compute embedding vectors in a hidden space for words

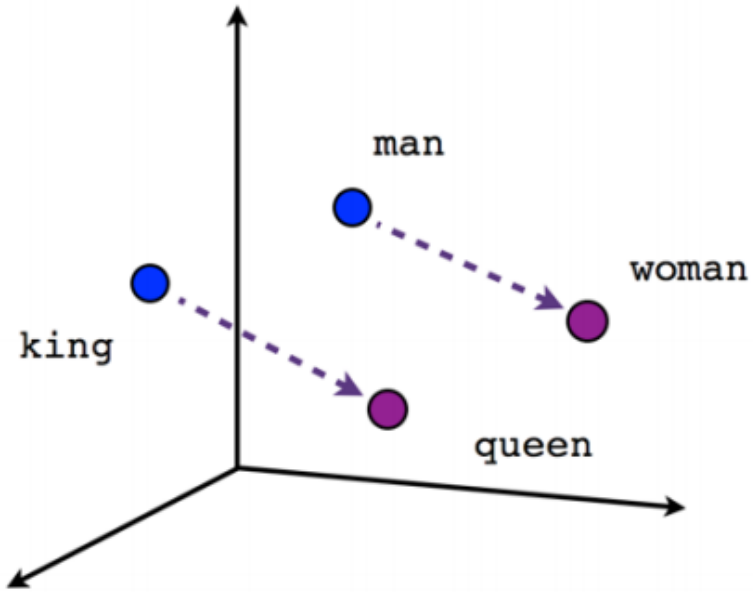
Deep Learning Approach

deep neural network to derive vectors



word2vec
Tool for computing continuous distributed representations of words.

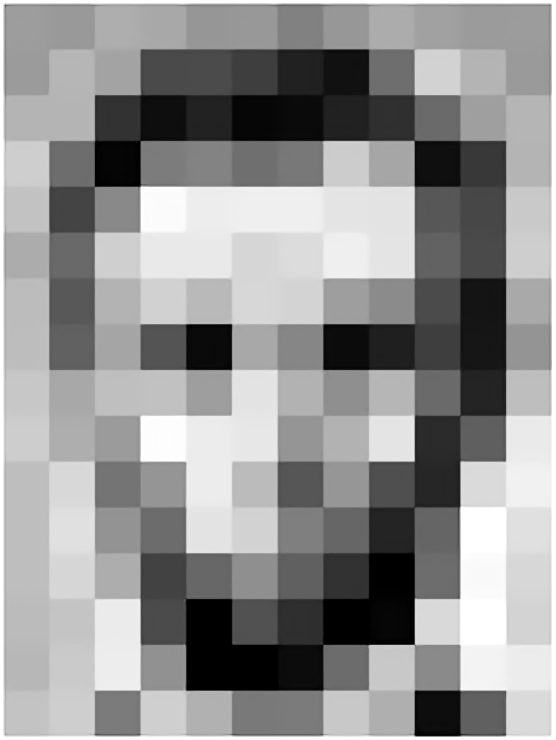
https://code.google.com/archive/p/word2vec/



An example from Tensorflow

Image Mining: Discovery from Pixels

An image consists of pixels (whose values can be features?!)



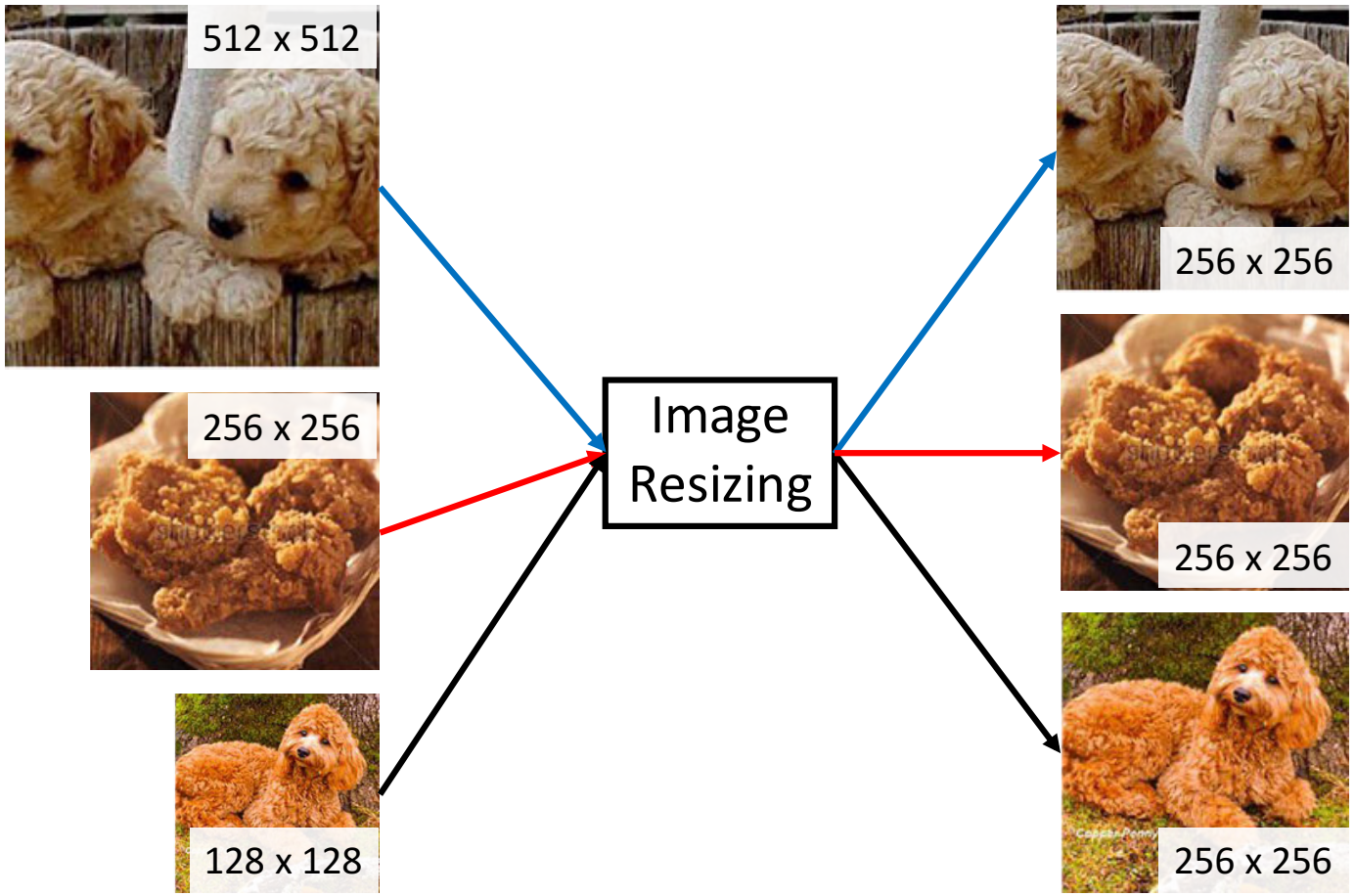
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

An example from openFrameworks

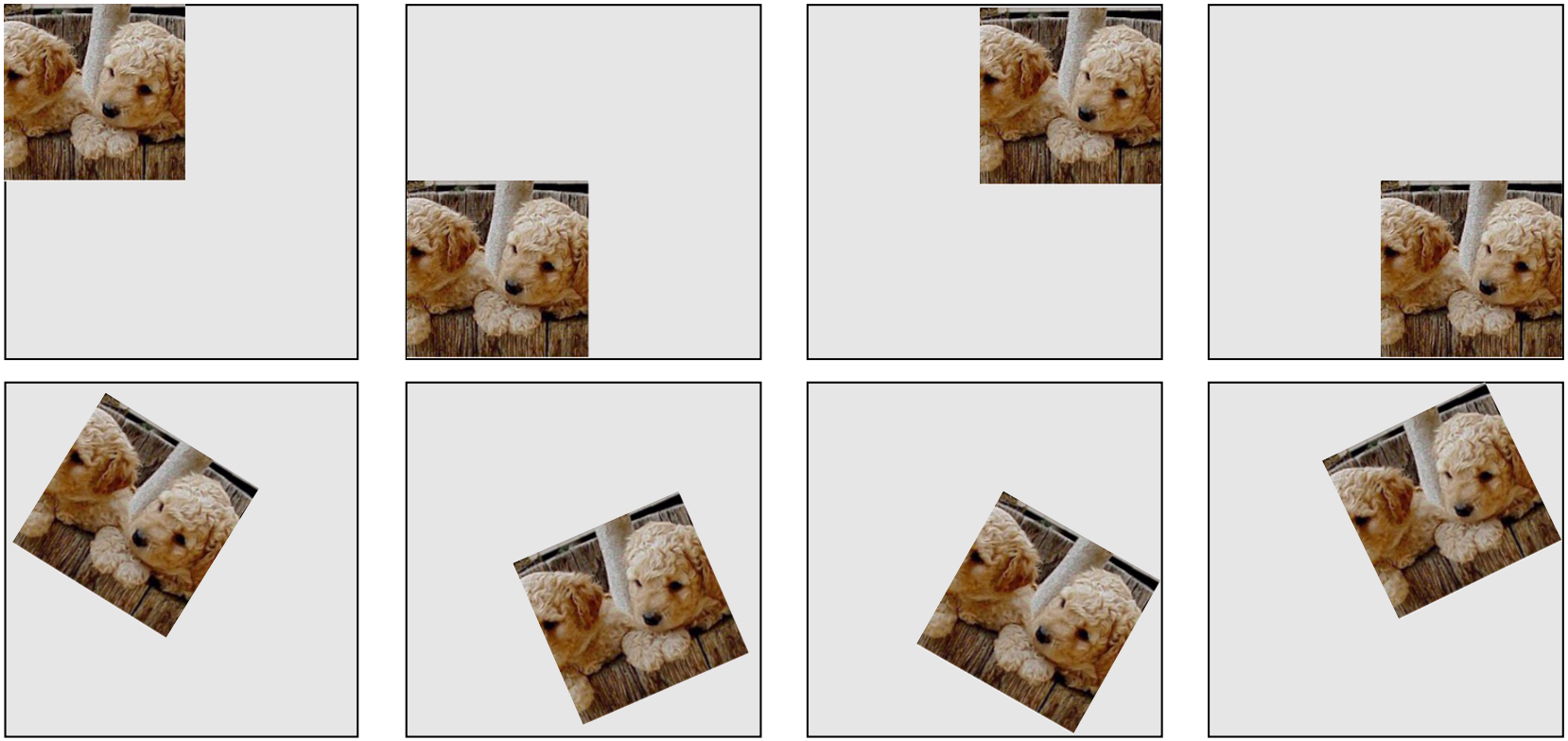
Resizing: Transform to an Identical Standard

Size of each image (i.e., # of pixels) may be different!



Pixel Positions are not Absolute!

Patterns are more important than values in specific positions.



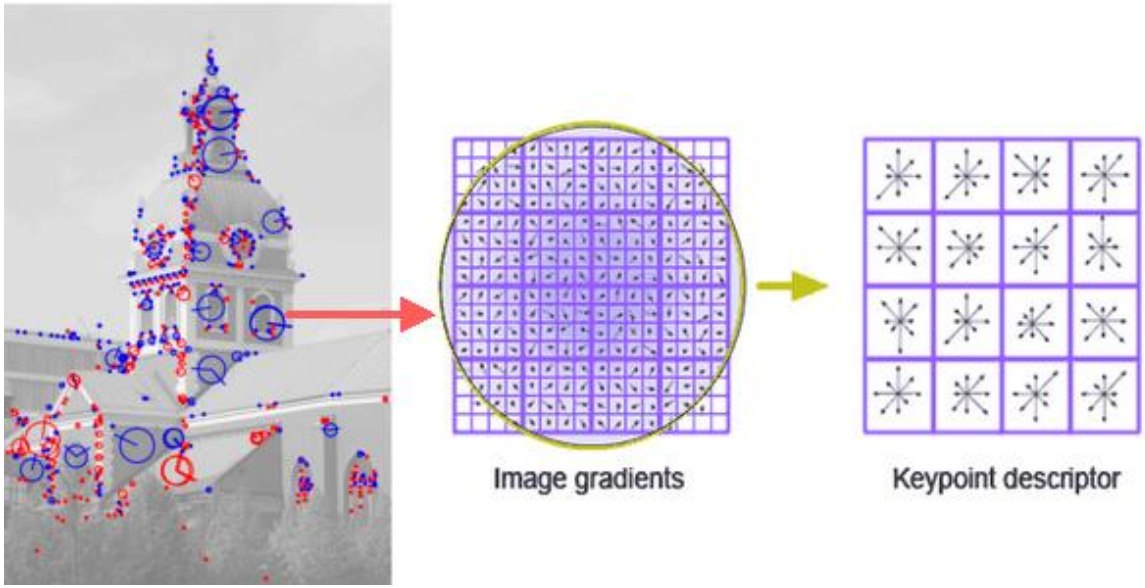
Feature vectors of only pixels can be much different!!

Scale-invariant Feature Transform (SIFT) [Lowe, ICCV '99]

Extract local keypoints (descriptors) as features

Invariant to

scaling, rotation and translation



http://www.codeproject.com/KB/recipes/619039/SIFT.JPG

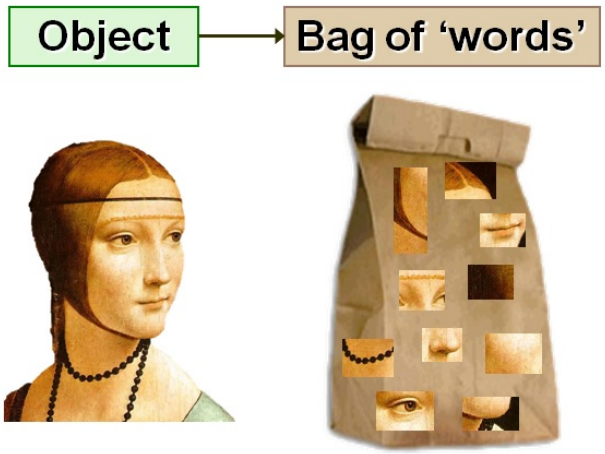
Bag of Visual Words

Apply bag-of-words to image mining

Visual Words

cluster local keypoints as words

TF-IDF works again!!



Credit: Gil Levi.

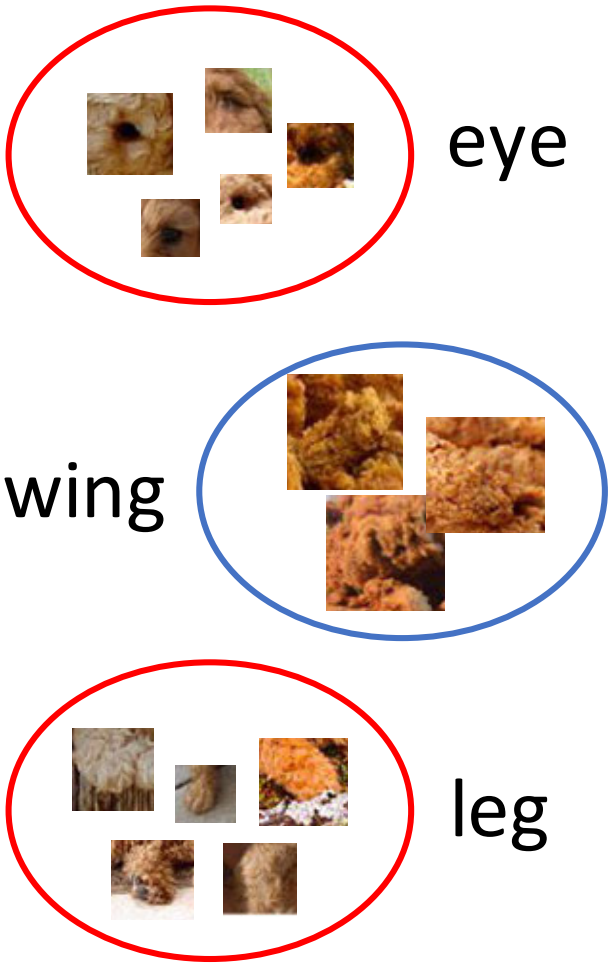
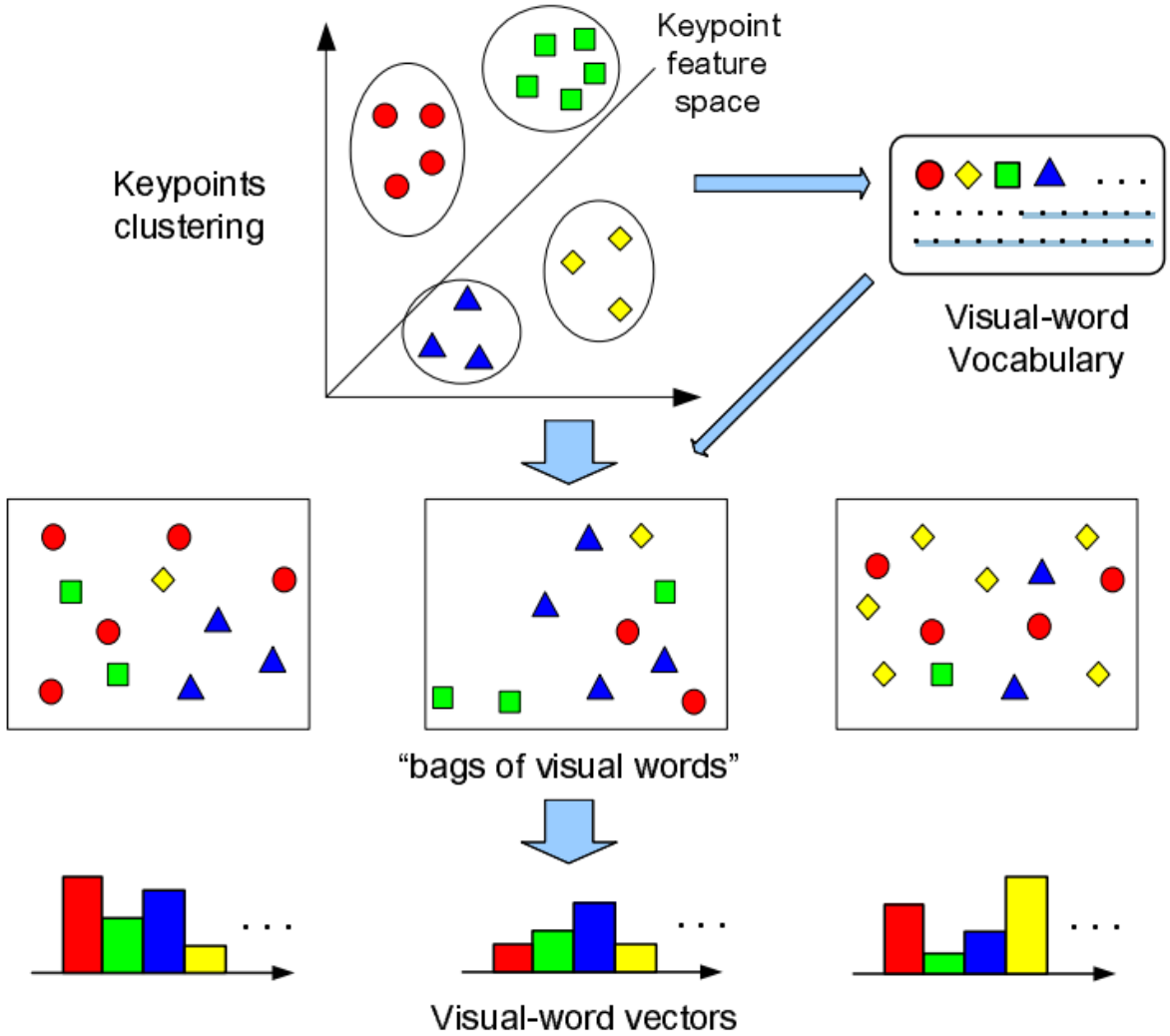
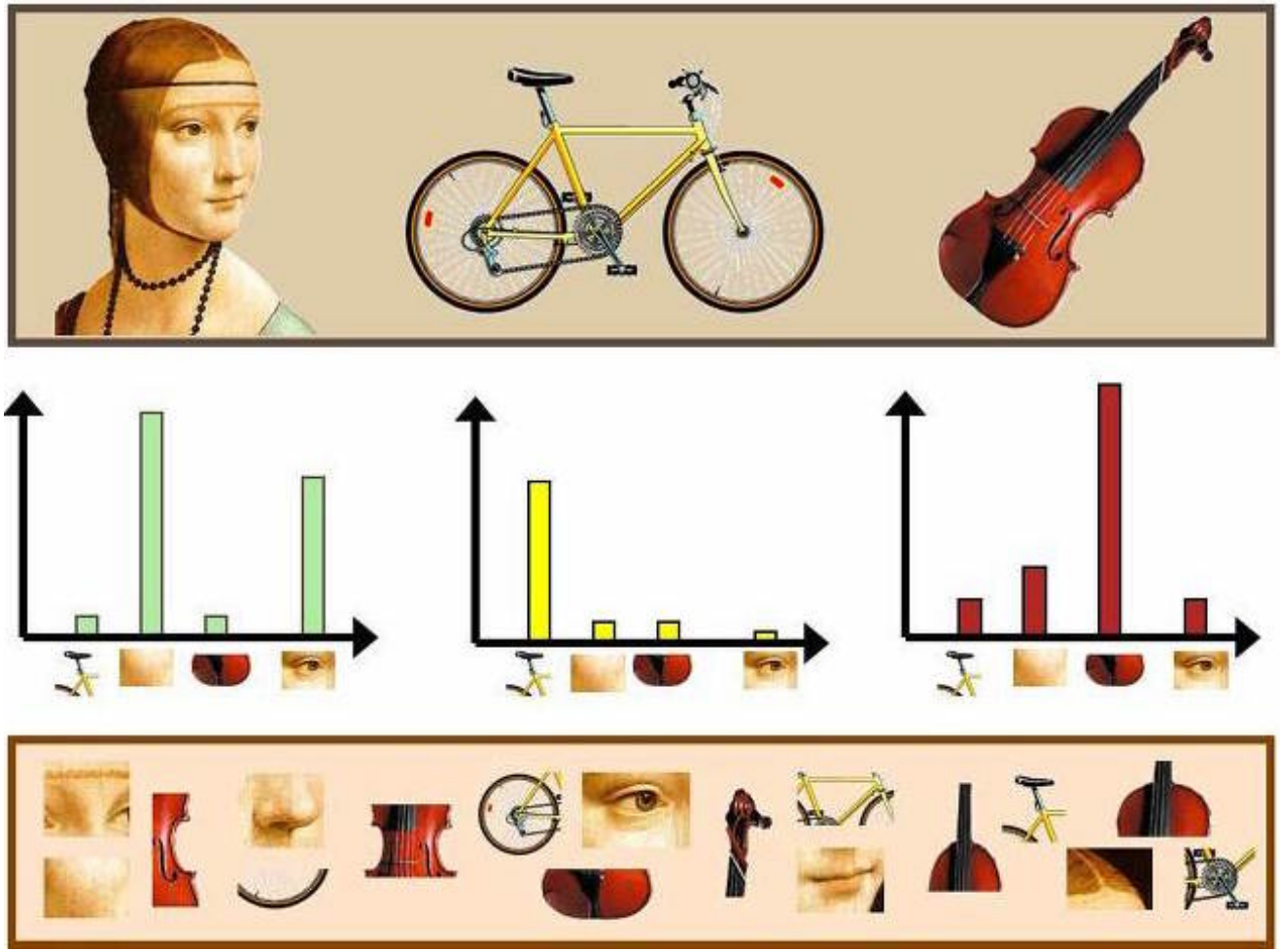


Illustration for Bag of Visual Words

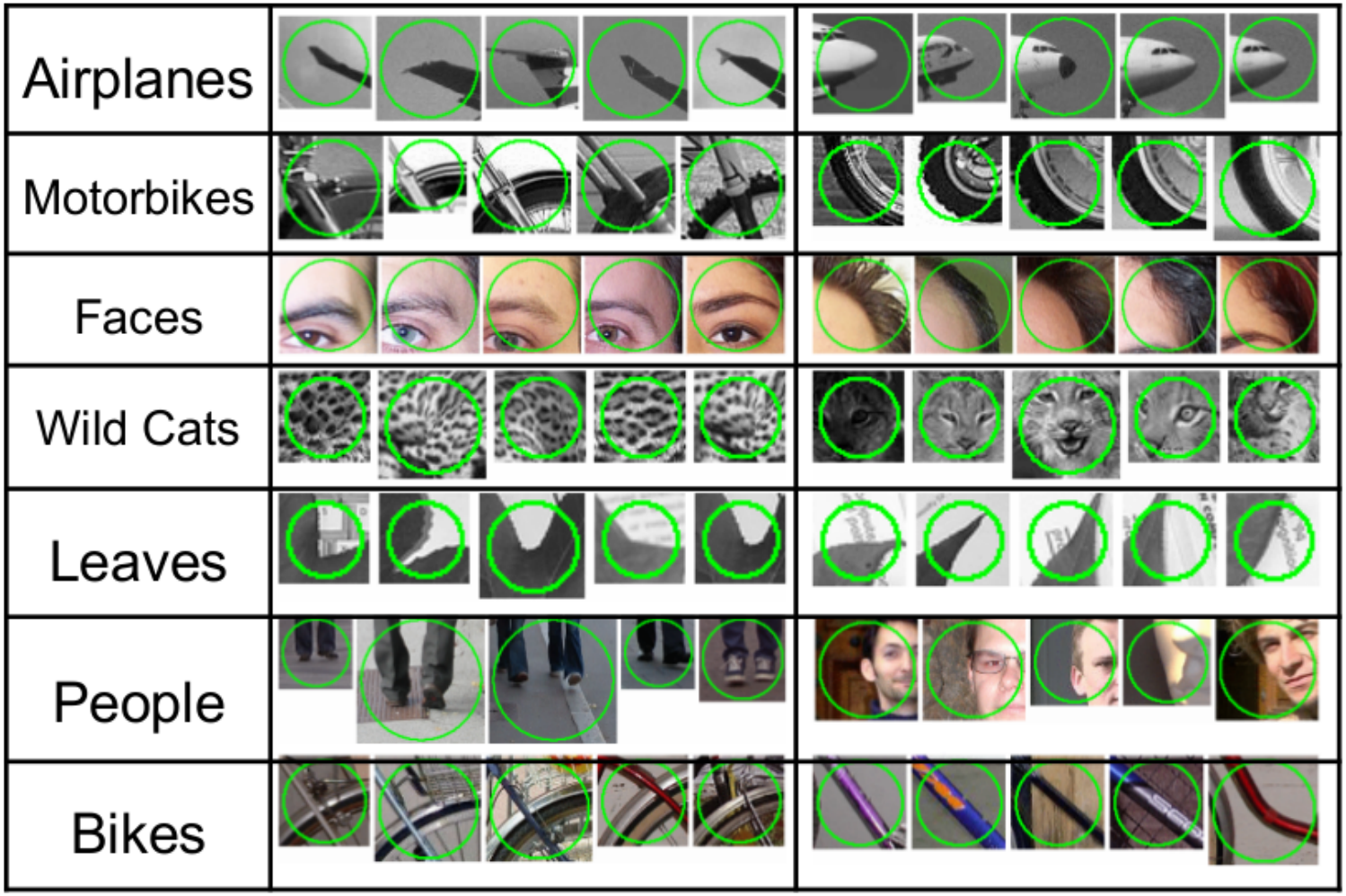


[Yang et al., MIR 2007]

Representation with Visual Words



More Visual Words



Convolutional Neural Network (CNN)

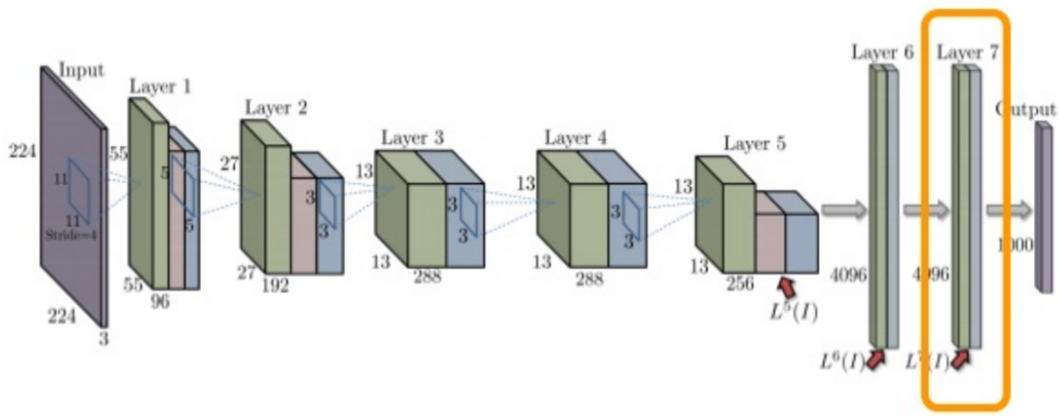
Deep features from multi-layer convolutional neural networks

Pre-train with other data

Apply the (last) hidden layers as features

Why it works?

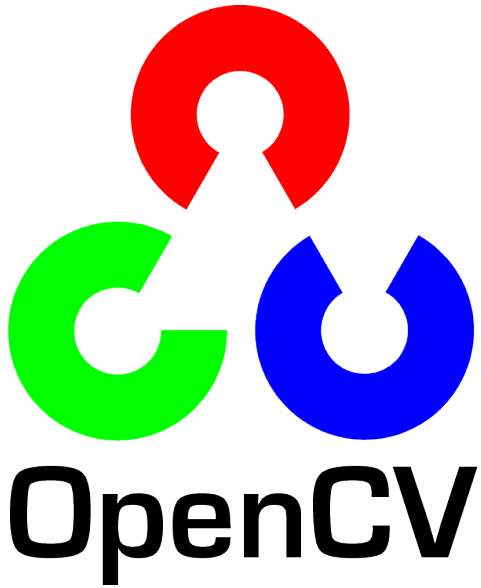
- Info from other data
- Learn local dependencies
- Layers with Non-linearity



[Jia et al., MM 2014]

OpenCV

- Open-sourced Computer Vision Library
- Cross-platforms and Various Interfaces
- Core implemented by C
- Update frequently (i.e., novel functions!!)
- `http://opencv.org/`



About Feature Extraction

- SIFT is a built-in function
- Able to link deep learning libraries for CNN

Deep Learning Libraries

- Tensorflow
  - Intuitive implementation
  - Slower Training Speed
  - Python Interface

- Theano
  - Longer Model Compilation Time
  - Faster Training Speed
  - Python Interface

theano

- Caffe
  - Designed for Computer Vision
  - Focus on CNN (effective in image processing)
  - Lots of pre-trained models (Caffe Model Zoo)
  - Various Interfaces

Caffe

Signal Data are Everywhere

The real world is full of signals!

Common Signal Data

- Sensor Data
- Musics
- Stocks
- Motions
- Images and Videos
- ... and more!

Problem

Most of them are variable-length time series data

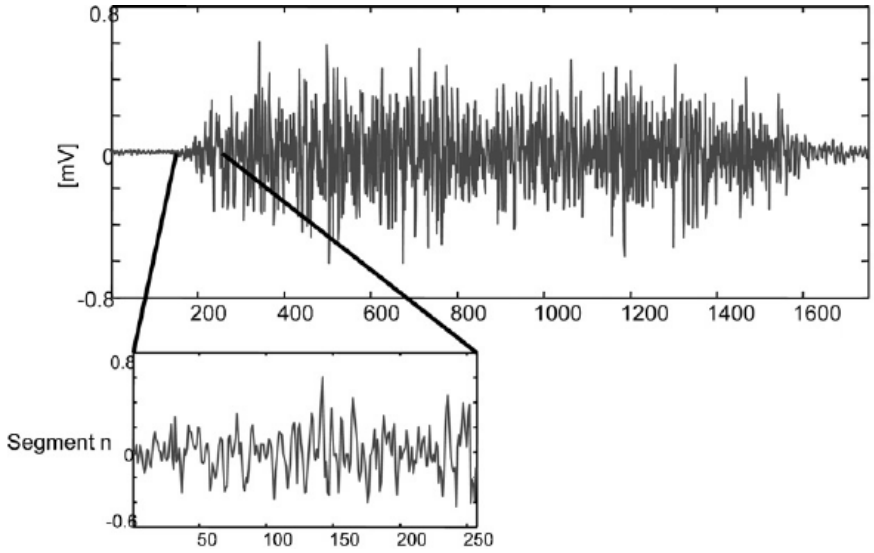
Sliding Window: Local Feature Extraction

Fixed-length Local Data form the **Interested Location**

- Raw signals as features
- Local dependencies
- Remove irrelevant signals

Problem

Local features cannot encode all data

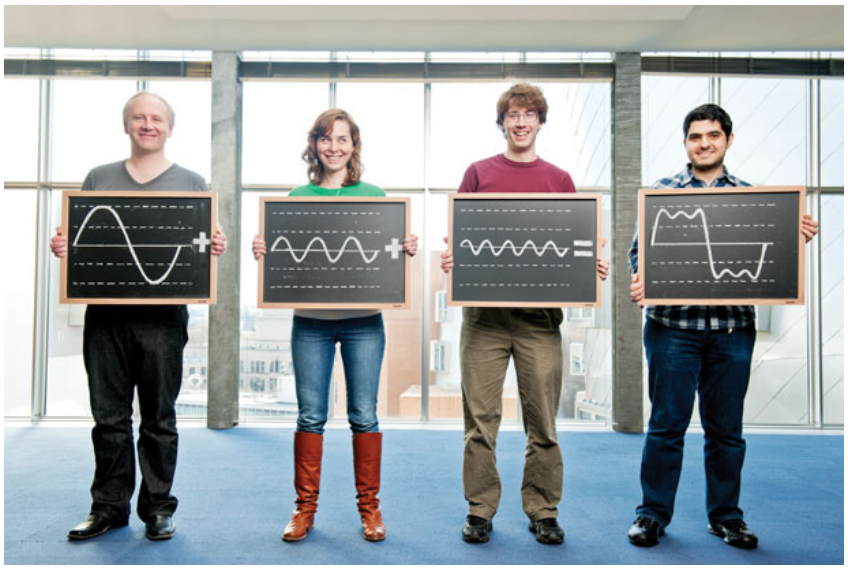


[Orosco et al., BSPC 2013]

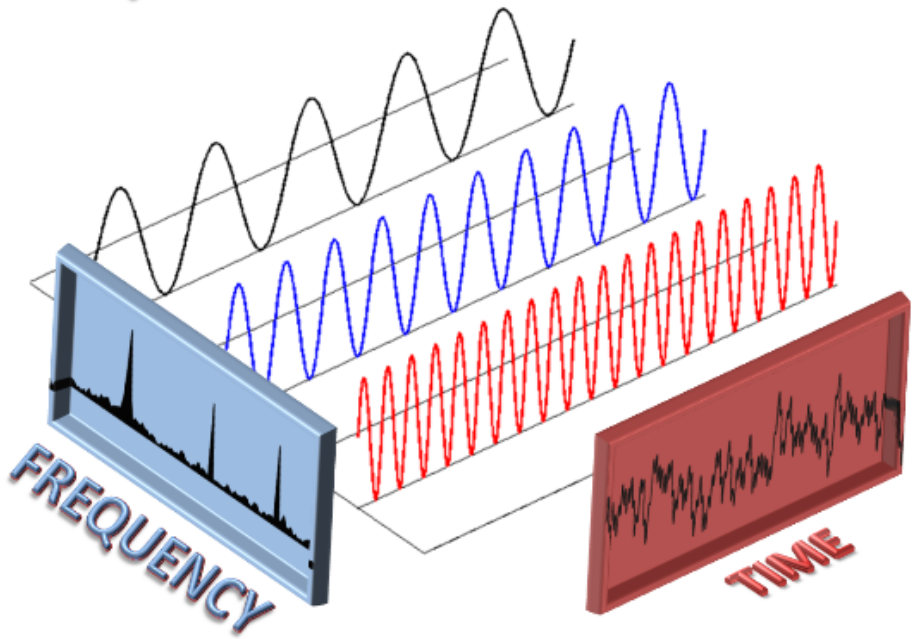
Fourier Transform: Observation from Other Aspect

Transform signals from time domain to frequency domain

- Decompose signal data into frequencies that make it up
- Properties of the whole data



Credit: MIT Technical Review



Outline

How to decide that features are good or bad?

Credit: `quickmeme.com`

Wrong features may lead to tragedies

Bad features cannot well represent the data!

Ridiculous, but fitting...

- Models still try to fit training data!
- But other data are not fitted!
  - Illusion of high performance

It is important to remove bad features!

Credit: ifunny.co

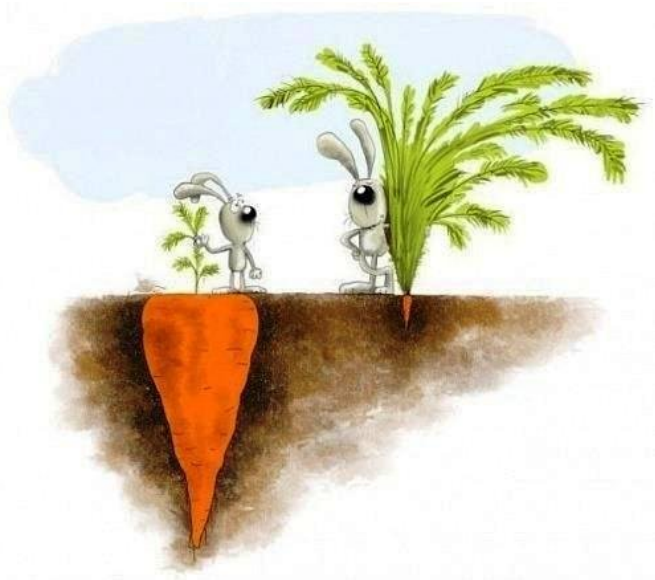
Good features lead to high performance

A Simple Criteria

select features with better performance!

Evaluation: Judge the Performance

- Don't evaluate results with eyes only
  - Illusions from partial data
- Compare with ground truths
- Utilize reasonable evaluation measures
- Different measures show performance in different sides



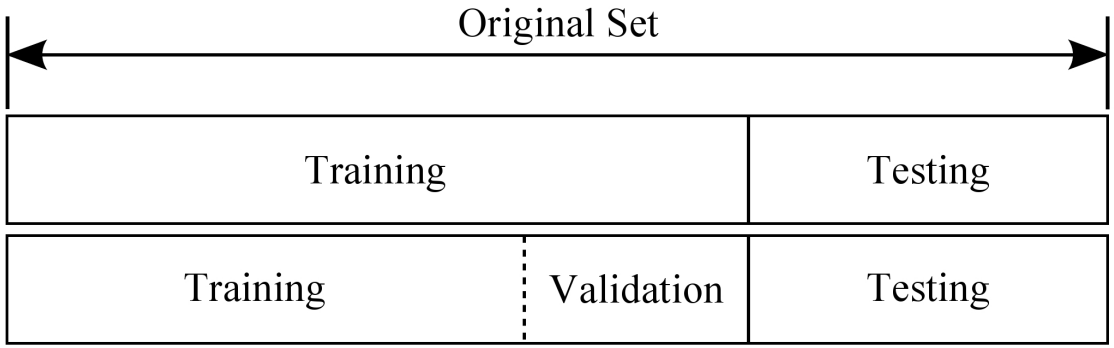
Problem

How to conduct a good evaluation?

Evaluation for Supervised Models

Data Preparation

- Data should be separated into three sets independent to each other
  - Training data: Learn the models
  - Validation data: Tune the parameters
  - Testing data: Judge the performance



Evaluation Measures in Classification

- Accuracy $= \frac{tp+fn}{tp+fp+fn+tn}$ (The ratio of correct predictions)
- Precision (for a class) $P = \frac{tp}{tp+fp}$
  - The ratio of correct predictions among positive prediction
- Recall (for a class) $R = \frac{tp}{tp+tn}$
  - The ratio of correct predictions among all such-class instances.
- F1-Score $F_1 = \frac{2 \cdot P \cdot R}{P+R}$
  - Consider precision and recall at the same time.

	Predicted = 1 (Positive)	Predicted = 0 (Negative)
Truth = 1 (True)	tp	tn
Truth = 0 (False)	fp	fn

Evaluation Measures in Regression

- Mean Absolute Error (MAE)
  - MAE measures how close the predictions f_i are to the ground truths y_i.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|f_i - y_i|$$

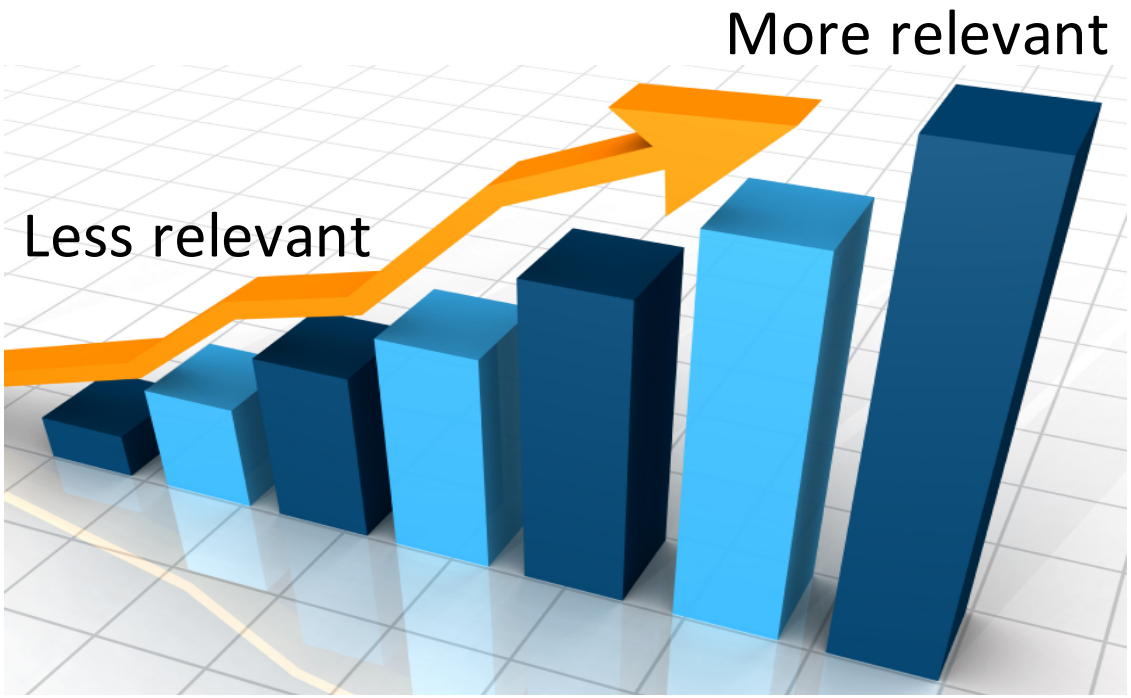
- Root-mean-square Error (RMSE)
  - RMSE amplifies and severely punishes large errors.

$$RMSE = \frac{1}{N}\sqrt{\frac{\sum_{i=1}^{N}(y_i - f_i)^2}{N}}$$

Evaluation Measures in Ranking

Key Problem of Ranking

- Assume instances have their relevance.
- Target – Ranking instances higher if they are more relevant.



Binary Relevance Ranking

Binary Relevance

Only two classes for instance (i.e., relevant and irrelevant)

- Mean Average Precision (MAP)

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{n_j} \sum_{i=1}^{n_j} P@R_{ji}$$

  - R_{ji} is the ranked position of i-th relevant instance
  - Consider the position of each relevant instance.

Binary Relevance Ranking (Cont'd)

- Mean Reciprocal Rank (MRR)

$$MRR(Q) = \frac{1}{|Q|} \sum_{i=1} \frac{1}{R_i}$$

  - Consider precision at the 1-st relevant instance
  - R_i is the position of the first relevant document in the i-th ranked list.
- Area under the Curve (AUC)
  - Pair-wise ranking errors between relevant and irrelevant instances
- Precision at k ($P@k$)
  - Precision of top k ranked instances

Graded Relevance Ranking

Graded Relevance

Relevance can be multi-scaled (e.g., scores 1 to 5)

- Normalized Discounted Cumulated Gain ($NDCG$)
  - Discounted Cumulative Gain (DCG)
    - $DCG = r_1 + \sum_{i=2}^{n} \frac{r_i}{\log_2 r_i}$
  - Ideal Discounted Cumulative Gain ($IDCG$)
    - Assume an ideal ranking R_i
    - $IDCG = r_1 + \sum_{i=2}^{n} \frac{R_i}{\log_2 R_i}$
  - $NDCG = \frac{DCG}{IDCG}$
  - $NDCG@k$ considers only top-k ranked instances.

Evaluation for Human Labeling

- Sometimes the ground truth may not be obtained.
- To obtain the ground truth, it needs human labeling.
- Each instance is labeled by at least two people.
- The process of labeling also has to be judged.

Cohen's Kappa Coefficient (κ)

Kappa evaluates the consistency of results labeled by two assessors.

$$\kappa = \frac{P(a) - P(e)}{1 - P(e)}$$

- $P(a)$ is the relative observed agreement among assessors.
- $P(e)$ is the hypothetical probability of chance agreement.
- $\kappa > 0.8$: good, $0.67 \leq \kappa \leq 0.8$: fair, $\kappa < 0.67$: dubious

Example

- $P(a) = \frac{20+15}{50} = 0.70$
- $P(e) = \frac{25}{50} \cdot \frac{30}{50} + \frac{25}{50} \cdot \frac{20}{50} = 0.5 \cdot 0.6 + 0.5 \cdot 0.4 = 0.3 + 0.2 = 0.5$
- $\kappa = \frac{0.7-0.5}{1-0.5} = 0.4$

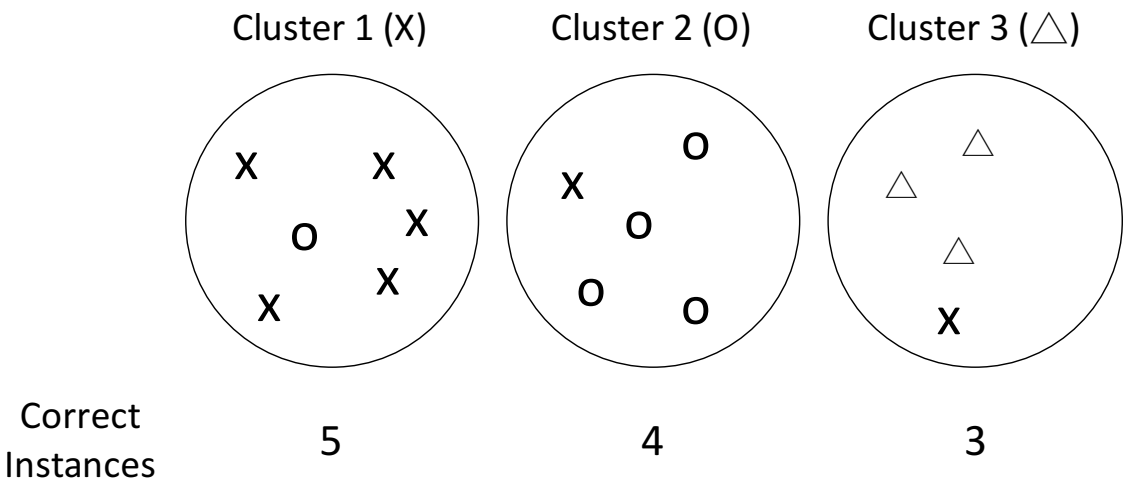
		B	
		Yes	No
A	Yes	20	5
	No	10	15

Evaluation Measures in Clustering

Clustering utilizes the corresponding classes C as ground truths.

- Purity
  - Each cluster $w_i \in \Omega$ is assigned to the most frequent class in the cluster
  - Compute accuracy by counting the correctly assigned instances

$$\text{purity}(\Omega, C) = \frac{1}{N} \sum_{w_k \in \Omega} \max_{c_j \in C} |w_k \cap c_j|$$



Evaluation Measures in Clustering (Cont'd)

Purity will be unsuitable when the number of clusters is large.

- Normalized Mutual Information (NMI)

$$\text{NMI}(\Omega, C) = \frac{I(\Omega, C)}{\frac{1}{2}(H(\Omega) + H(C))}$$

  - $I(\Omega, C)$ is the mutual information
  - $H(\cdot)$ is the entropy

Short Summary

In the former of lecture 2, you have learned ...

- The importance of features in data mining
- How to extract features from different sources
- Some useful toolkits for feature extraction
- How to evaluate the performance of models

Next: How to select good features from numerous features?

Tea break and communication time!

Outline

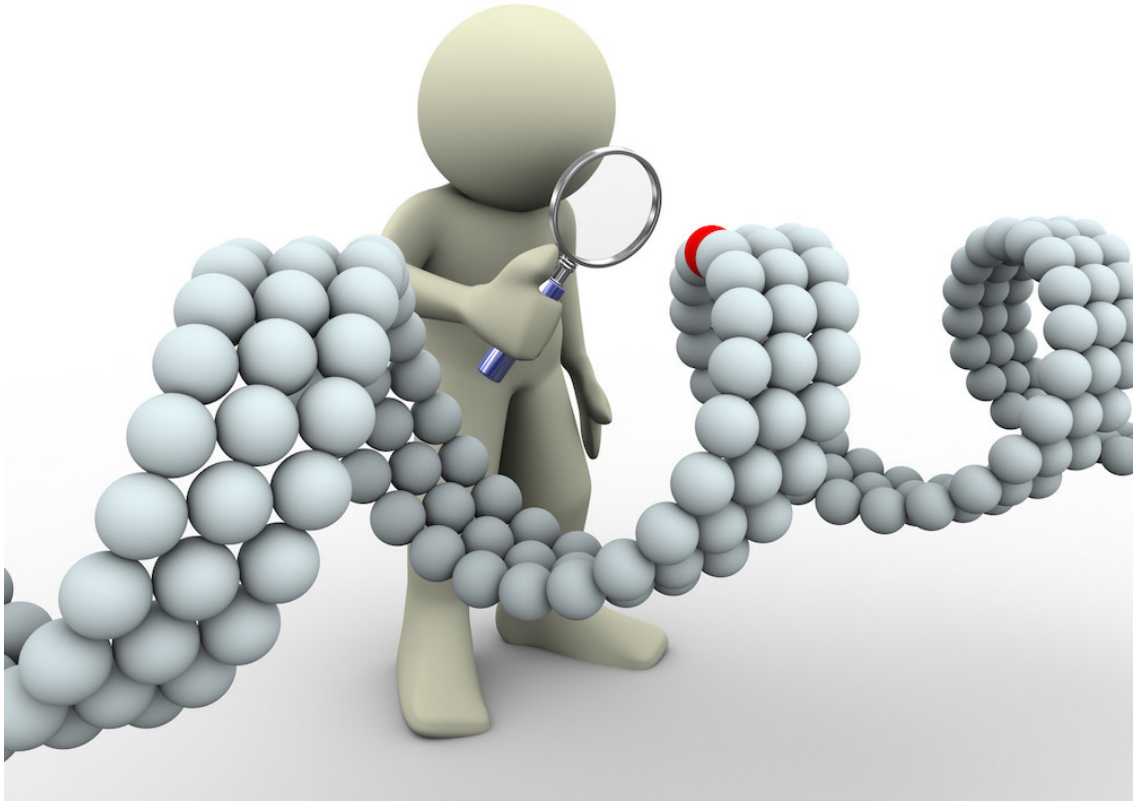
Features can be numerous

Feature selection should be systematic and automatic!

Credit: Selene Thought

Feature analysis is also important

To understand why the model works!



Credit: nasirkhan

Wrapper Method

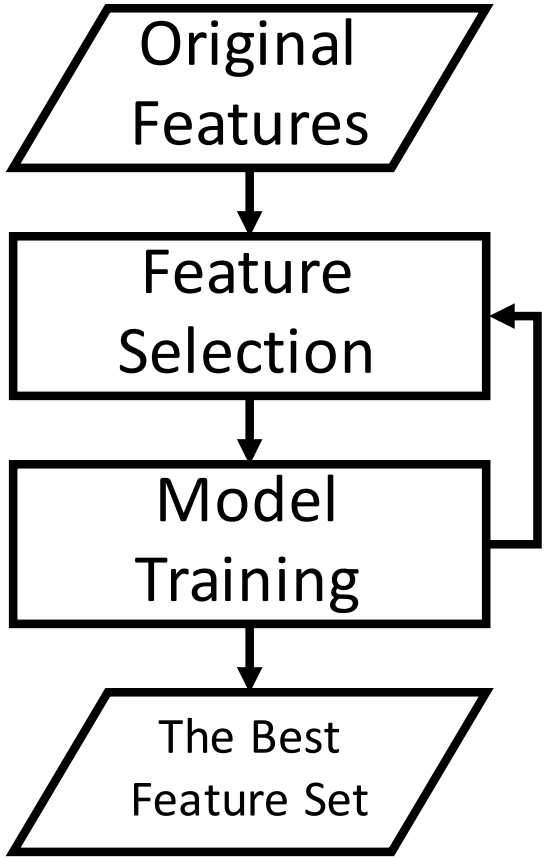
Try all available feature subsets to find the best one

Model-dependent Approach

- Evaluate "好度" of each subset directly
- Feedback from the model
- Can intuitively find the best feature set

Limitations

- Total 2^N combinations!
- High computational complexity



Leave-one-out Analysis

A variation of wrapper method

Discard each feature separately for evaluating the feature

Leave-one-out Approach

- Performance loss = Feature importance
- Negative loss (扯後腿) → Bad feature

Feature Analysis

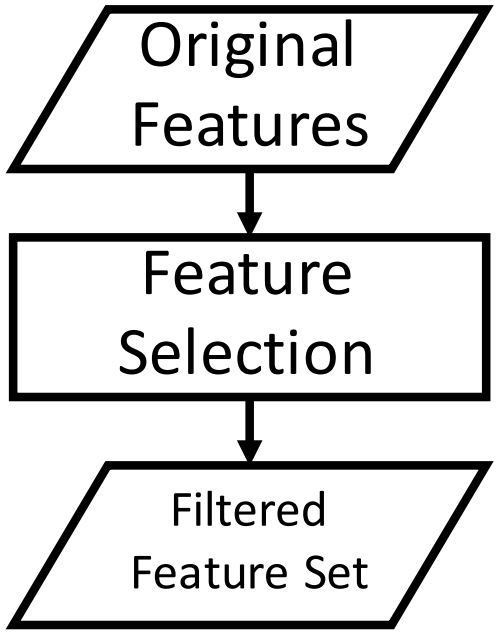
- Evaluate importance of each feature
- Rank features for feature analysis

Filter Method

Select features before applying the model

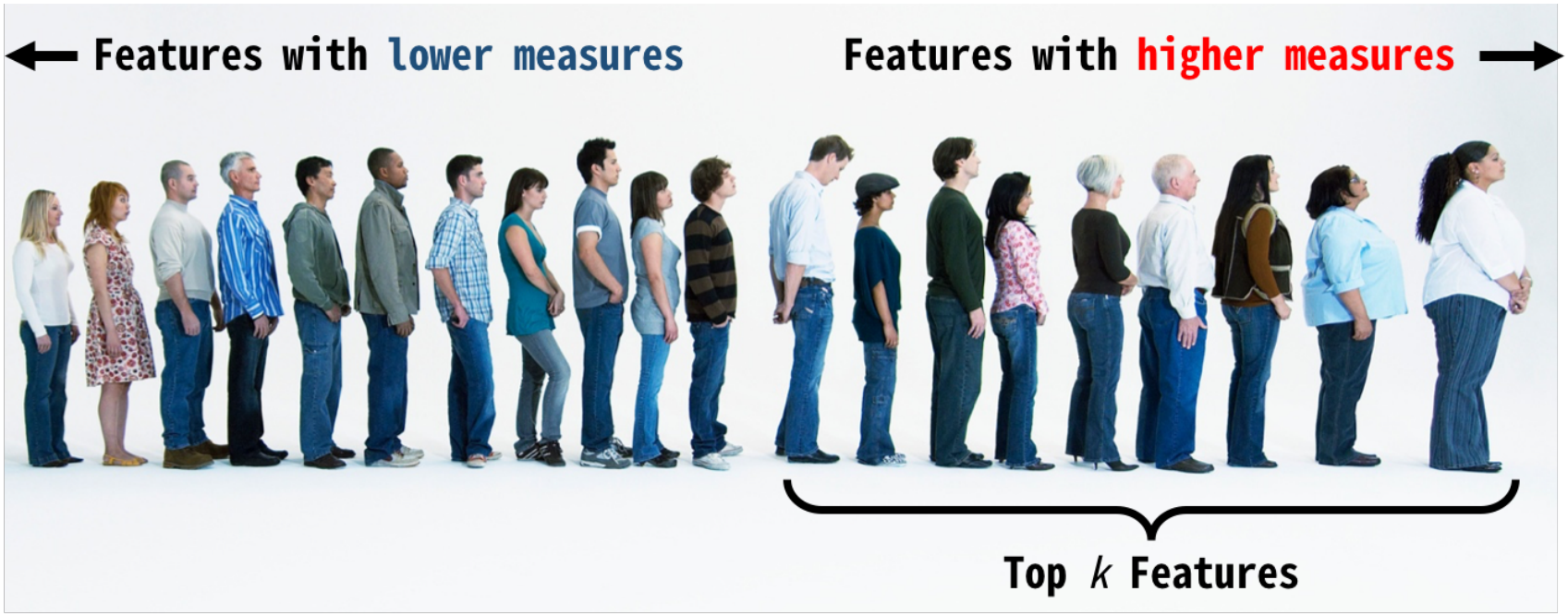
Compared to Wrapping Method...

- More efficient (i.e., model-independent)
- Do not consider relations among features
- Need other measures to filter features



Greedy Inclusion Algorithm

Greedily select top K features by a measure



Criteria for Discrete Features

Evaluate whether the existence of a feature is important

Common Criteria [Yang and Pedersen, ICML 1997]

- Document Frequency (DF)
- χ^2 (chi-square) Statistics (CHI)
- Mutual Information (MI)
- Information Gain (IG)

Document Frequency (DF)

- Remove rare features appearing only few times
  - May be just noises
  - Not influential in the final decisions
  - Unlikely to appear in new data

Problem

- Too ad-hoc for feature selection
- Independent to ground truth (i.e., the target)

χ^2 (chi-square) Statistics

χ^2 Statistics

Check whether a characteristic is dependent to being in one of groups

χ^2 in Feature Selection

- Groups:
  - Group 1: instances of the class c
  - Group 2: other instances
- Characteristic
  - The instance contains the feature t

χ^2 (chi-square) Statistics (Cont'd)

Null Hypothesis

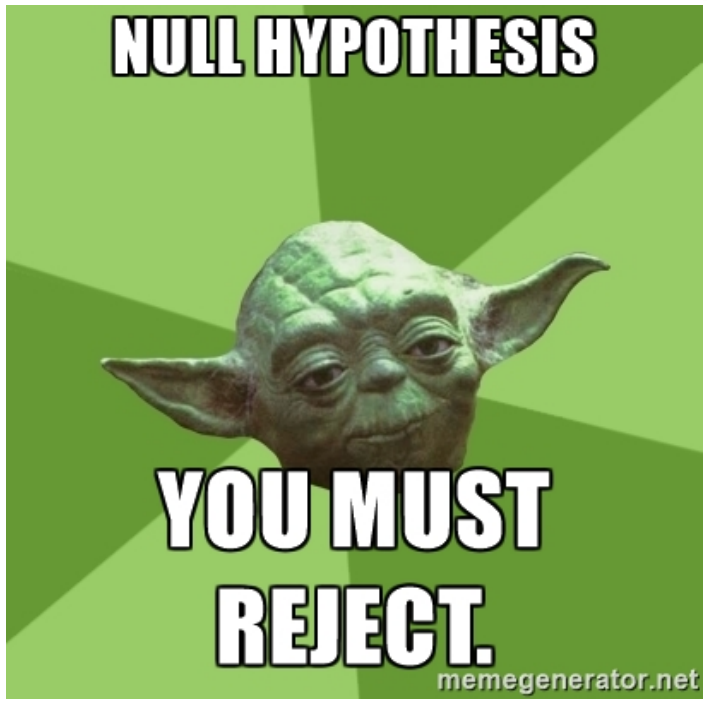
The feature t is independent to whether instances are of the class c

If hypothesis is upheld...

$$P(t, c) = P(t) \times P(c)$$

Reality vs. Hypothesis

- Data-driven probability estimation
- Farther from the assumed probability
  - The feature is more important!



Credit: memegenerator.net

Mutual Information (MI)

Measure the dependence between discrete random variables X and Y

$$I(X,Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) \log \frac{P(x,y)}{P(x) \cdot P(y)}$$

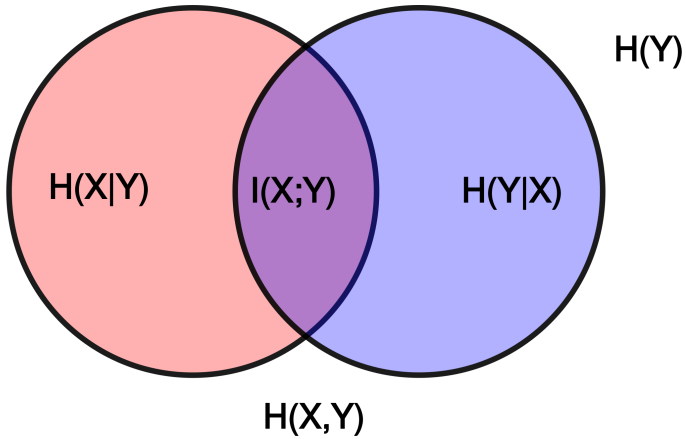


Illustration from Wikipedia

MI in Feature Selection

- Dependence between the feature and the ground truths
- Utilize data to estimate probabilities

Information Gain (IG)

Information gained by knowing where the feature is present

Entropy

- Measure the information
- Foundation of information theory

$$H(X) = E[-\log P(X)] = \sum_{x \in X} P(x) \log P(x)$$

Gain by the Feature

- Decompose cases by the feature status

$$H(c \mid \text{知道 feature } t) - H(c) =$$

$$P(t) \cdot H(c \mid t) + P(\neg t) \cdot H(c \mid \neg t) - H(c)$$

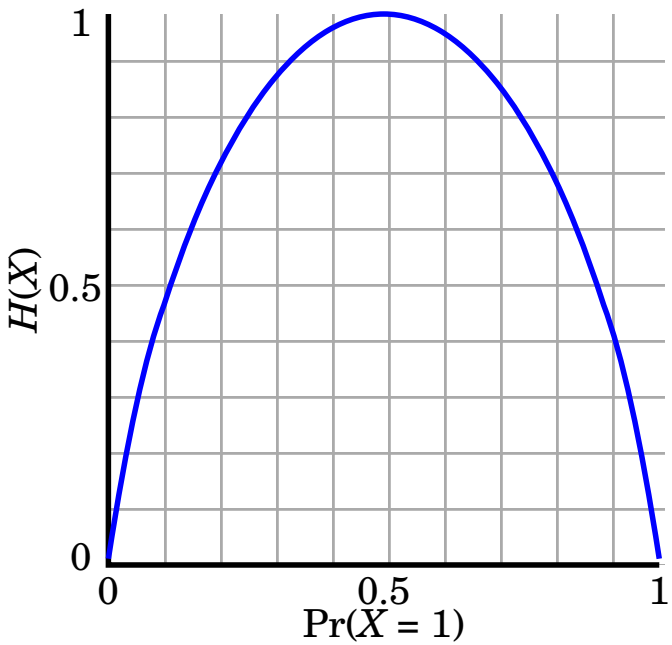


Illustration from Wikipedia

Criteria for Numerical Features

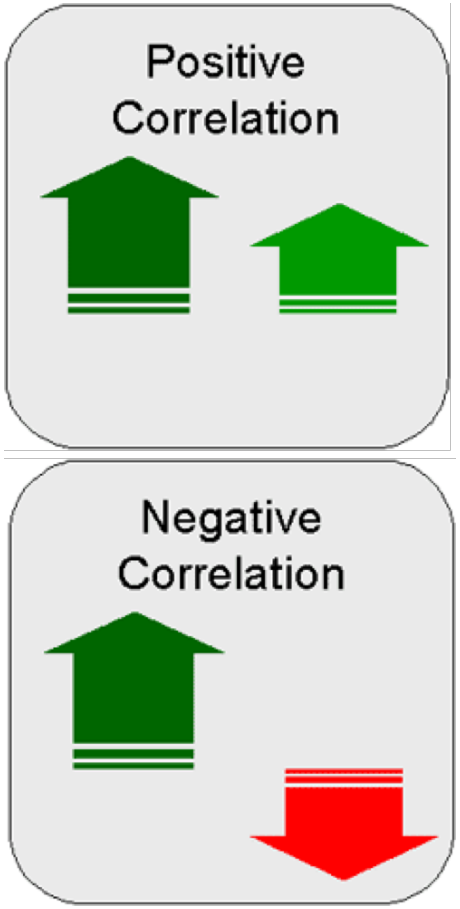
Evaluate the dependency between feature values and ground truths

Correlation: Relationship between 2 Variables

- Positive or negative correlation?
- Native in the regression task
- 0 and 1 can be also informative

Common Correlation Coefficients

- Pearson Correlation (γ)
- Spearman Rank Correlation (ρ)
- Kendall Rank Correlation (τ)



Pearson Correlation (γ)

Linearity Assumption

- Assume linear relationship between variables
  - e.g., Greater weights, higher heights.
- Assume variables are normal distributed

Pearson Correlation

$$\gamma = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

- μ is mean, and σ is standard deviation.
- Useful in evaluating the effect size
- Efficient computation in $O(N)$

Effect Size	$\lvert\gamma\rvert$
Small	0.10
Medium	0.30
Large	0.50

Spearman Rank Correlation (ρ)

Correlation for Ordinal Variables

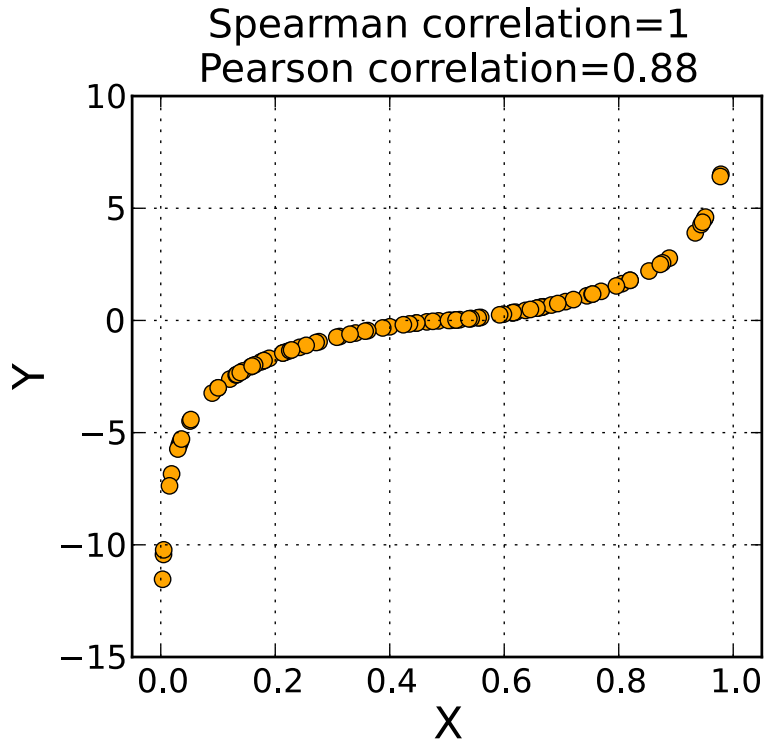
Compare the rank order of two variables

Spearman Rank Correlation

$$\rho = 1 - \frac{6 \sum (\text{rank}(X_i) - \text{rank}(Y_i))^2}{N(N^2 - 1)}$$

- $\text{rank}(X_i)$ represent the rank in data
- Assess the monotonic relationship
- $O(N \log N)$ time complexity

How about ties (identical rank)



Kendall Rank Correlation (τ)

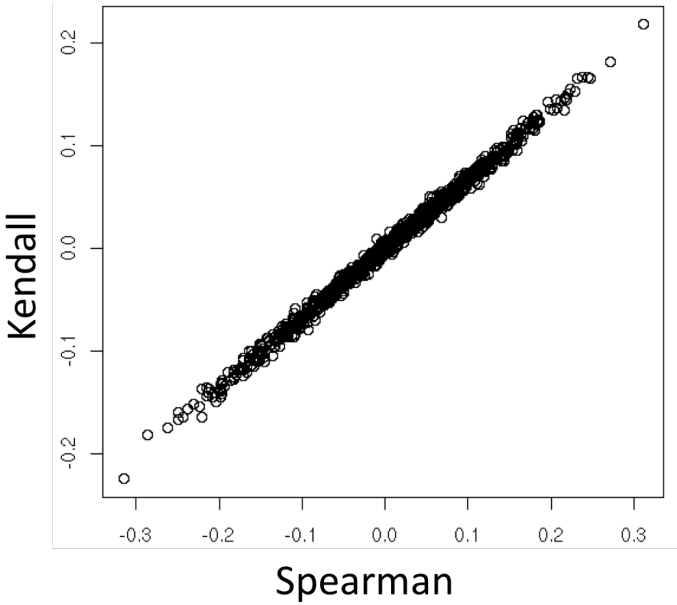
Correlation for Ordinal Variables

Compare the rank consistency of two variables

$$\tau = \frac{(\#\text{ of concordant pairs}) - (\#\text{ of discordant pairs})}{\frac{1}{2}n(n-1)}$$

Kendall Rank Correlation

- Measure pairwise correlation
- 0 if variables are independent
- More inefficient computation in $O(N^2)$



Outline

Large-scale data may have countless features

High-dimensional Data

- High-resolution Images
  - $9600 \times 7200 \approx 2^{26}$ pixels
- Social Networks
  - $65{,}608{,}366 \approx 2^{26}$ nodes (Friendster)
- Large-scale Text Corpus
  - $\sim 2^{60}$ 5-gram Bag-of-Word terms

Edited from One Piece

More dimensions, Longer training time

K times dimensions generally need $O(K)$ times training time

Sparseness Problem in Discrete Features

Example: Bag of ~~Chips~~ Words

2^{60} dimensions!! But usually only tens of them are non-zero...

Captured from a news of CTS

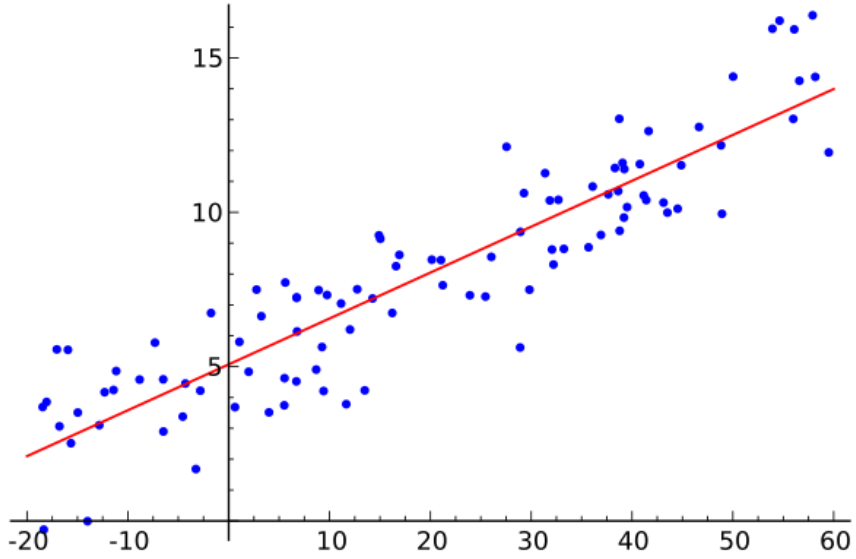
Dimension Reduction

Goal

Reduce the number of features and obtain most information

What should be kept?

- Principal Information
  - to represent most information
- Uncorrelated Information
  - to reduce the number of features



Credit: http://usa-da.blogspot.tw/

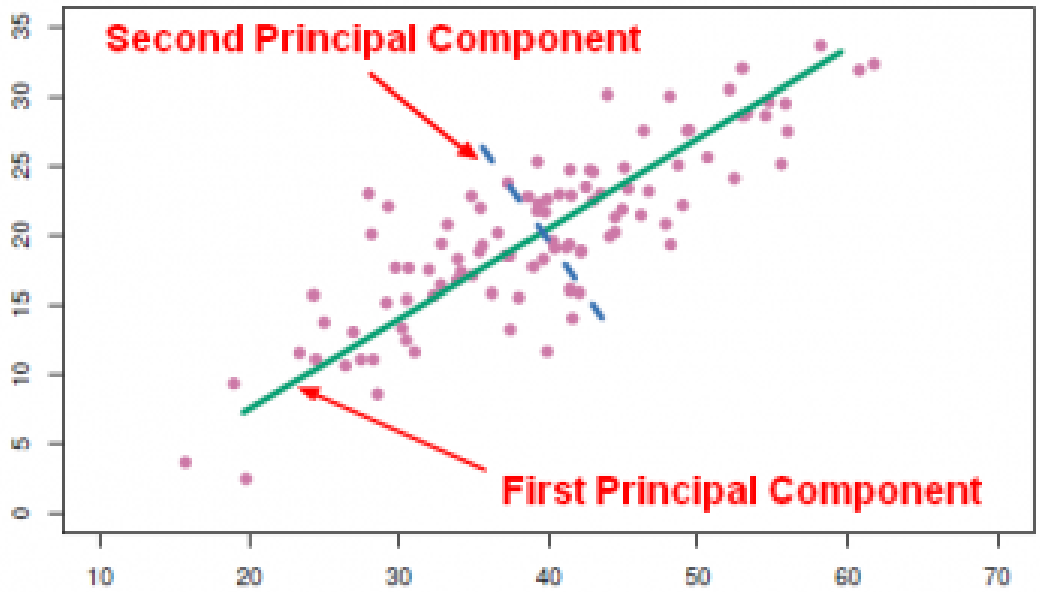
1 dimension can represent 2D data!

Principal Component Analysis (PCA)

Transform data into a lower-dimensional space with principal components

Principal Components?

- First largest component
- Second largest one and orthogonal to the above
- ...



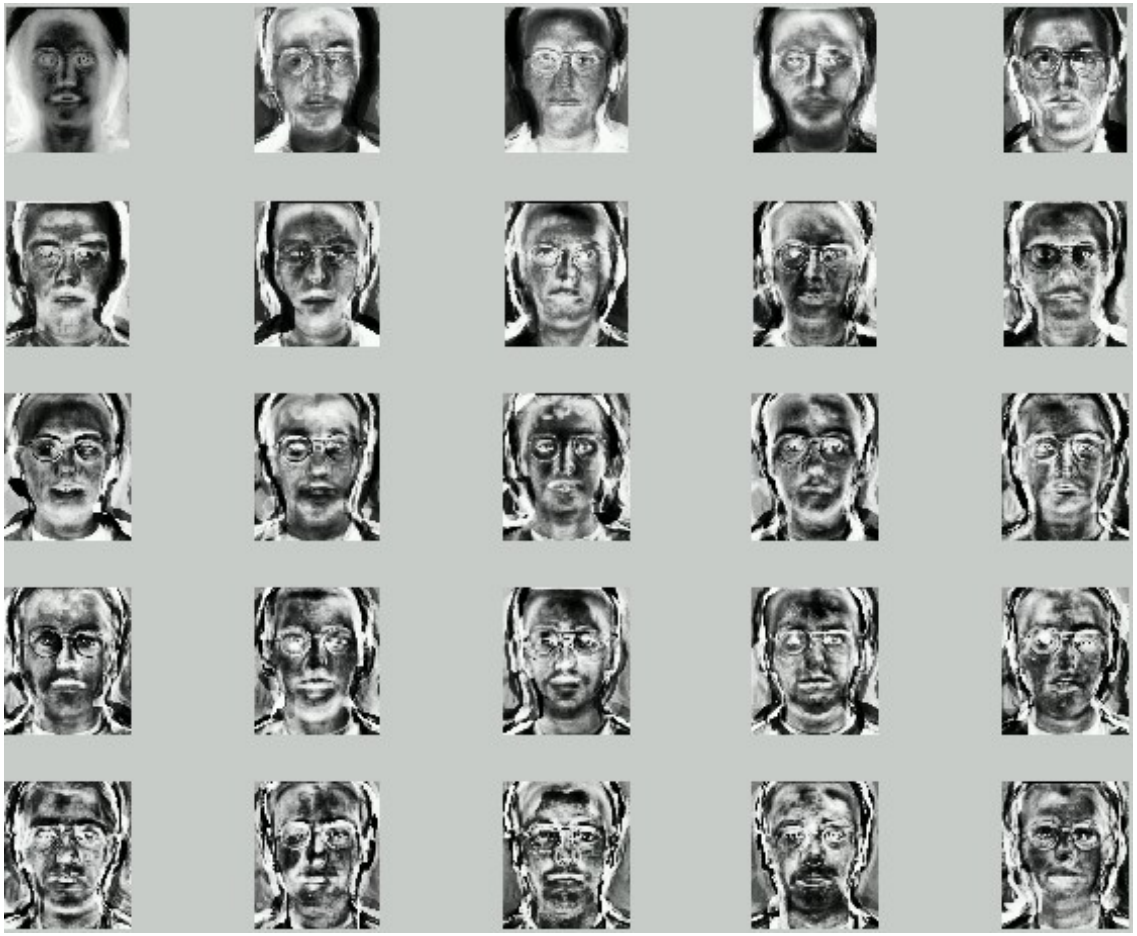
Implemented in the scikit-learn library

How many principal components do you need?

80/20 Rule

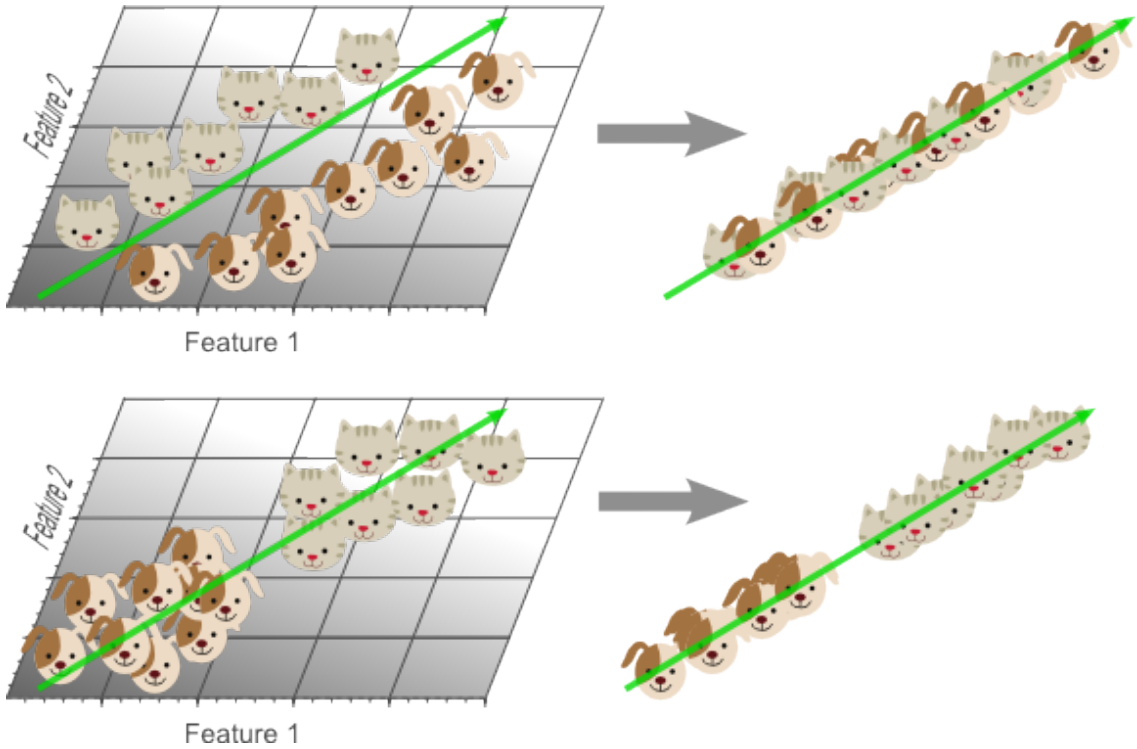
Most variance can be explained by only few principal components

Application: Eigenface – The basis of faces



Basis with only 7 features [Turk and Pentland, JCN 1991]

PCA may not be the panacea



http://www.visiondummy.com/2014/05/feature-extraction-using-pca/

Sometimes should consider class information.

Short Summary

In the latter of lecture 2, you have learned ...

- Why feature selection and analysis are important
- How to select features from numerous data
- How to reduce dimensions but keep information

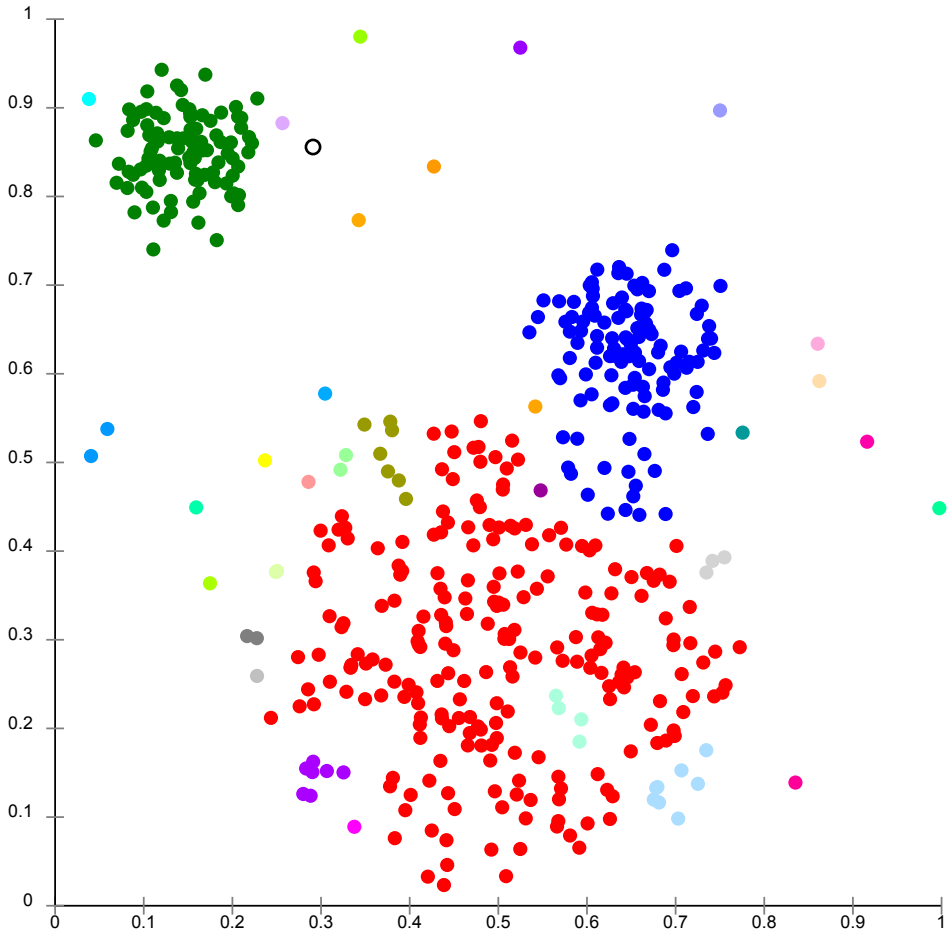
Next: How to cluster data and apply clustering to solve problems

Outline

Outline

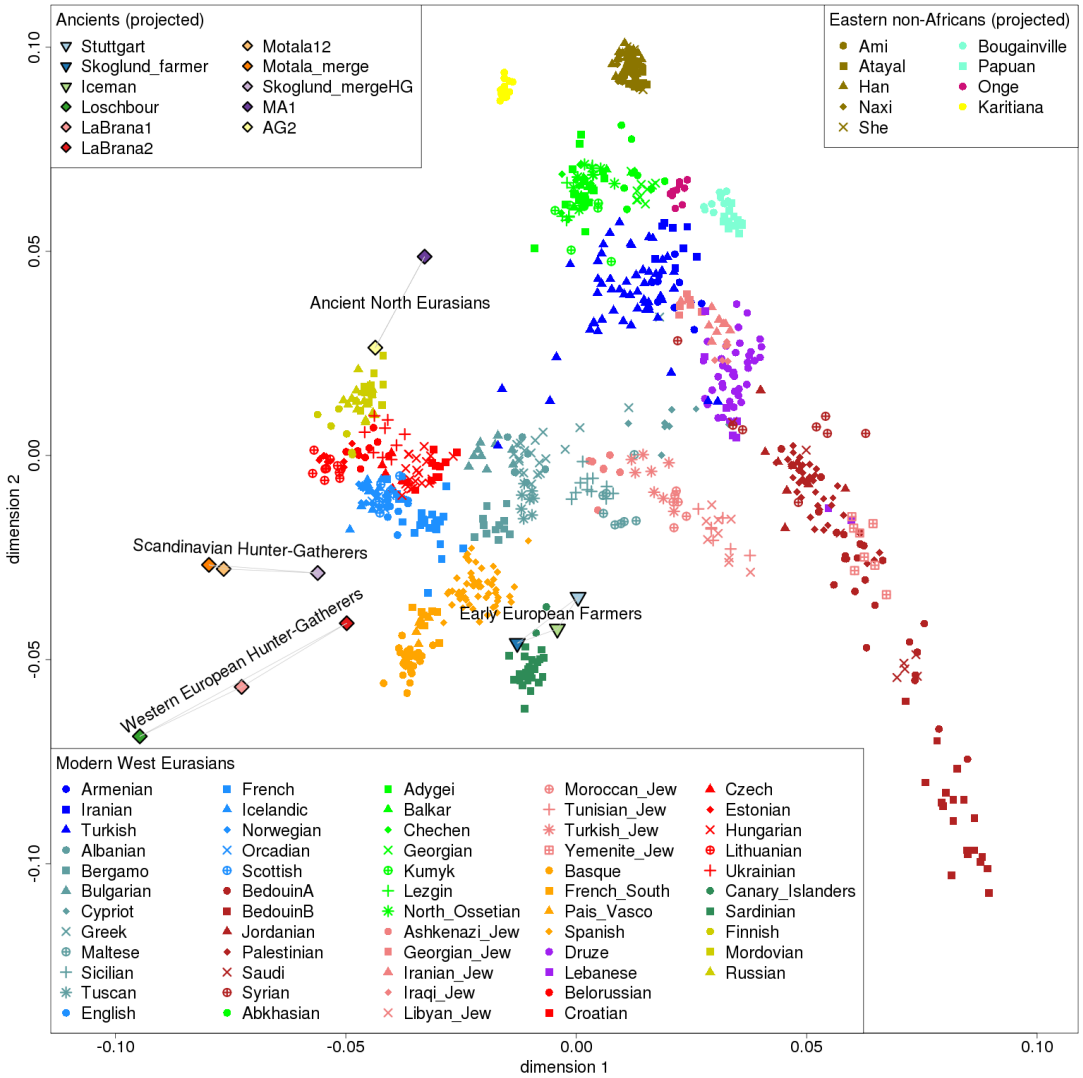
Clustering: An unsupervised learning problem

group subsets of instances based on some notion of similarity



Credit: Wikipedia

Origin: DNA clustering



http://dienekes.blogspot.tw/2013/12/europeans-neolithic-farmers-mesolithic.html

More Specific Definition

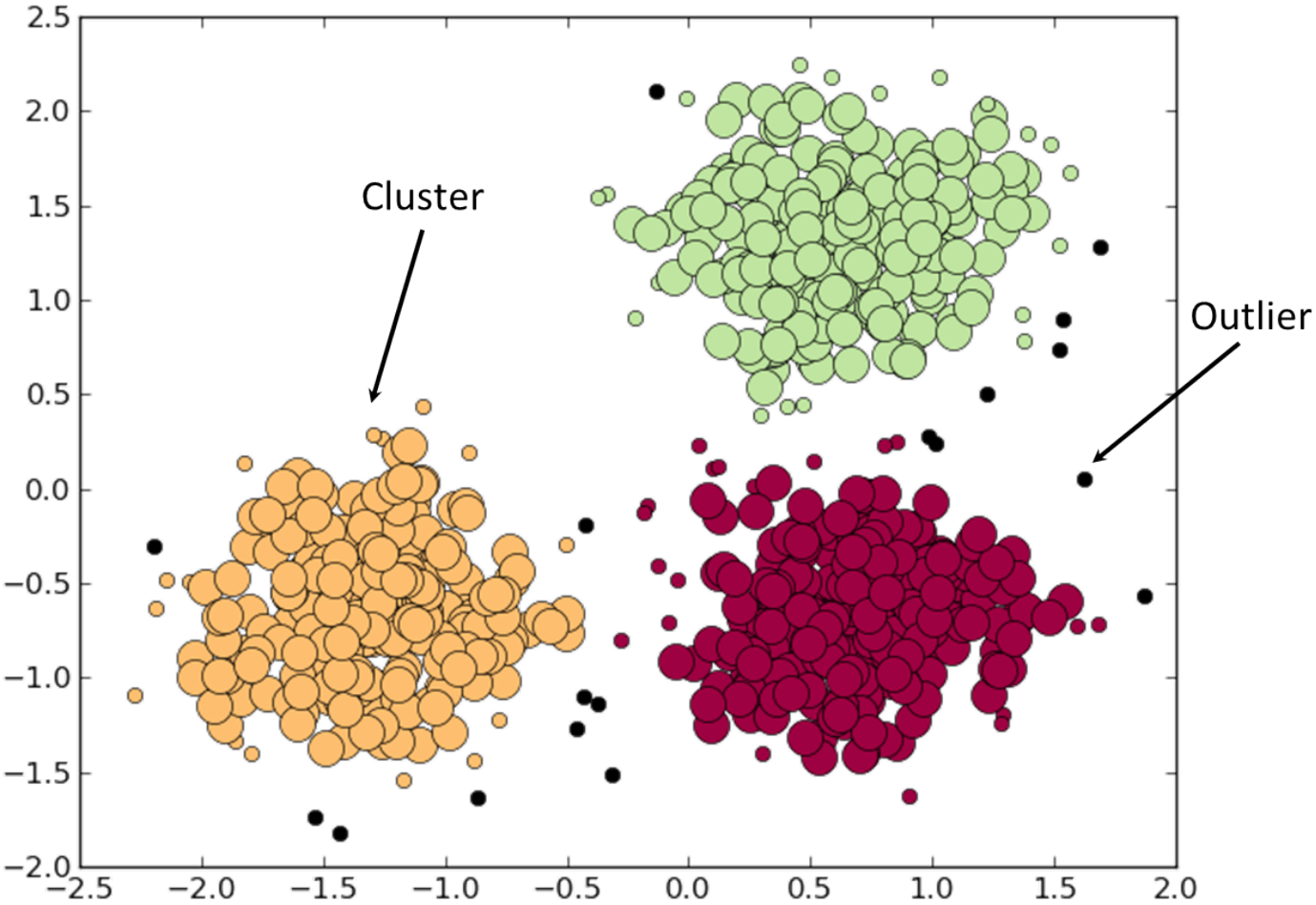
- Discover groups and structures (clusters) in the data.

Definition

Given a set of points (features), group the points into clusters such that

- Points in a cluster are close/similar to each other.
- Points in different clusters are dissimilar.

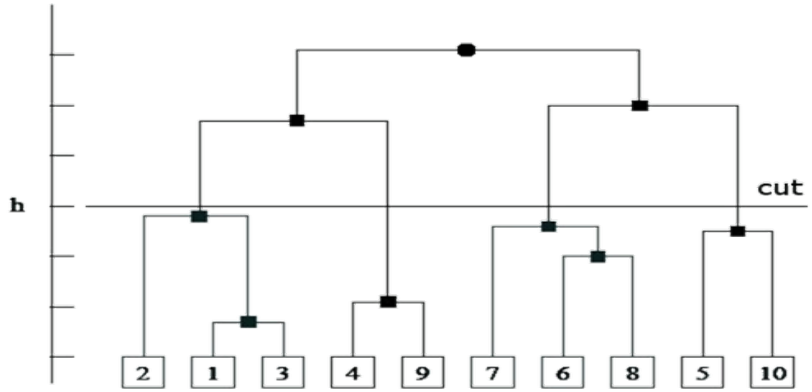
- Usually,
  - Data might be in high-dimensional space.
  - Similarity is defined as distance measures.


Cluster
Outlier
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

Types of Clustering Methods

Hierarchical Clustering

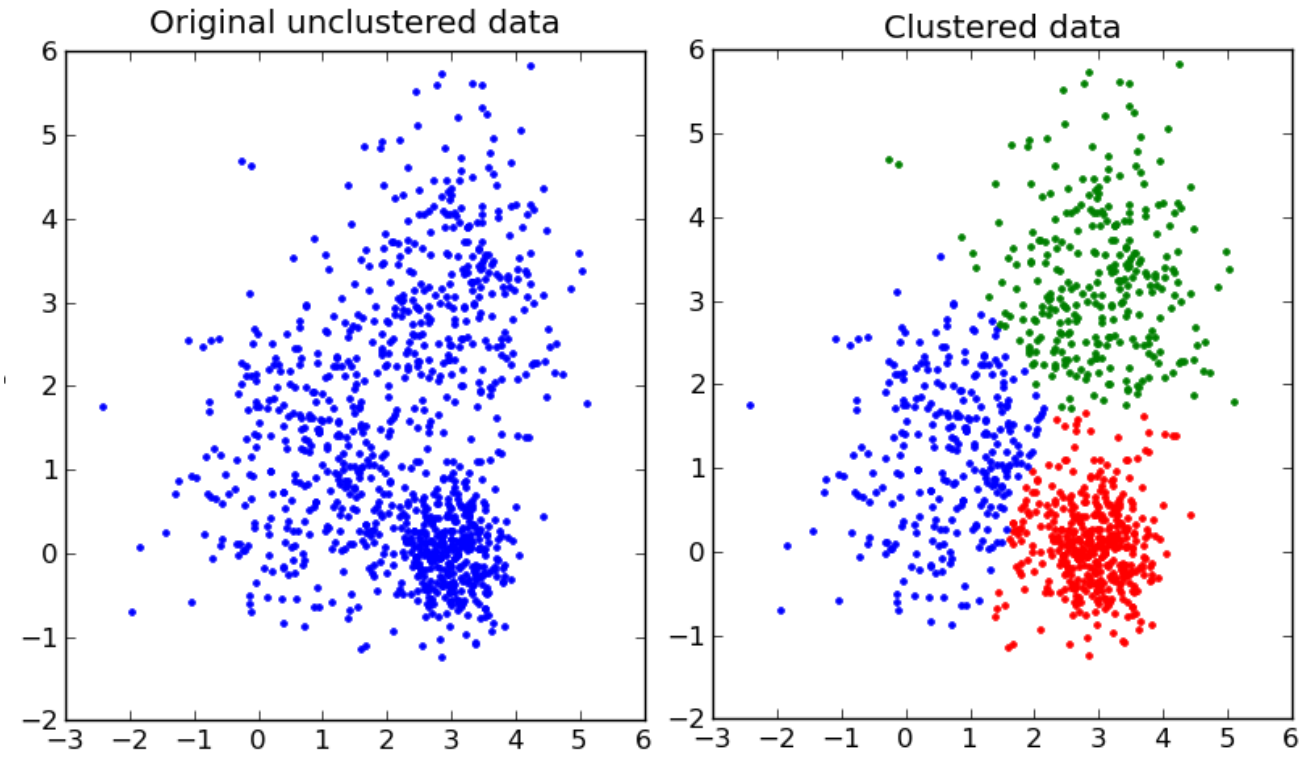
- Assume clusters can be represented by a cluster tree.
- Agglomerative (bottom-up)
  - Initially, each point is a cluster
  - Repeatedly combine the two "nearest" clusters into one
- Divisive (top-down)
  - Start with one cluster and recursively split it



Types of Clustering Methods (Cont'd)

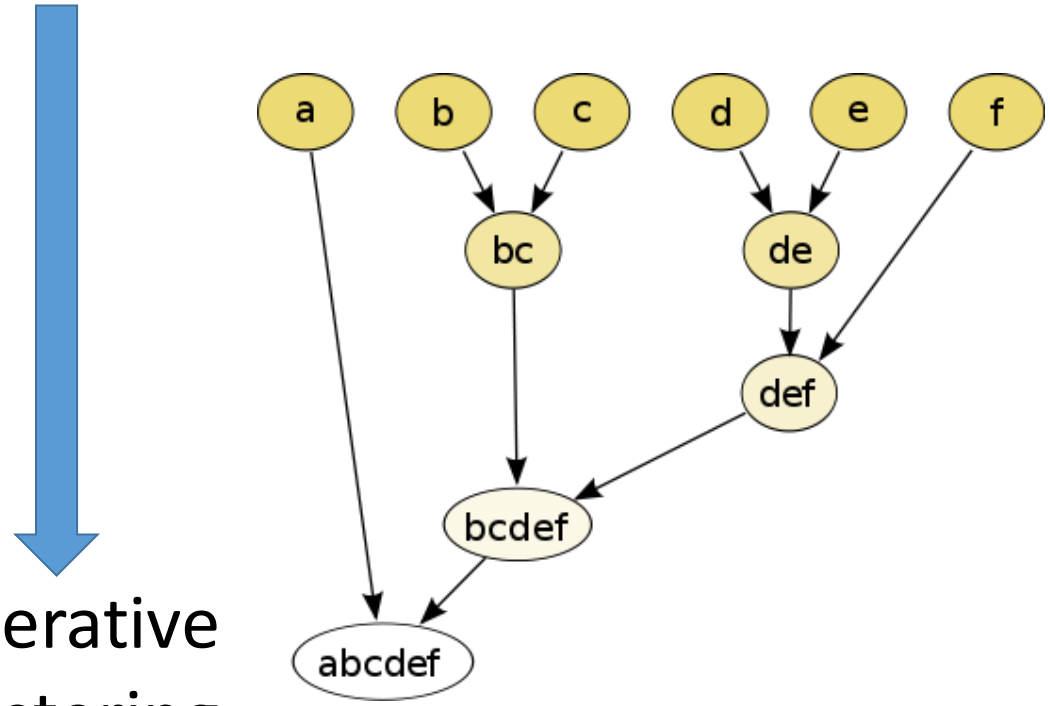
Partitional Clustering

- Maintain a set of clusters
- Points belong to "nearest" cluster



Outline

Two Types of Hierarchical Clustering



Agglomerative Clustering

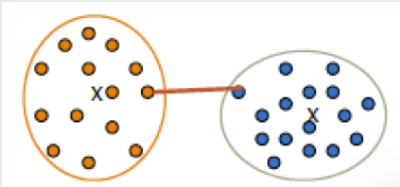
Divisive Clustering

Distance between Two Clusters

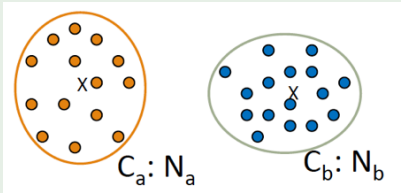
Need to know distances for splitting and merging clusters

Common Approaches

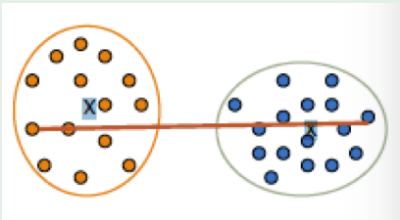
- Single-link
  - Distance of the closest points



- Average-link
  - Average distance between pairs



- Complete-link
  - Distance of the furthest points



- Centroid
  - Distance of centroids

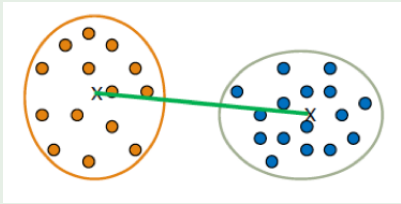


Illustration from Online Course "Cluster Analysis in Data Mining" in Coursera

Implementation of Hierarchical Clustering

Time Complexity

- Divisive clustering is $O(2^n)$
  - Use an exhaustive search to try splitting clusters.
- Agglomerative clustering is $O(n^3)$ or $O(n^2 \log n)$ with a heap.
  - At each step, compute pairwise distances between all pairs of clusters.
- Suitable for smaller data, but too expensive large-scale data

Not difficult to implement, and also implemented in scikit-learn.

Outline

K-Means Algorithm

- K-Means is a partitional clustering algorithm.
- Assume Euclidean space/distance.
- It partitions the given data into k clusters.
  - k is a user-specific parameter.

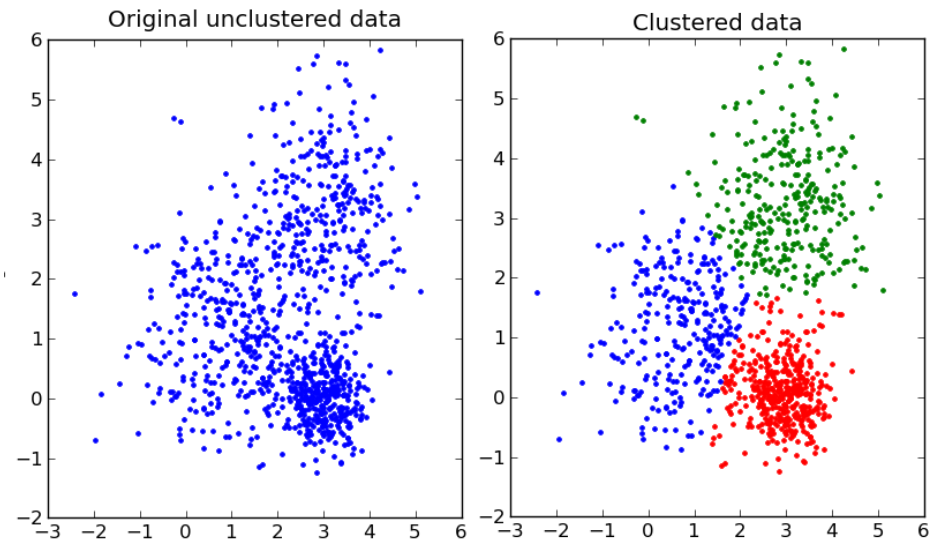


Figure: Clustering results with $k = 3$.

K-Means Algorithm

- Given k, the k-means works as follows:

K-Means Algorithms

- (1) Randomly choose k points to be the initial centroids.
- (2) Assign each point to the nearest cluster with centroids.
- (3) Update centroids of clusters with current cluster constituents.
- (4) If the stopping criterion is not met, go to (2).

- Stopping Criterion
  - No point is assigned to different cluster
  - Centroids are stable.

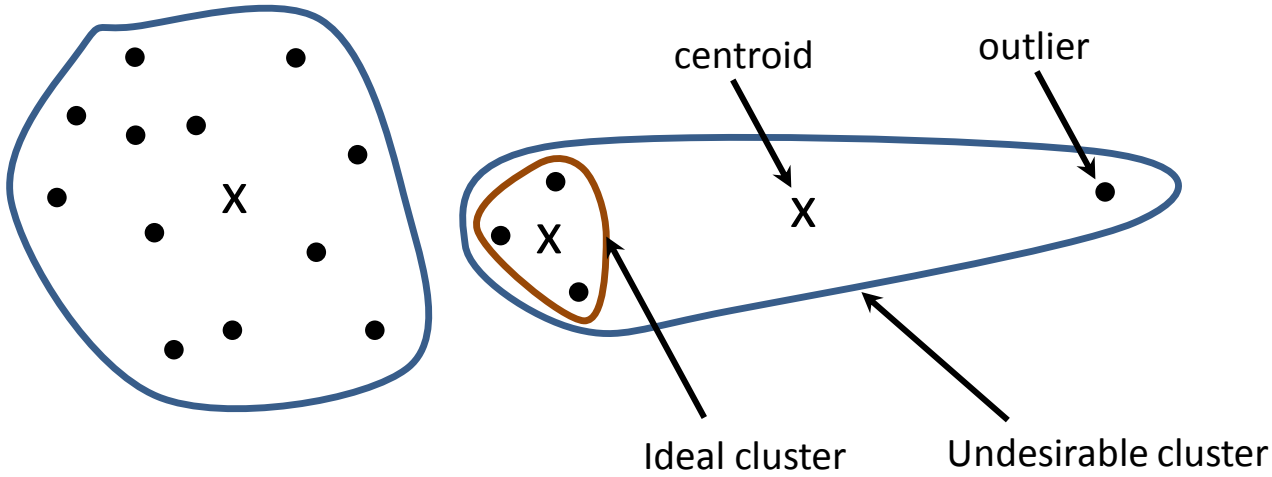
Advantages and Disadvantages of K-Means

Pros

- easy to learn and implement
- efficient – $O(kn)$ for t iterations
- almost linear time with small k

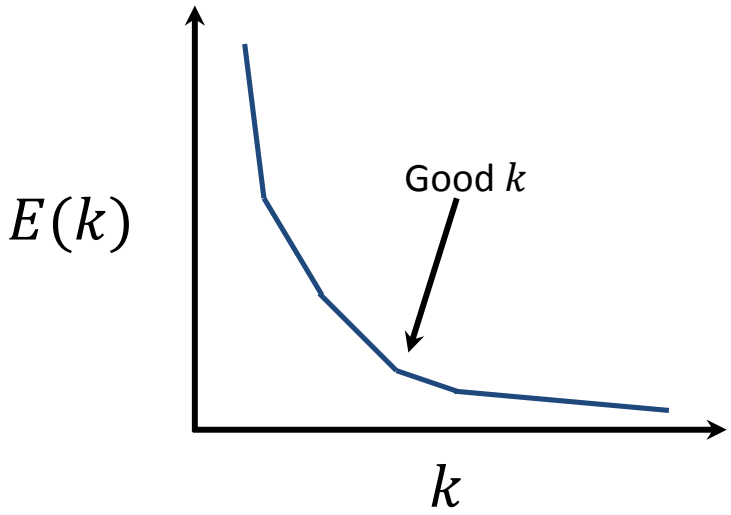
Cons

- k is needed to be specified.
- sensitive to outliers
- sensitive to initial centroids



How to select a good K

- Approach 1
  - Let $E(k)$ represent the average distance to centroids.
  - $E(k)$ decreases while k is increasing.
  - $E(k)$ falls rapidly until right k, then changes little.
- Approach 2
  - Semi-automatic selection
  - Consider distortion and complexity in an equation at the same time.
  - $K = \text{argmin}_k(E(k) + \lambda K)$, where λ is a weighting parameter
  - λ can be shared with similar data in the past.



BFR Algorithm

- BFR [Bradley-Fayyad-Reina] is a variant of k-means [Bradley et al., KDD 1998]
- Designed to handle large-scale and high-dimensional data sets.

Assumption

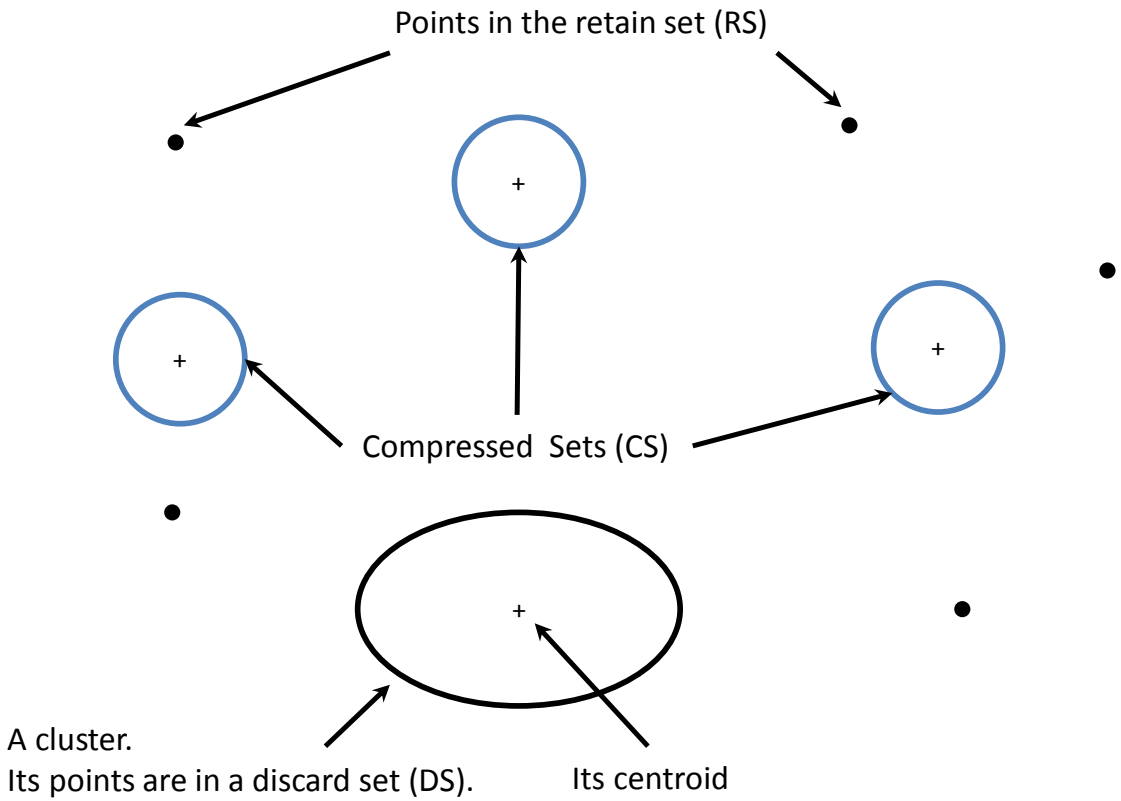
- Clusters are normally distributed around a centroid in a Euclidean space.

- Most loaded points are summarized by simple statistics.
- Efficient – $O(clusters)$ space complexity but $O(data)$ time complexity.

BFR Algorithm

- Initialization
  - Start from a K-Means
  - Sample some points and cluster optimally.
- Classify points into 3 sets
  - Discard Set (DS):
    Points close enough to a centroid to be summarized.
  - Compression Set (CS):
    Groups of points that are close together but not close to any existing centroid. These points are summarized, but not assigned to a cluster.
  - Retained Set (RS):
    Isolated points waiting to be assigned to a compression set.

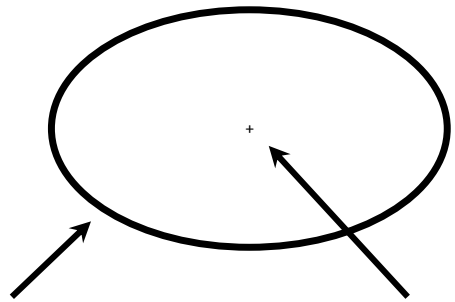
Three Sets in BFR Algorithm



- Discard Set (DS): Close enough to a centroid to be summarized
- Compression Set (CS): Summarized, but not assigned to a cluster
- Retained Set (RS): Isolated points

Summarizing Sets of Points

- For each cluster, the discard set (DS) is summarized by:
  - The number of points, N
  - The vector SUM,
    $SUM(i)$ is the sum of the coordinates of the points in the i-th dimension
  - The vector $SUMSQ$,
    $SUMSQ(i)$ = sum of squares of coordinates in i-th dimension.
- Variance of a cluster's discard set in dimension i
  - $\frac{SUMSQ(i)}{N} - \left(\frac{SUM(i)}{N}\right)^2$



A cluster.
Its points are in a discard set (DS).

Its centroid

The "Memory-Load" of Points

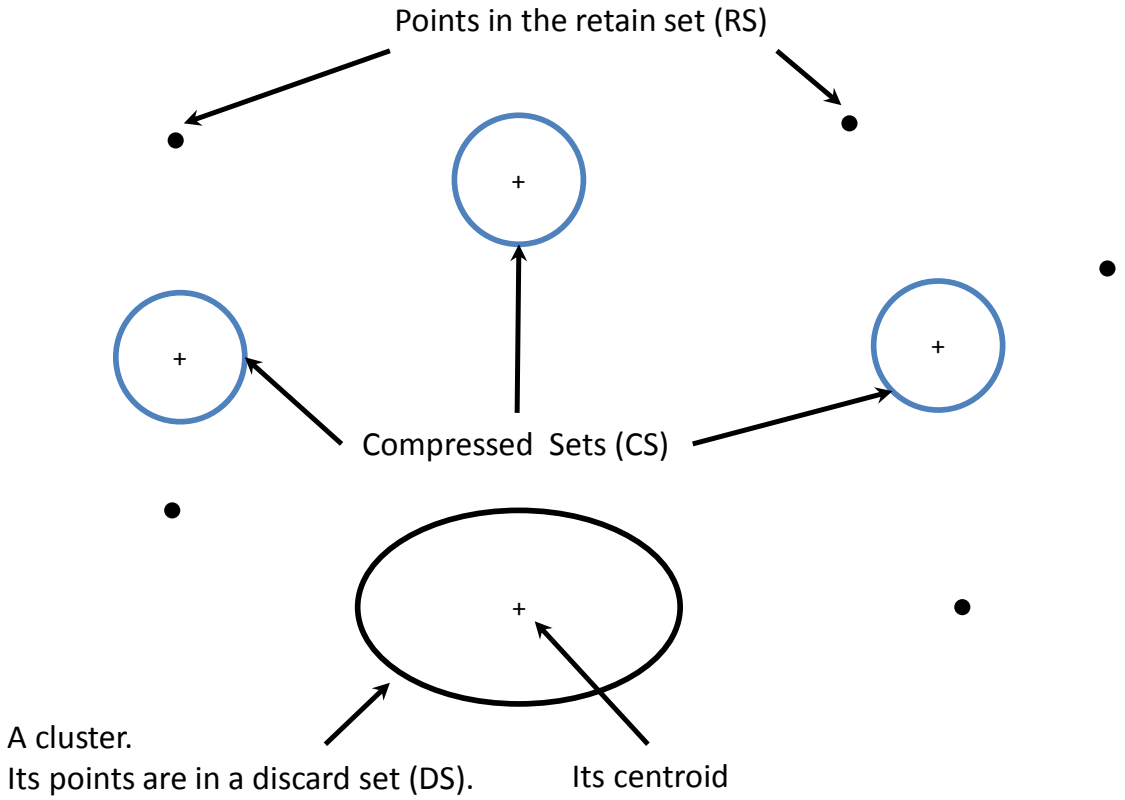
- Points are read from disk one main-memory-full at a time.

BFR Algorithm

- (1) Add "sufficiently close" points into each cluster and its DS
- (2) Cluster (by K-Means or others) remaining points and the old RS
  - Clusters become CS, outliers become RS.
- (3) In the last round, merge CS and RS into their nearest cluster.

- Discard Set (DS): Close enough to a centroid to be summarized
- Compression Set (CS): Summarized, but not assigned to a cluster
- Retained Set (RS): Isolated points

Three Sets in BFR Algorithm



- Discard Set (DS): Close enough to a centroid to be summarized
- Compression Set (CS): Summarized, but not assigned to a cluster
- Retained Set (RS): Isolated points

How Close is Close Enough?

- The Mahalanobis distance is less than a threshold.
- High likelihood of the point belonging to currently nearest centroid.

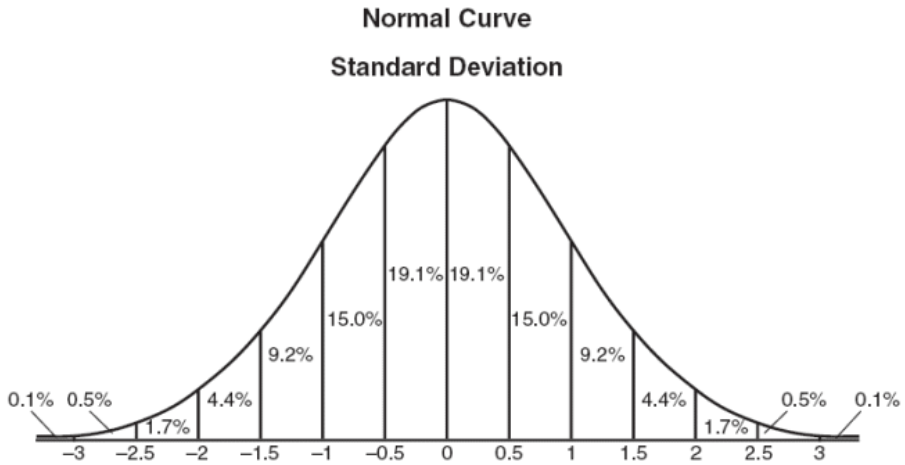
Normalized Euclidean distance from centroid

- For a point x and a centroid c
  - (1) Normalize in each dimension, $y_i = (x_i - c_i)/\sigma_i$
  - (2) Take sum of the squares of y_i
  - (3) The the square root

$$d(x, c) = \sqrt{\sum_{i=1}^{d} \left(\frac{x_i - c_i}{\sigma_i} \right)^2}$$

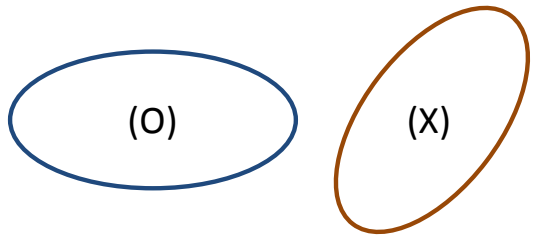
Mahalanobis Distance

- Clusters are normally distributed in d-dimensions
  - One standard deviation $= \sqrt{d}$ after transformation.
  - 68% of points of the cluster will have a Mahalanobis distance $< \sqrt{d}$.
- Decide the threshold with standard deviations and Mahalanobis distance.
  - e.g., 2 standard deviations.

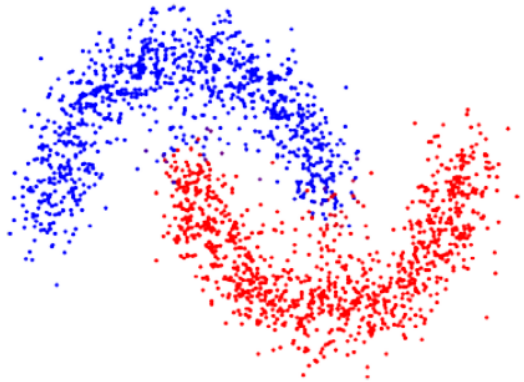


CURE Algorithm

- Problem in BFR/K-Means:
  - Assume clusters are normally distributed in each dimension.



- CURE (Clustering Using REpresentatives) [Guha et al., SIGMOD 1998]
  - Allows clusters to assume any shape
  - Uses a collection of representative points to represent clusters.



CURE Algorithm

CURE Algorithm

- (0) Pick a random sample of points that fit in main memory.
- (1) Initialize Clusters
  - Cluster points hierarchically – group nearest points and clusters.
- (2) Pick Representatives
  - For each cluster, pick a sample of points.
  - Pick representatives by moving them 20% toward the centroid.
- (3) Find Closest Cluster
  - For each point p, find the closest representative and and assign p to representative's cluster.

Summary in Clustering Algorithms

- Discover groups and structures (clusters) in the data.
- Hierarchical Clustering
  - Agglomerative Clustering (bottom-up, $O(n^2 \log n)$)
  - Divisive Clustering (top-down, $O(2^n)$)
- K-Means
  - Simplest partitional clustering method
- BFR Algorithm
  - A variant of k-means for large-scale and high-dimensional data
- CURE Algorithm
  - Allows clusters to assume any shape

Outline

Applications of Clustering

Key Idea of Clustering Applications

- Things in a cluster share similar properties.
  - Contents of data
  - Communities in social networks (Social Network Analysis)
  - Relevance of documents (Information Retrieval)
  - Meanings of photos (Multimedia)
  - Locations of geo-tagged objects
  - Styles of musics
  - ...

- It implies...
  - (Properties Discovery)
    A cluster might represent some specific properties.
  - (Properties Inference)
    We can inference properties of unknown data from same-cluster data.

POI Identification

- Identify point-of-interests (special locations) to geo-located photos.
- Photos in close locations (same cluster) represent the same POI.

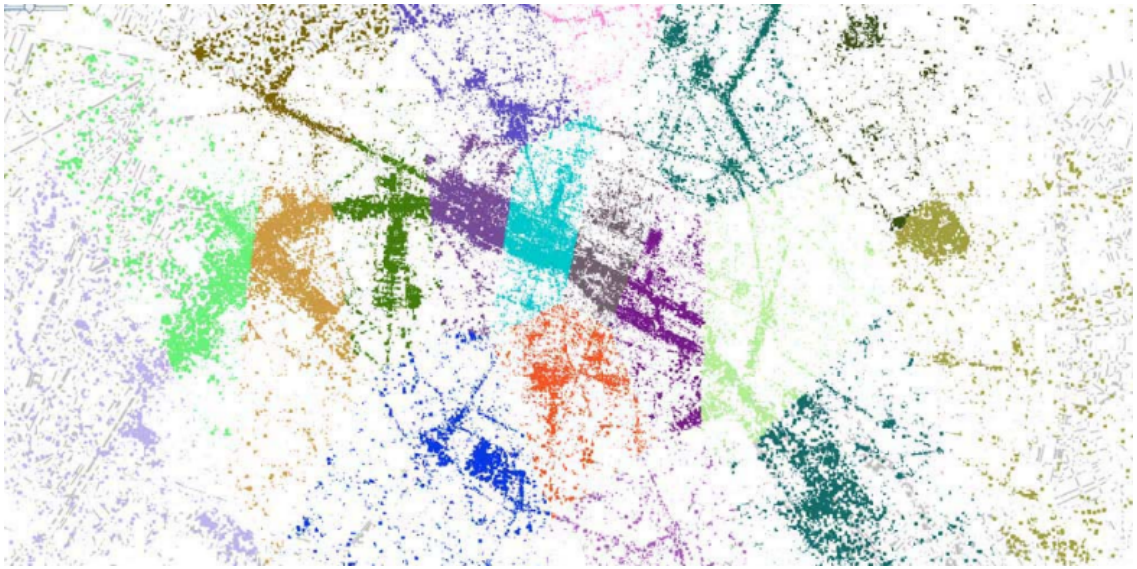


Figure: Identify POIs to Photos with Clustering [Yang et al., SIGIR 2011]

Community Detection

- Identify nodes in social networks into several communities.
- Nodes in a community have more interactions and connections.
- Cluster nodes into groups with related information.
- Nodes in a cluster represent a community.

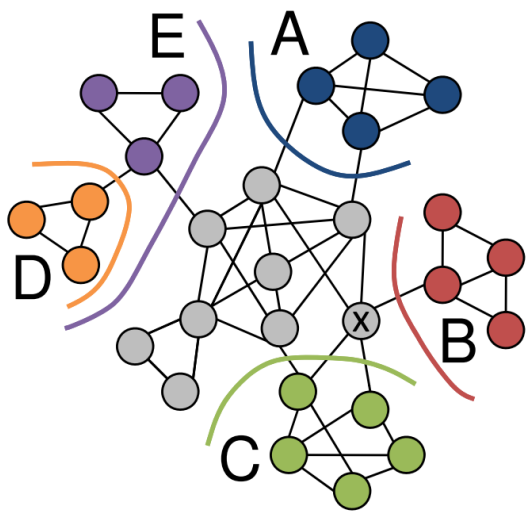


Figure: Detect communities by clustering [Leskovec et al., WWW 2010]

Document Type Discovery

- While a document describes an entity, it might have types.
- Same-cluster documents might share same types

Born	Thomas Cruise Mapother IV July 3, 1962 (age 51) Syracuse, New York, United States
Occupation	Actor, producer, writer
Years active	1981–present
Religion	Scientology
Spouse(s)	Mimi Rogers (m. 1987–90) Nicole Kidman (m. 1990–2001) Katie Holmes (m. 2006–12)
Children	3 (two adopted)
	Website
	TomCruise.com

Figure: Discover types of Wikipages by clustering infoboxes [Nguyen et al., CIKM '12]

Clustering in Information Retrieval

- Core Problem of Information Retrieval
  - Given many documents and a query, rank them by their relevance.

Clustering Hypothesis

- Closely associated documents tend to be relevant to the same requests

[van Rijsbergen, 1979]

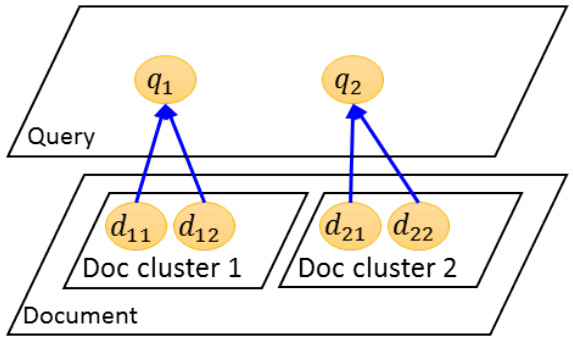


Figure: Illustration of clustering hypothesis.

Search Results Clustering

- Cluster search results into several groups.
- Relevance of documents can inference to same-cluster documents.
- More effective information presentation to users.

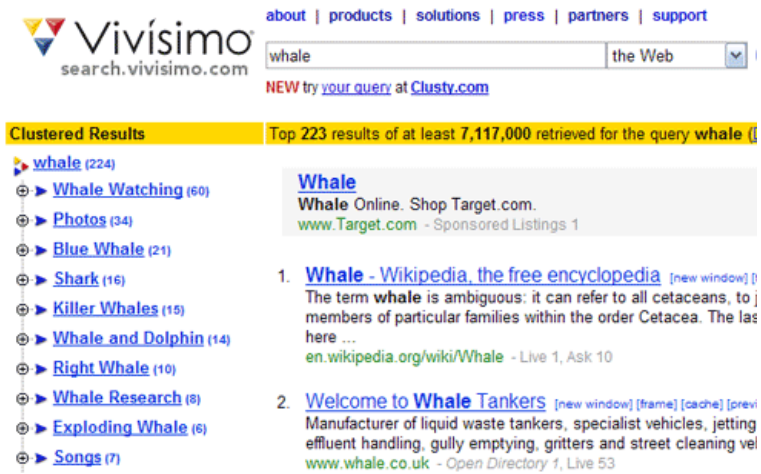


Figure: Vivisimo, a search engine clustering search results.

Short Summary

In the lecture 3, you have learned ...

- What is clustering
- Two types of clustering methods
- Two hierarchical and three partitional methods
- How to apply clustering to solve problems

Next: Recommender systems with no features

Tea break and communication time!

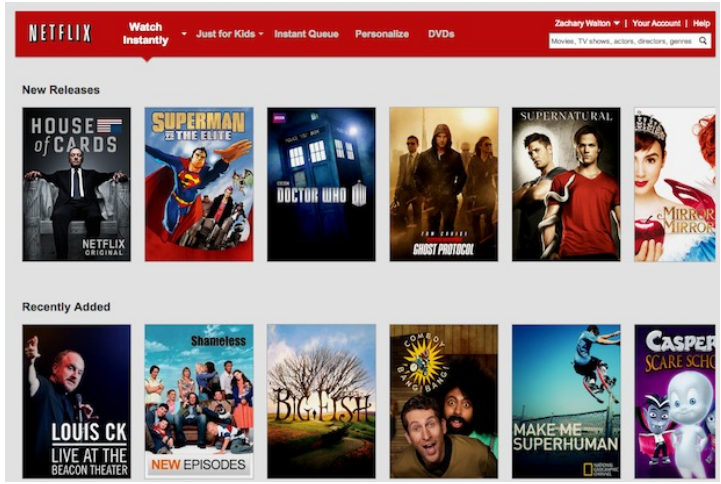
Outline

Outline

Stories of Video Providers

- Internet changes business models and customer behaviors.

Past



Present

- External reading: Internet Kills the Video Store
  - `http://www.nytimes.com/2013/11/07/business/media/internet-kills-the-video-store.html`

Services in the Era of Internet

More Users

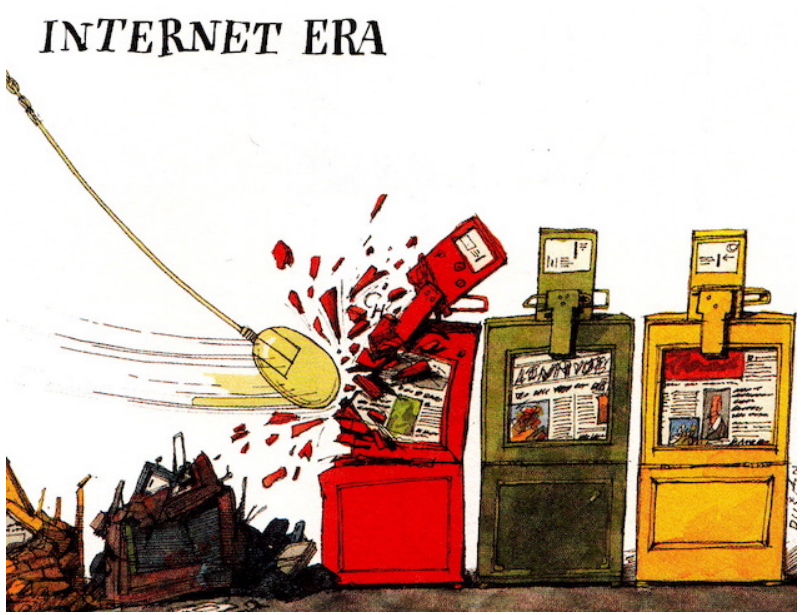
- Not to be limited by physical stores
- More information can be mined

More Items

- More choices for users
- Decision making becomes difficult

More Online Services

- Users can use services conveniently.
- Personalized services are possible.



Credit: Alison Garwood-Jones

Can the services do something more with their characteristics?

Real case: Amazon predicts orders in the future

- Amazon has a patent predicting orders in the future.
- Items can be moved to nearby hubs before you actually make the order.

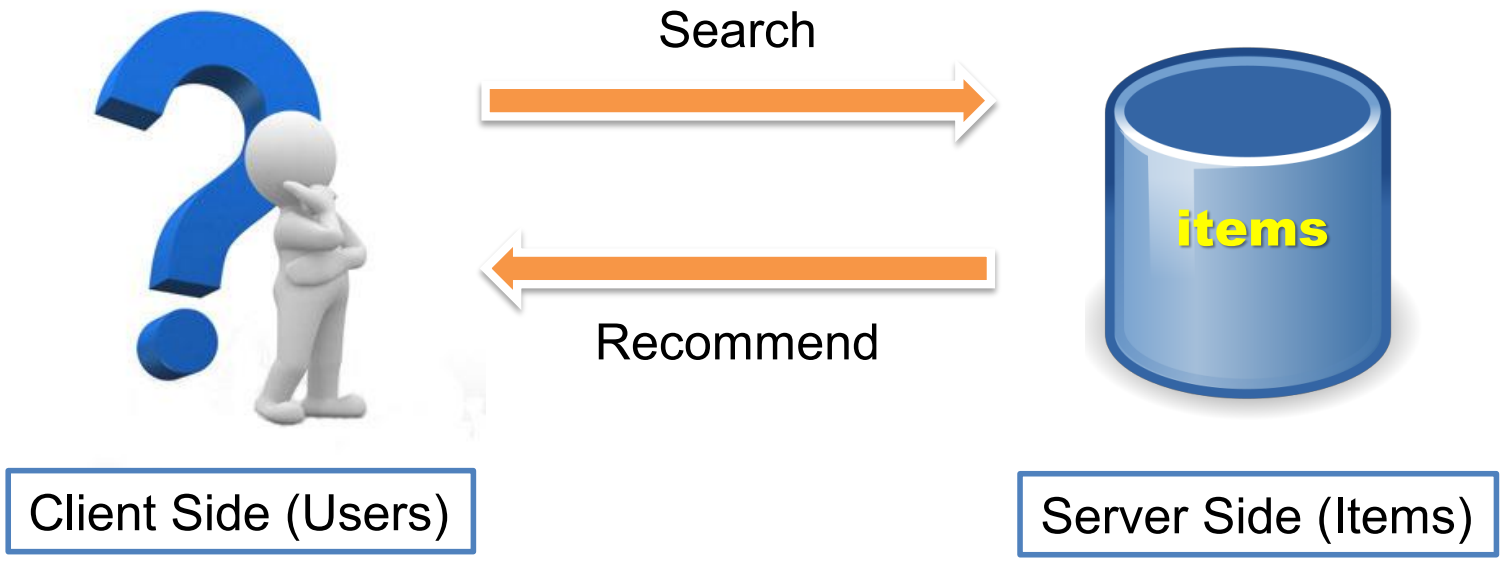
- External reading
  - www.digitaltrends.com/web/amazon-patented-anticipatory-shipping-system-predicts-orders/

Motivation: Recommender Systems

- Predict users' preferences on items.
- Widely used in many popular commercial on-line systems.
- Play the main role on the system performance.

By recommender systems...

- Systems can recognize users' favors exactly.
- Better user experiences in ordering items.
- Discover potential orders of users.



Types of Recommender Systems

Editorial Systems

- List of hand-curated essential items

Global Recommendation

- Simple statistics and aggregation
- Popularity: Top 10, Most popular
- Temporal: Recent uploads

Personalized Systems

- Tailored to individual users
- Amazon, Netflix, Facebook, ...

Credit: `https://icrunchdatanews.com/`

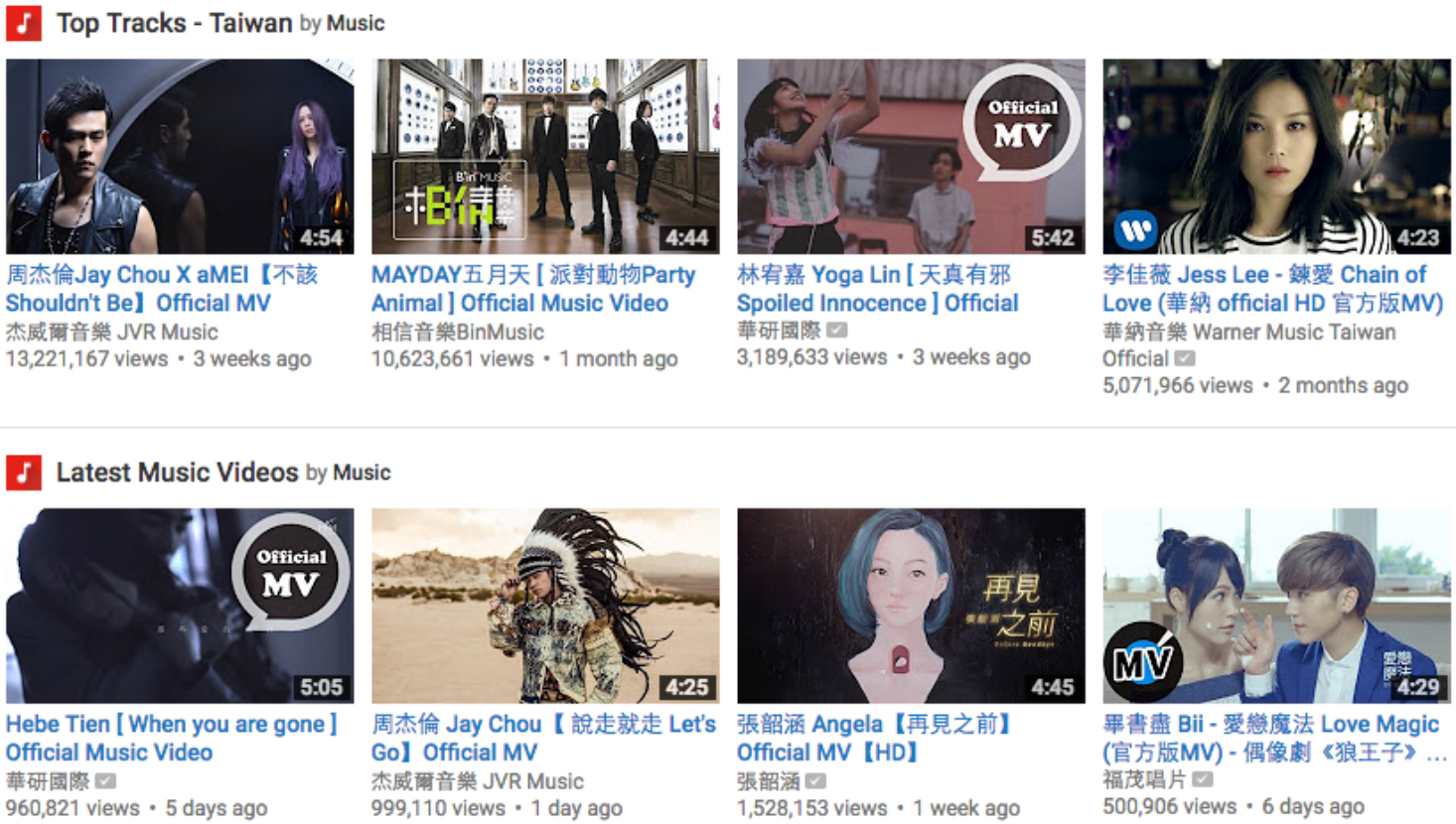
Editorial Systems

小編很辛苦...

An ad from VoiceTube

Global Recommendation

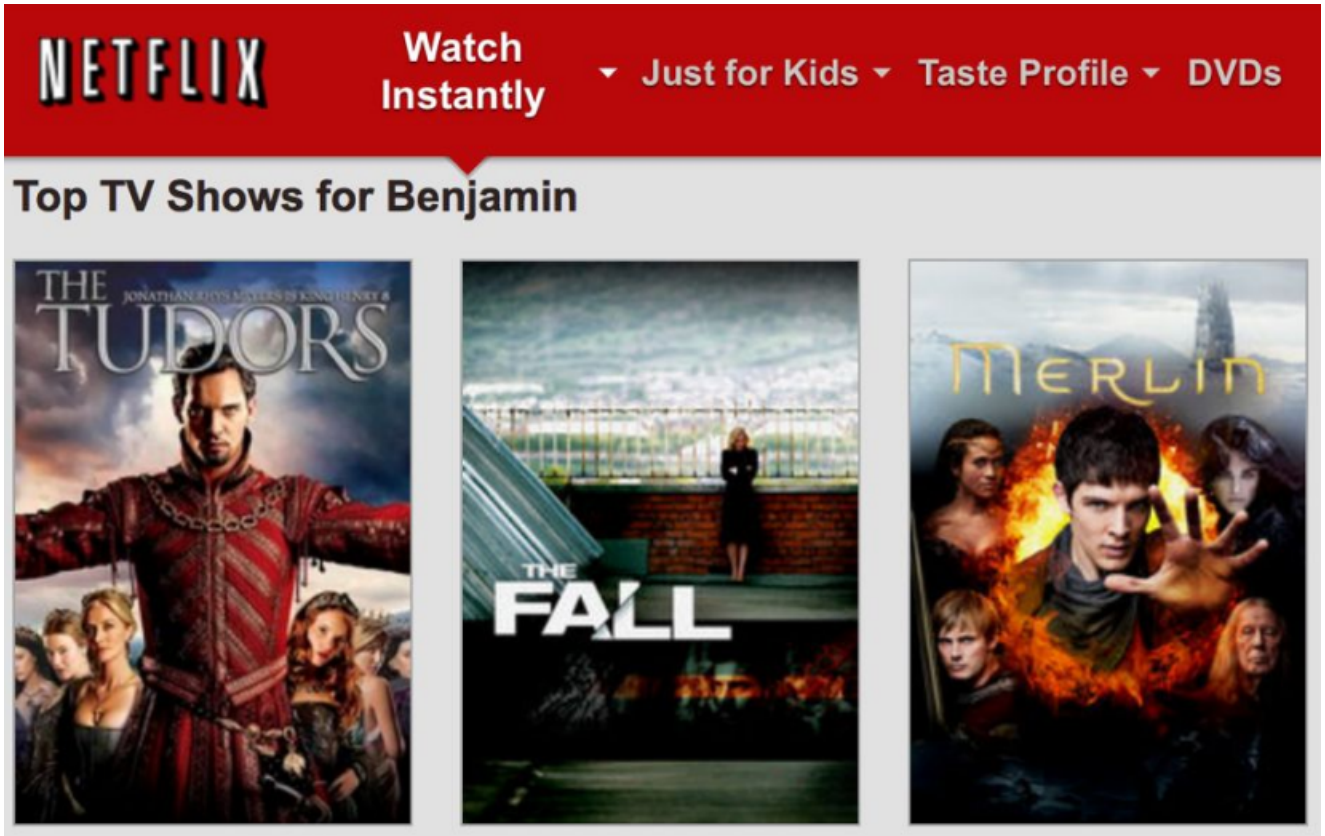
Popular and new items are always interesting for everyone?



Captured from Youtube

Personalized Recommendation

Personalization is useful and practical, but difficult.



Captured from Netflix

Problem Formulation

- Assume users' preferences can be represented by a matrix.
- Let X be the set of users, S be the set of items

Utility Function R

- $X \times S \rightarrow R$
- R is the set of ratings.
- R is a totally ordered set
  - 0-5 stars, ranks, $\cdots$

Utility Matrix and Filtering

Finally, the problem becomes filtering the utility matrix.

Key Steps in Recommender Systems

Gathering Known Ratings

- Collect ratings from raw data
- Contruct the utility matrix

Unknown Rating Prediction

- Learn the model with known ratings
- Predict unknown ratings with the model

Evaluation

- Measure the performance of a method

Gathering Known Ratings

Collect known data in the utility matrix

Explicit Ratings

- Ideal Case – Ratings in services
- Ask users to label ratings → But users will feel bad :(

Implicit Ratings

- No exact ratings from users directly
- Learn ratings from user actions

Explicit Ratings: Netflix

- A system with real ratings
- Users rate movies from 1 to 5 (integral) stars.

Implicit Ratings: PChome Online Shopping

- A system without real ratings
- User Actions:
  - Click an item
  - Archive an item in the favorite list
  - Put an item in market basket
  - Purchase an item
- Actions show positive feedback (higher ratings).
- How to find lower ratings?

Evaluation with Explicit Ratings

Regression-oriented Evaluation

- Closer to the ground truths, better performance!
- Root Mean Square Error (RMSE)
  - Widely used in numerical rating prediction.

$$RMSE = \sqrt{\frac{\sum (r_{ui} - \hat{r}_{ui})^2}{n}}$$

Ranking-oriented Evaluation

- Higher rankings, greater preferences
- Normalized Discounted Cumulative Gain (NDCG)
- Rank Correlation
  - Spearman ρ and Kendall τ

Evaluation with Implicit Ratings

Implicit ratings usually have only positive ground truths.

Relevance-oriented Evaluation

- Rank positive items as higher as possible
- Precision at k, Recall at k
- Mean Average Precision (MAP)

Evaluation as Binary Classification

- Treat remaining items as negative ones
- Evaluation focusing on positive items
- Precision, Recall and F-Score

Unknown rating prediction is difficult

Sparseness Problem

- Most users rate few or no items.
- Sparse utility matrix

Cold-start Problem

- New items have no rating.
- New users have no history.

Two Main Approaches

- Content-based Filtering
- Collaborative Filtering

Credit: Daniel Tunkelang

Outline

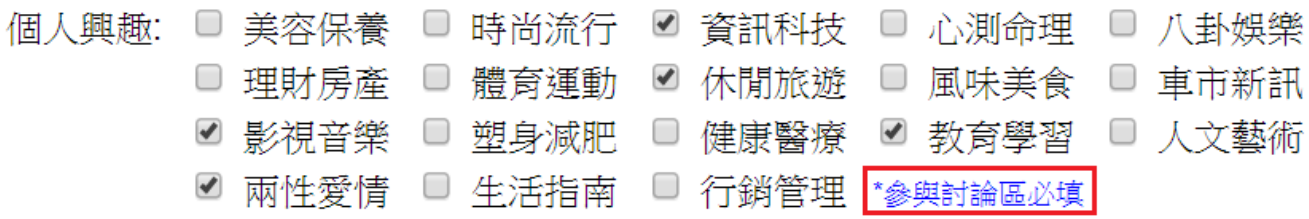
Content-based Filtering

Main Concept

- Build a system based on users' and items' characteristic information.

Big Problem

- Writing profiles is annoying for users.
- Users may refuse to provide information → No user features!



Captured from Mobile01

Pseudo User Profile

Build users' profiles from their actions automatically.

Main Idea

Recommend items similar to previously rated items.

Example

- Movies shared same actors
- News with similar contents

※你可能還想看：

公投會：95.5%贊成入俄 克里米亞17日申請入俄

Captured from CNA news.

Illustration of Building Pseudo User Profile

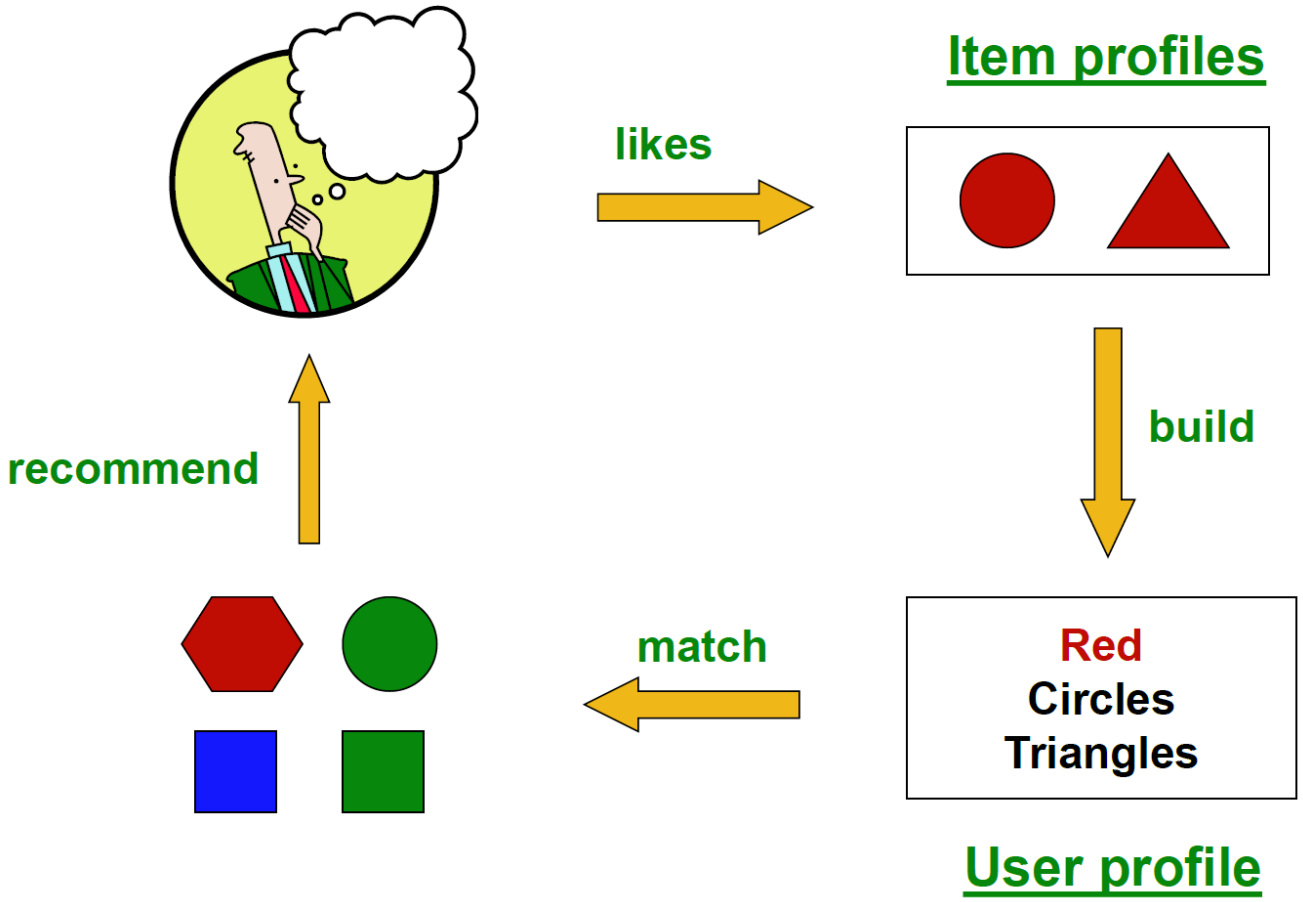


Figure: The flowchart captured from Prof. Leskovec's slides.

Item Profile Representation

Featurize the Items

- For each item, create an item profile.
- Represent an item as a feature vector
  - Movies – author, title, actor, director, $\cdots$
  - Texts – bag of words in contents.

Credit: memegenerator.net

Feature selection is important!

User Profiles and Heuristic Prediction

Pseudo User Profile

- Weighted average of rated item features
- Represent a user with rated items

Heuristic Prediction

- Estimate similarity between profiles in feature space
- Cosine similarity between user profile $\boldsymbol{x}$ and item profile $\boldsymbol{s}$

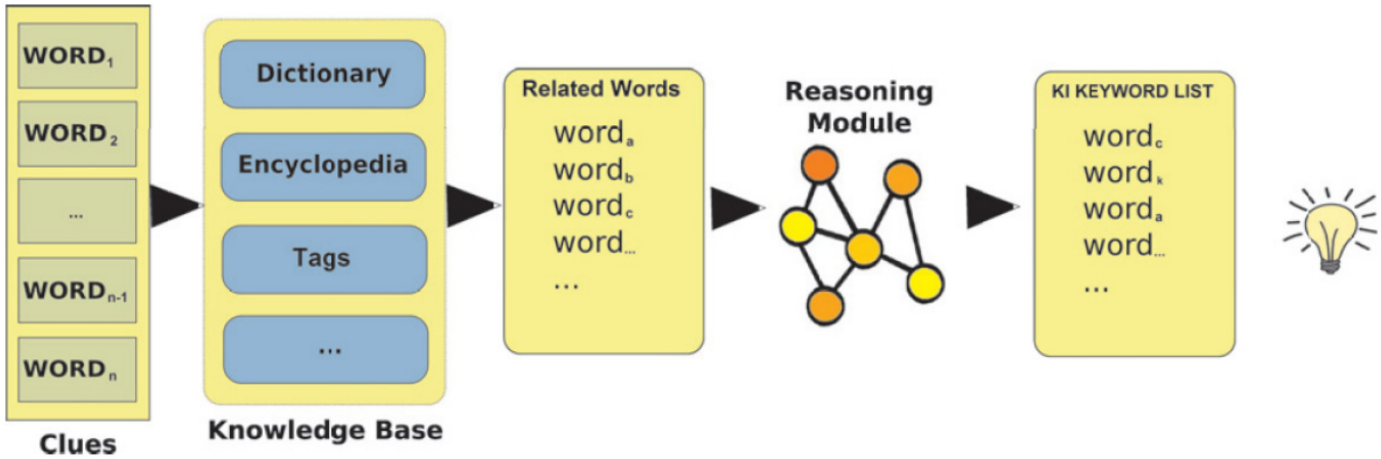
$$u(\boldsymbol{x}, \boldsymbol{s}) = cos(\boldsymbol{x}, \boldsymbol{s}) = \frac{\boldsymbol{x} \cdot \boldsymbol{s}}{||\boldsymbol{x}|| \cdot ||\boldsymbol{s}||}$$

More Information, More Features

- Can easily import external information
- More abundant and precise features, better performance

Example

- Knowledge-based systems contain copious information.
- e.g., Wikipedia, Wordnet, Freebase, ...



Knowledge Infusion Process of Item Profiles [Semeraro et al., RecSys '09]

Short Summary for Content-based Filtering

Advantages

- Recommend to each user respectively
- Able to recommend if user's taste is unique
- No cold-start problem in item side.
- Easy to provide explanations of recommendations

Disadvantages

- Difficult to define appropriate features
- New user might have no user profile
- Users might have multiple interests outside their profiles
- Privacy issues and personal information

Outline

Collaborative Filtering (CF)

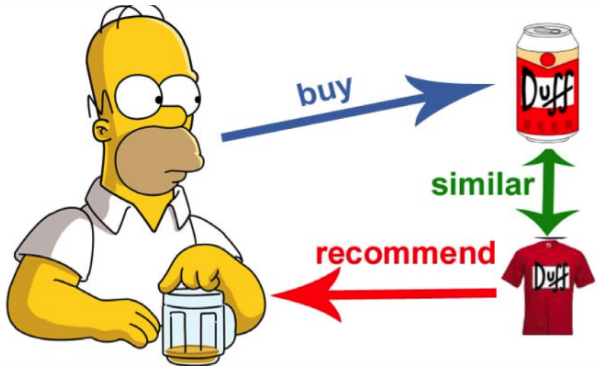
Collaborative: Consider multiple users (or items) at the same time.

Assumption

- Similar items receive similar ratings.
- Similar users have similar preference.

No Feature Extraction

- CF relies on past user behaviors only.
- Predict with user relations and item relations



Back to Utility Matrix

Ratings can be inferred from similar users/items.

An example of utility matrix

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
U_1	5		1		1	2	4	5		
U_2		4		5			4	5	5	4
U_3	4	2	2					4	2	4

k-Nearest Neighbors (kNN)

Main Idea

More similar users/items (neighbors), more correlate rating/rated behavior

User-based kNN vs. Item-based kNN

- Similarity between users or items
- Item-based kNN is used more often
  - Users have multiple tastes
  - #items vs. #users

Two items are similar if both are rated by similar subsets of users.

Similarity between two Items

- Set Correlation

$$c^{Set}(i,j) = \frac{|D(i,j)|}{\min(|D(i)|, |D(j)|)}$$

- Common User Support (CUS)

$$c^{CUS}(i,j) = \frac{n_{ij} \cdot U}{n_i \cdot n_j}, \text{ where } n_{ij} = \sum_{u \in D(i,j)} \frac{1}{|D(u)|}, n_i = \sum_{u \in D(i)} \frac{1}{|D(u)|}$$

- Pearson Correlation (consider rating scores)

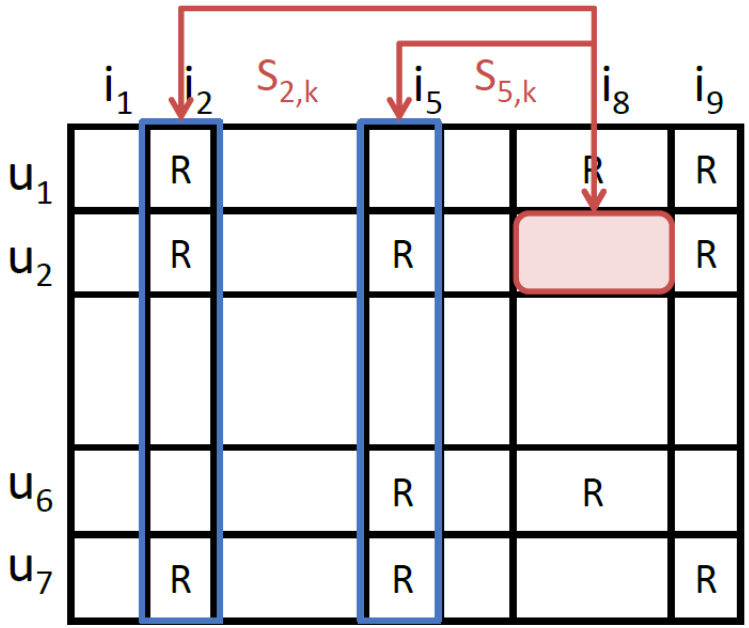
$$c^{Pearson}(i,j) = \frac{\sum_{u \in D(i,j)} (r_{ui} - \overline{r}_i)(r_{uj} - \overline{r}_j)}{\sqrt{\sum_{u \in D(i)} (r_{ui} - \overline{r}_i)^2} \sqrt{\sum_{u \in D(j)} (r_{uj} - \overline{r}_j)^2}}$$

kNN Prediction

- Weighted aggregations of ratings with similarities

$$\hat{r}_{ui} = \frac{\sum_{j \in G_k(u,i)} w_{ij} \cdot r_{uj}}{\sum_{j \in G_k(u,i)} w_{ij}}$$

- k is a parameter deciding the top-k group $G_k(u, i)$.
- Correlation function w decides similarities w_{ij}.



Advantages and Disadvantages of kNN

Advantages

- Easy to understand and implement
- Do not need feature selection

Disadvantages

- Long computation time to calculate similarity
- Popularity bias
- Cold-start problem
- Difficult to find users who rated same items if matrix is too sparse

Outline

The other viewpoint of CF

Collaborative Filtering

- Fill in missing entries in a large matrix
- The matrix has low-rank
  - rows/columns are linearly dependent

New Idea

Fill in the missing entries with matrix algebra

1	2	3	4
2	3	4	?
1	2	?	4

An example of low-rank matrix

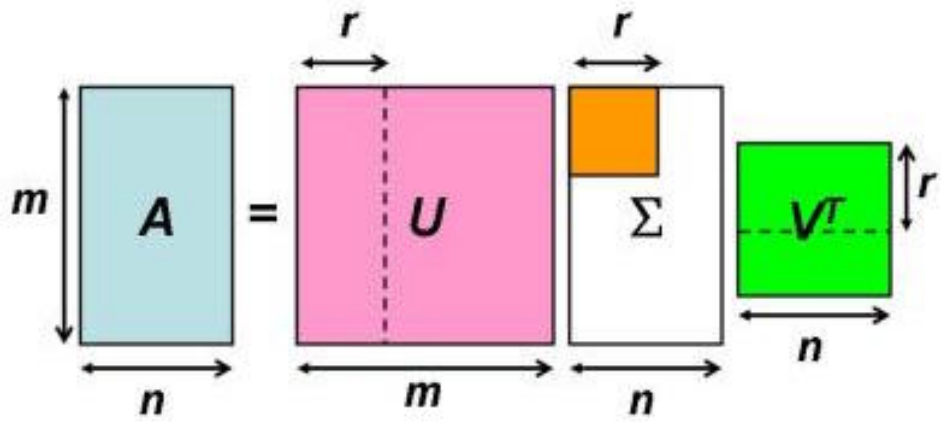
Singular Value Decomposition (SVD)

- For any $m \times n$ matrix, it can be decomposed as:

$$R_{mn} = U_{mm} \Sigma_{mn} V_{nn}^T$$

- U and V are orthonormal matrices ($U^T U = I$, $V^T V = I$)
- Σ is a diagonal matrix with non-negative real values (singular values)

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq \sigma_{r+1} = \cdots = \sigma_n = 0$$



Credit: https://ccjou.wordpress.com/

SVD and Low-rank Approximation

Use fewer singular values and dimensions to approximate R

Low-rank Approximation

- Keep the largest k singular values

$$U_{mk}\Sigma_{kk}V_{nk}^T$$

- The best k-rank approximation of R
- Useful in many applications and algorithms
  - e.g., Principle Component Analysis, Latent Sematic Analysis, ...

A naïve approach

Given an incomplete matrix, fill in some missing entries

Algorithm

- R_0 = fill in missing values by random/zero/mean
- Do SVD on R_0, save its low-rank approximation as R_1
- Do SVD on R_1, save its low-rank approximation as R_2
- ...

Drawbacks

- Bad Performance
  - Bad initial R_0, optimize with wrong values
  - 1% sparsity in Netflix prize dataset
- Very slow (Do SVD in many times...)

New Idea: Latent Vectors (Hidden Factors)

Latent Factor Model

- Represent a user u with a k-dimensional user latent vector p_u
- Represent an item i with a k-dimensional item latent vector q_i

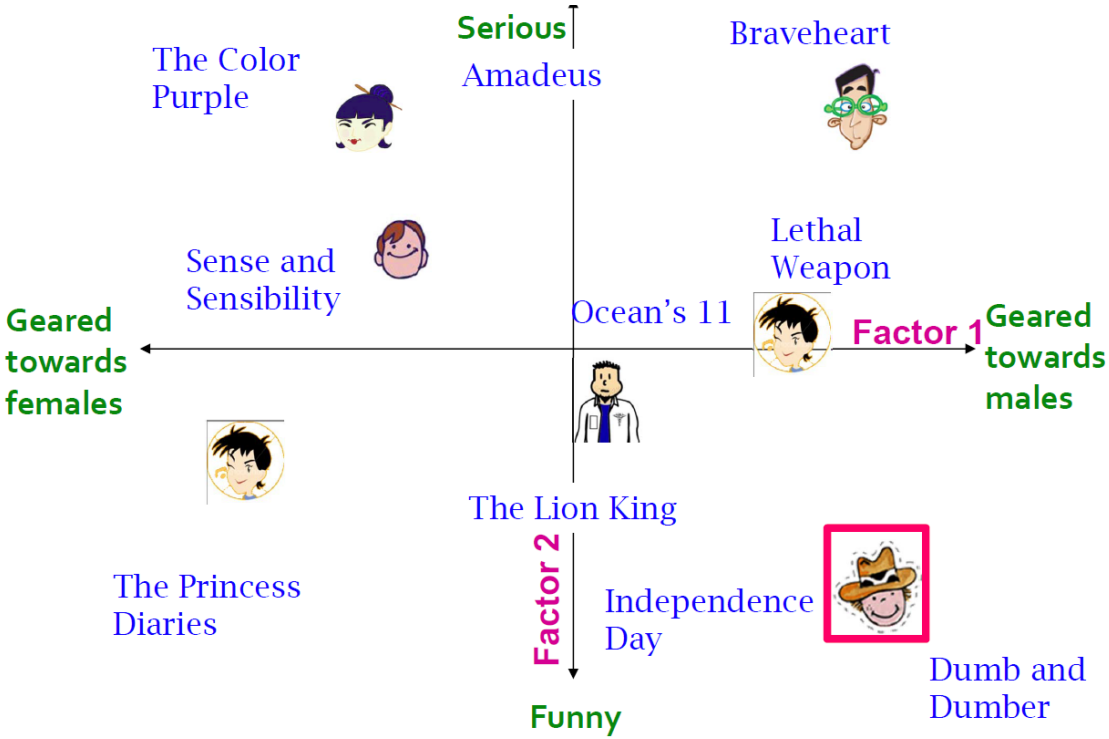


Figure captured from [Koren et al., IEEE Computer Society '09]

Matrix Factorization

Matrix Factorization (MF)

- Approximate R by multiplying latent matrices P and Q

$$R \approx \hat{R} = P \times Q$$

- To predict a rating r_{ui}, multiply the latent vectors

$$\hat{r}_{ui} = p_u^T \cdot q_i$$

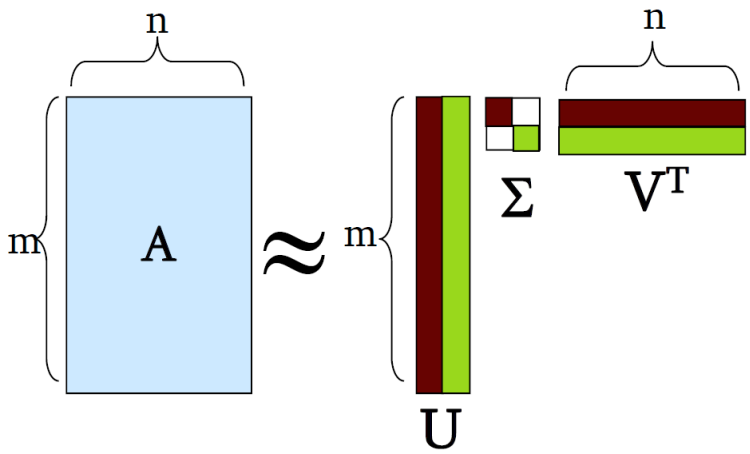
Measure the similarity in latent space (inner product)

From SVD to MF

- We can simply find a form of MF with SVD
  - $R = A$, $P = U$, $Q = \Sigma V^T$

Problem

- SVD cannot be computed while there are some missing entries.
- Filling in random/zero/mean values might lead bad performance.



Find Latent Factors via Optimization

- Lower evaluation measure (RMSE), better performance
- Optimize the objective function based on evaluation measure

Goal

- Find matrices P and Q such that

$$\min_{P,Q} \sum_{(u,i)\in R} \frac{1}{2}\left(r_{ui} - p_u^T \cdot q_i\right)^2$$

Generalization and Overfitting

Generalization in Machine Learning

- Distributions of training and testing data are similar
- Expect good performance in testing data if it is good in training
- Focus on optimizing errors in training data

Overfitting

- Noises must exist in real world
- Too much freedom (parameters) makes the models start fitting noises
- Fitting too well might NOT generalize to unseen testing data

Reduce Overfitting – Regularization

- To avoid overfitting, regularized terms are applied into optimization
  - Allow rich model where are sufficient data
  - Shrink aggressively where data are scarce

$$\min_{P,Q} \sum_{(u,i)\in R} \frac{1}{2}\left(r_{ui} - p_u^T \cdot q_i\right)^2 + \left[\frac{1}{2}\lambda_p \sum_u ||p_u||^2 + \frac{1}{2}\lambda_q \sum_i ||q_i||^2\right]$$

- Consider "length" (complexity penalty) in the latent space

Solving MF: Alternating Least Square (ALS)

Main Idea [Zhou et al., AAIM '08]

- Given P, Q can be solved mathematically [Least Square Problem]
- Given Q, P can be solved mathematically [Least Square Problem]

Least Square Problem

$$\min_{w} \sum \left(y - \boldsymbol{w}^T \boldsymbol{x} \right)^2 + \lambda \boldsymbol{w}^T \boldsymbol{w}$$

Algorithm

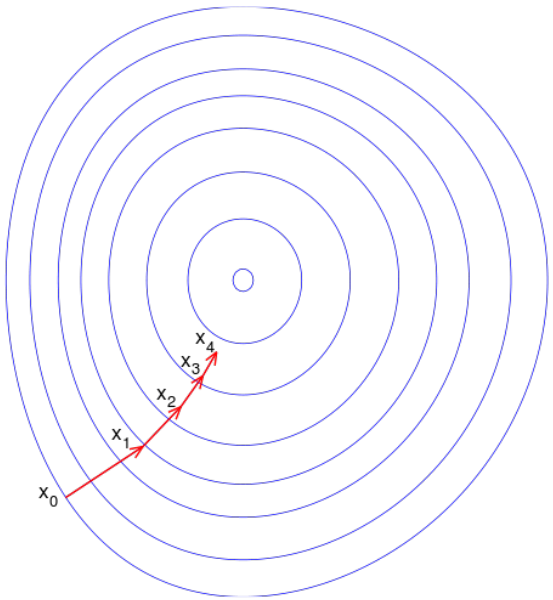
- Randomly initialize P_0
- Solve Q_0 with P_0
- Solve P_1 with Q_0
- ..., until convergence.

Solving MF: Gradient Descent (GD)

- Recap the target of optimization

$$\min_{P,Q} \sum_{(u,i)\in R} \frac{1}{2}\left(r_{ui} - p_u^T \cdot q_i\right)^2 + \left[\lambda_p \sum_u \frac{1}{2}||p_u||^2 + \lambda_q \sum_i \frac{1}{2}||q_i||^2\right]$$

- Initialize P and Q
- Do Gradient Descent
  - $P \leftarrow P - \nabla P$
  - $Q \leftarrow Q - \nabla Q$
  - ∇P and ∇Q are gradients of P and Q
  - $\nabla q_{ik} = \sum_{(u,i)\in R} -(r_{ui} - p_u^T q_i)p_{uk} + \lambda_q q_{ik}$



Problem in Gradient Descent

Slow to compute gradients!

Solving MF: Stochastic Gradient Descent (SGD)

Main Idea

- Total error is sum of individual error
- Calculate gradient using only one rating and update the model

- For each rating,

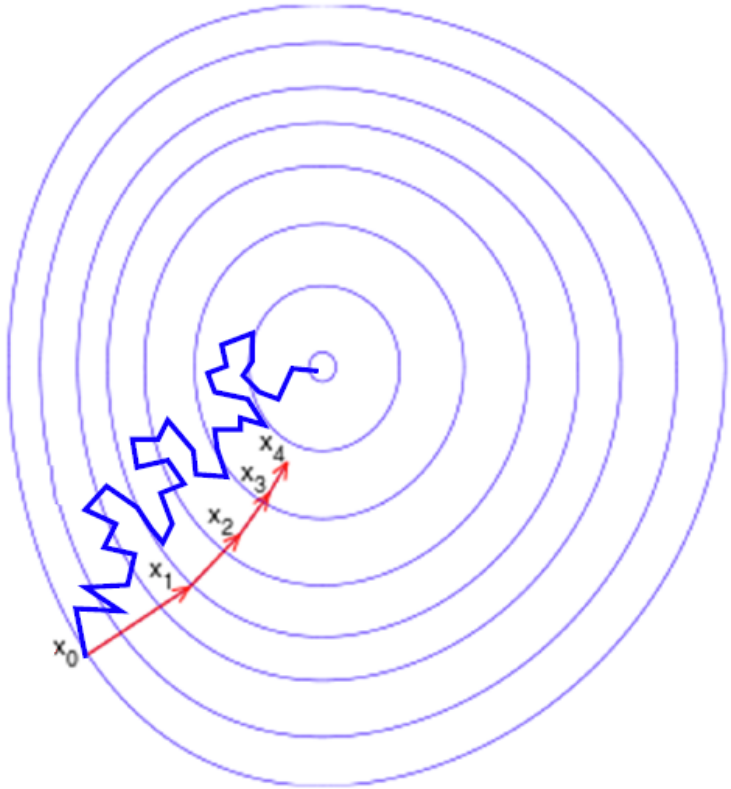
$$p_{uk} \leftarrow p_{uk} - \eta \left(- \left(r_{ui} - p_u^T q_i \right) q_{ik} + \lambda_p p_{uk} \right)$$

$$q_{ik} \leftarrow q_{ik} - \eta \left(- \left(r_{ui} - p_u^T q_i \right) p_{uk} + \lambda_q q_{uk} \right)$$

- Why SGD?
  - Noise but faster to convergence

Convergence of GD and SGD

More steps, but faster!



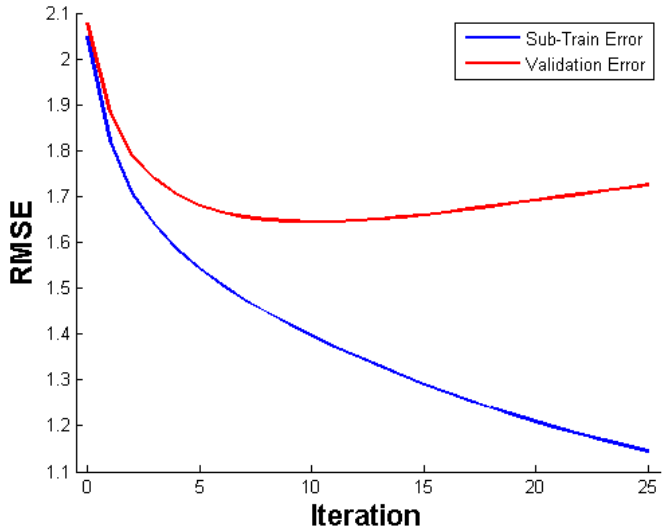
Parameters in SGD

- η, λ are predefined parameters
  - Different parameters make **large differences**
- Try different parameters and observe performance in validation set
  - Validation set: an internal testing set
  - Search parameter exponentially
    - (O) 0.1, 0.05, 0.01, 0.005, ... [exponentially]
    - (X) 0.1, 0.2, 0.3, ... [linearly]

Well tune parameters before you think MF is useless!

Stopping Criterion

- How many iterations are suitable?
  - Too many iterations also make overfitting
- Early Stop
  - Train on a subset of training data
  - Stop when validation RMSE increase
  - Re-train on whole training data and stop on same iterations



Biases in Matrix Factorization

- Include external information into model

Usual Biases

- Overall mean rating μ
- User biases b_u: rating scale
- Item biases b_i: popularity

- Re-write the rating function

$$\hat{r}_{ui} = \mu + b_u + b_i + p_u^T q_i$$

- Re-write the objective function

$$\min_{P,Q} \sum_{(u,i)\in R} \frac{1}{2}(r_{ui} - \hat{r}_{ui})^2 + \left[\lambda_p \sum_u \frac{1}{2}||p_u||^2 + \lambda_q \sum_i \frac{1}{2}||q_i||^2 + \lambda_{b_u} \sum_u \frac{1}{2}||b_u||^2 + \lambda_{b_i} \sum_i \frac{1}{2}||b_i||^2\right]$$

Outline

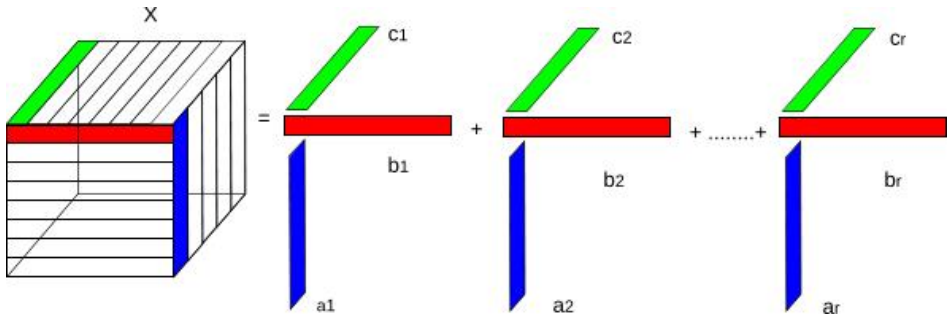
MF with Context: Tensor Factorization

Add more dimensions in factorization models

$$r_{abc} = \sum_k A_{ak} \cdot B_{bk} \cdot C_{ck}$$

Example

- Tag Recommendation [Kendle et al., KDD '09]
- Temporal Factors [Xiong et al., SDM '10]
- Online Review Recommendation [Moghaddam et al., WSDM '12]



Implicit Ratings: One-Class Collaborative Filtering

Recap about Implicit Ratings

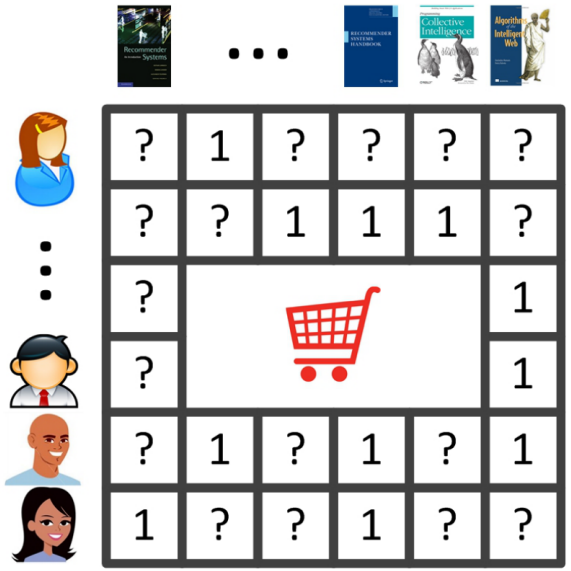
Data of implicit ratings generally have only positive feedback.

Problem

How to generate negative feedback?

Implicit Feedback

- Purchase v.s Non-purchase
- Learn from relations to unseen ratings



Credit: Dr. Weike Pan's Slides

Bayesian Personalized Ranking (BPR)

Optimize AUC (i.e., pair-wise errors) but RMSE

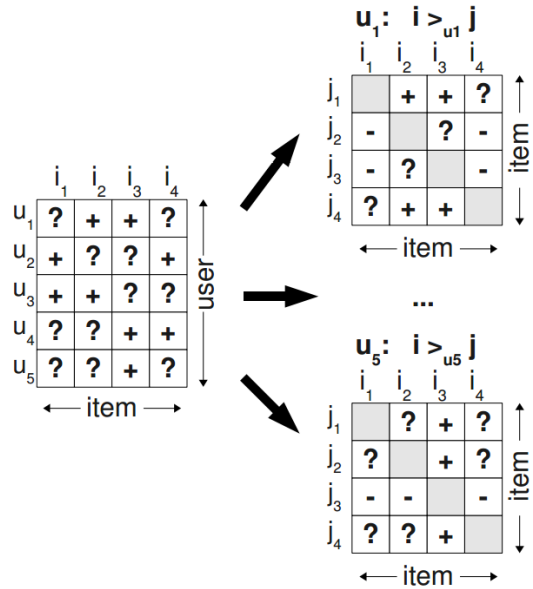
Principle of BPR

If AUC between known and unknown ratings improves,
AUC among unknown ratings will also improve.

BPR Algorithm

- In each iteration of SGD
- Random select a unknown rating for each known rating
- Optimize the pairwise relationship

- More closer to real situations



[Rendle et al., UAI '09]

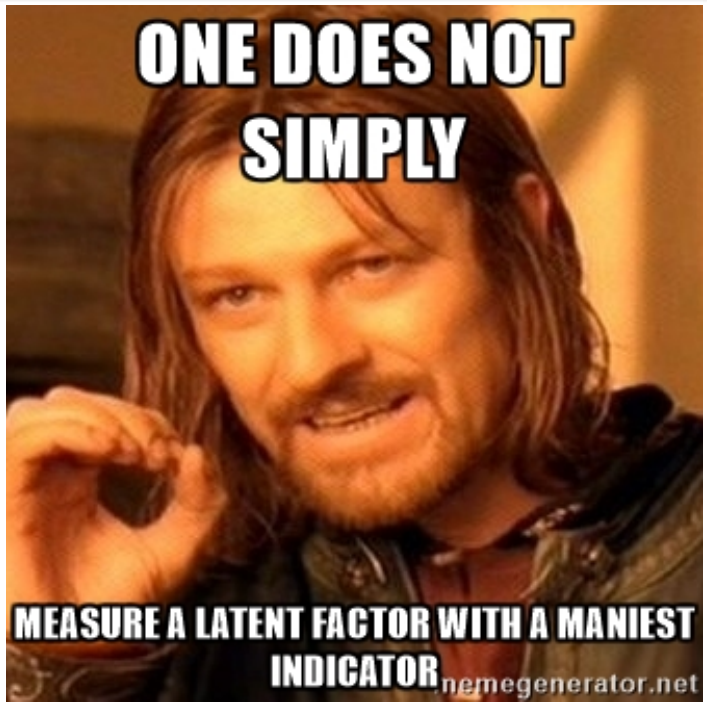
Latent Feature Extraction

Applications of Matrix Factorization

- MF is also helpful for scenarios without explicit features.
- Latent factors can be treated as latent features.
- Some problems can also treated as recommendation.

Example

- Link Prediction [Menon et al., ECML '11]
- Query Suggestion [Ma et al., CIKM '08]
- Advertising [Menon et al., KDD '11]
- Malware Detection [Jiang et al., CCS '14]



Credit: memegenerator.net

Short Summary

In the lecture 4, you have learned ...

- What are recommender systems
- How to gather ratings and evaluate a system
- Algorithms of recommender systems in two types
- Some variations of latent factor models

Thank You!

Any Questions? :)