Other Data Mining Tasks

Dr. Yi Long (Neal)

Most contents (text or images) of course slides are from the following textbook

Provost, Foster, and Tom Fawcett. Data Science for Business: What you need to know about data mining and data-analytic thinking. " O'Reilly Media, Inc.", 2013

Adjustment in Course Assessment

5. **Assessment method**

Component/ method	**% weight**
Assignments and weekly quiz	60
~~Final Exam~~ Group Project	20
~~Course Project~~ Individual Project	20

Last homework will be released on May 15, with one week to finish

Outline

- Lift and Association Rules
- Feature Reduction & Application
- Feature Extraction in Text Mining
- Lab Quiz

A Model of Evidence "Lift"

- Assuming full feature independence:

$$p(c \mid \mathbf{E}) = \frac{p(e_1 \mid c) \cdot p(e_2 \mid c) \cdots p(e_k \mid c) \cdot p(c)}{p(\mathbf{E})} = \frac{p(e_1 \mid c) \cdot p(e_2 \mid c) \cdots p(e_k \mid c) \cdot p(c)}{p(e_1) \cdot p(e_2) \cdots p(e_k)}$$

- Above calculation can be viewed as a product of evidence lifts

$$p(C = c \mid \mathbf{E}) = p(C = c) \cdot \text{lift}_c(e_1) \cdot \text{lift}_c(e_2) \cdots$$

Where,

$$\text{lift}_c(x) = \frac{p(x \mid c)}{p(x)} = \frac{p(x|c) \cdot p(c)}{p(x) \cdot p(c)} = \frac{p(x \wedge c)}{p(x) \cdot p(c)} = \frac{p(c|x) \cdot p(x)}{p(c) \cdot p(x)} = \frac{p(c|x)}{p(c)}$$

Evidence Lifts from Facebook “Likes”

- What people “Like” on Facebook is quite predictive of[1]:
  - ✓ How they score on intelligence tests
  - ✓ Whether they drink alcohol or smoke
  - ✓ Their religion and political views
  - ✓ Whether they are (openly) gay
  - ✓ Whether they drink alcohol or smoke
  - ✓ …

[1] Kosinski, Michal, David Stillwell, and Thore Graepel. "Private traits and attributes are predictable from digital records of human behavior." Proceedings of the National Academy of Sciences 110.15 (2013): 5802-5805.

Evidence Lifts For Target (IQ>130)

Like	Lift	Like	Lift
Lord Of The Rings	1.69	Wikileaks	1.59
One Manga	1.57	Beethoven	1.52
Science	1.49	NPR	1.48
Psychology	1.46	*Spirited Away*	1.45
The Big Bang Theory	1.43	Running	1.41
Paulo Coelho	1.41	Roger Federer	1.40
The Daily Show	1.40	*Star Trek*	1.39
Lost	1.39	Philosophy	1.38
Lie to Me	1.37	*The Onion*	1.37
How I Met Your Mother	1.35	*The Colbert Report*	1.35
Doctor Who	1.34	*Star Trek*	1.32
Howl's Moving Castle	1.31	Sheldon Cooper	1.30
Tron	1.28	*Fight Club*	1.26
Angry Birds	1.25	*Inception*	1.25
The Godfather	1.23	*Weeds*	1.22

Wal-Mart: How to Arrange

- It is intuitive to sell baby-related products to new parents.
  - ✓ The arrival of a new baby in a family is one point where people do change their shopping habits significantly. In the Target analyst's word, "As soon as we get them buying diapers from us, they're going to start buying everything else too".
- "Men often bought beer at the same time they bought diapers." (very famous, known as market basket analysis)

Co-occurrences and Associations

- Co-occurrence grouping or association discovery attempts to find associations between entities based on transactions involving them
  - ✓ Transactions of entities
  - ✓ Association rule: beer -> diaper

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

market basket transactions

Metrics for Association Rules

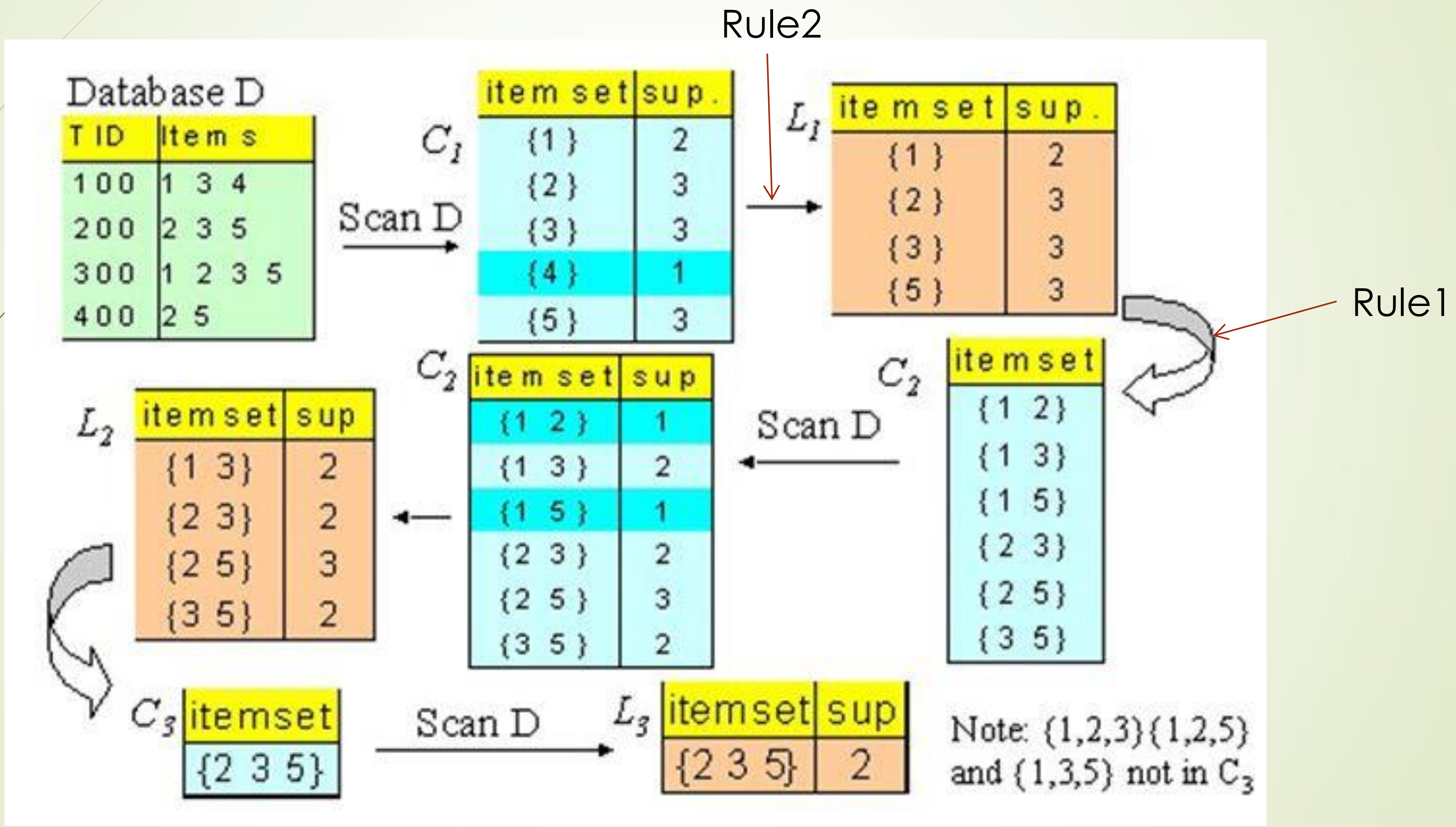
- **Support** is an indication of how frequently these items appears in the dataset
  - ✓ Let's say that we require the items of a rule should appear in at least 0.01% of all transactions
- **Confidence** or **strength** of a rule is an indication of how often the rule has been found to be true
  - ✓ Confidence(X->Y) = support(X,Y)/support(X)
- **Lift** of a rule is the ratio of the observed support to that expected if X and Y were independent
  - ✓ Lift(X<->Y) = support(X,Y)/(support(X)* support(Y))

$$\text{lift}_c(x) = \frac{p(x \mid c)}{p(x)} = \frac{p(x|c) \cdot p(c)}{p(x) \cdot p(c)} = \frac{p(x \wedge c)}{p(x) \cdot p(c)} = \frac{p(c|x) \cdot p(x)}{p(c) \cdot p(x)} = \frac{p(c|x)}{p(c)}$$

Apriori Algorithm

- Apriori algorithm assumes that
  - ✓ Rule 1: All subsets of a frequent itemset must be frequent
  - ✓ Rules 2: For any infrequent itemset, all its supersets must be infrequent too
- Apriori algorithm identify those frequent itemset with support higher than a given threshold value as follows:
  - ✓ Initialization: Start with itemsets containing just a single item, such as {beer} and {diaper}
  - ✓ Step 1. Determine the support for itemsets, and remove itemsets that do not meet your minimum support threshold (Rule2).
  - ✓ Step 2. Using the itemsets you have kept from Step 1, generate all the possible itemset configurations, and delete some new itemsets (supersets) based on Rule1 .
  - ✓ Step 3. Repeat Steps 1 & 2 until there are no more new itemsets.

Example: Apriori Algorithm



Identify Rules with High Confidence/Lift

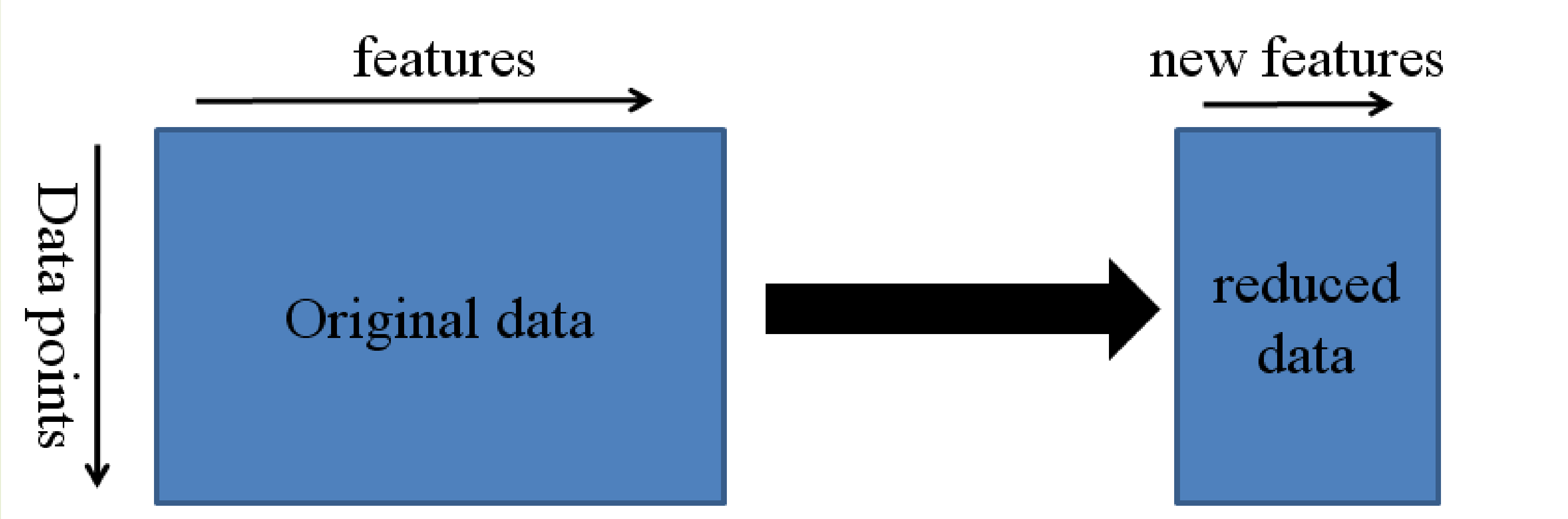
- The same principle can also be used to identify item associations with high confidence or lift.
- Once high-support itemsets have been identified, computing confidence or lift is easy because confidence and lift values are calculated using support values
- Disadvantage：
  - ✓ Computationally Expensive. Even though the apriori algorithm reduces the number of candidate itemsets to consider, this number could still be huge when store inventories are large or when the support threshold is low
  - ✓ Spurious Associations. Analysis of large inventories would involve more itemset configurations, and the support threshold might have to be lowered to detect certain associations. However, lowering the support threshold might also increase the number of spurious associations detected.

Outline

- Lift and Association Rules
- Feature Reduction & Application
- Feature Extraction in Text Mining
- Lab Quiz

Data Reduction and Latent Dimensions

- We would like to take a large set of data and replace it with a smaller set that
  - ✓ Preserves much of the important information in the original data while mitigating the impact of noise in data (may sacrifice some information as well)
  - ✓ The smaller dataset may be easier to deal with or to process



Data Reduction via Clustering

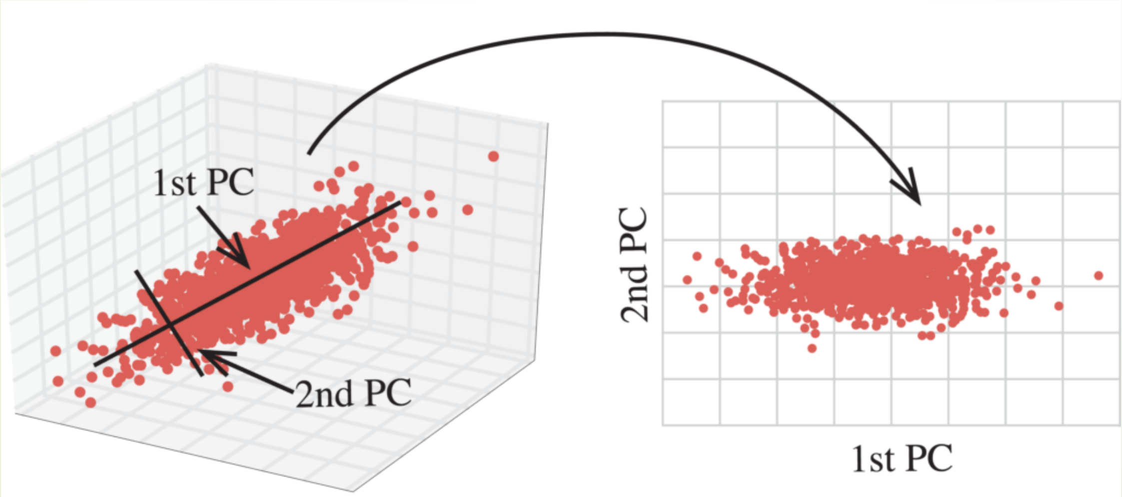
- cluster.FeatureAgglomeration applies Hierarchical clustering to group together features that behave similarly.

User/movie	小时代1	小时代2	Lord of The Rings 1	Lord of The Rings 2	Lord of The Rings 3
Lily	1	0	1	0	1
Jim	0	1	1	1	1
…	…	…	…	…	…

User/movie	小时代	Lord of The Rings
Lily	1	1
Jim	1	1
…	…	…

Data Reduction via PCA

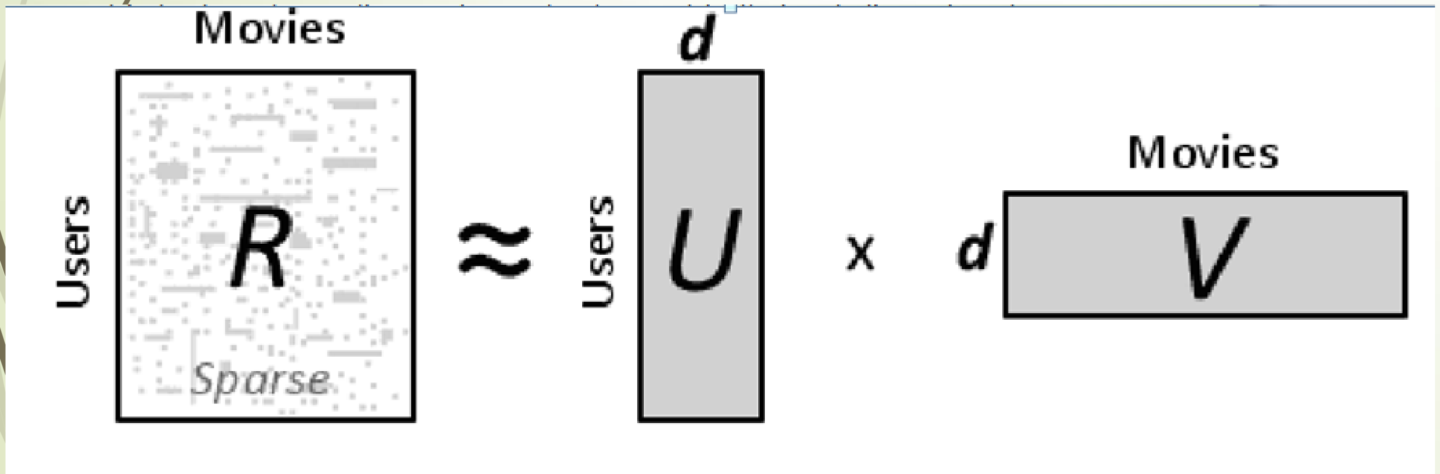
- Using a technique called principal components analysis (or PCA), we can reduced the dimensionality of a dataset, while preserving as much of its precious variance as possible.
  - ✓ Convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components (**PC**)
  - ✓ The first PC has the largest possible variance, then the second PC, the third …

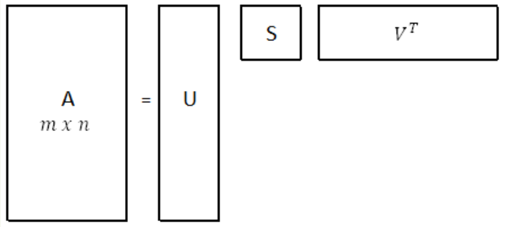


decomposition.PCA in Sklearn.

Latent Information via Reduction

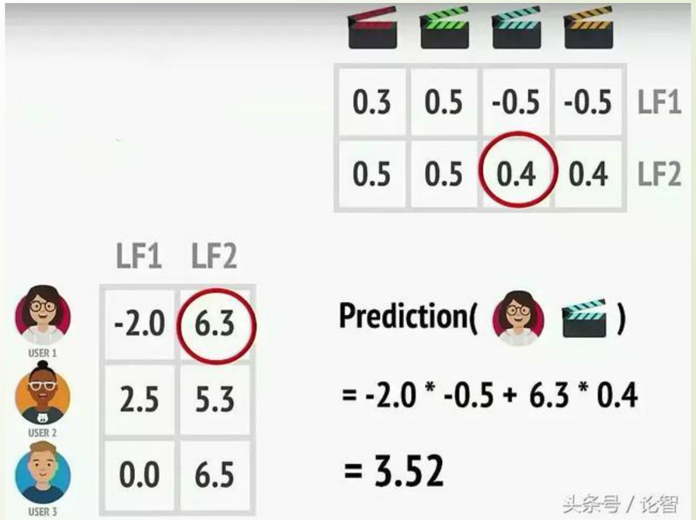
- Singular Value Decomposition (SVD) can decompose the user-movie preference matrix to two low-dimensioned sub-matrix (latent dimesions)
  - ✓ Represent each movie as a feature vector using the latent *d*-dimensions
  - ✓ Represent each user's preferences as a feature vector using the same d-latent dimensions as well



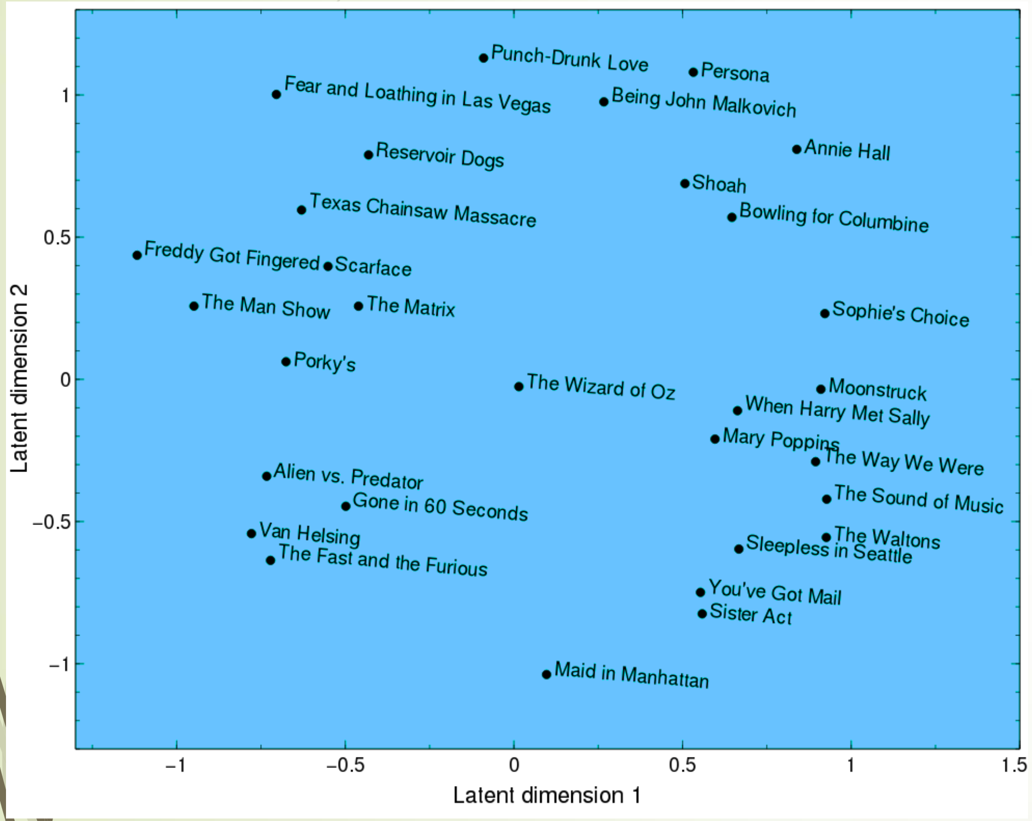


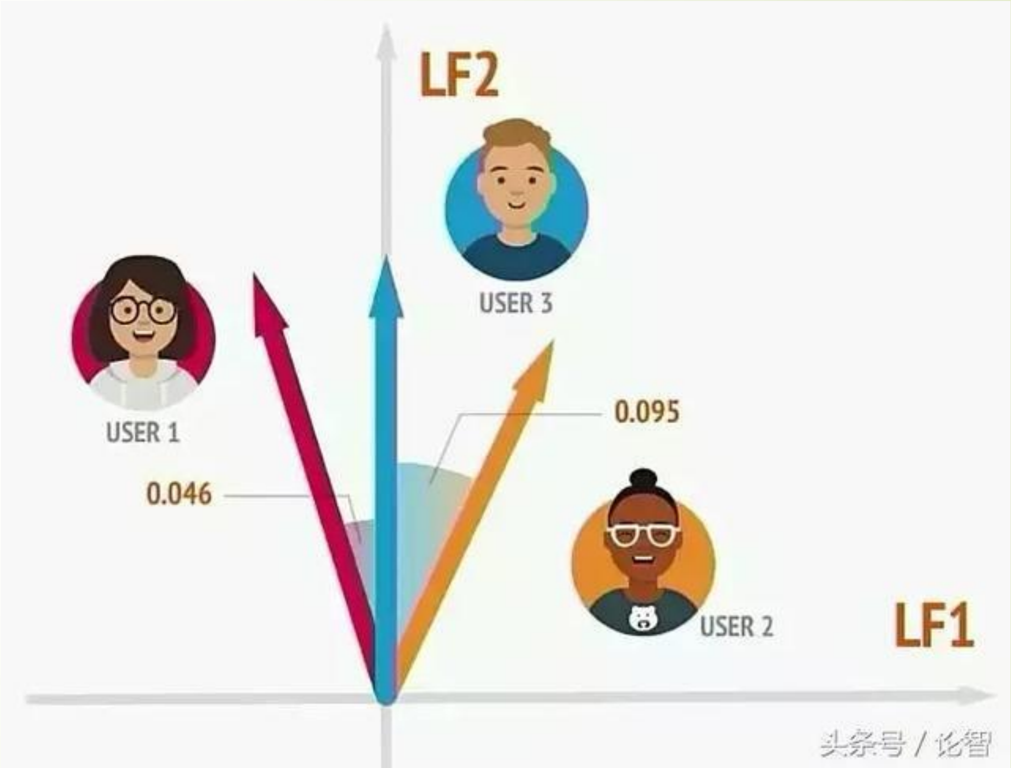
Latent Factor (LF) for Recommendation

?



Latent Factor (LF) for Similarity





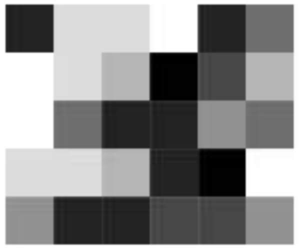
A collection of movies placed in a “taste space” defined by the two strongest latent dimensions mined from the Netflix Challenge data.

Outline

- Lift and Association Rules
- Feature Reduction & Application
- Feature Extraction in Text Mining
- Lab Quiz

Feature Extraction

- Transforming arbitrary data, such as text or images, into numerical features usable for machine learning.
  - ✓ Unstructured -> structured data



$$\begin{bmatrix} 6 & 1 & 1 & 0 & 6 & 4 \\ 0 & 1 & 2 & 7 & 5 & 2 \\ 0 & 4 & 6 & 6 & 3 & 4 \\ 1 & 1 & 2 & 6 & 7 & 0 \\ 3 & 6 & 6 & 5 & 5 & 3 \end{bmatrix}$$

6.2.4. Image feature extraction ¶

6.2.4.1. Patch extraction

The `extract_patches_2d` function extracts patches from an image stored as a two-dimensional array, or three-dimensional with color information along the third axis. For rebuilding an image from all its patches, use `reconstruct_from_patches_2d`. For example let use generate a 4x4 pixel picture with 3 color channels (e.g. in RGB format):

https://scikit-learn.org/stable/modules/feature_extraction.html

Text Mining Is Important

- Most of our knowledge and information are represented and transmitted via text naturally
  - ✓ Books, newspaper, reports, research paper, medical records, consumer complaint logs …
  - ✓ Internet contains a vast amount of text in the form of personal web pages, Twitter feeds, email, Facebook status updates, product descriptions and blog postings …

证券研究报告

华泰证券
HUATAI SECURITIES

公司研究 / 公告点评
2016年08月02日

建筑 / 建筑装饰Ⅱ

投资评级：买入（维持评级）	
当前价格(元):	26.19
合理价格区间(元):	33.7~37.1

鲍荣富 执业证书编号: S0570515120002
研究员 021-28972085

外延并购再下一城，省外市场拓展加速

东易日盛(002713)

拟现金收购上海创域，外延并购加速

公司公告拟以现金 11,220 万元收购上海创域实业 51%股权，并约定 3 年业绩承诺期满达到约定条件的，公司将继续收购剩余 29%股权。创域实业整合的关联方上海关镇

Text Mining in Accounting

No.C2018013 2018-11-15

文本大数据分析在经济学和金融学中的应用：

一个文献综述

沈艳 、陈赟、黄卓
北京大学国家发展研究院

Journal *of* Accounting Research

DOI: 10.1111/1475-679X.12123
Journal of Accounting Research
Vol. No. xxxx 2016
Printed in U.S.A.

CHICAGO BOOTH

Textual Analysis in Accounting and Finance: A Survey

TIM LOUGHRAN* AND BILL MCDONALD*

Text Mining is Difficult

- Text data is unstructured data
  - ✓ Text is of different length, and is not like records with fields having fixed meanings
  - ✓ Linguistic structure of text is intended for human consumption, not for computers
- Text data is relatively dirty
  - ✓ People write ungrammatically: misspell words, run words together, abbreviate unpredictably, and punctuate randomly
  - ✓ Contain synonyms (multiple words with the same meaning) and homographs (one spelling shared among multiple words with different meanings)

    Synonym: “good”, “nice” Homographs: bear (verb) – to carry, bear (noun) – the animal
- “Context” matters
  - ✓ Take sentiment analysis for example: Is “incredible” positive or negative

    “我的汽车音响声音很大” vs. “我的电脑风扇声音很大”

Text Mining is More Difficult For Chinese

- In general, words in English is separated by white space
  - ✓ Except for proper nouns, such as New York
- Word segmentation is usually the first step when handling Chinese text
  - ✓ Definition of "word/phrase" is subjective: 随地吐痰者 is a word or phrase
  - ✓ Word segmentation is difficult

    "乒乓球拍/卖/完了" vs "乒乓球/拍卖/完了"

    "说/的/确实/在理" "说/的确/实在/理"
  - ✓ Word segmentation is important for understanding

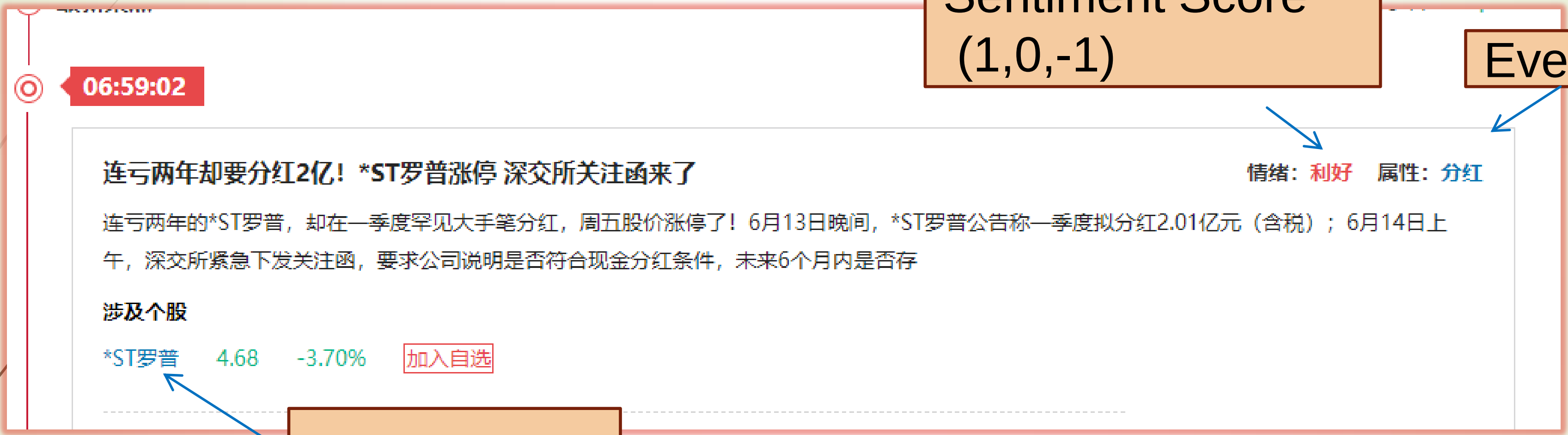
Text Preprocessing

- ***Tokenization***/word segmentation
- The case should be normalized (***case normalization***)
  - ✓ Every term is in lowercase such that iPhone = iphone = IPHONE
- Words should be ***stemmed***
  - ✓ Suffixes are removed, such as noun plurals are transformed to singular forms
  - ✓ cats = cat, see = saw, running = run …
- ***Stop-words*** should be removed
  - ✓ A stop-word is a very common word (should be careful)
  - ✓ Typical words such as the words "*the, and, of, on (or 是, 的, 呢,啊)* " are removed

Text Analytics Based on Dictionary?

Sentiment Score (1,0,-1)

Event


06:59:02

连亏两年却要分红2亿！*ST罗普涨停 深交所关注函来了

情绪：利好 属性：分红

连亏两年的*ST罗普，却在一季度罕见大手笔分红，周五股价涨停了！6月13日晚间，*ST罗普公告称一季度拟分红2.01亿元（含税）；6月14日上午，深交所紧急下发关注函，要求公司说明是否符合现金分红条件，未来6个月内是否存

涉及个股

*ST罗普 4.68 -3.70% 加入自选

Related Firm

“该项新措施可降低风险，减少损失”

Negative

“英国经济“脱欧红利”恐难持久”

Positive

大湾区水域洪涝灾害跨区域救援实战演练举行

Flood Alert

Text Representation

- Text representation: taking a set of documents and turning it into our familiar feature-vector form (each document is an instance)

Doc1:广汽集团携手腾讯发展智能汽车
Doc2: 深交所发函质疑大连友谊资产重组
Doc3: 支付宝发大招限制现金贷利率
…

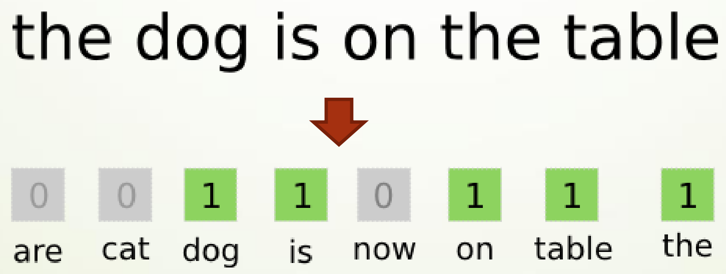
0	1	1
0	1	1
1	1	0

…

- A collection of documents is called a ***corpus***
- A ***document*** is a relatively freeform sequence of individual tokens
- A ***token*** is an instance of a sequence of characters that are grouped together as a useful semantic unit for processing
  - ✓ Words (assume to be words here): “不明觉厉”，”New York”
  - ✓ Numbers: 69,236.12, 27%
  - ✓ Emoticons: 
  - ✓ Punctuations
  - ✓ …

“Bag of Words”

- Treat every document as just a collection of individual words
  - ✓ Ignore grammar, word order, sentence structure, and (usually) punctuation
  - ✓ Treat every word in a document as a potentially important keyword of the document
- Each feature in the feature vector corresponds to a fixed token, and
  - ✓ The feature of a document is represented by a one (if the token is present in the document) or a zero (the token is not present in the document)



Term Frequency

- We can set the value of features is bag-of –word models by word count (frequency) in the document instead of just a zero or one
  - ✓ Differentiates between how many times a word is used
  - ✓ The importance of a term in a document should increase with the number of times that term occurs

d1	jazz music has a swing rhythm
d2	swing is hard to explain
d3	swing rhythm is a natural rhythm

➡

	a	explain	hard	has	is	jazz	music	natural	rhythm	swing	to
d1	1	0	0	1	0	1	1	0	1	1	0
d2	0	1	1	0	1	0	0	0	0	1	1
d3	1	0	0	0	1	0	0	1	2	1	0

Normalized Term Frequency

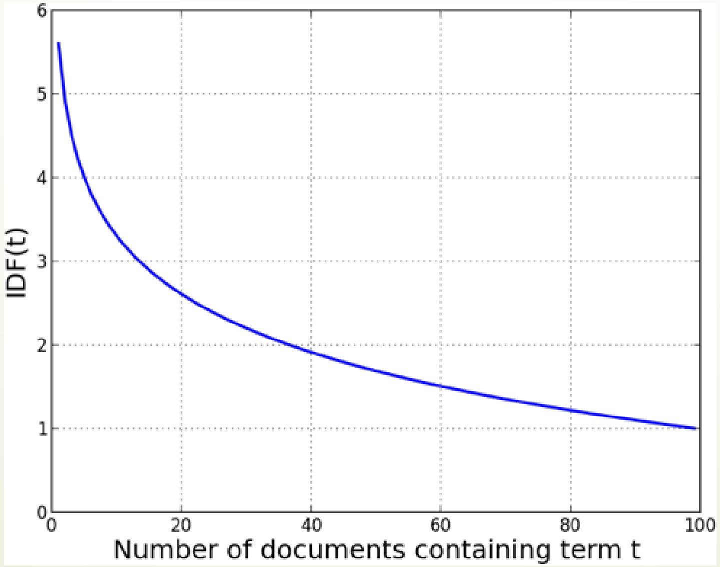
- Documents are of various lengths
  - ✓ Long documents usually will have more words—and thus more word occurrences—than shorter ones
  - ✓ The purpose of term frequency is to represent the relevance of a term to a document.
- The raw term frequencies are normalized in some way
  - ✓ Divided by the total number of words in the document: *tf = raw_tf/total_number_of_words*
  - ✓ Divided by l1-norm or l2-norm as (p=1 or 2) $tf = \frac{raw_t f}{\|raw\_tf\_vector\|_p}$
  - ✓ Logarithmically scaled frequency: *tf = log(1+raw_tf)*
- It matters that how common/sparse a word is in the entire corpus we're mining
  - ✓ Words cannot be too common, otherwise cannot distinguish documents effectively
  - ✓ Words cannot be too rare: only appears in one document cannot used for meauring similarity

IDF

- Inverse document frequency (IDF) is widely used to measure the sparseness of a term *t*

$$\mathrm{IDF}(t) = 1 + \log\left(\frac{\text{Total number of documents}}{\text{Number of documents containing } t}\right)$$ or $$\mathrm{IDF}(t) = \log\left(\frac{\text{Total number of documents}}{\text{Number of documents containing } t}\right)$$

- IDF decreases quickly as *t* becomes more common in documents, and finally approach 1.0 (it appears in all documents)



TF-IDF

- A very popular representation for text is the product of Term Frequency (TF) and Inverse Document Frequency (IDF), commonly referred to as TFIDF or (TF-IDF)
- The TFIDF value of a term *t* in a given document *d* is

$$\mathrm{TFIDF}(t, d) = \mathrm{TF}(t, d) \times \mathrm{IDF}(t)$$

- Note that the TFIDF value is specific to a single document (*d*) whereas IDF depends on the entire corpus
  - ✓ Extracting keywords of a document: select words appears frequently in the document, but uncommon in other documents

	angeles	los	new	post	times	york
d1	0	0	1	0	1	1
d2	0	0	1	1	0	1
d3	1	1	0	0	1	0

	angeles	los	new	post	times	york
d1	0	0	0.584	0	0.584	0.584
d2	0	0	0.584	1.584	0	0.584
d3	1.584	1.584	0	0	0.584	0

$$log_2(3/2) = log_2(1.5) \approx 0.584$$

Bag-of-words Approach

- The bag-of-words approach treats every word in a document as an independent potential keyword (feature) of the document, the values can be (you determine):
  - ✓ Binary (1 for appearance, 0 for absence)
  - ✓ Term frequency (either raw or normalized frequency)
  - ✓ TF-IDF score
- The bag-of-words approach is
  - ✓ Straightforward representation
  - ✓ Inexpensive to generate
  - ✓ Tends to work well for many tasks
  - ✓ Sometimes too simple to capture the text structure

Example: Jazz Musicians

- 15 prominent jazz musicians and excerpts of their biographies from Wikipedia

Charlie Parker

Charles "Charlie" Parker, Jr., was an American jazz saxophonist and composer. Miles Davis once said, "You can tell the history of jazz in four words: Louis Armstrong. Charlie Parker." Parker acquired the nickname "Yardbird" early in his career and the shortened form, "Bird," which continued to be used for the rest of his life, inspired the titles of a number of Parker compositions, [...]

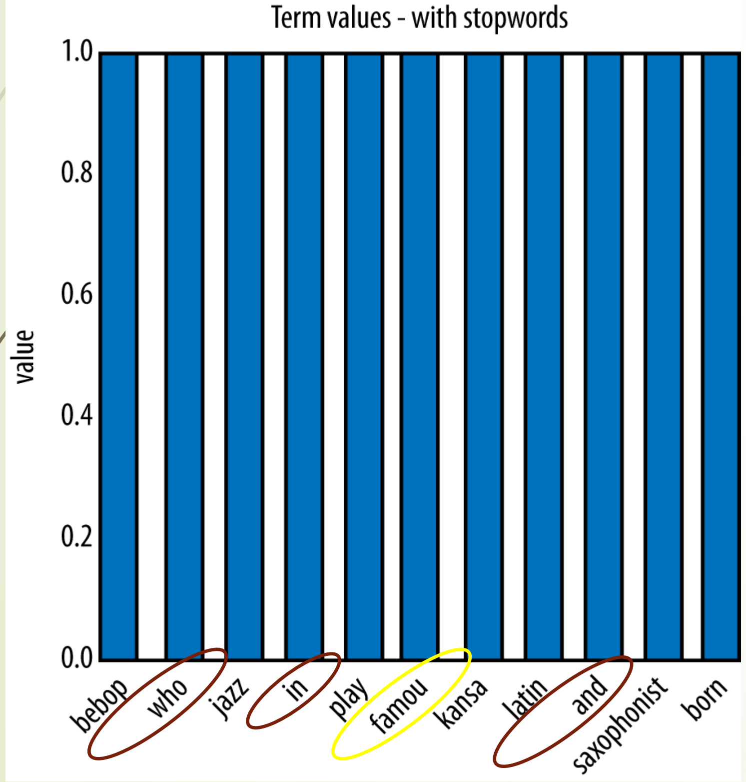
Duke Ellington

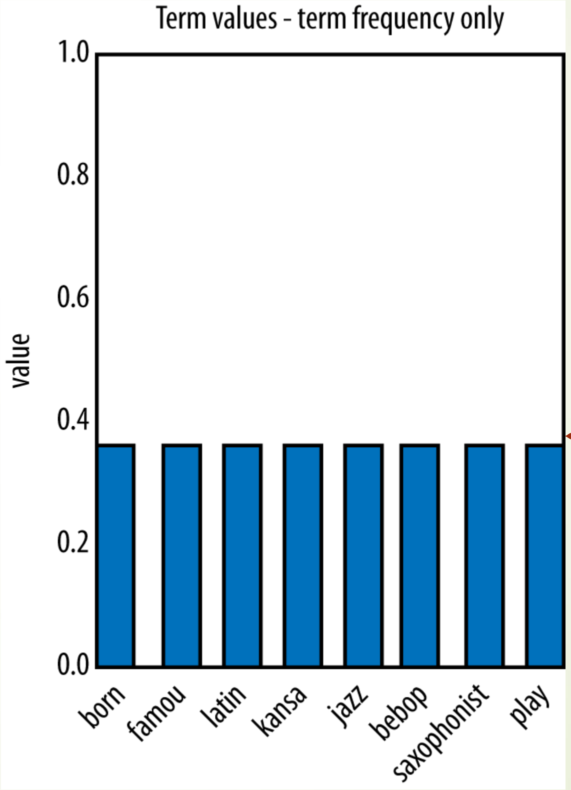
Edward Kennedy "Duke" Ellington was an American composer, pianist, and big-band leader. Ellington wrote over 1,000 compositions. In the opinion of Bob Blumenthal of *The Boston Globe*, "in the century since his birth, there has been no greater composer, American or otherwise, than Edward Kennedy Ellington." A major figure in the history of jazz, Ellington's music stretched into various other genres, including blues, gospel, film scores, popular, and classical.[...]

- Nearly 2,000 features after stemming and stop-word removal!
- Consider the sample phrase "Famous jazz saxophonist born in Kansas who played bebop and latin"

Example: Jazz Musicians (TF)

- Raw count vs. l2-normalized frequency without stop words



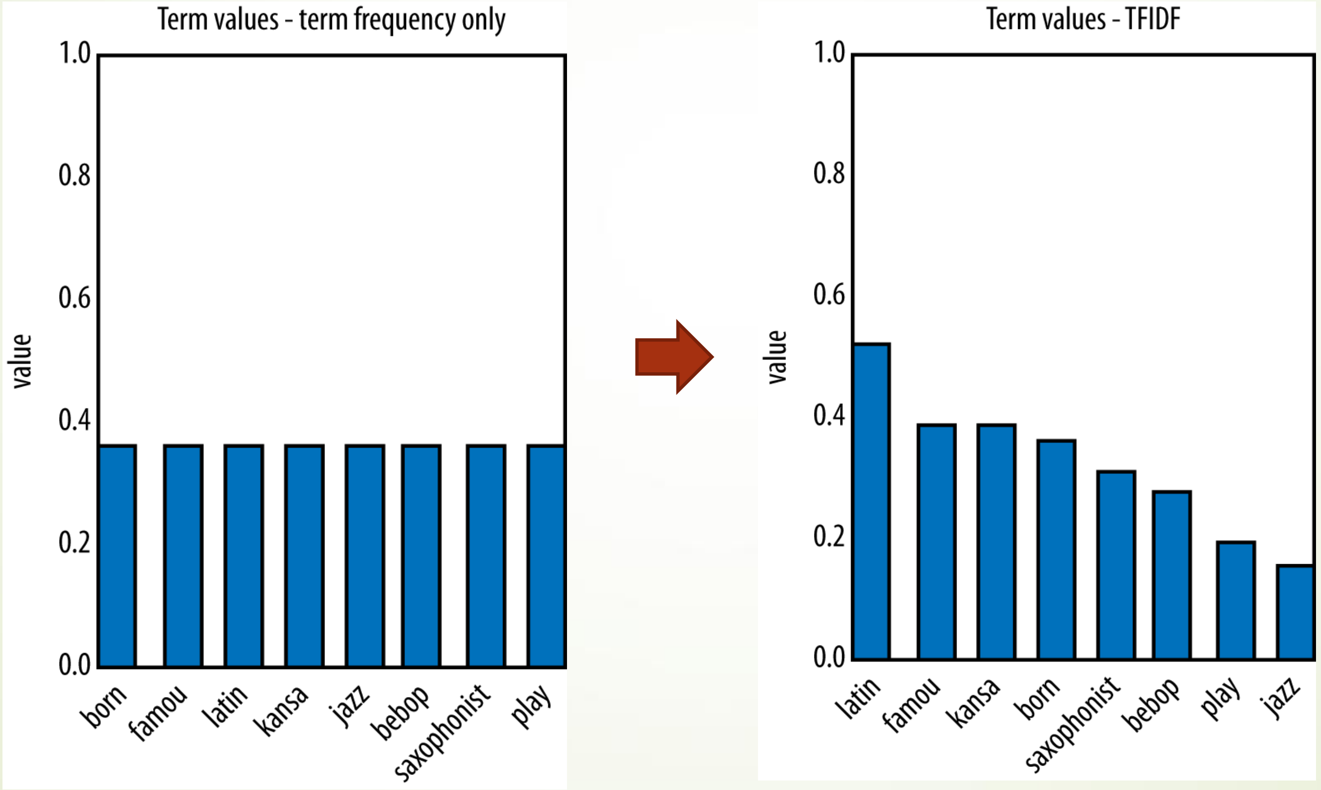


$$\frac{1}{\sqrt{8}} \approx 0.353$$

Representation of the query "Famous jazz saxophonist born in Kansas who played bebop and Latin" after stemming.

Example: Jazz Musicians (TFIDF)

- We can further compute TFIDF score



Representation of the query "Famous jazz saxophonist born in Kansas who played bebop and Latin" after stemming.

Example: Jazz Musicians (Similarity)

- Similarity of each musician's text to the query 'Famous jazz saxophonist born in Kansas who played bebop and latin,' ordered by decreasing **cosine similarity**.

Musician	Similarity	Musician	Similarity
Charlie Parker	0.135	Count Basie	0.119
Dizzie Gillespie	0.086	John Coltrane	0.079
Art Tatum	0.050	Miles Davis	0.050
Clark Terry	0.047	Sun Ra	0.030
Dave Brubeck	0.027	Nina Simone	0.026
Thelonius Monk	0.025	Fats Waller	0.020
Charles Mingus	0.019	Duke Ellington	0.017
Benny Goodman	0.016	Louis Armstrong	0.012

$$\text{Cosine\_similarity}(\mathbf{X},\mathbf{Y}) = \frac{\mathbf{X}\cdot\mathbf{Y}}{\|\mathbf{X}\|_2 \cdot \|\mathbf{Y}\|_2}$$

Drawbacks of BoW

China defeats **Brazil**, and qualify for the next round in FIFA Worldcup!

|| ?

Brazil defeats **China** , and qualify for the next round in FIFA Worldcup!

N-gram Model

- In some cases, **word order** is important and you want to preserve some information about it in the representation
- We can include sequences of ***n*** adjacent words as terms/tokens (***n-grams***)
  - ✓ Adjacent pairs are commonly called **bi-grams.**

    "The quick brown fox jumps" ➡ {quick, brown, fox, jumps, quick_brown, brown_fox, fox_jumps}
  - ✓ "bag of n-grams up to three" it simply means representing each document using as features its individual words, adjacent word pairs, and adjacent word triples.
- N-grams greatly increase the size of the feature set

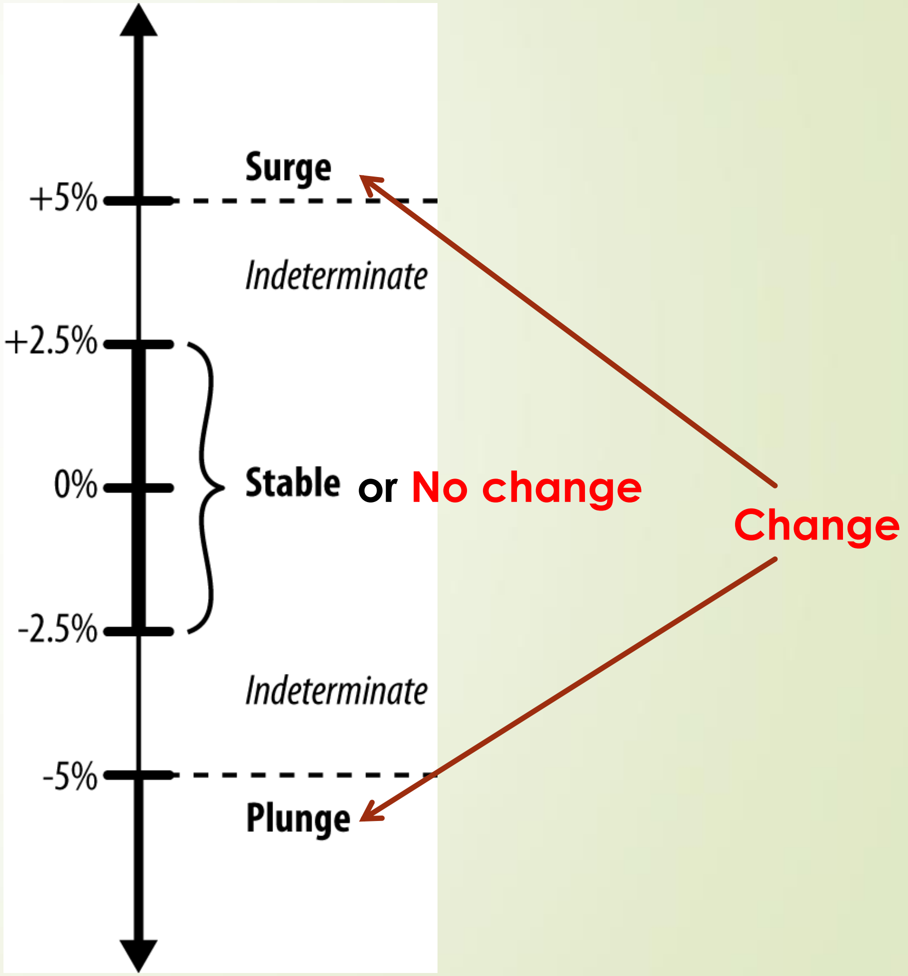
When using <u>up to</u> bi-gram

"Is this good" as {this, is, good, is_this, this_good}

"this Is good" as {this, is, good, this_is, is_good}

Example: Predicting the Stock Market (1)

- **Task**: predict the stock market based on the stories that appear on the news
  - ✓ It is difficult to predict the effect of news far in advance. Hence we'll try to predict what effect a news story will have on stock price the same day.
  - ✓ It is difficult to predict exactly what the stock price will be. Hence we'll simplify predict whether the stock price will **change** and **no change**.
  - ✓ We will assume that only news stories mentioning (usually inaccurate) a specific stock will affect that stock's price.



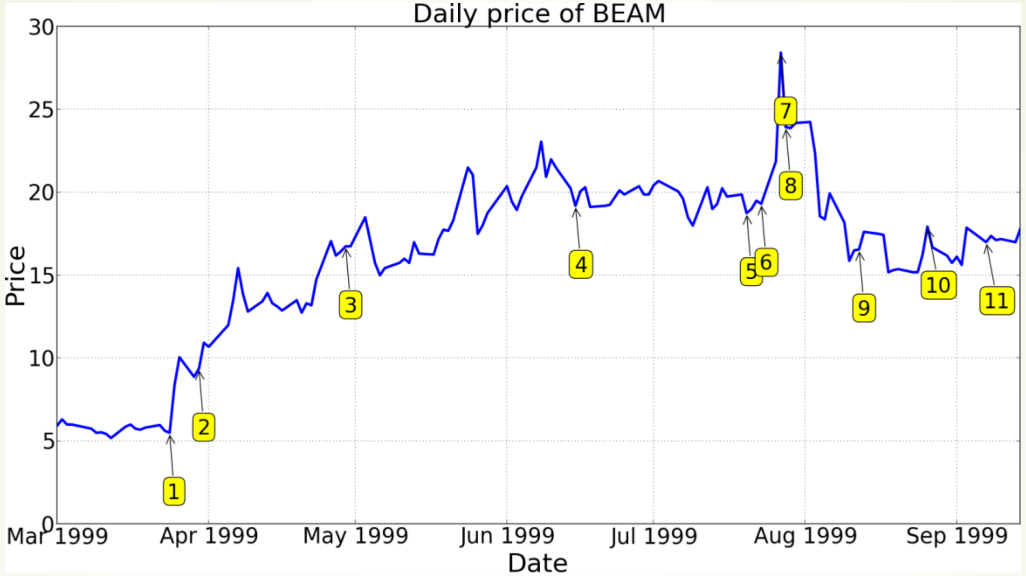
Example: Predicting the Stock Market (2)

- Using bi-grams to represent the news
  - ✓ Compute the daily percentage change - divide the difference between the day's prices at 4 pm (closing) and 10 am (opening) by the stock's closing price, this becomes the. Hence each news is tagged with a label (_change_ or _no change_)
  - ✓ Nearly 36,000 news in 1999, including timestamps, mentioned stocks
  - ✓ Each word was case-normalized and stemmed, and stopwords were removed
  - ✓ Using n-grams up to two, with TFIDF score
- Class prior: 25% of the news were followed by a significant price change to the stocks involved, and 75% were not.

```
1999-03-30 14:45:00
WALTHAM, Mass.--(BUSINESS WIRE)--March 30, 1999--Summit Technology,
Inc. (NASDAQ:BEAM) and Autonomous Technologies Corporation
(NASDAQ:ATCI) announced today that the Joint Proxy/Prospectus for
Summit's acquisition of Autonomous has been declared effective by the
```

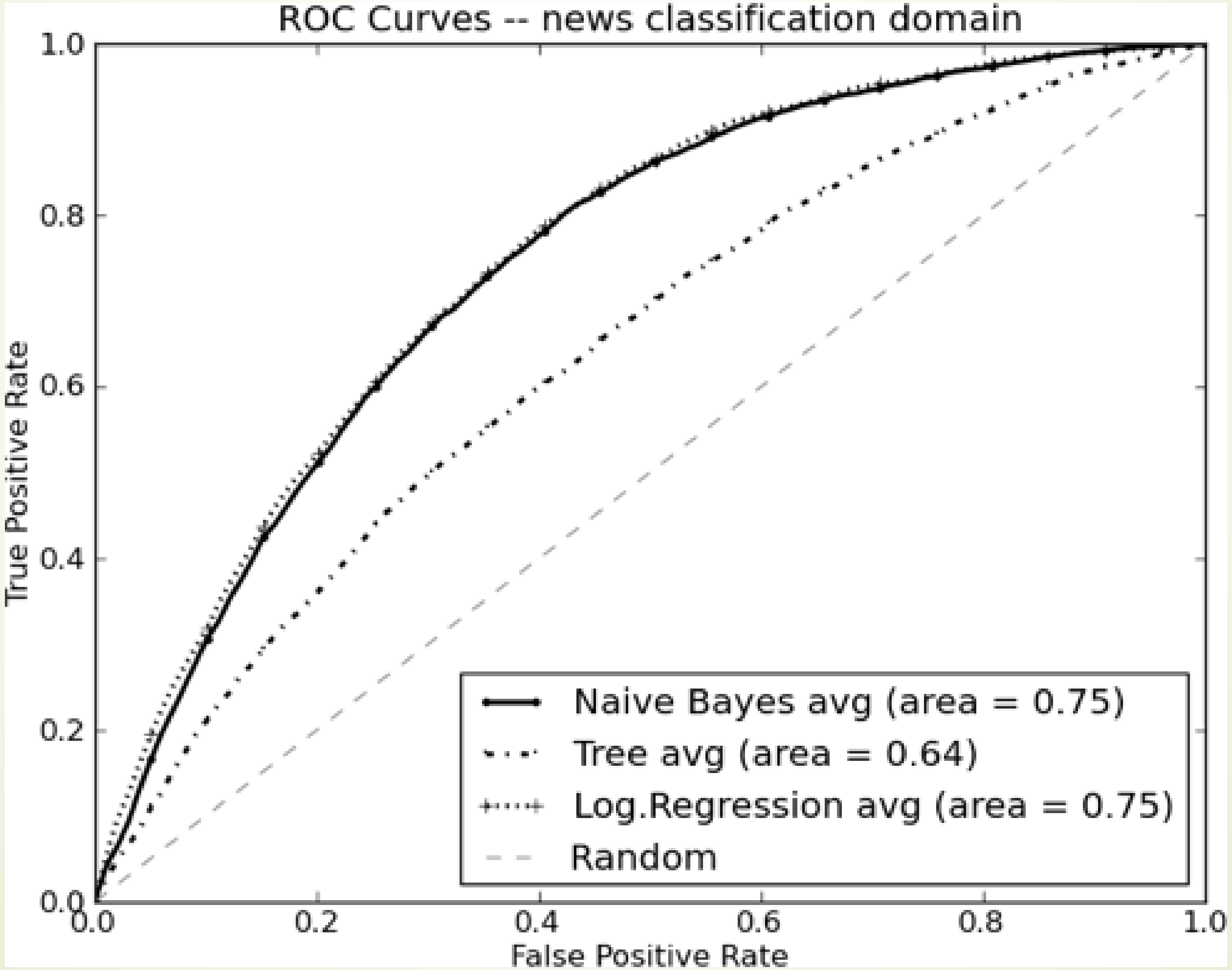
…

Example: Predicting the Stock Market (3)



1	Summit Tech announces revenues for the three months ended Dec 31, 1998 were $22.4 million, an increase of 13%.
2	Summit Tech and Autonomous Technologies Corporation announce that the Joint Proxy/Prospectus for Summit's acquisition of Autonomous has been declared effective by the SEC.
3	Summit Tech said that its procedure volume reached new levels in the first quarter and that it had concluded its acquisition of Autonomous Technologies Corporation.
4	Announcement of annual shareholders meeting.
5	Summit Tech announces it has filed a registration statement with the SEC to sell 4,000,000 shares of its common stock.
6	A US FDA panel backs the use of a Summit Tech laser in LASIK procedures to correct nearsightedness with or without astigmatism.
7	Summit up 1-1/8 at 27-3/8.

Example: Predicting the Stock Market (4)



10-fold cross-validation

Text Mining Pipelines

https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html

Working With Text Data

Tutorial setup

Loading the 20 newsgroups dataset

Extracting features from text files

Training a classifier

Building a pipeline

Evaluation of the performance on the test set

Parameter tuning using grid search

Exercise 1: Language identification

Exercise 2: Sentiment Analysis on movie reviews

Exercise 3: CLI text classification utility

Where to from here

Useful Python Packages

- General package (NER, Segmentation, POS tagging)
  - ✓ Stanford NLP (https://nlp.stanford.edu/software/)
- Text preprocessing/ classification
  - ✓ Sklearn
  - ✓ NLTK
- Topic models
  - ✓ Gensim
- Chinese segmentation/POS tagging
  - ✓ Jieba

5% Lab Quiz

- **Deadline**: 17:59 p.m., May. 8, 2020
- **Upload** the **answer worksheet** and the accomplished **Python files** to the **Blackboard**
- You may submit **unlimited times** but only the **LAST** submission will be considered
- **Only the answers in answer sheet** will be referred for grading
- Note： **MUST attach ALL** the required files in every submission/resubmission, otherwise other files will be missing.

Ranking Score

predict_proba(*self, X*) [source]

Probability estimates.

The returned estimates for all classes are ordered by the label of classes.

For a multi_class problem, if multi_class is set to be "multinomial" the softmax function is used to find the predicted probability of each class. Else use a one-vs-rest approach, i.e calculate the probability of each class assuming it to be positive using the logistic function. and normalize these values across all the classes.

Parameters:	**X : *array-like of shape (n_samples, n_features)*** Vector to be scored, where `n_samples` is the number of samples and `n_features` is the number of features.
Returns:	**T : *array-like of shape (n_samples, n_classes)*** Returns the probability of the sample for each class in the model, where classes are ordered as they are in `self.classes_`.

decision_function(*self, X*)

Predict confidence scores for samples.

The confidence score for a sample is the signed distance of that sample to the hyperplane.