Mid-Semester Survey

- https://www.surveymonkey.com/r/G32K2PT

- Please complete/submit the survey after this lecture (that we will finish all chapters before mid-term).
- If >25 students submitted survey before Oct. 3 11:59pm (HW3 due), we would have the only question of this Chapter (on advanced pattern mining) off the mid-term exam.

- Short talk today: "SciBot" Project: Task 1 to 4 in 75 minutes

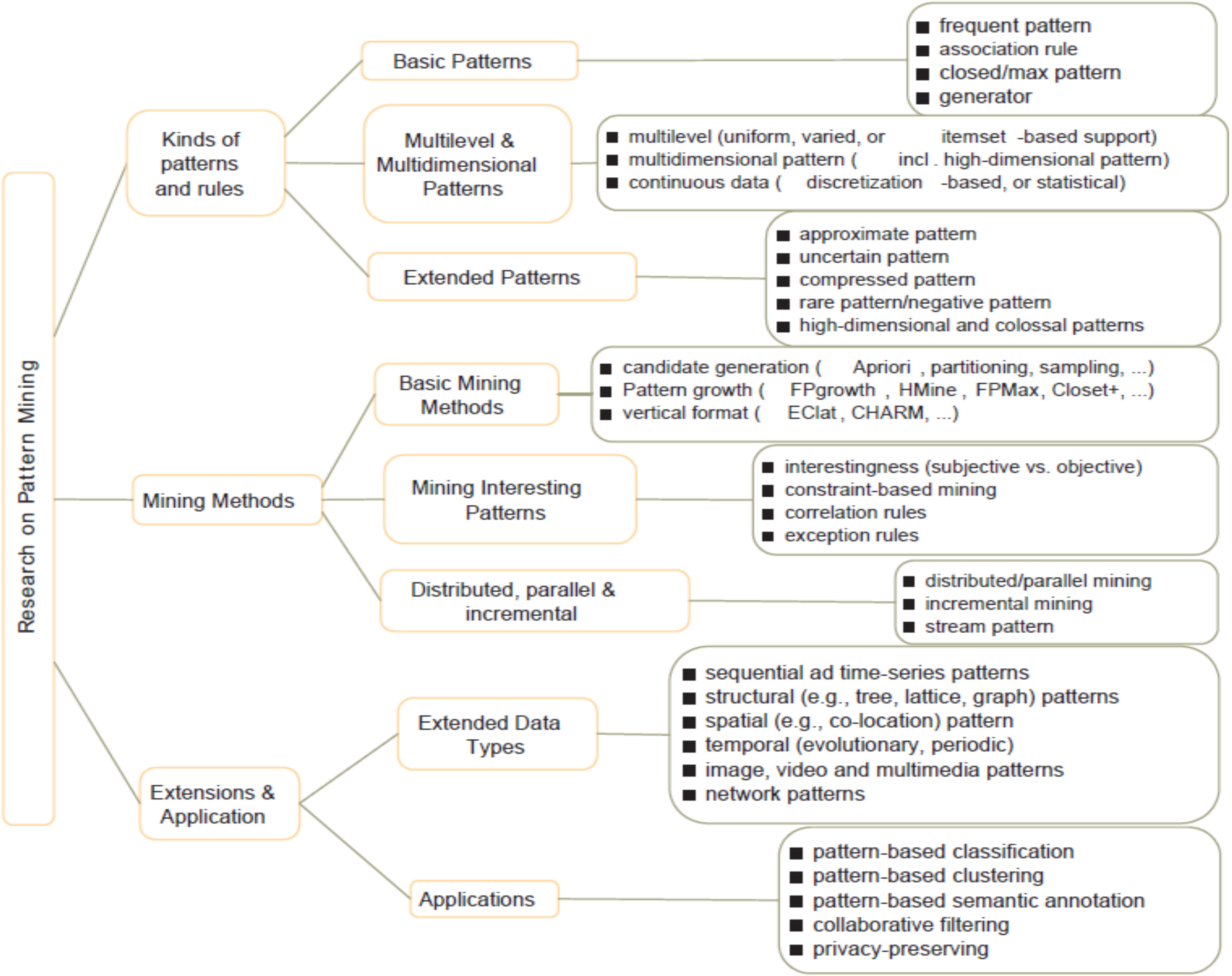
Chapter 7. Advanced Frequent Pattern Mining: Diverse Patterns

Meng Jiang

CSE 40647/60647 Data Science Fall 2017

Introduction to Data Mining

Research on Pattern Mining: A Road Map



Advanced Frequent Pattern Mining

- **Mining Diverse Patterns**
- Constraint-Based Frequent Pattern Mining
- Sequential Pattern Mining
- Graph Pattern Mining

Mining Diverse Patterns

- Mining Multiple-Level Associations
- Mining Multi-Dimensional Associations
- Mining Quantitative Associations
- Mining Negative Correlations

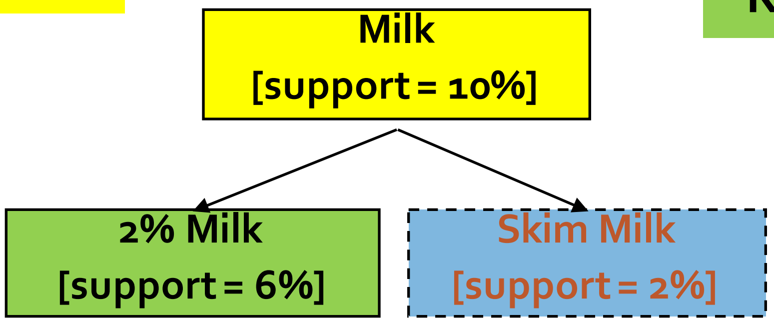
Mining Multiple-Level Frequent Patterns

- Items often form hierarchies
  - Ex.: Dairyland 2% milk; Wonder wheat bread
- How to set min-support thresholds?
  - Uniform min-support across multiple levels (reasonable?)
  - Level-reduced min-support: Items at the lower level are expected to have lower support

Uniform support

Level 1
min_sup = 5%

Level 2
min_sup = 5%



Reduced support

Level 1
min_sup = 5%

Level 2
min_sup = 1%

Redundancy Filtering at Mining Multi-Level Associations

- Multi-level association mining may generate many redundant rules
- Redundancy filtering: Some rules may be redundant due to "ancestor" relationships between items

  (Suppose the 2% milk sold is about ¼ of milk sold in gallons)

  – milk ⇒ wheat bread [support = 8%, confidence = 70%] (1)
  – 2% milk ⇒ wheat bread [support = 2%, confidence = 72%] (2)
- A rule is *redundant* if its support is close to the "expected" value, according to its "ancestor" rule, and it has a similar confidence as its "ancestor"
  – Rule (1) is an ancestor of rule (2), which one to prune?

Customized Min-Supports for Different Kinds of Items

- We have used the same min-support threshold for all the items or item sets to be mined in each association mining
- In reality, some items (e.g., diamond, watch, …) are valuable but less frequent
- It is necessary to have customized min-support settings for different kinds of items
- One Method: Use group-based “individualized” min-support
  - E.g., {diamond, watch}: 0.05%; {bread, milk}: 5%; …

Mining Multi-Dimensional Associations

- Single-dimensional rules (e.g., items are all in “product” dimension)
  - buys(X, “milk”) ⇒ buys(X, “bread”)
- Multi-dimensional rules (i.e., items in ≥ 2 dimensions or predicates)
  - Inter-dimension association rules (*no repeated predicates*)
    - age(X, “18-25”) ∧ occupation(X, “student”) ⇒ buys(X, “coke”)
  - Hybrid-dimension association rules (*repeated predicates*)
    - age(X, “18-25”) ∧ buys(X, “popcorn”) ⇒ buys(X, “coke”)

Mining Quantitative Associations

- Mining quantitative associations
  - Ex.: Gender = female ⇒ Wage: mean=$7/hr (overall mean = $9)
  - LHS: a subset of the population
  - RHS: an ***extraordinary*** behavior of this subset
- Rule condition can be categorical or numerical
  - Ex.: (Gender = female) ^ (South = yes) ⇒ mean wage = $6.3/hr
  - Ex.: Education in [14-18] (yrs) ⇒ mean wage = $11.64/hr
- Data cube technology?

Rare Patterns vs. Negative Patterns

- Rare patterns
  - Very low support but interesting (e.g., buying Rolex watches)
- Negative patterns
  - Negatively correlated: Unlikely to happen together
  - Ex.: Since it is unlikely that the same customer buys both a Ford Expedition (an SUV car) and a Ford Fusion (a hybrid car), buying a Ford Expedition and buying a Ford Fusion are likely negatively correlated patterns
  - How to define negative patterns?

Defining Negative Correlated Patterns

- A support-based definition
  - If itemsets A and B are both frequent but rarely occur together, i.e., $sup(A \cup B) \ll sup(A) \times sup(B)$
  - Then A and B are negatively correlated

Does this remind you the definition of *lift*?

- Is this a good definition for large transaction datasets?
- Ex.: Suppose a store sold two needle packages A and B 100 times each, but only one transaction contained both A and B
  - When there are in total 200 transactions, we have
    - $s(A \cup B) = 0.005$, $s(A) \times s(B) = 0.25$, $s(A \cup B) \ll s(A) \times s(B)$
  - But when there are 10^5 transactions, we have
    - $s(A \cup B) = 1/10^5$, $s(A) \times s(B) = 1/10^3 \times 1/10^3$, $s(A \cup B) > s(A) \times s(B)$
  - What is the problem? — Null transactions: The support-based definition is not null-invariant!

Defining Negative Correlation: Need Null-Invariance in Definition

- A good definition on negative correlation should take care of the null-invariance problem
  - Whether two itemsets A and B are negatively correlated should not be influenced by the number of null-transactions
- A Kulczynski measure-based definition
  - If itemsets A and B are frequent but $(P(A|B) + P(B|A))/2 < \epsilon$, where ϵ is a negative pattern threshold, then A and B are negatively correlated
- For the same needle package problem:
  - No matter there are in total 200 or 10^5 transactions
  - If $\epsilon = 0.01$, we have $(P(A|B) + P(B|A))/2 = (0.01 + 0.01)/2 < \epsilon$

Advanced Frequent Pattern Mining

- Mining Diverse Patterns
- Constraint-Based Frequent Pattern Mining
- **Sequential Pattern Mining**
- Graph Pattern Mining

Pattern Mining Methods

Pattern	Closed Pattern (Concepts)	Idea 1: Pattern candidate generation and pruning	Idea 2: Pattern growth
Frequent pattern (itemset)	?	?	?
Sequential pattern	?	?	?
Graph pattern	?	?	?

Pattern Mining Methods

Pattern	**Closed Pattern (Concepts)**	**Idea 1: Pattern candidate generation and pruning**	**Idea 2: Pattern growth**
Frequent pattern (itemset)	Closed frequent itemset	Apriori (1994)	FP-Growth (2000)
Sequential pattern	Closed seq. pattern	GSP (1996)	PrefixSpan (2004)
Graph pattern	Closed graph pattern	FSG (2000-2001)	gSpan (2002)

Sequential Patterns: Applications

- Sequential pattern mining has broad applications
  - Customer shopping sequences
    - Purchase a laptop first, then a digital camera, and then a smartphone, within 6 months
  - Medical treatments, natural disasters (e.g., earthquakes), science & engineering processes, stocks and markets, ...
  - Weblog click streams, calling patterns, …
  - Software engineering: Program execution sequences, …
  - Biological sequences: DNA, protein, …

Sequential Pattern and Sequential Pattern Mining

- Sequential pattern mining: Given a set of sequences, find the complete set of frequent subsequences (i.e., satisfying the min_sup threshold)

A *sequence database*

SID	Sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

A *sequence*: < (ef) (ab) (df) c b >

- An element may contain a set of items (also called events)
- Items within an element are unordered and we list them alphabetically

<a(bc)dc> is a *subsequence* of <a(abc)(ac)d(cf)>

- Given support threshold min_sup = 2, <(ab)c> is a sequential pattern

http://hanj.cs.illinois.edu/pdf/span01.pdf

Sequence vs Element/Itemset/Event vs Item/Instance

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of all **items**. An **itemset** is a subset of items. A **sequence** is an ordered list of itemsets. A sequence s is denoted by $\langle s_1 s_2 \cdots s_l \rangle$, where s_j is an itemset, i.e., $s_j \subseteq I$ for $1 \leq j \leq l$. s_j is also called an **element** of the sequence, and denoted as $(x_1 x_2 \cdots x_m)$, where x_k is an item, i.e., $x_k \in I$ for $1 \leq k \leq m$. For brevity, the brackets are omitted if an element has only one item. That is, element (x) is written as x. An item can occur at most once in an element of a sequence, but can occur multiple times in different elements of a sequence. The

Sequential Pattern Mining Algorithms

- Algorithm requirement: Efficient, scalable, finding complete set, incorporating various kinds of user-specific constraints
- The Apriori property still holds: If a subsequence s_1 is infrequent, none of s_1's super-sequences can be frequent
- Representative algorithms
  - Apriori-based Generalized Sequential Patterns: GSP (Srikant & Agrawal @ EDBT'96)
  - Pattern-growth methods: PrefixSpan (Pei, et al. @TKDE'04)
- Mining **closed** sequential patterns: CloSpan (Yan, et al. @SDM'03)
- Constraint-based sequential pattern mining

GSP: Apriori-Based Sequential Pattern Mining

SID	Sequence
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>

- Initial candidates: All singleton sequences
  - <a>, <b>, <c>, <d>, <e>, <f>, <g>, <h>
- Scan DB once, count support for each candidate
- Generate length-2 candidate sequences

min_sup = 2

Cand.	sup
<a>	3
<b>	5
<c>	4
<d>	3
<e>	3
<f>	2
<g>	1
<h>	1

	<a>	<b>	<c>	<d>	<e>	<f>
<a>	<aa>	<ab>	<ac>	<ad>	<ae>	<af>
<b>	<ba>	<bb>	<bc>	<bd>	<be>	<bf>
<c>	<ca>	<cb>	<cc>	<cd>	<ce>	<cf>
<d>	<da>	<db>	<dc>	<dd>	<de>	<df>
<e>	<ea>	<eb>	<ec>	<ed>	<ee>	<ef>
<f>	<fa>	<fb>	<fc>	<fd>	<fe>	<ff>

Length-2 candidates:
36 + 15= 51
Without Apriori pruning:
8*8+8*7/2=92 candidates

	<a>	<b>	<c>	<d>	<e>	<f>
<a>		<(ab)>	<(ac)>	<(ad)>	<(ae)>	<(af)>
<b>			<(bc)>	<(bd)>	<(be)>	<(bf)>
<c>				<(cd)>	<(ce)>	<(cf)>
<d>					<(de)>	<(df)>
<e>						<(ef)>
<f>						

GSP (Generalized Sequential Patterns): Srikant & Agrawal @ EDBT'96)

GSP Mining and Pruning

- Repeat (for each level (i.e., length-k))
  - Scan DB to find length-k frequent sequences
  - Generate length-(k+1) candidate sequences from length-k frequent sequences using Apriori
  - set k = k+1
- Until no frequent sequence or no candidate can be found

PrefixSpan: A Pattern-Growth Approach

- Prefix and suffix
  - Given <a(abc)(ac)d(cf)>
  - Prefixes: <a>, <aa>, <a(ab)>, <a(abc)>, …
  - Prefixes-based projection
- PrefixSpan Mining: Prefix Projections
  - Step 1: Find length-1 sequential patterns
    - <a>, <b>, <c>, <d>, <e>, <f>
  - Step 2: Divide search space and mine each projected DB
    - <a>-projected DB,
    - <b>-projected DB,
    - …
    - <f>-projected DB, …

SID	Sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

Prefix	Suffix (Projection)
<a>	<(abc)(ac)d(cf)>
<aa>	<(_bc)(ac)d(cf)>
<ab>	<(_c)(ac)d(cf)>

PrefixSpan (Prefix-projected Sequential pattern mining) Pei, et al. @TKDE'04

http://hanj.cs.illinois.edu/pdf/span01.pdf

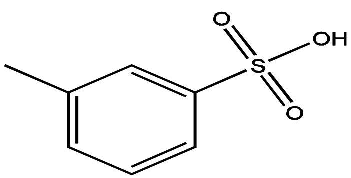
Advanced Frequent Pattern Mining

- Mining Diverse Patterns
- Constraint-Based Frequent Pattern Mining
- Sequential Pattern Mining
- **Graph Pattern Mining**

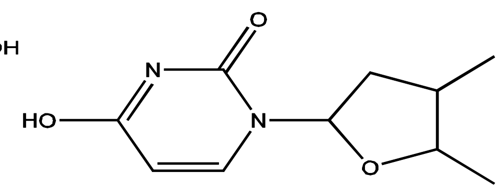
Frequent (Sub)Graph Patterns

- Given a labeled graph dataset $D = \{G_1, G_2, \ldots, G_n\}$, the supporting graph set of a subgraph g is $D_g = \{G_i \mid g \subseteq G_i, G_i \in D\}$.
  - $support(g) = |D_g| / |D|$
- A (sub)graph g is ***frequent*** if $support(g) \geq min\_sup$ Ex.: Chemical structures
- Alternative:
  - Mining frequent subgraph patterns from a single large graph or network

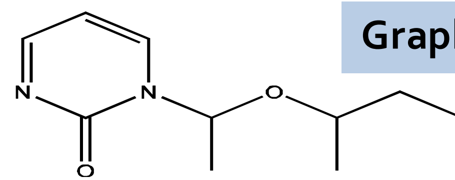
Graph Dataset



(A)



(B)

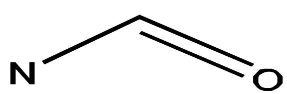


(C)

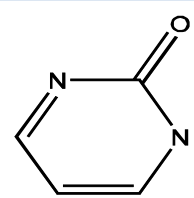
min_sup = 2

Frequent Graph Patterns

(1)



(2)

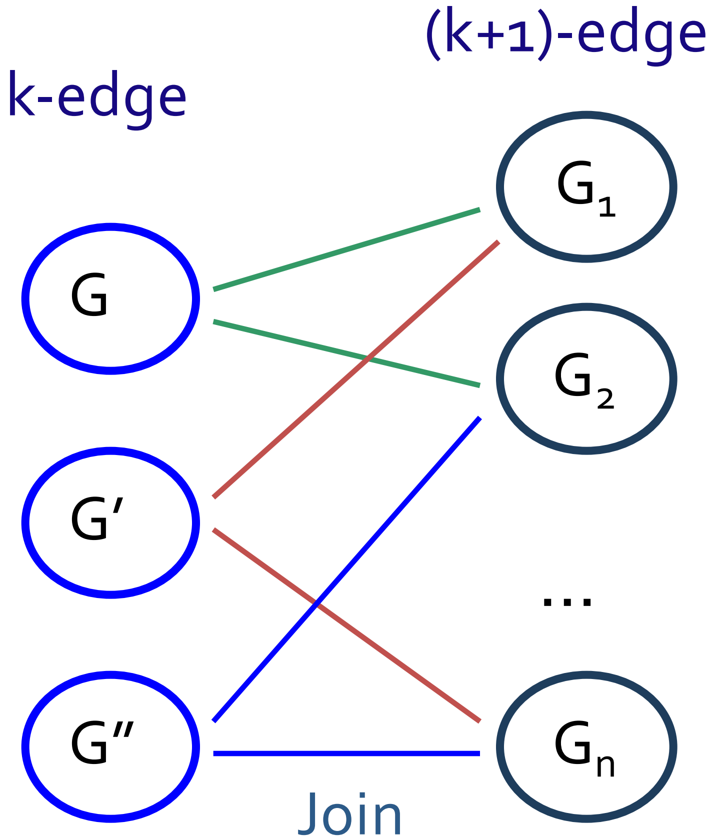


Graph Pattern Mining: Applications

- Bioinformatics
  - Gene networks, protein interactions, metabolic pathways
- Chem-informatics: Mining chemical compound structures
- Social networks, web communities, tweets, …
- Cell phone networks, computer networks, …
- Web graphs, XML structures, semantic Web, information networks
- Software engineering: program execution flow analysis
- Building blocks for graph classification, clustering, compression, comparison, and correlation analysis
- Graph indexing and graph similarity search

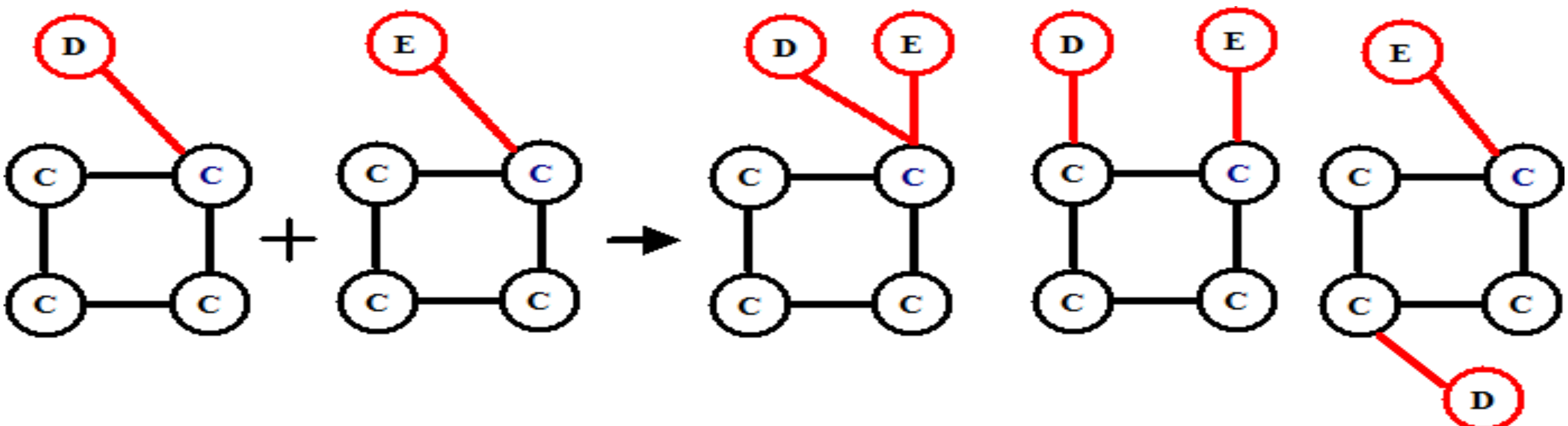
Apriori-Based Approach

- The Apriori property (anti-monotonicity): A size-*k* subgraph is frequent if and only if all of its subgraphs are frequent
- A candidate size-(*k*+1) edge/vertex subgraph is generated if its corresponding two *k*-edge/vertex subgraphs are frequent
- Iterative mining process:
  - Candidate-generation → candidate pruning → support counting → candidate elimination



Candidate Generation: Vertex Growing vs. Edge Growing

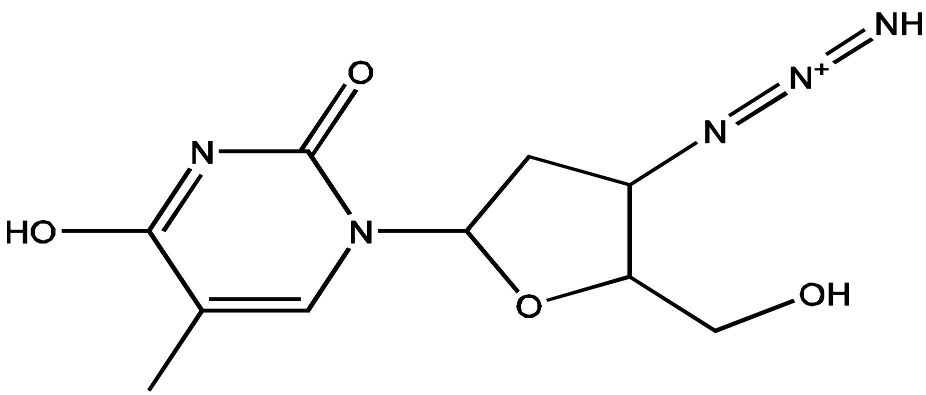
- Methodology: breadth-search, Apriori joining two size-k graphs
  - Many possibilities at generating size-(k+1) candidate graphs



- Generating new graphs with one more vertex
  - AGM (Inokuchi, et al., PKDD'00)
- Generating new graphs with one more edge
  - FSG (Kuramochi and Karypis, ICDM'01)
- Performance shows via edge growing is more efficient

Why Mining Closed Graph Patterns?

- Challenge: An **n**-edge frequent graph may have 2^n subgraphs
- Motivation: Explore *closed frequent subgraphs* to handle graph pattern explosion problem
- A frequent graph G is *closed* if there exists no supergraph of G that carries the same support as G



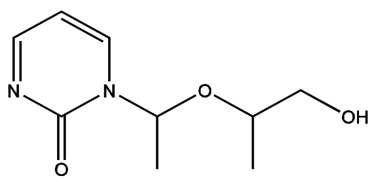
If this subgraph is *closed* in the graph dataset, it implies that none of its frequent super-graphs carries the same support

- *Lossless compression:* Does not contain non-closed graphs, but still ensures that the mining result is complete
- Algorithm CloseGraph: Mines closed graph patterns directly

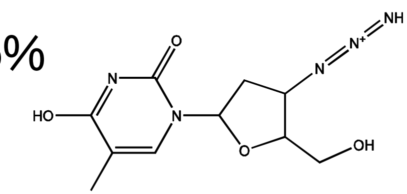
Experiment and Performance Comparison

- The AIDS antiviral screen compound dataset from NCI/NIH
- The dataset contains 43,905 chemical compounds
- Discovered Patterns: The smaller minimum support, the bigger and more interesting subgraph patterns discovered

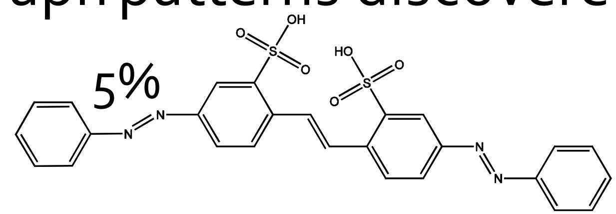
20%



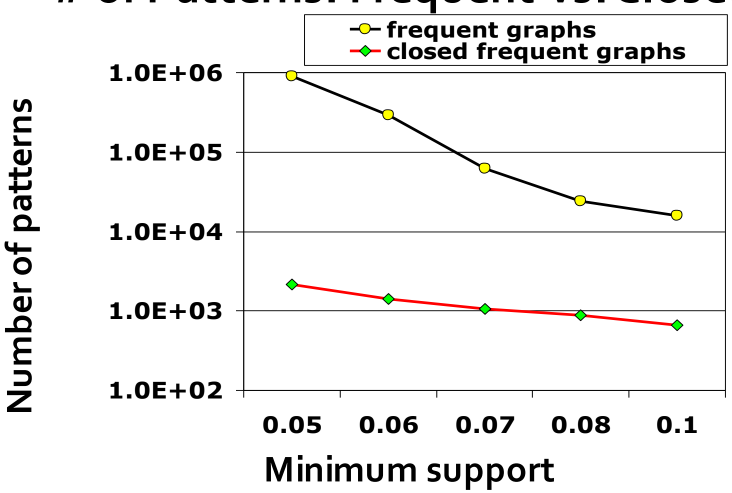
10%



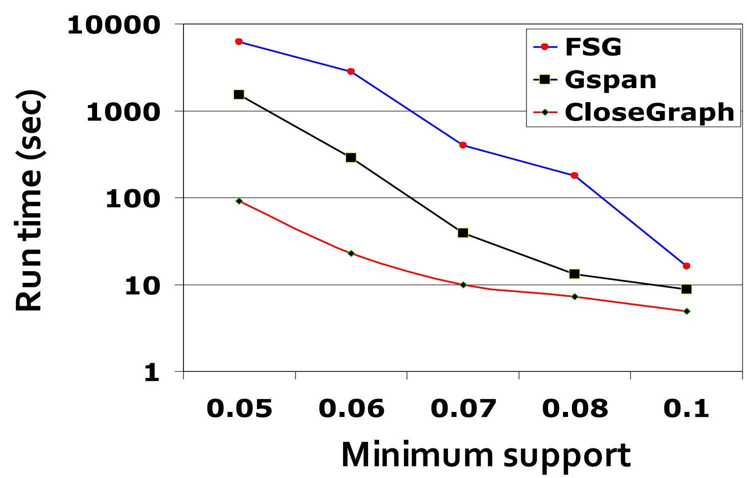
5%



# of Patterns: Frequent vs. Closed



Runtime: Frequent vs. Closed



References: Mining Diverse Patterns

- R. Srikant and R. Agrawal, “Mining generalized association rules”, VLDB'95
- Y. Aumann and Y. Lindell, “A Statistical Theory for Quantitative Association Rules”, KDD'99
- K. Wang, Y. He, J. Han, “Pushing Support Constraints Into Association Rules Mining”, IEEE Trans. Knowledge and Data Eng. 15(3): 642-658, 2003
- D. Xin, J. Han, X. Yan and H. Cheng, "On Compressing Frequent Patterns", Knowledge and Data Engineering, 60(1): 5-29, 2007
- D. Xin, H. Cheng, X. Yan, and J. Han, "Extracting Redundancy-Aware Top-K Patterns", KDD'06
- J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent Pattern Mining: Current Status and Future Directions", Data Mining and Knowledge Discovery, 15(1): 55-86, 2007
- F. Zhu, X. Yan, J. Han, P. S. Yu, and H. Cheng, “Mining Colossal Frequent Patterns by Core Pattern Fusion”, ICDE'07

References: Constraint-Based Frequent Pattern Mining

- R. Srikant, Q. Vu, and R. Agrawal, "Mining association rules with item constraints", KDD'97
- R. Ng, L.V.S. Lakshmanan, J. Han & A. Pang, "Exploratory mining and pruning optimizations of constrained association rules", SIGMOD'98
- G. Grahne, L. Lakshmanan, and X. Wang, "Efficient mining of constrained correlated sets", ICDE'00
- J. Pei, J. Han, and L. V. S. Lakshmanan, "Mining Frequent Itemsets with Convertible Constraints", ICDE'01
- J. Pei, J. Han, and W. Wang, "Mining Sequential Patterns with Constraints in Large Databases", CIKM'02
- F. Bonchi, F. Giannotti, A. Mazzanti, and D. Pedreschi, "ExAnte: Anticipated Data Reduction in Constrained Pattern Mining", PKDD'03
- F. Zhu, X. Yan, J. Han, and P. S. Yu, "gPrune: A Constraint Pushing Framework for Graph Pattern Mining", PAKDD'07

References: Sequential Pattern Mining

- R. Srikant and R. Agrawal, “Mining sequential patterns: Generalizations and performance improvements”, EDBT’96
- M. Zaki, “SPADE: An Efficient Algorithm for Mining Frequent Sequences”, Machine Learning, 2001
- J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach", IEEE TKDE, 16(10), 2004
- X. Yan, J. Han, and R. Afshar, “CloSpan: Mining Closed Sequential Patterns in Large Datasets”, SDM'03
- J. Pei, J. Han, and W. Wang, "Constraint-based sequential pattern mining: the pattern-growth methods", J. Int. Inf. Sys., 28(2), 2007
- M. N. Garofalakis, R. Rastogi, K. Shim: Mining Sequential Patterns with Regular Expression Constraints. IEEE Trans. Knowl. Data Eng. 14(3), 2002
- H. Mannila, H. Toivonen, and A. I. Verkamo, “Discovery of frequent episodes in event sequences”, Data Mining and Knowledge Discovery, 1997

References: Graph Pattern Mining

- C. Borgelt and M. R. Berthold, Mining molecular fragments: Finding relevant substructures of molecules, ICDM'02
- J. Huan, W. Wang, and J. Prins. Efficient mining of frequent subgraph in the presence of isomorphism, ICDM'03
- A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data, PKDD'00
- M. Kuramochi and G. Karypis. Frequent subgraph discovery, ICDM'01
- S. Nijssen and J. Kok. A Quickstart in Frequent Structure Mining can Make a Difference. KDD'04
- N. Vanetik, E. Gudes, and S. E. Shimony. Computing frequent graph patterns from semistructured data, ICDM'02
- X. Yan and J. Han, gSpan: Graph-Based Substructure Pattern Mining, ICDM'02
- X. Yan and J. Han, CloseGraph: Mining Closed Frequent Graph Patterns, KDD'03
- X. Yan, P. S. Yu, J. Han, Graph Indexing: A Frequent Structure-based Approach, SIGMOD'04
- X. Yan, P. S. Yu, and J. Han, Substructure Similarity Search in Graph Databases, SIGMOD'05

“SciBot” Project:
Task 1 to 4 in 75 minutes

This is just a base.

Task 1 to 4

- Task 1: Data Cleaning and Integration (**10 minutes**)
- Task 2: Entity name recognition (**30 minutes**)
  - 2-1: Entity name candidate generation (20 minutes)
  - 2-2: Entity name quality assessment (10 minutes)
- Task 3: Entity typing (**15 minutes**)
- Task 4: Collaboration discovery (**20 minutes**)

Task 1 (10 minutes)

def task_1(files):

…

return

pid2txt

pid2title_year_conf

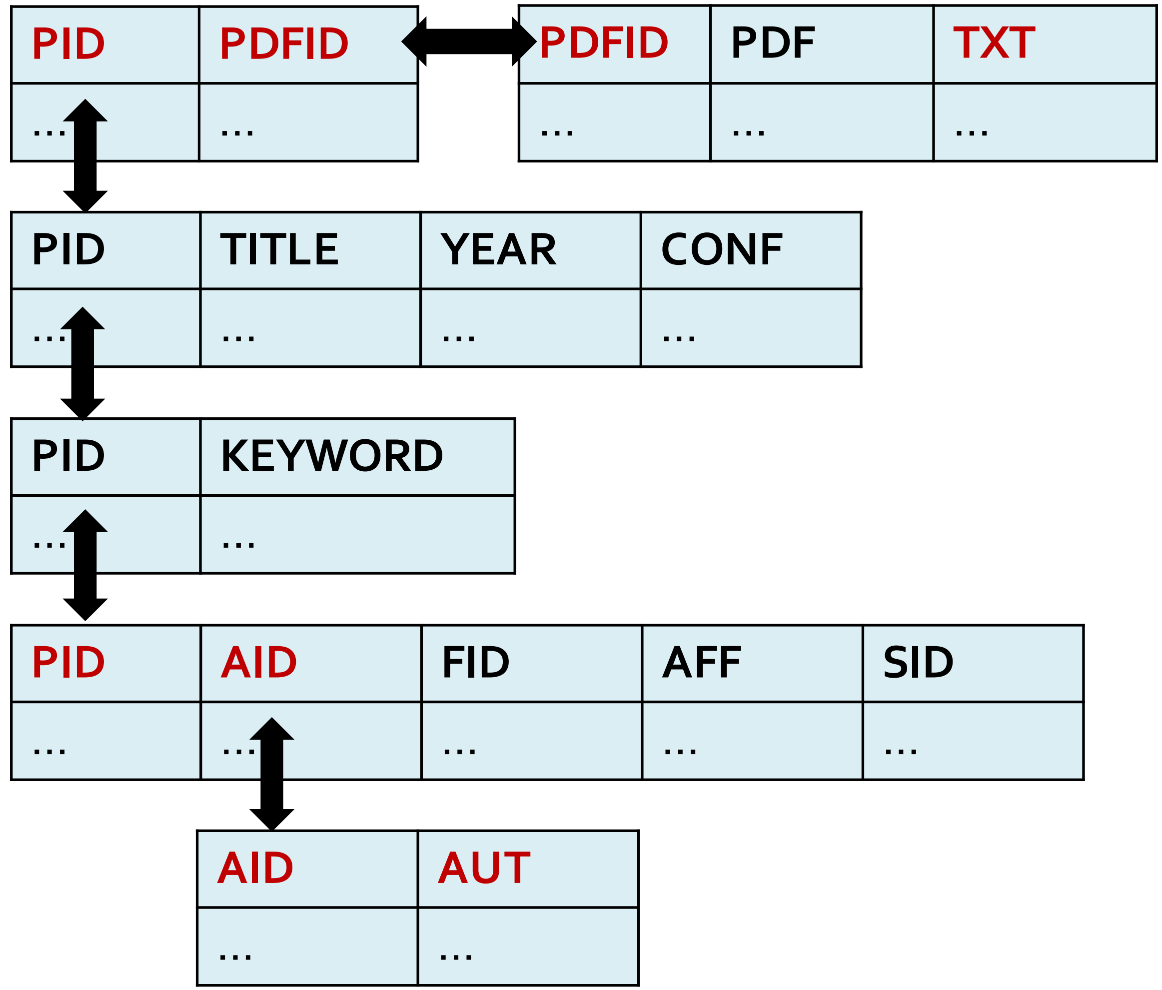
pid2keyword

pid2authorseq

… word2pidlist

… aid2pidlist

aid2authorname



An Integrated and Cleaned Database

PID	YEAR	CONF	TITLE	KEYWORDS
776E2648	2010	kdd	new perspectives	machine learning|networks|supervised learning|variance reduction
784B7EF4	2014	kdd	improving manag	clustering|data mining|invasive species|networks|risk assessment
7E395F14	2008	icdm	start globally opti	data mining|global optimization|learning artificial intelligence|optimization|probabili

SEQ_AUTHOR_AFFS
1:109A673C:ryan n lichtenwalter:066A71BC:university of notre dame|2:7DA9ABBD:jake t lussier:066A71BC:university of notre da
1:7E5C680D:jian xu:066A71BC:university of notre dame|2:5DAE606C:thanuka l wickramarathne:066A71BC:university of notre da
1:7776FD94:david a cieslak:066A71BC:university of notre dame|2:76014D6E:nitesh v chawla:066A71BC:university of notre dame

PID	PDFID
…	…

PDFID	PDF	TXT
…	…	…

PID	TITLE	YEAR	CONF
…	…	…	…

PID	KEYWORD
…	…

PID	AID	FID	AFF	SID
…	…	…	…	…

AID	AUT
…	…

Qs in HW3 on Task 1

- a) How many unique papers and how many unique authors are there in your integrated and cleaned dataset?
- b) Find “matrix” experts: List the top three authors who published at least 3 papers AND used the word “matrix” the most frequently in their papers (i.e., the highest average number of “matrix” in their publications).
- c) Find “long-title” authors: List the top three authors who published at least 3 papers AND preferred long titles in their papers (i.e., the highest average length of paper titles).

Task 2-1: Entity Name Candidate Generation (20 minutes)

- Tech 1: Hand-crafted Rules
  - **20 minutes**

def tech_1(pid2txt):

...

return

name2abbr2count

... Support Vector Machines (**SVMs**) ...

latent_dirichlet_allocation	247	LDA:247
support_vector_machine	218	SVM:218
support_vector_machines	214	SVM:125|SVMs:89
singular_value_decomposition	150	SVD:150
world_wide_web	145	WWW:145
information_retrieval	141	IR:141
mean_average_precision	124	MAP:124
collaborative_filtering	110	CF:110
expectation_maximization	105	EM:105
principal_component_analysis	104	PCA:104
resource_description_framework	95	RDF:95
neural_information_processing_systems	93	NIPS:93
stochastic_gradient_descent	91	SGD:91
minimum_description_length	78	MDL:78
natural_language_processing	77	NLP:77
normalized_mutual_information	76	NMI:76
mean_reciprocal_rank	76	MRR:76
document_object_model	71	DOM:71
mean_absolute_error	69	MAE:69
logistic_regression	67	LR:67
normalized_discounted_cumulative_gain	66	NDCG:66
latent_semantic_indexing	63	LSI:63
maximum_a_posteriori	62	MAP:62
mean_squared_error	62	MSE:62
receiver_operating_characteristic	58	ROC:57|RoC:1
dynamic_time_warping	58	DTW:58
markov_chain_monte_carlo	57	MCMC:57
naive_bayes	57	NB:57
transactions_on_information_systems	56	TOIS:56
probabilistic_latent_semantic_analysis	52	PLSA:52
directed_acyclic_graph	51	DAG:51
open_directory_project	50	ODP:50
non-negative_matrix_factorization	50	NMF:50
national_science_foundation	50	NSF:50
hidden_markov_models	47	HMMs:26|HMM:21
maximum_likelihood_estimation	47	MLE:47
root_mean_square_error	47	RMSE:47
hidden_markov_model	45	HMM:45
conditional_random_fields	45	CRFs:26|CRF:19
matrix_factorization	44	MF:44
information_extraction	42	IE:42
named_entity_recognition	42	NER:42
root_mean_squared_error	39	RMSE:39
cumulative_distribution_function	39	CDF:39
nonnegative_matrix_factorization	39	NMF:39

Task 2-1: Entity Name Candidate Generation (20-60 minutes)

- Tech 2: Frequent pattern mining
  - **40 minutes**

def tech_2(pid2txt, name2abbr2count, min_sup):

name2abbr2count → {"vector":16, "support":3, "machine":20…}

pid2txt →

seed words: Number of entity names containing the word

"…. use feature **vector** …": > min_sup!

"… **vector** efficiently …": < min_sup

" … into **vector** space …": > min_sup!

" … into **vector** space …": > min_sup, but "into" is a *stopword* ☹

→ name_sup = [["feature vector", 251], ["vector space", 176]…]

→ 2-grams to 3-grams to 4…

[Apriori+: write code? call package?]

return name_sup

Find a stopword list on the web (Google, GitHub…)

Task 2-2: Entity name quality assessment (10-30 minutes)

Support (0-10 minutes): #sentences (paragraphs/documents)

latent_dirichlet_allocation	247	LDA:247
support_vector_machine	218	SVM:218
support_vector_machines	214	SVM:125|SVMs:89
singular_value_decomposition	150	SVD:150
world_wide_web	145	WWW:145
information_retrieval	141	IR:141
mean_average_precision	124	MAP:124
collaborative_filtering	110	CF:110
expectation_maximization	105	EM:105
principal_component_analysis	104	PCA:104
resource_description_framework	95	RDF:95
neural_information_processing_systems	93	NIPS:93
stochastic_gradient_descent	91	SGD:91
minimum_description_length	78	MDL:78
natural_language_processing	77	NLP:77
normalized_mutual_information	76	NMI:76
mean_reciprocal_rank	76	MRR:76
document_object_model	71	DOM:71
mean_absolute_error	69	MAE:69
logistic_regression	67	LR:67
normalized_discounted_cumulative_gain	66	NDCG:66
latent_semantic_indexing	63	LSI:63
maximum_a_posteriori	62	MAP:62
mean_squared_error	62	MSE:62
receiver_operating_characteristic	58	ROC:57|RoC:1
dynamic_time_warping	58	DTW:58
markov_chain_monte_carlo	57	MCMC:57
naive_bayes	57	NB:57
transactions_on_information_systems	56	TOIS:56
probabilistic_latent_semantic_analysis	52	PLSA:52
directed_acyclic_graph	51	DAG:51
open_directory_project	50	ODP:50
non-negative_matrix_factorization	50	NMF:50
national_science_foundation	50	NSF:50
hidden_markov_models	47	HMMs:26|HMM:21
maximum_likelihood_estimation	47	MLE:47
root_mean_square_error	47	RMSE:47
hidden_markov_model	45	HMM:45
conditional_random_fields	45	CRFs:26|CRF:19
matrix_factorization	44	MF:44
information_extraction	42	IE:42
named_entity_recognition	42	NER:42
root_mean_squared_error	39	RMSE:39
cumulative_distribution_function	39	CDF:39
nonnegative_matrix_factorization	39	NMF:39

Outlier-ness measure (a significance score): (10 minutes)

$$sig(P_1, P_2) \approx \frac{f(P_1 \oplus P_2) - \mu_0(P_1, P_2)}{\sqrt{f(P_1 \oplus P_2)}}$$

We have 10000 words.

“feature”: 300, “vector”: 200

“feature vector”:100

sig(“feature”,”vector”)

= (100 - 10000 * 300/10000 * 200/10000) / sqrt(100)

= (100 – 6)/10 = 9.4

How about 3-grams?

Qs in HW3 on Task 2

- d) How many unique case-insensitive entity names (like “support vector machines”) have you discovered in the dataset?
  - List the *top 20 entity names* and their *support* (i.e., the number of papers that have at least one such entity name) if you have.
- e) Briefly explain your technique(s).

Task 3: Entity Typing (15 minutes)

- Trigger words
  - METHOD: method algorithm model approach framework process scheme implementation procedure strategy architecture
  - PROBLEM: problem technique process system application task evaluation tool paradigm benchmark software
  - DATASET: data dataset database
  - METRIC: value score measure metric function parameter
- *Classification* using *trigger-word features*: *Majority-voting*

```
collaborative_filtering 967     METHOD:729|PROBLEM:217|DATASET:18|METRIC:3
feature_selection       895     METHOD:708|PROBLEM:138|METRIC:48|DATASET:1
link_prediction 641     PROBLEM:348|METHOD:204|METRIC:89
active_learning 588     METHOD:492|PROBLEM:96
latent_dirichlet_allocation     538     METHOD:530|PROBLEM:6|DATASET:2
matrix_factorization    518     METHOD:354|PROBLEM:164
supervised_learning     503     METHOD:347|PROBLEM:153|METRIC:2|DATASET:1
logistic_regression     490     METHOD:443|PROBLEM:31|METRIC:16
expectation_maximization        434     METHOD:426|PROBLEM:5|METRIC:3
social_network  391     DATASET:280|METHOD:68|PROBLEM:40|METRIC:3
binary_classification   391     PROBLEM:328|METHOD:48|DATASET:11|METRIC:4
resource_description_framework  367     DATASET:267|METHOD:92|PROBLEM:8
random_walk     367     METHOD:313|DATASET:27|METRIC:14|PROBLEM:13
```

equivalent to a simple Multi-class Decision Tree

Some Entity Typing Results:

Clustering and then typing?
(+40 minutes):
"s_v_m" and "s_v_ms"

OR

Pattern-based classification?
(+40 minutes):
more features "problem of $PROBLEM"

```
METHOD  latent_dirichlet_allocation     247     LDA:247
METHOD  support_vector_machine  218     SVM:218
METHOD  support_vector_machines 214     SVM:125|SVMs:89
METHOD  singular_value_decomposition    150     SVD:150
DATASET world_wide_web  145     WWW:145
PROBLEM information_retrieval   141     IR:141
METRIC  mean_average_precision  124     MAP:124
METHOD  collaborative_filtering 110     CF:110
METHOD  expectation_maximization        105     EM:105
METHOD  principal_component_analysis    104     PCA:104
DATASET resource_description_framework  95      RDF:95
DATASET neural_information_processing_systems   93      NIPS:93
METHOD  stochastic_gradient_descent     91      SGD:91
METRIC  minimum_description_length      78      MDL:78
PROBLEM natural_language_processing     77      NLP:77
METRIC  normalized_mutual_information   76      NMI:76
METRIC  mean_reciprocal_rank    76      MRR:76
METHOD  document_object_model   71      DOM:71
METRIC  mean_absolute_error     69      MAE:69
METHOD  logistic_regression     67      LR:67
METRIC  normalized_discounted_cumulative_gain   66      NDCG:66
METHOD  latent_semantic_indexing        63      LSI:63
METHOD  maximum_a_posteriori    62      MAP:62
METRIC  mean_squared_error      62      MSE:62
METRIC  receiver_operating_characteristic       58      ROC:57|RoC:1
METHOD  dynamic_time_warping    58      DTW:58
METHOD  markov_chain_monte_carlo        57      MCMC:57
METHOD  naive_bayes     57      NB:57
ENTITY  transactions_on_information_systems     56      TOIS:56
METHOD  probabilistic_latent_semantic_analysis  52      PLSA:52
METHOD  directed_acyclic_graph  51      DAG:51
DATASET open_directory_project  50      ODP:50
METHOD  non-negative_matrix_factorization       50      NMF:50
DATASET national_science_foundation     50      NSF:50
METHOD  hidden_markov_models    47      HMMs:26|HMM:21
METHOD  maximum_likelihood_estimation   47      MLE:47
METRIC  root_mean_square_error  47      RMSE:47
METHOD  hidden_markov_model     45      HMM:45
METHOD  conditional_random_fields       45      CRFs:26|CRF:19
METHOD  matrix_factorization    44      MF:44
PROBLEM information_extraction  42      IE:42
PROBLEM named_entity_recognition        42      NER:42
METRIC  root_mean_squared_error 39      RMSE:39
METRIC  cumulative_distribution_function        39      CDF:39
METHOD  nonnegative_matrix_factorization        39      NMF:39
METHOD  vector_space_model      38      VSM:38
ENTITY  defense_advanced_research_projects_agency       37      DARP
METRIC  information_gain        37      IG:37
```

Task 4: Collaboration Discovery (15 minutes)

pid2aidlist, keyword2aidlist, conference2aidlist

Given the *paper/keyword/conference-author* data, find *frequent author-sets (as patterns)* : which two/three/four authors often collaborate together?

- Frequent pattern mining: Apriori or **FP-Growth**
- Transactions: Papers
- Items: Authors
- min_sup?

Advisor-Advisee Discovery (+5 minutes):
- Ranking 2-itemsets by *Kulc* measure.
Evaluation: Subjective: top 10 item-pairs?

Time and Performance

		Time	
Task 1	Cleaning and Integration	10 mins	
Task 2	Entity name candidate generation	20 mins (Abbreviation rules)	+40 mins (Apriori)
	Entity name quality assessment	10 mins (support + 2-gram sig.)	+20 mins (n-gram sig.)
Task 3	Entity typing	15 mins (majority voting)	+40 mins (clustering+typing) OR (pattern-based typing)
Task 4	Collaboration discovery	20 mins (FP-Growth)	+5 mins (Kulc for 2-itemsets)
		75 mins	+105 mins = 180 mins = **3 hours**
Grading		A, B+, A-, B+, A-	A, A, A, A, A
	× (professor/student): 0.5 to 3.0	**38 mins** – 3h 45mins	1h 30mins – **9 hours**