Contrast Data Mining: Methods and Applications

Kotagiri Ramamohanarao and James Bailey, NICTA Victoria Laboratory and The University of Melbourne

Guozhu Dong, Wright State University

Contrast data mining - What is it ?

Contrast - ``To compare or appraise in respect to differences'' (Merriam Webster Dictionary)

Contrast data mining - The mining of patterns and models contrasting two or more classes/conditions.

Contrast Data Mining - What is it ?

Cont.

``*Sometimes it's good to contrast what you like with something else. It makes you appreciate it even more''*

Darby Conley, *Get Fuzzy, 2001*

What can be contrasted ?

- Objects at different *time* periods
  - ``Compare ICDM papers published in 2006-2007 versus those in 2004-2005''
- Objects for different *spatial* locations
  - ``Find the distinguishing features of location *x* for human DNA, versus location *x* for mouse DNA''
- Objects *across* different *classes*
  - ``Find the differences between people with brown hair, versus those with blonde hair''

What can be contrasted ? Cont.

- Objects *within* a class
  - ``Within the academic profession, there are few people older than 80'' (*rarity*)
  - ``Within the academic profession, there are no rich people'' (*holes*)
  - ``Within computer science, most of the papers come from USA or Europe'' (*abundance)*
- Object positions in a *ranking*
  - ``Find the differences between high and low income earners''
- Combinations of the above

Alternative names for contrast data mining

- Contrast={change, difference, discriminator, classification rule, …}

- Contrast data mining is related to topics such as:

  Change detection, class based association rules, contrast sets, concept drift, difference detection, discriminative patterns, (dis)similarity index, emerging patterns, high confidence patterns, (in)frequent patterns, top k patterns,……

Characteristics of contrast data mining

- Applied to multivariate data
- Objects may be relational, sequential, graphs, models, classifiers, combinations of these
- Users may want either
  - To find *multiple* contrasts (*all*, or top *k*)
  - A *single* measure for comparison
    - ``The degree of difference between the groups (or models) is 0.7''

Contrast characteristics Cont.

- *Representation* of contrasts is important. Needs to be
  - Interpretable, non redundant, potentially actionable, expressive
  - Tractable to compute
- *Quality* of contrasts is also important. Need
  - Statistical significance, which can be measured in multiple ways
  - Ability to rank contrasts is desirable, especially for classification

How is contrast data mining used ?

- Domain understanding
  - ``Young children with diabetes have a greater risk of hospital admission, compared to the rest of the population
- Used for building *classifiers*
  - Many different techniques - to be covered later
  - Also used for weighting and ranking instances
- Used in construction of *synthetic instances*
  - Good for rare classes
- Used for alerting, notification and monitoring
  - ``Tell me when the dissimilarity index falls below 0.3''

Goals of this tutorial

- Provide an overview of contrast data mining
- Bring together results from a number of disparate areas.
  - Mining for different types of data
    - Relational, sequence, graph, models, …
  - Classification using discriminating patterns

By the end of this tutorial you will be able to …

- Understand some principal techniques for representing contrasts and evaluating their quality
- Appreciate some mining techniques for contrast discovery
- Understand techniques for using contrasts in classification

Don’t have time to cover ..

- String algorithms
- Connections to work in inductive logic programming
- Tree-based contrasts
- Changes in data streams
- Frequent pattern algorithms
- Connections to granular computing
- …

Outline of the tutorial

- Basic notions/univariate contrasts
- Pattern and rule based contrasts
- Contrast pattern based classification
- Contrasts for rare class datasets
- Data cube contrasts
- Sequence based contrasts
- Graph based contrasts
- Model based contrasts
- Common themes + open problems + summary

Basic notions and univariate case

- Feature selection and feature significance tests can be thought of as a basic contrast data mining activity.
  - ``Tell me the discriminating features''
    - Would like a single quality measure
    - Useful for feature ranking
  - Emphasis is less on *finding* the contrast and more on *evaluating* its power

Sample Feature-Class

ID	*Height (cm)*	*Class*
9004	150	Happy ☺
1005	200	Sad ☹
9006	137	Happy ☺
4327	120	Happy ☺
3325	……	…..

Discriminative power

- Can assess discriminative power of *Height* feature by
  - Information measures (signal to noise, information gain ratio, …)
  - Statistical tests (t-test, Kolmogorov-Smirnov, Chi squared, Wilcoxon rank sum, …). Assessing whether
    - The mean of each class is the same
    - The samples for each class come from the same distribution
    - How well a dataset fits a hypothesis

No single test is best in all situations !

Example Discriminative Power Test - Wilcoxon Rank Sum

- Suppose n_1 happy, and n_2 sad instances
- Sort the instances according to height value:

  $h_1 <= h_2 <= h_3 <= \ldots h_{n1+n2}$
- Assign a rank to each instance, indicating how many values in the other class are less than it
- For each class
  - Compute the S=Sum(ranks of all its instances)
  - Null Hypothesis: The instances are from the same distribution
  - Consult statistical significance table to determine whether value of S is significant

Rank Sum Calculation Example

ID	Height(cm)	Class	Rank
324	220	Happy ☺	3
481	210	Sad ☹	2
660	190	Sad ☹	2
321	177	Happy ☺	1
415	150	Sad ☹	1
816	120	Happy ☺	0

Happy: RankSum=3+1+0=4 Sad:RankSum=2+2+1=5

Wilcoxon Rank Sum Test Cont.

- This test
  - Non parametric (no normal distribution assumption)
  - Requires an ordering on the attribute values
- Value of S is also equivalent to area under ROC curve for using the selected feature as a classifier

Discriminating with attribute values

- Can alternatively focus on significance of attribute *values*, with either

1) *Frequency/infrequency* (high/low counts)
   - Frequent in one class and infrequent in the other.
     - There are 50 happy people of height 200cm and only two sad people of height 200cm

2) *Ratio* (high ratio of support)
   - Appears 25 times more in one class than the other assuming equal class sizes
     - There are 25 times more happy people of height 200cm than sad people

Attribute/Feature Conversion

- Possible to form a new binary feature based on attribute value and then apply feature significance tests
  - Blur distinction between attribute and attribute value

150cm	200cm	…	Class
Yes	No	…	Happy ☺
No	Yes	…	Sad ☹

Discriminating Attribute Values in a Data Stream

- Detecting changes in attribute values is an important focus in data streams
  - Often focus on univariate contrasts for efficiency reasons
  - Finding when change occurs (non stationary stream).
  - Finding the magnitude of the change. E.g. How big is the distance between two samples of the stream?
  - Useful for signaling necessity for model update or an impending fault or critical event

Odds ratio and Risk ratio

- Can be used for comparing or measuring effect size
- Useful for binary data
- Well known in clinical contexts
- Can also be used for quality evaluation of multivariate contrasts (will see later)
- A simple example given next

Odds and risk ratio Cont.

Gender (feature)	Exposed (event)
Male	Yes
Female	No
Male	No
…	…

Odds Ratio Example

- Suppose we have 100 men and 100 women and 70 men and 10 women have been exposed
  - Odds of exposure(male)=0.7/0.3=2.33
  - Odds of exposure(female)=0.1/0.9=0.11
  - Odds ratio=2.33/.11=21.2
- Males have 21.2 times the odds of exposure than females
- Indicates exposure is much more likely for males than for females

Relative Risk Example

- Suppose we have 100 men and 100 women and 70 men and 10 women have been exposed
  - Relative risk of exposure (male)=70/100=0.7
  - Relative risk of exposure(female)=10/100=0.1
  - The relative risk=0.7/0.1=7
- Men 7 times more likely to be exposed than women

Pattern/Rule Based Contrasts

- Overview of ``relational'' contrast pattern mining
- Emerging patterns and mining
  - Jumping emerging patterns
  - Computational complexity
  - Border differential algorithm
    - Gene club + border differential
    - Incremental mining
  - Tree based algorithm
  - Projection based algorithm
  - ZBDD based algorithm
- Bioinformatic application: cancer study on microarray gene expression data

Overview

- Class based association rules (Cai et al 90, Liu et al 98, ...)
- Version spaces (Mitchell 77)
- Emerging patterns (Dong+Li 99) – many algorithms (later)
- Contrast set mining (Bay+Pazzani 99, Webb et al 03)
- Odds ratio rules & delta discriminative EP (Li et al 05, Li et al 07)
- MDL based contrast (Siebes, KDD07)
- Using statistical measures to evaluate group differences (Hilderman+Peckman 05)
- Spatial contrast patterns (Arunasalam et al 05)
- …… see references

Classification/Association Rules

- Classification rules -- special association rules (with just one item – class -- on RHS):
  - X → C (s,c)
    - X is a pattern,
    - C is a class,
    - s is support,
    - c is confidence

Version Space (Mitchell)

- Version space: the set of all patterns consistent with given (D+,D-) – patterns separating D+, D-.
  - The space is delimited by a *specific* & a *general boundary*.
  - Useful for searching *the true hypothesis*, which lies somewhere b/w the two boundaries.
  - Adding +ve examples to D+ makes the specific boundary more general; adding -ve examples to D- makes the general boundary more specific.
- Common pattern/hypothesis language operators: conjunction, disjunction
- Patterns/hypotheses are crisp; need to be generalized to deal with percentages; hard to deal with noise in data

STUCCO, MAGNUM OPUS for contrast pattern mining

- STUCCO (Bay+Pazzani 99)
  - Mining contrast patterns X (called contrast sets) between k>=2 groups: |suppi(X) – suppj(X)| >= minDiff
  - Use Chi2 to measure statistical significance of contrast patterns
    - cut-off thresholds change, based on the level of the node and the local number of contrast patterns
  - Max-Miner like search strategy, plus some pruning techniques
- MAGNUM OPUS (Webb 01)
  - An association rule mining method, using Max-Miner like approach (proposed before, and independently of, Max-Miner)
  - Can mine contrast patterns (by limiting RHS to a class)

Contrast patterns vs decision tree based rules

- It has been recognized by several authors (e.g. Bay+Pazzani 99) that
  - rules generation from decision trees can be good contrast patterns,
  - but may miss many good contrast patterns.
  - Random forests can address this problem
- Different contrast set mining algorithms have different thresholds
  - Some have min support threshold
  - Some have no min support threshold; low support patterns may be useful for classification, etc

Emerging Patterns

- Emerging Patterns (EPs) are contrast patterns between two classes of data whose support changes significantly between the two classes. Change significantly can be defined by:
  - big support ratio:
    - supp2(X)/supp1(X) >= minRatio ← similar to Relative Risk; +: allowing patterns with small overall support
  - big support difference:
    - |supp2(X) – supp1(X)| >= minDiff (as defined by Bay+Pazzani 99)
- If supp2(X)/supp1(X) = infinity, then X is a *jumping EP*.
  - jumping EP occurs in some members of one class but never occur in the other class.
- Conjunctive language; extension to disjunctive EP later

A typical EP in the Mushroom dataset

- The Mushroom dataset contains two classes: edible and poisonous.
- Each data tuple has several features such as: odor, ring-number, stalk-surface-bellow-ring, etc.
- Consider the pattern

  {odor = none,
  stalk-surface-below-ring = smooth,
  ring-number = one}

Its support increases from 0.2% in the poisonous class to 57.6% in the edible class (a growth rate of 288).

Example EP in microarray data for cancer

Normal Tissues

g1	g2	g3	g4
L	H	L	H
L	H	L	L
H	L	L	H
L	H	H	L

Cancer Tissues

g1	g2	g3	g4
H	H	L	H
L	H	H	H
L	L	L	H
H	H	H	L

binned data

Jumping EP: Patterns w/ high support ratio b/w data classes

E.G. {g1=L,g2=H,g3=L}; suppN=50%, suppC=0

Top support *minimal* jumping EPs for colon cancer

Colon Cancer EPs

{1+ 4- 112+ 113+} 100%
{1+ 4- 113+ 116+} 100%
{1+ 4- 113+ 221+} 100%
{1+ 4- 113+ 696+} 100%
{1+ 108- 112+ 113+} 100%
{1+ 108- 113+ 116+} 100%
{4- 108- 112+ 113+} 100%
{4- 109+ 113+ 700+} 100%
{4- 110+ 112+ 113+} 100%
{4- 112+ 113+ 700+} 100%
{4- 113+ 117+ 700+} 100%
{1+ 6+ 8- 700+} 97.5%

Very few 100% support EPs.

Colon Normal EPs

{12- 21- 35+ 40+ 137+ 254+} 100%
{12- 35+ 40+ 71- 137+ 254+} 100%
{20- 21- 35+ 137+ 254+} 100%
{20- 35+ 71- 137+ 254+} 100%
{5- 35+ 137+ 177+} 95.5%
{5- 35+ 137+ 254+} 95.5%
{5- 35+ 137+ 419-} 95.5%
{5- 137+ 177+ 309+} 95.5%
{5- 137+ 254+ 309+} 95.5%
{7- 21- 33+ 35+ 69+} 95.5%
{7- 21- 33+ 69+ 309+} 95.5%
{7- 21- 33+ 69+ 1261+} 95.5%

These EPs have 95%--100% support in one class but 0% support in the other class.

Minimal: Each proper subset occurs in both classes.

EPs from Mao+Dong 2005 (gene club + border-diff).

Colon cancer dataset (Alon et al, 1999 (PNAS)): 40 cancer tissues, 22 normal tissues. 2000 genes

A potential use of minimal jumping EPs

- Minimal jumping EPs for normal tissues

  ➔ Properly expressed gene groups important for normal cell functioning, but destroyed in all colon cancer tissues

  ➔ Restore these ➔ ?cure colon cancer?

- Minimal jumping EPs for cancer tissues

  ➔ Bad gene groups that occur in some cancer tissues but never occur in normal tissues

  ➔ Disrupt these ➔ ?cure colon cancer?

- ? Possible targets for drug design ?

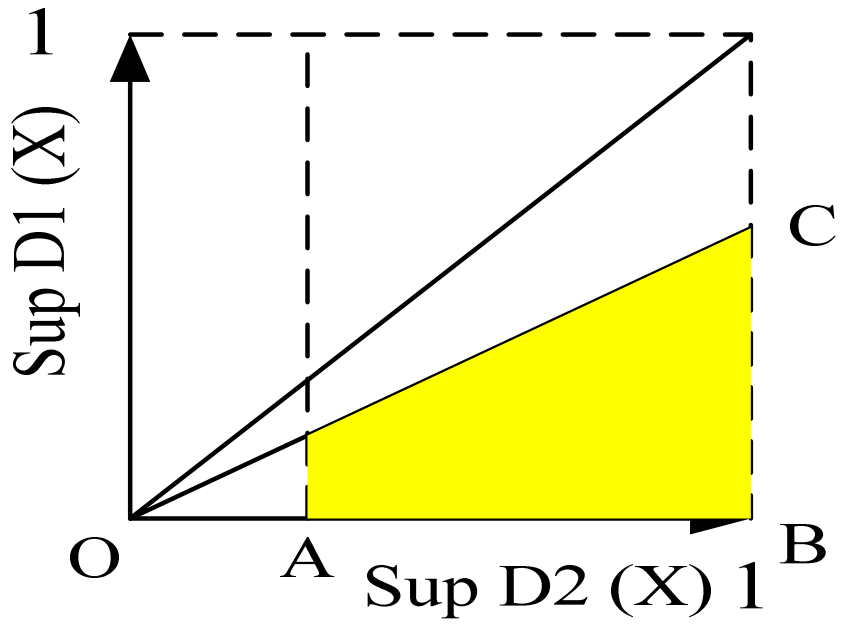
Li+Wong 2002 proposed "gene therapy using EP" idea: therapy aims to destroy bad JEP & restore good JEP

Usefulness of Emerging Patterns

- EPs are useful
  - for building highly accurate and robust classifiers, and for improving other types of classifiers
  - for discovering powerful distinguishing features between datasets.
- Like other patterns composed of conjunctive combination of elements, EPs are easy for people to understand and use directly.
- EPs can also capture patterns about change over time.
- Papers using EP techniques in Cancer Cell (cover, 3/02).
- Emerging Patterns have been applied in medical applications for diagnosing acute Lymphoblastic Leukemia.

The landscape of EPs on the support plane, and challenges for mining

Landscape of EPs



Challenges for EP mining

- EP minRatio constraint is neither monotonic nor anti-monotonic (but exceptions exist for special cases)
- Requires smaller support thresholds than those used for frequent pattern mining

Odds Ratio and Relative Risk Patterns [Li and Wong PODS06]

- May use odds ratio/relative risk to evaluate compound factors as well
  - May be no single factor with high relative risk or odds ratio, but a combination of factors
    - Relative risk patterns - Similar to emerging patterns
    - Risk difference patterns - Similar to contrast sets
    - Odds ratio patterns

Mining Patterns with High Odds Ratio or Relative Risk

- Space of odds ratio patterns and relative risk patterns are not convex in general
- Can become convex, if stratified into plateaus, based on support levels

EP Mining Algorithms

- Complexity result (Wang et al 05)
- Border-differential algorithm (Dong+Li 99)
- Gene club + border differential (Mao+Dong 05)
- Constraint-based approach (Zhang et al 00)
- Tree-based approach (Bailey et al 02, Fan+Ramamohanarao 02)
- Projection based algorithm (Bailey el al 03)
- ZBDD based method (Loekito+Bailey 06).

Complexity result

- The complexity of finding emerging patterns (even those with the highest frequency) is MAX SNP-hard.
  - This implies that polynomial time approximation schemes do not exist for the problem unless P=NP.

Borders are concise representations of convex collections of itemsets

- < minB={**12,13**}, maxB={*12345,12456*}>

	123, 1234	
12	124, 1235	*12345*
	125, 1245	*12456*
	126, 1246	
13	134, 1256	
	135, 1345	

A collection S is convex: If for all X,Y,Z (X in S, Y in S, X subset Z subset Y) ➔ Z in S.

Border-Differential Algorithm

- <{{}},{*1234*}> - **<{{}},{23,24,34}>**
  = <{1,234},{*1234*}>

  {}

  1, **2, 3, 4**

  12, 13, 14, **23, 24, 34**

  123, 124, 134, 234

  1234

Algorithm:

- Use iterations of expansion & minimization of "products" of differences
- Use tree to speed up minimization

  - Find minimal subsets of 1234 that are not subsets of 23, 24, 34.
  - {1,234} = min ({1,4} X {1,3} X {1,2})

- Good for: Jumping EPs; EPs in "rectangle regions," …

Iterative expansion & minimization can be viewed as optimized Berge hypergraph transversal algorithm

Gene club + Border Differential

- Border-differential can handle up to 75 attributes (using 2003 PC)
- For microarray gene expression data, there are thousands of genes.
- (Mao+Dong 05) used border-differential after finding many gene clubs -- one gene club per gene.
- A gene club is a set of k genes strongly correlated with a given gene and the classes.
- Some EPs discovered using this method were shown earlier. Discovered more EPs with near 100% support in cancer or normal, involving many different genes. Much better than earlier results.

Tree-based algorithm for JEP mining

- Use tree to compress data and patterns.
- Tree is similar to FP tree, but it stores two counts per node (one per class) and uses different item ordering
- Nodes with non-zero support for positive class and zero support for negative class are called base nodes.
- For every base node, the path's itemset is a potential JEP. Gather negative data containing root item and item for based nodes on the path. Call border differential.
- Item ordering is important. Hybrid (support ratio ordering first for a percentage of items, frequency ordering for other items) is best.

Projection based algorithm

- Form dataset H to contain the differences {p-ni | i=1…k}.
  - p is a positive transaction, n1, …, nk are negative transactions.
- Let x1<…<xm be increasing item frequency (in H) ordering.
- For i=1 to m
  - let Hxi be H with all items y > xi projected out & with all transactions containing xi removed (data projection).
  - remove non minimal transactions in Hxi.
  - if Hxi is small, do iterative expansion and minimization.
  - Otherwise, apply the algorithm on Hxi.

Let H be:
a b c d
b e d
b c e
c d e

Item ordering:
a < b < c < d < e

Ha is H with all items > a (red items) projected out and also edge with a removed, so Ha={}.

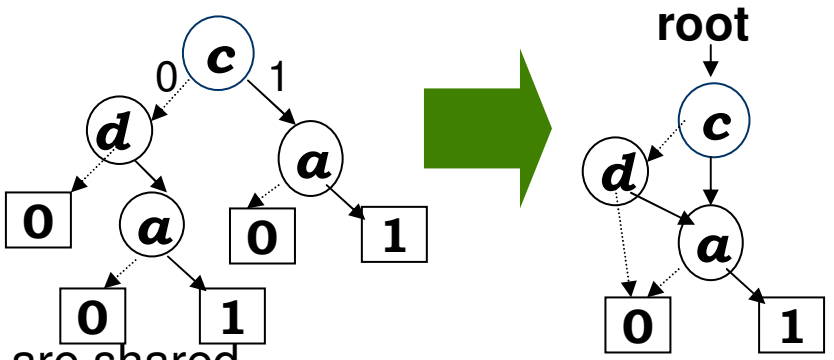
ZBDD based algorithm to mine disjunctive emerging patterns

- **Disjunctive Emerging Patterns:** allowing disjunction as well as conjunction of simple attribute conditions.
  - **e.g. Precipitation =** (*gt-norm* **OR** *lt-norm*) **AND Internal discoloration =** (*brown* **OR** *black)*
  - *Generalization of EPs*
- ZBDD based algorithm uses Zero Surpressed Binary Decision Diagram for efficiently mining disjunctive EPs.

Binary Decision Diagrams (BDDs)

- Popular in Boolean SAT solvers and reliability eng.
- Canonical DAG representations of Boolean formulae

f = (c ∧ a) ∨ (d ∧ a)



dotted (or 0) edge: don't link the nodes (in formulae)

- **Node sharing**: identical nodes are shared
- **Caching principle**: past computation results are automatically stored and can be retrieved

  Efficient BDD implementations available, e.g. CUDD (U of Colorado)

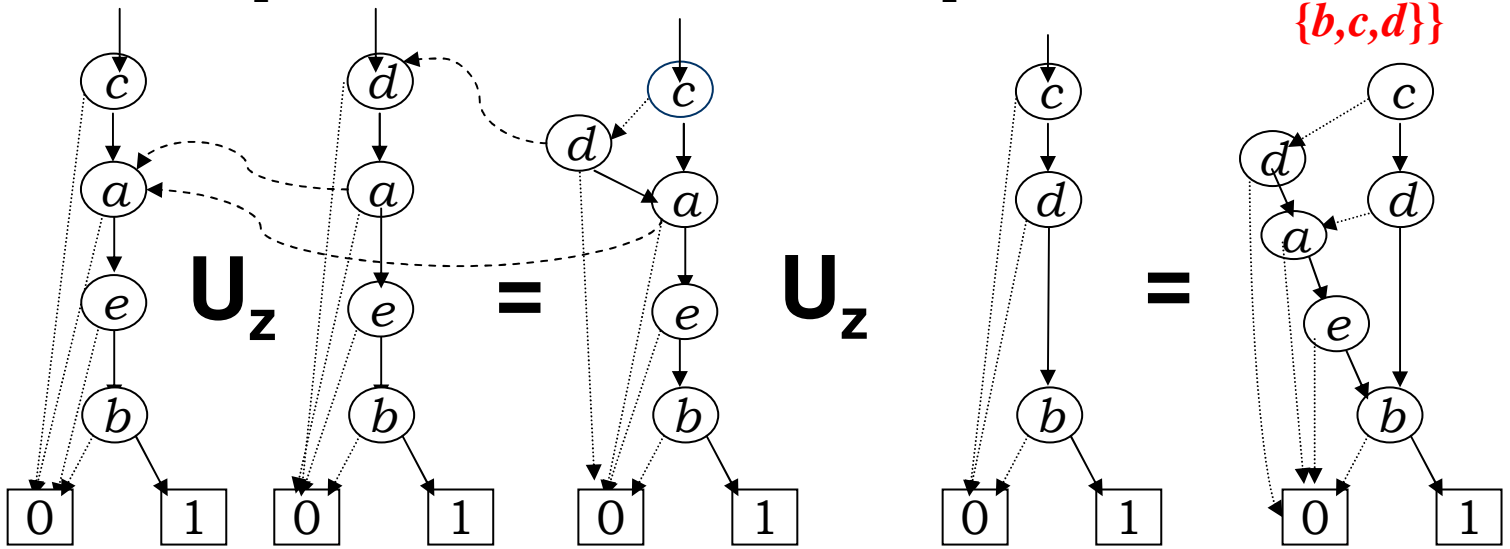
ZBDD Representation of Itemsets

James: what's the use of 0 edges? How do we reconstruct data?

- **Zero-suppressed BDD, ZBDD** : A BDD variant for manipulation of item combinations
- E.g. Building a ZBDD for $\{\{a,b,c,e\},\{a,b,d,e\},\{b,c,d\}\}$

Ordering : *c < d < a < e < b*

$\{\{a,b,c,e\}\} \cup_z \{\{a,b,d,e\}\} = \{\{a,b,c,e\},\{a,b,d,e\}\} \cup_z \{\{b,c,d\}\} = \{\{a,b,c,e\},\{a,b,d,e\},\{b,c,d\}\}$

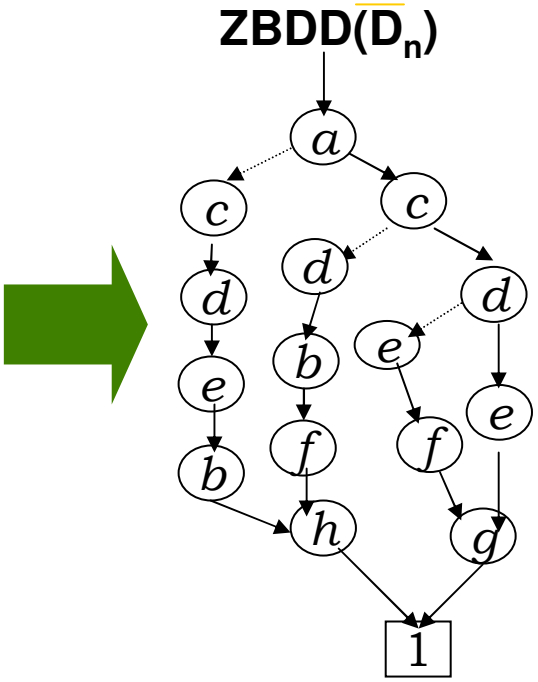


$\cup_z$ = ZBDD set-union

ZBDD based mining example

Use solid paths in ZBDD(D_n) to generate candidates, and use Bitmap of D_p to check frequency support in D_p.

D_p			D_n		
A1	A2	A3	A1	A2	A3
a	e	g	a	f	g
a	d	i	b	d	h
b	f	h	b	f	h
c	e	h	c	e	g



Bitmap

		a	b	c	d	e	f	g	h	i
D_p=	P1:	1	0	0	0	1	0	1	0	0
	P2:	1	0	0	1	0	0	0	0	1
	P3:	0	1	0	0	0	1	0	1	0
	P4:	0	0	1	0	1	0	0	1	0
D_n=	N1:	1	0	0	0	0	1	1	0	0
	N2:	0	1	0	1	0	0	0	1	0
	N3:	0	1	0	0	0	1	0	1	0
	N4:	0	0	1	0	1	0	1	0	0

Ordering: a<c<d<e<b<f<g<h

Contrast pattern based classification -- history

- Contrast pattern based classification: Methods to build or improve classifiers, using contrast patterns
  - CBA (Liu et al 98)
  - CAEP (Dong et al 99)
  - Instance based method: DeEPs (Li et al 00, 04)
  - Jumping EP based (Li et al 00), Information based (Zhang et al 00), Bayesian based (Fan+Kotagiri 03), improving scoring for >=3 classes (Bailey et al 03)
  - CMAR (Li et al 01)
  - Top-ranked EP based PCL (Li+Wong 02)
  - CPAR (Yin+Han 03)
  - Weighted decision tree (Alhammady+Kotagiri 06)
  - Rare class classification (Alhammady+Kotagiri 04)
  - Constructing supplementary training instances (Alhammady+Kotagiri 05)
  - Noise tolerant classification (Fan+Kotagiri 04)
  - EP length based 1-class classification of rare cases (Chen+Dong 06)
  - …
- Most follow the aggregating approach of CAEP.

EP-based classifiers: rationale

- Consider a typical EP in the Mushroom dataset, {odor = none, stalk-surface-below-ring = smooth, ring-number = one}; its support increases from 0.2% from “poisonous” to 57.6% in “edible” (growth rate = 288).
- Strong differentiating power: if a test T contains this EP, we can predict T as edible with high confidence 99.6% = 57.6/(57.6+0.2)
- A single EP is usually sharp in telling the class of a small fraction (e.g. 3%) of all instances. Need to aggregate the power of many EPs to make the classification.
- EP based classification methods often out perform state of the art classifiers, including C4.5 and SVM. They are also noise tolerant.

CAEP (Classification by Aggregating Emerging Patterns)

- Given a test case T, obtain T's scores for each class, by aggregating the discriminating power of EPs contained by T; assign the class with the maximal score as T's class.
- The discriminating power of EPs are expressed in terms of supports and growth rates. Prefer large supRatio, large support
- The contribution of one EP X (support weighted confidence):

$$strength(X) = sup(X) * supRatio(X) / (supRatio(X)+1)$$

Compare CMAR: Chi2 weighted Chi2

- Given a test T and a set E(Ci) of EPs for class Ci, the aggregate score of T for Ci is $score(T, Ci) = \Sigma\ strength(X)$ (over X of Ci matching T)
  - For each class, using median (or 85%) aggregated value to normalize to avoid bias towards class with more EPs

How CAEP works? An example

- Given a test T={a,d,e}, how to classify T?

- T contains EPs of class 1 : {a,e} (50%:25%) and {d,e} (50%:25%), so Score(T, class1) =

  0.5*[0.5/(0.5+0.25)] + 0.5*[0.5/(0.5+0.25)] = 0.67

- T contains EPs of class 2: {a,d} (25%:50%), so Score(T, class 2) = 0.33;

- T will be classified as class 1 since Score1>Score2

Class 1 (D1)

a	c	d	e
a	e		
b	c	d	e
b			

Class 2 (D2)

a	b		
a	b	c	d
c	e		
a	b	d	e

DeEPs (Decision-making by Emerging Patterns)

- An instance based (lazy) learning method, like k-NN; but does not use normal distance measure.
- For a test instance T, DeEPs
  - First project each training instance to contain only items in T
  - Discover EPs from the projected data
  - Then use these EPs to select training data that match some discovered EPs
  - Finally, use the proportional size of matching data in a class C as T's score for C
- Advantage: disallow similar EPs to give duplicate votes!

DeEPs : Play-Golf example (data projection)

Test = {sunny, mild, high, true}

Original data

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
rain	cool	normal	true	N
sunny	mild	high	false	N
rain	mild	high	true	N
overcast	hot	high	FALSE	P
rain	mild	high	FALSE	P
rain	cool	normal	FALSE	P
overcast	cool	normal	TRUE	P
sunny	cool	normal	FALSE	P
rain	mild	normal	FALSE	P
sunny	mild	normal	TRUE	P
overcast	mild	high	TRUE	P
overcast	hot	normal	FALSE	P

Projected data

Outlook	Temperature	Humidity	Windy	Class
sunny		high		N
sunny		high	true	N
			true	N
sunny	mild	high		N
	mild	high	true	N
		high		P
	mild	high		P
			TRUE	P
sunny				P
	mild			P
sunny	mild		TRUE	P
	mild	high	TRUE	P

Discover EPs and derive scores using the projected data

PCL (Prediction by Collective Likelihood)

- Let X1,…,Xm be the m (e.g. 1000) most general EPs in descending support order.
- Given a test case T, consider the list of all EPs that match T. Divide this list by EP's class, and list them in descending support order:

  P class: Xi1, …, Xip

  N class: Xj1, …, Xjn
- Use k (e.g. 15) top ranked matching EPs to get score for T for the P class (similarly for N):

$$\text{Score}(T,P) = \Sigma_{t=1}^{k} \text{suppP}(Xit) / \text{supp}(Xt)$$

normalizing factor

EP selection factors

- There are many EPs, can't use them all. Should select and use a good subset.
- EP selection considerations include
  - Keep minimal (shortest, most general) ones
  - Remove syntactic similar ones
  - Use support/growth rate improvement (between superset/subset pairs) to prune
  - Use instance coverage/overlap to prune
  - Using only JEPs
  - ……

Why EP-based classifiers are good

- Use discriminating power of **low** support EPs, together with high support ones
- Use **multi-feature** conditions, not just single-feature conditions
- Select from **larger pools** of discriminative conditions
  - Compare: Search space of patterns for decision trees is limited by early greedy choices.
- **Aggregate/combine** discriminating power of a diversified committee of “experts” (EPs)
- Decision is highly **explainable**

Some other works

- CBA (Liu et al 98) uses one rule to make a classification prediction for a test
- CMAR (Li et al 01) uses **aggregated** (Ch2 weighted) Chi2 of matching rules
- CPAR (Yin+Han 03) uses aggregation by averaging: it uses the average accuracy of top k rules for each class matching a test case
- …

Aggregating EPs/rules vs bagging (classifier ensembles)

- Bagging/ensembles: a committee of classifiers vote
  - Each classifier is fairly accurate for a large population (e.g. >51% accurate for 2 classes)
- Aggregating EPs/rules: matching patterns/rules vote
  - Each pattern/rule is accurate on a very small population, but inaccurate if used as a classifier on all data; e.g. 99% accurate on 2% of data, but 2% accurate on all data

Using contrasts for rare class data

[Al Hammady and Ramamohanarao 04,05,06]

- Rare class data is important in many applications
  - Intrusion detection (1% of samples are attacks)
  - Fraud detection (1% of samples are fraud)
  - Customer click thrus (1% of customers make a purchase)
  - …..

Rare Class Datasets

- Due to the class imbalance, can encounter some problems
  - Few instances in the rare class, difficult to train a classifier
  - Few contrasts for the rare class
  - Poor quality contrasts for the majority class
- Need to either *increase the instances* in the rare class or *generate extra contrasts* for it

Synthesising new contrasts (new emerging patterns)

- Synthesising new emerging patterns by superposition of high growth rate items
  - Suppose that attribute A2=`a' has high growth rate and that {A1=`x', A2=`y'} is an emerging pattern. Then create a new emerging pattern {A1='x', A2='a'} and test its quality.
- A simple heuristic, but can give surprisingly good classification performance

Synthesising new data instances

- Can also use previously found contrasts as the basis for constructing new rare class instances
  - Combine overlapping contrasts and high growth rate items
- Main idea - intersect and `cross product' the emerging patterns and high growth rate (support ratio) items
  - Find emerging patterns
  - Cluster emerging patterns into groups that cover all the attributes
  - Combine patterns within each group to form instances

Synthesising new instances

- E1{A1=1, A2=X1}, E2{A5=Y1,A6=2,A7=3},
  E3{A2=X2,A3=4,A5=Y2} - this is a group

  V4 is a high growth item for A4

  Combine E1+E2+E3+{A4=V4} to get four synthetic instances.

A1	A2	A3	A4	A5	A6	A7
1	X1	4	V4	Y1	2	3
1	X1	4	V4	Y2	2	3
1	X2	4	V4	Y1	2	3
1	X2	4	V4	Y2	2	3

Measuring instance quality using emerging patterns

[Al Hammady and Ramamohanarao 07]

- Classifiers usually assume that data instances are related to only a single class (crisp assignments).
- However, real life datasets suffer from noise.
- Also, when experts assign an instance to a class, they first assign scores to each class and then assign the class with the highest score.
- Thus, an instance may in fact be related to several classes

Measuring instance quality Cont.

- For each instance *i*, assign a weight that represents its strength of membership in each class. Can use emerging patterns to determine appropriate weights for instances
  - Use aggregation of EPs divided by mean value for instances in that class to give an instance weight
- Use these weights in a modified version of classifier, e.g. a decision tree
  - Modify information gain calculation to take weights into account

Using EPs to build Weighted Decision Trees

- Instead of crisp class membership,
  - let instances have weighted class membership,
  - then build weighted decision trees, where probabilities are computed from the weighted membership.
- DeEPs and other EP based classifiers can be used to assign weights.

An instance Xi's membership in k classes: (Wi1,…,Wik)

$$\hat{P}(T) = (\hat{p}_1(T) = \frac{\sum_{i \in T} Wi1}{|T|}, ..., \hat{p}_k(T) = \frac{\sum_{i \in T} Wik}{|T|})$$

$$Info_{WDT}(\hat{P}(T)) = -\sum_{j=1}^{k} \hat{p}_j(T) * \log_2(\hat{p}_j(T))$$

$$Info_{WDT}(A,T) = \sum_{l=1}^{m} \frac{|T_l|}{|T|} Info(\hat{P}(T_l))$$

Measuring instance quality by emerging patterns Cont.

- More effective than k-NN techniques for assigning weights
  - Less sensitive to noise
  - Not dependent on distance metric
  - Takes into account all instances, not just close neighbors

Data cube based contrasts

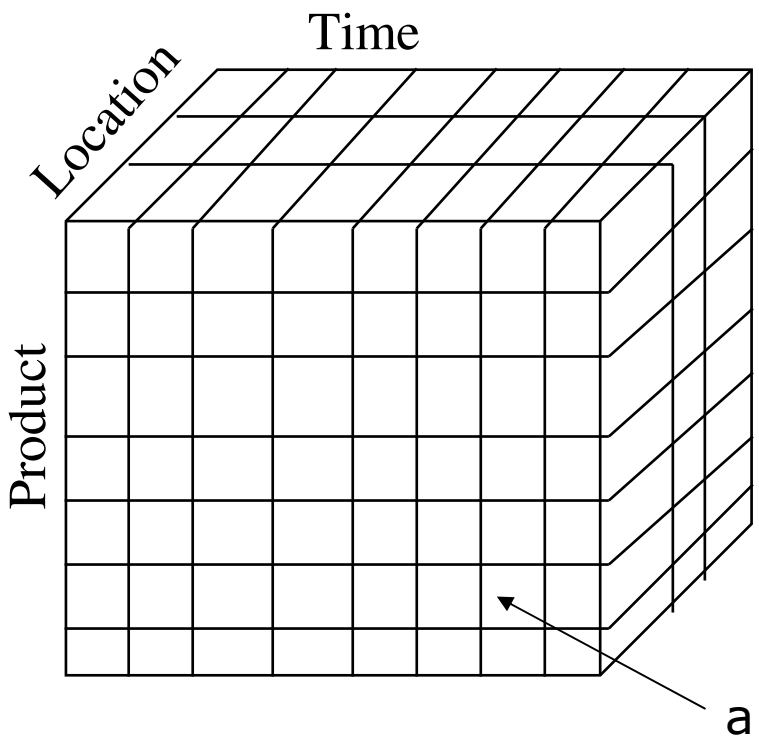
- Gradient (Dong et al 01), cubegrade (Imielinski et al 02 – TR published in 2000):
  - Mining syntactically similar cube cells, having significantly different measure values
  - Syntactically similar: ancestor-descendant or sibling-sibling pair
  - Can be viewed as "**conditional contrasts**": two neighboring patterns with big difference in performance/measure
- Data cubes useful for analyzing multi-dimensional, multi-level, time-dependent data.
- Gradient mining useful for MDML analysis in marketing, business, medical/scientific studies

Decision support in data cubes

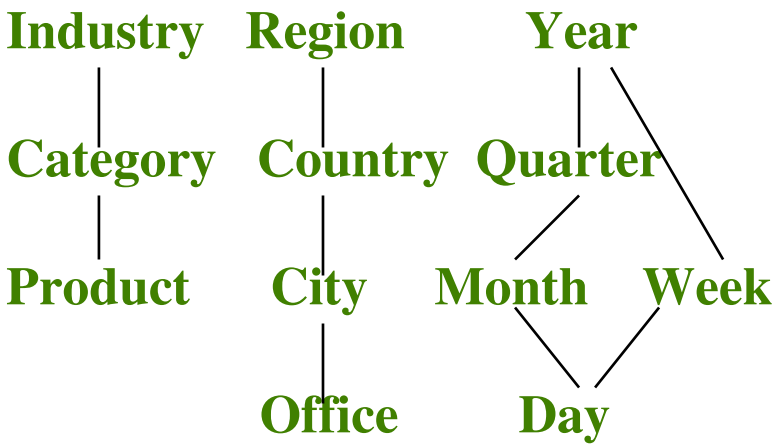
- Used for discovering patterns captured in consolidated historical data for a company/organization:
  - rules, anomalies, unusual factor combinations
- Focus on modeling & analysis of data for decision makers, not daily operations.
- Data organized around major subjects or factors, such as customer, product, time, sales.
- Cube “contains” huge number of MDML “segment” or “sector” summaries at different levels of details
- Basic OLAP operations: Drill down, roll up, slice and dice, pivot

Data Cubes: Base Table & Hierarchies

- Base table stores sales volume (*measure*), a function of product, time, & location (dimensions)

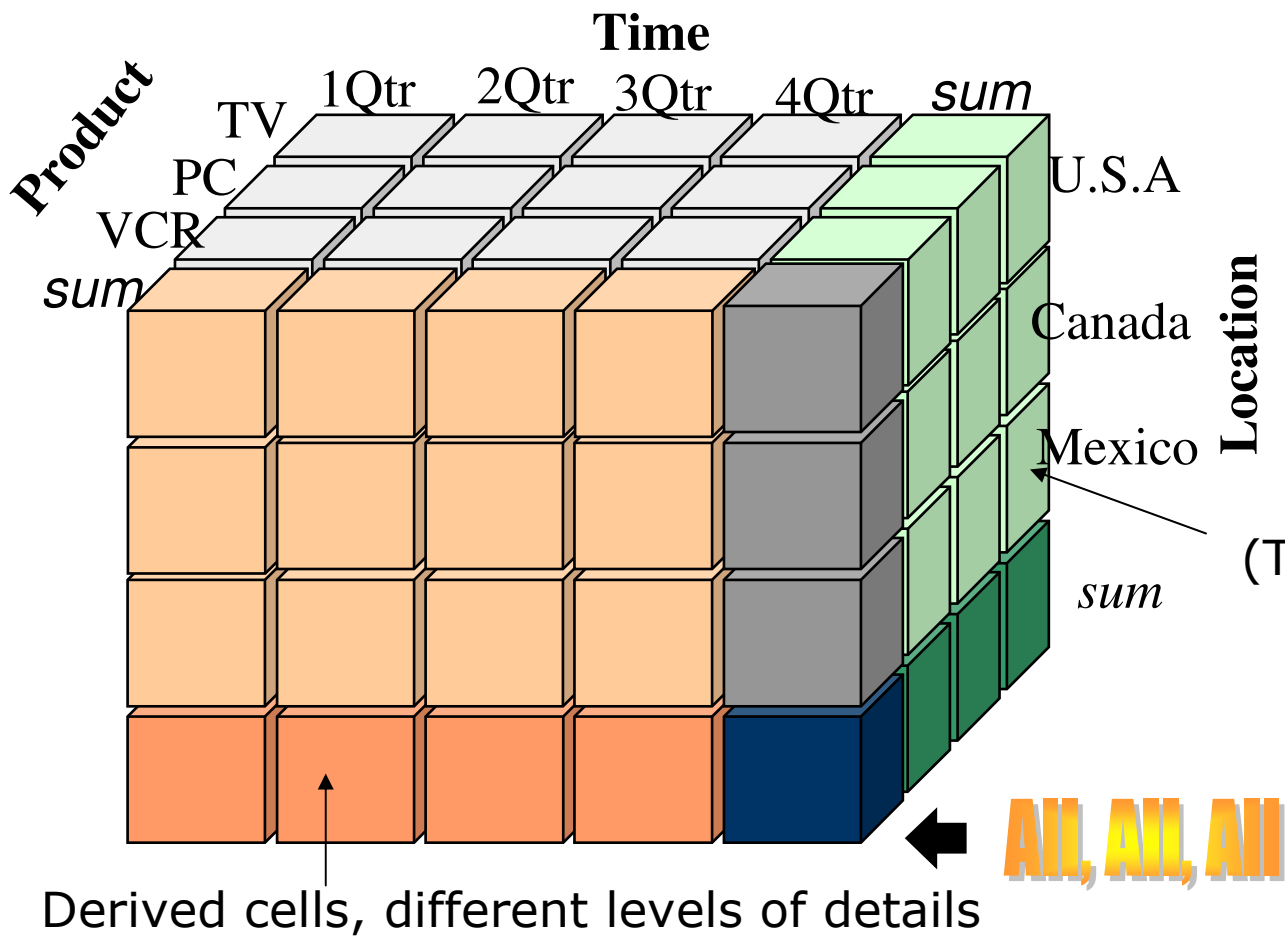


Hierarchical summarization paths



*: all (as top of each dimension)

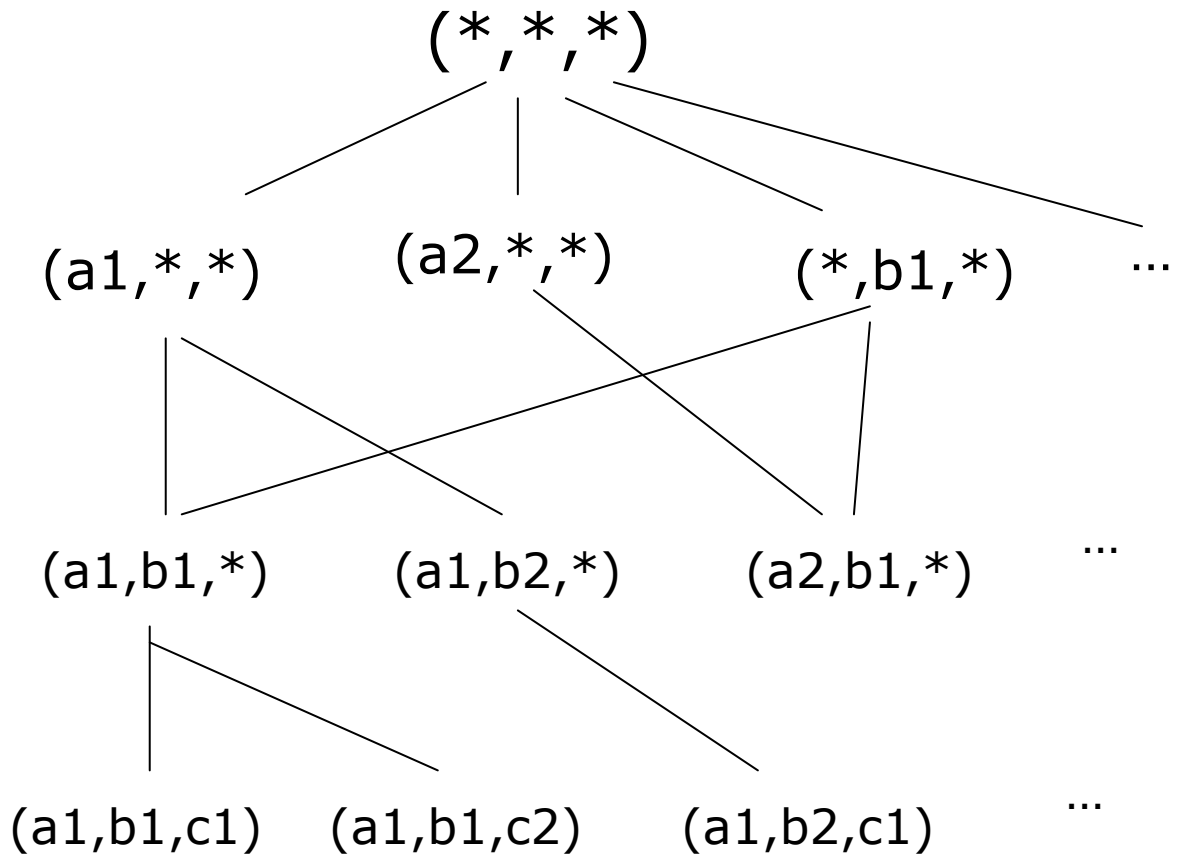
Data Cubes: Derived Cells



Measures:
sum, count,
avg, max,
min, std, …

Data Cubes: Cell Lattice

Compare:
cuboid lattice



Gradient mining in data cubes

- Users want: more powerful (OLAM) support: Find potentially interesting cells from the billions!
  - OLAP operations used to help users search in huge space of cells
  - Users do: *mousing*, *eye-balling*, *memoing, decisioning, …*
- Gradient mining: Find syntactically similar cells with significantly different measure values
  - (teen clothing,California,2006), total-profit=100K
  - vs (teen clothing,Pensylvania,2006), total profit = 10K

  - A specific OLAM task

LiveSet-Driven Algorithm for constrained gradient mining

- Set-oriented processing; traverse the cube while carrying the *live set* of cells having potential to match descendants of the current cell as gradient cells
  - A gradient compares two cells; one is the probe cell, & the other is a gradient cell. Probe cells are ancestor or sibling cells
- Traverse the cell space in a coarse-to-fine manner, looking for matchable gradient cells with potential to satisfy gradient constraint
- Dynamically prune the live set during traversal
- Compare: Naïve method checks each possible cell pair

Pruning probe cells using dimension matching analysis

- Defn: Probe cell p=(a_1,…,a_n) is matchable with gradient cell g=(b_1, …, b_n) iff
  - No solid-mismatch, or
  - Only one solid-mismatch but no *-mismatch
- A solid-mismatch: if $a_j \neq b_j$ + none of a_j or b_j is *
- A *-mismatch: if a_j=* and $b_j \neq$*

p=(00, Tor, *, *) : 1 solid
g=(00, Chi, *,PC) : 1 *

- Thm: cell p is matchable with cell g iff p may make a probe-gradient pair with some descendant of g (using only dimension value info)

Sequence based contrasts

- We want to compare sequence datasets:
  - bioinformatics (DNA, protein), web log, job/workflow history, books/documents
  - e.g. compare protein families; compare bible books/versions
- Sequence data are very different from relational data
  - order/position matters
  - unbounded number of “flexible dimensions”
- Sequence contrasts in terms of 2 types of comparison:
  - Dataset based: Positive vs Negative
    - Distinguishing sequence patterns with gap constraints (Ji et al 05, 07)
    - Emerging substrings (Chan et al 03)
  - Site based: Near marker vs away from marker
    - Motifs
    - May also involve data classes

Roughly: A site is a position in a sequence where a special marker/pattern occurs

Example sequence contrasts

When comparing the two protein families *zf-C2H2* and *zf-CCHC*, we discovered a protein MDS *CLHH* appearing as a subsequence in *141* of *196* protein sequences of *zf-C2H2* but never appearing in the *208* sequences in *zf-CCHC*.

When comparing the first and last books from the Bible, we found the subsequences (with gaps) “having horns”, “face worship”, “stones price” and “ornaments price” appear multiple times in sentences in the Book of Revelation, but never in the Book of Genesis.

Sequence and sequence pattern occurrence

- A sequence $S = e_1 e_2 e_3 \cdots e_n$ is an ordered list of items over a given alphabet.
  - E.G. "$AGCA$" is a DNA sequence over the alphabet {A, C, G, T}.
  - "AC" is a subsequence of "$AGCA$" but not a substring;
  - "GCA" is a substring
- Given sequence S and a subsequence pattern S', an occurrence of S' in S consists of the positions of the items from S' in S.
- EG: consider S = "$ACACBCB$"
  - <1,5>, <1,7>, <3,5>, <3,7> are occurrences of "AB"
  - <1,2,5>, <1,2,7>, <1,4,5>, … are occurrences of "ACB"

Maximum-gap constraint satisfaction

- A (maximum) gap constraint: specified by a positive integer g.
- Given S & an occurrence o_s = <i_1, … i_m>, if $i_{k+1} - i_k <= g + 1$ for all 1 <= k <m, then o_s fulfills the g-gap constraint.
- If a subsequence S' has one occurrence fulfilling a gap constraint, then S' satisfies the gap constraint.
  - The <3,5> occurrence of “AB” in S = “AC**A**CB**B**CB”, satisfies the maximum gap constraint g=1.
  - The <3,4,5> occurrence of “ACB” in S = “AC**ACB**CB” satisfies the maximum gap constraint g=1.
  - The <1,2,5>, <1,4,5>, <3,4,5> occurrences of “ACB” in S = “ACACBCB” satisfy the maximum gap constraint g=2.
- One sequence contribute to at most one to count.

g-MDS Mining Problem

Given two sets pos & neg of sequences, two support thresholds *minp* & *maxn*, & a maximum gap g, a pattern p is a *Minimal Distinguishing Subsequence* with g-gap constraint (g-MDS), if these conditions are met:

1. Frequency condition: $supp_{pos}(p, g)$ >= minp;
2. Infrequency condition: $supp_{neg}(p, g)$ <= maxn;
3. Minimality condition: There is no subsequence of p satisfying 1 & 2.

Given pos, neg, *minp, minn* and g, the g-MDS mining problem is to find all the g-MDSs.

Example g-MDS

- Given *minp*=1/3, *maxn*=0, *g*=1,
  - *pos* = {CBAB, AACCB, BBAAC},
  - neg = {BCAB,ABACB}
- 1-MDS are: BB, CC, BAA, CBA
  - *"ACC"* is frequent in *pos* & non-occurring in *neg*, but it is not minimal (its subsequence "*CC*" meets the first two conditions).

g-MDS mining : Challenges

- The support thresholds in mining distinguishing patterns need to be lower than those used for mining frequent patterns.
  - Min supports offer very weak pruning power on the large search space.
- Maximum gap constraint is neither monotone nor anti-monotone.
- Gap checking requires clever handling.

ConSGapMiner

- The ***ConSGapMiner*** algorithm works in three steps:

1. Candidate Generation:
   Candidates are generated without duplication. Efficient pruning strategies are employed.

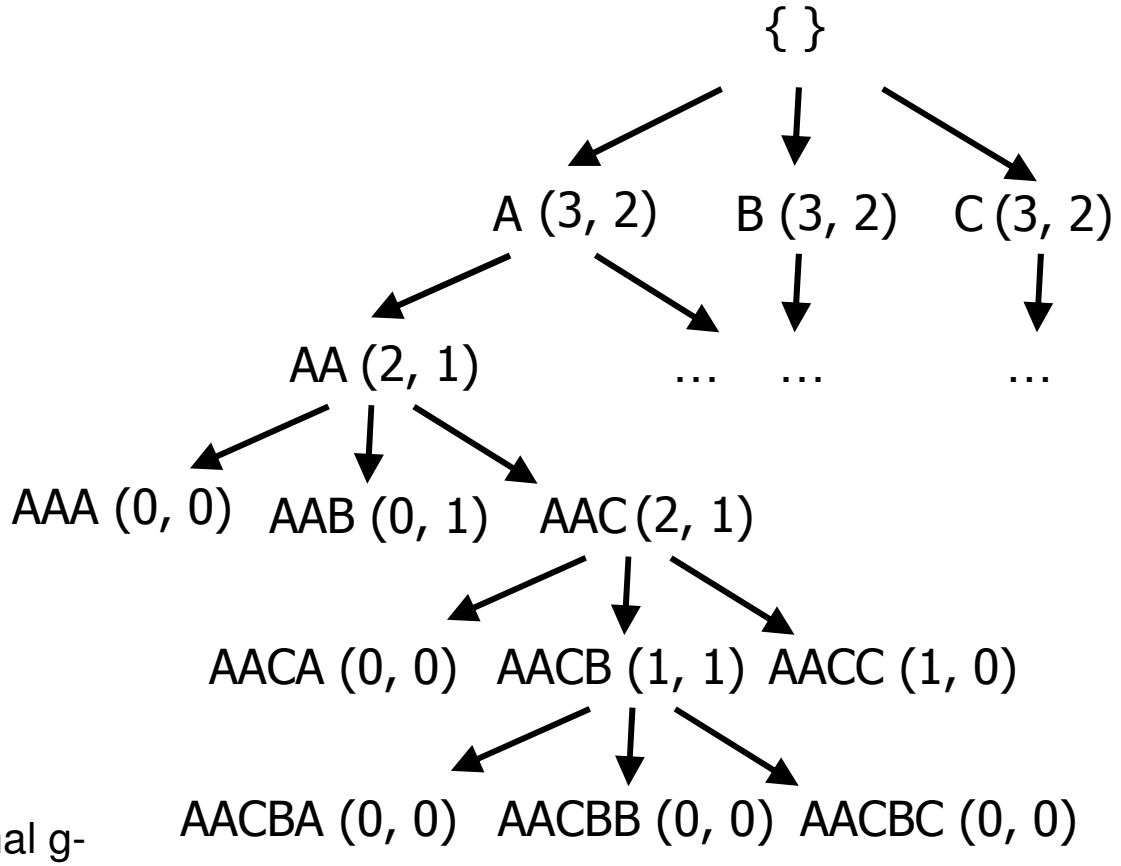
2. Support Calculation and Gap Checking:
   For each generated candidate c, $supp_{pos}(c, g)$ and $supp_{neg}(c, g)$ are calculated using bitset operations.

3. Minimization:
   Remove all the non-minimal patterns (using pattern trees).

ConSGapMiner : Candidate Generation

ID	Sequence	Class
1	*CBAB*	*pos*
2	*AACCB*	*pos*
3	*BBAAC*	*pos*
4	*BCAB*	*neg*
5	*ABACB*	*neg*

- DFS tree
- Two counts per node/pattern
- Don’t extend pos-infrequent patterns
- Avoid duplicates & certain non-minimal g-MDS (e.g. don’t extend g-MDS)



Use Bitset Operation for Gap Checking

Storing projected suffixes and performing scans is expensive.

e.g. Given a sequence ***ACTGTATTACCAGTATCG***

to check whether ***AG*** is a subsequence for ***g*=1**:

We encode the occurrences' ending positions into a bitset and use a series of bitwise operations to generate a new candidate sequence's bitset.

Projections with prefix *A* :

ACTGTATTACCAGTATCG
ATTACCAGTATCG
ACCAGTATCG
AGTATCG
ATCG

Projections with *AG obtained from the above*:

AGTATCG

ConSGapMiner: Support & Gap Checking (1)

- Initial Bitset Array Construction: For each item x, construct an array of bitsets to describe where x occurs in each sequence from *pos* and *neg*.

Dataset

ID	Sequence	Class
1	*CBAB*	*pos*
2	*AACCB*	*pos*
3	*BBAAC*	*pos*
4	*BCAB*	*neg*
5	*ABACB*	*neg*

Initial Bitset Array

single-item *A*
0010
11000
00110
0010
10100

ConSGapMiner: Support & Gap Checking (2)

Two steps: (1) g+1 right shifts; (2) OR them

EG: generate mask bitset for X = "A" in sequence 5 (with max gap $g = 1$):

ID	Sequence	Class
1	*C B A B*	***pos***
2	*A A C C B*	***pos***
3	*B B A A C*	***pos***
4	*B C A B*	***neg***
5	*A B A C B*	***neg***

$1\ 0\ 1\ 0\ 0 \gg 0\ 1\ 0\ 1\ 0$

$0\ 1\ 0\ 1\ 0 \gg 0\ 0\ 1\ 0\ 1$

OR

Mask bitset for X : $0\ 1\ 1\ 1\ 1$

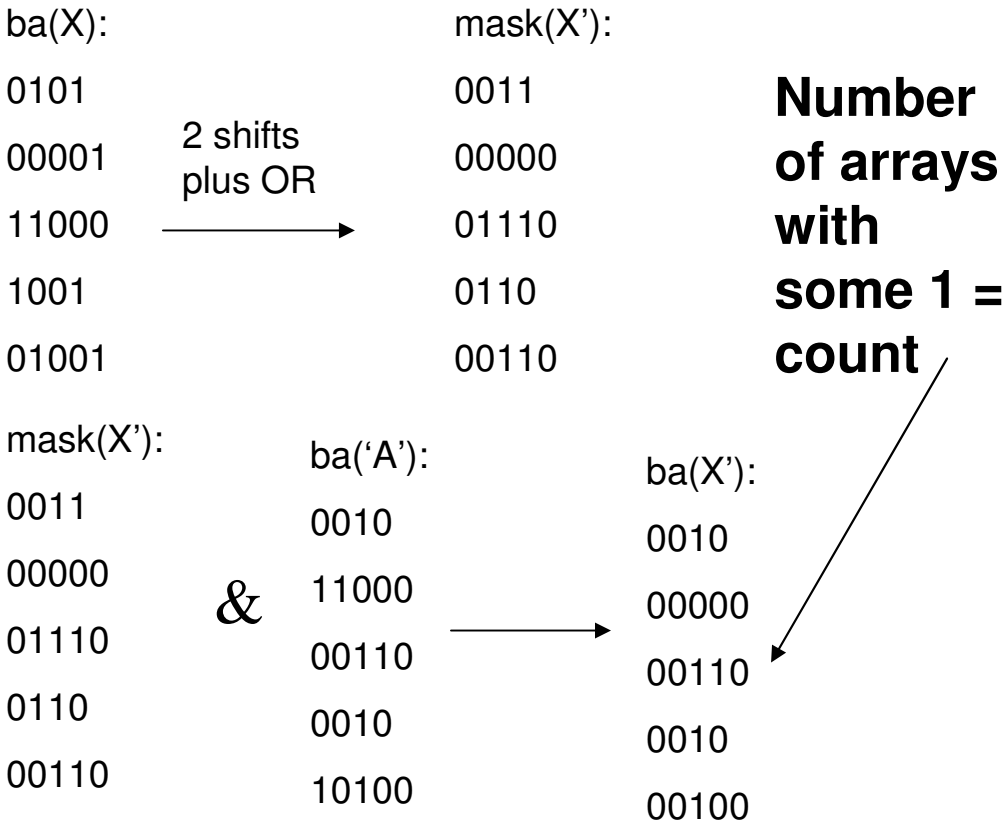
Mask bitset: all the legal positions in the sequence at most (g+1)-positions away from tail of an occurrence of the (maximum prefix of the) pattern.

ConSGapMiner: Support & Gap Checking (3)

EG: Generate bitset array (ba) for $X' = \text{“}BA\text{”}$ from $X = \text{‘}B\text{’}$ $(g = 1)$

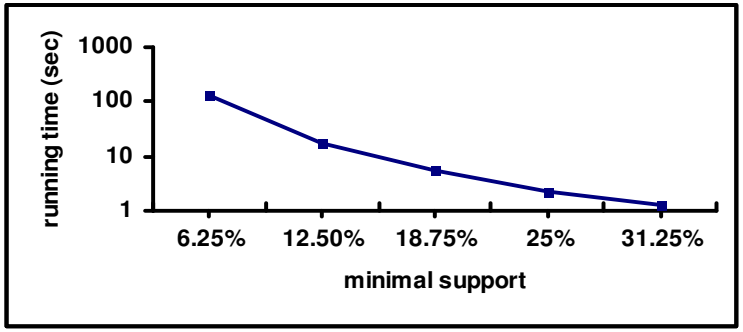
1. Get ba for X=‘B’
2. Shift ba(X) to get mask for X’ = ‘BA’
3. AND ba(‘A’) and mask(X’) to get ba(X’)

ID	Sequence	Class
1	*C B A B*	***pos***
2	*A A C C B*	***pos***
3	*B B A A C*	***pos***
4	*B C A B*	***neg***
5	*A B A C B*	***neg***

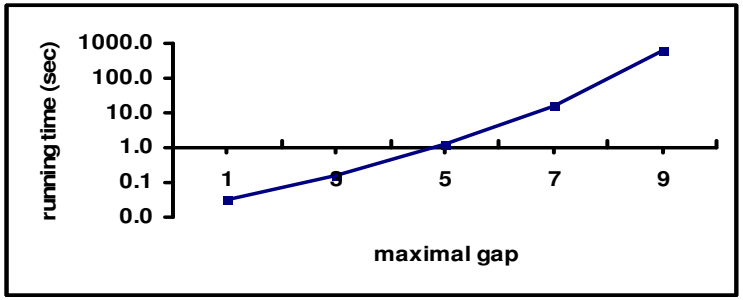


Execution time performance on protein families

Pos(#)	Neg(#)	Avg. Len. (Pos, Neg)
DUF1694 (16)	DUF1695 (5)	(123, 186)

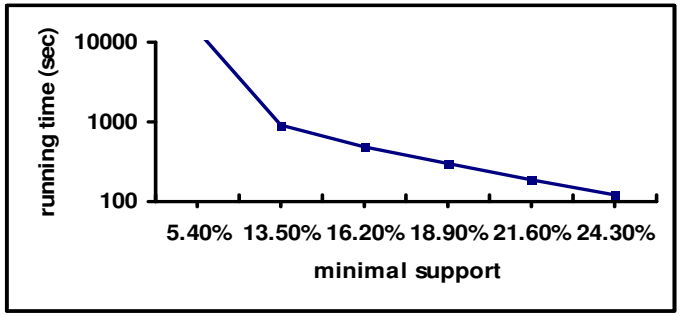


runtime vs support, for g = 5

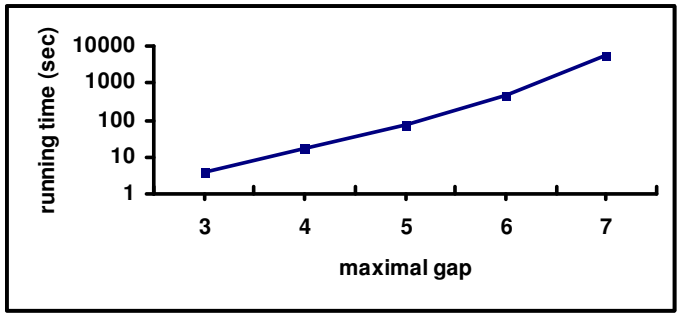


runtime vs g, for α = 0.3125(5)

Pos(#)	Neg(#)	Avg. Len. (Pos, Neg)
TatC (74)	TatD_DNase(119)	(205, 262)



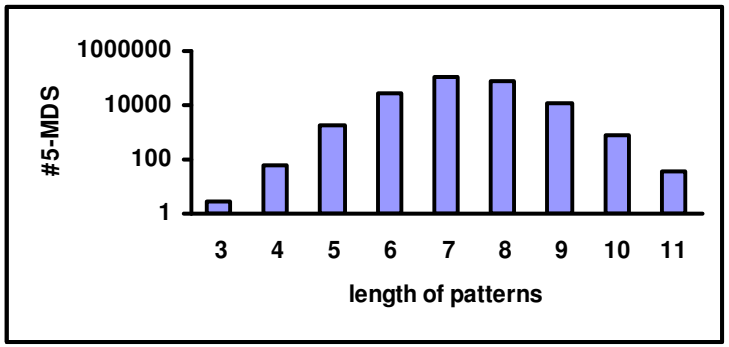
runtime vs support, for g = 5



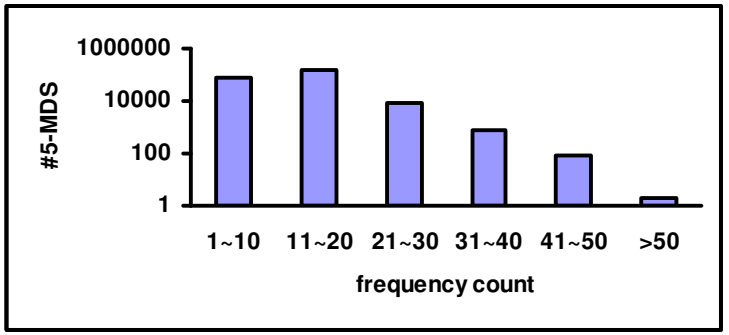
runtime vs g, for α = 0.27(20)

Pattern Length Distribution -- Protein Families

The length and frequency distribution of patterns: TaC vs TatD_DNase, $g = 5$, $\alpha = 13.5\%$.

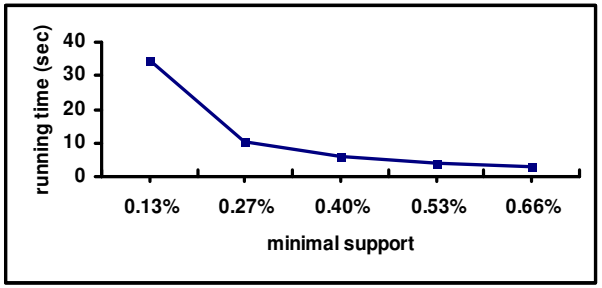


Length distribution

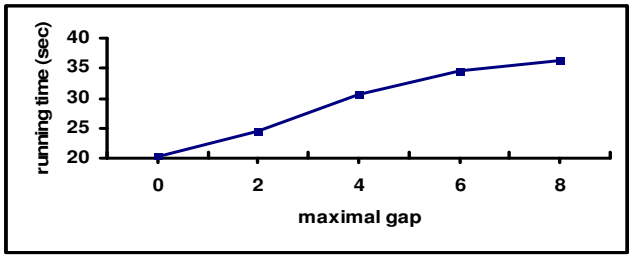


Frequency distribution

Bible Books Experiment



runtime vs support, for g = 6.



runtime vs g, for α = 0.0013.

New Testament (Matthew, Mark, Luke and John) vs Old Testament (Genesis, Exodus, Leviticus and Numbers):

#Pos	#Neg	Alphabet	Avg. Len.	Max. Len.
3768	4893	3344	7	25

Some interesting terms found from the Bible books (New Testament vs Old Testament):

Substrings (count)	Subsequences (count)
eternal life (24)	seated hand (10)
good news (23)	answer truly (10)
Forgiveness in (22)	Question saying (13)
Chief priests (53)	Truly kingdom (12)

Extensions

- Allowing min gap constraint
- Allowing max window length constraint
- Considering different minimization strategies:
  - Subsequence-based minimization (described on previous slides)
  - Coverage (matching tidset containment) + subsequence based minimization
  - Prefix based minimization

Motif mining

- Find sequence patterns frequent around a site marker, but infrequent elsewhere
- Can also consider two classes:
  - Find patterns frequent around site marker in +ve class, but in frequent at other positions, and infrequent around site marker in –ve class
  - Often, biological studies use background probabilities instead of a real -ve dataset
- Popular concept/tool in biological studies

Contrasts for Graph Data

- Can capture structural differences
  - Subgraphs appearing in one class but not in the other class
    - Chemical compound analysis
    - Social network comparison

Contrasts for graph data Cont.

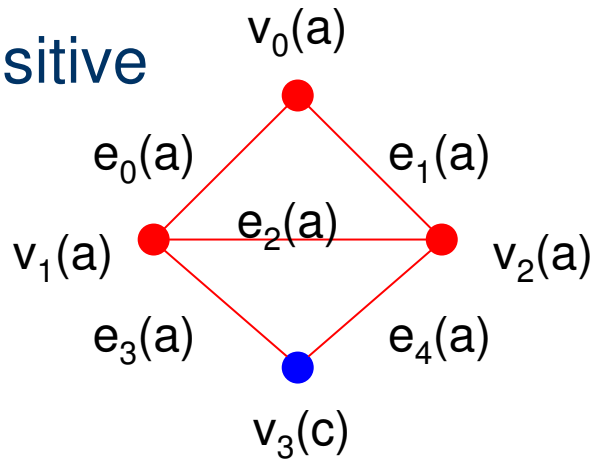
- Standard frequent subgraph mining
  - Given a graph database, find connected subgraphs appearing frequently
- Contrast subgraphs particularly focus on discrimination and minimality

Minimal contrast subgraphs [Ting and Bailey 06]

- A contrast graph is a subgraph appearing in once class of graphs and never in another class of graphs
  - Minimal if none of its subgraphs are contrasts
  - May be disconnected
    - Allows succinct description of differences
    - But requires larger search space
- Will focus on one versus one case

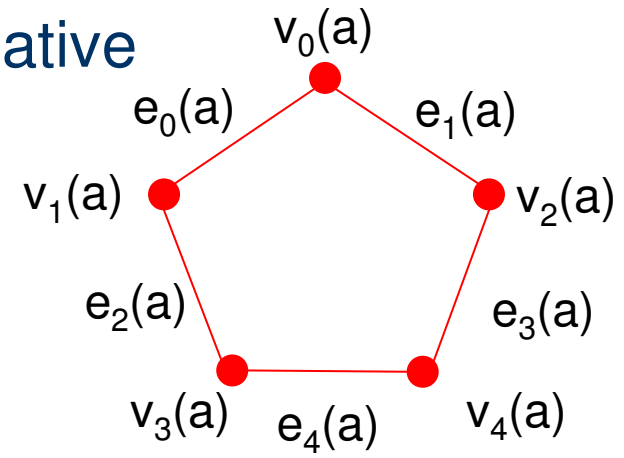
Contrast subgraph example

Positive



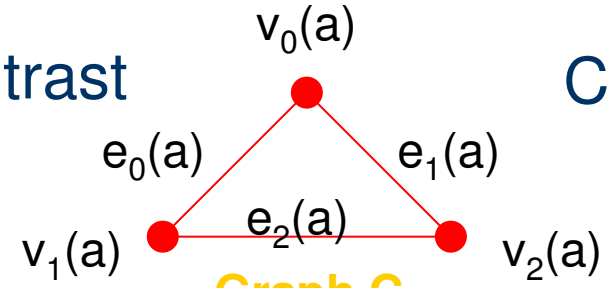
Graph A

Negative



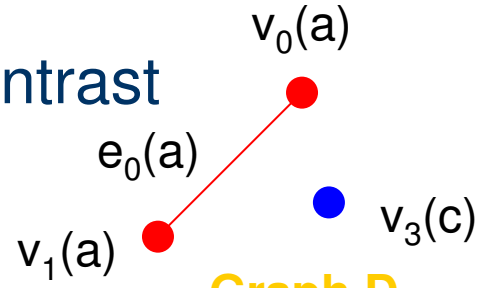
Graph B

Contrast



Graph C

Contrast



Graph D

Contrast

Graph E

Minimal contrast subgraphs

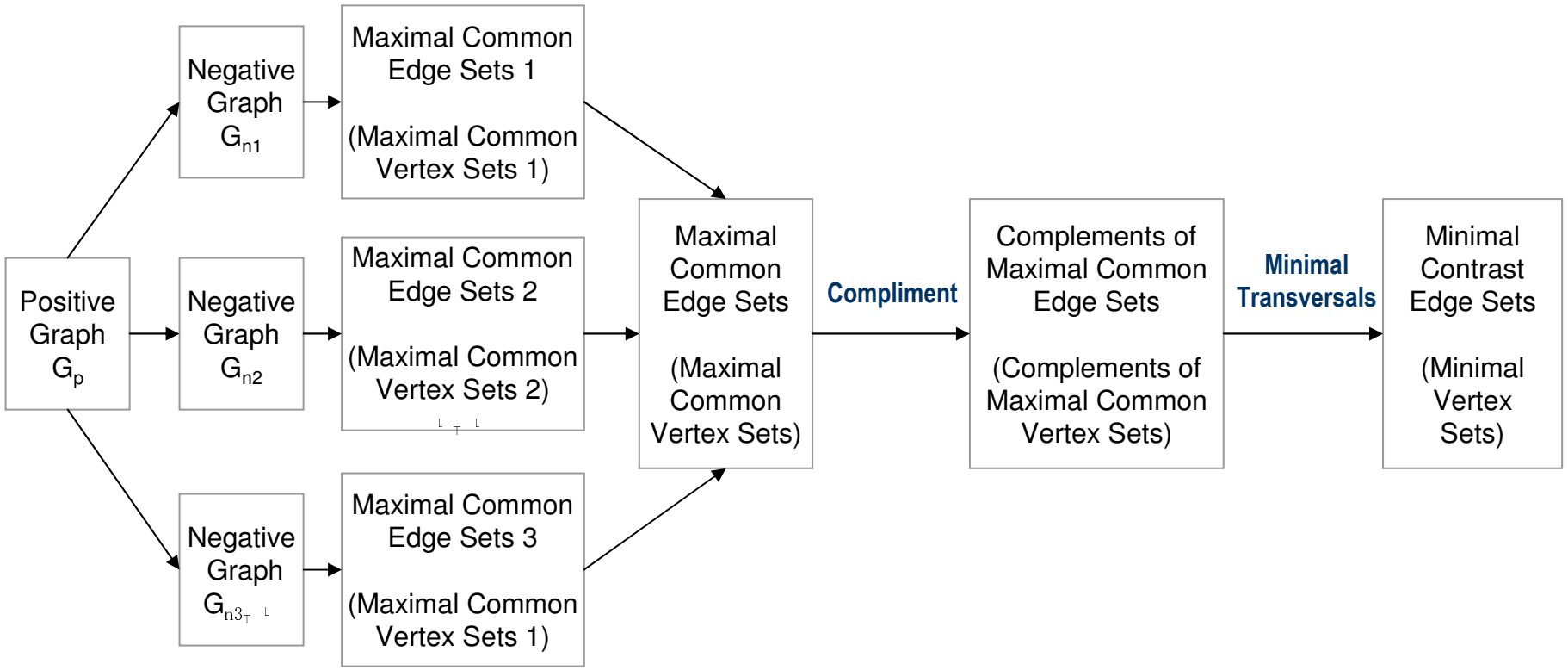
- From the example, we can see that for the 1-1 case, contrast graphs are of two types
  - Those with only vertices (*a vertex set*)
  - Those without isolated vertices (*edge sets*)
- Can prove that for 1-1 case, the *minimal* contrast subgraphs are the union of

Min. Vertex Sets + Minimal Edge Sets

Mining contrast subgraphs

- Main idea
  - Find the maximal common edge sets
    - These may be disconnected
  - Apply a minimal hypergraph transversal operation to derive the minimal contrast edge sets from the maximal common edge sets
  - Must compute minimal contrast vertex sets separately and then minimal union with the minimal contrast edge sets

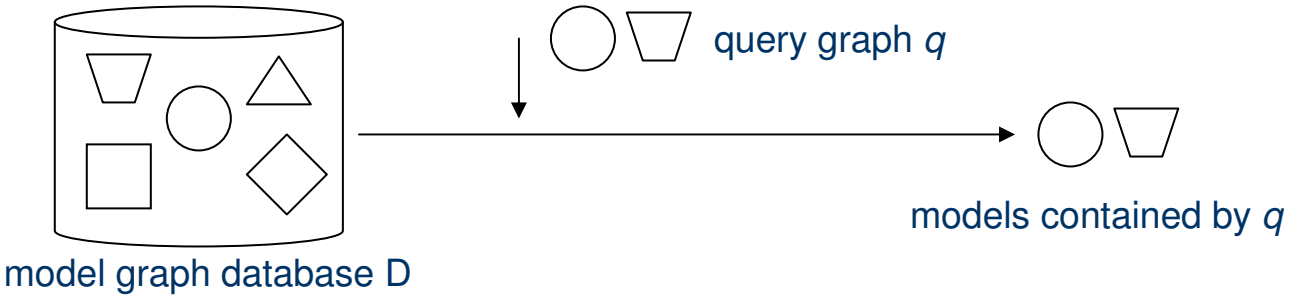
Contrast graph mining workflow



Using discriminative graphs for containment search and indexing

[Chen et al 07]

- Given a graph database and a query *q*. Find all graphs in the database contained in *q*.



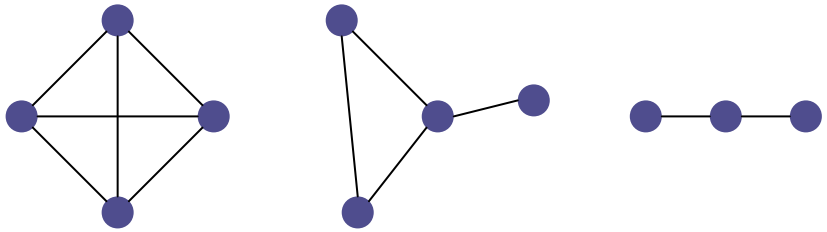
- Applications
  - Querying image databases represented as attributed relational graphs. Efficiently find all objects from the database contained in a given scene (query).

Discriminative graphs for indexing

Cont.

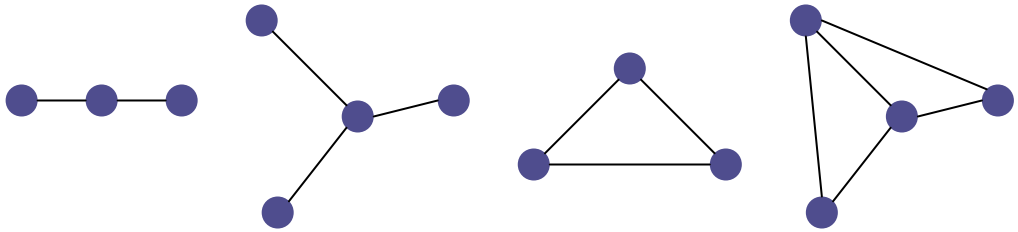
- Main idea:
  - Given a query graph q and a database graph g
    - If a feature *f* is not contained in *q* and *f* is contained in *g*, then *g* is not contained in *q*
- Also exploit similarity between graphs.
  - If *f* is a common substructure between *g1* and *g2*, then if *f* is not contained in the query, both *g1* and *g2* are not contained in the query

Graph Containment Example [From Chen et al 07]



(g_a) (g_b) (g_c)

A Sample Database



(f_1) (f_2) (f_3) (f_4)

Features

	g_a	g_b	g_c
f_1	1	1	1
f_2	1	1	0
f_3	1	1	0
f_4	1	0	0

Discriminative graphs for indexing

- Aim to select the ``contrast features'' that have the most pruning power (save most isomorphism tests)
- These are features that are contained by many graphs in the database, but are unlikely to be contained by a query graph.
- Generate lots of candidates using a frequent subgraph mining and then filter output graphs for discriminative power

Generating the Index

- After the contrast subgraphs have been found, select a subset of them
  - Use a set cover heuristic to select a set that ``covers'' all the graphs in the database, in the context of a given query *q*
  - For multiple queries, use a maximum coverage with cost approach

Contrasts for trees

- Special case of graphs
  - Lower complexity
  - Lots of activity in the document/XML area, for change detection.
- Notions such as edit distance more typical for this context

Contrasts of models

- Models can be clusterings, decision trees, …
- Why is contrasting useful here ?
  - Contrast/compare a user generated model against a known reference model, to evaluate accuracy/degree of difference.
  - May wish to compare degree of difference between one algorithm using varying parameters
  - Eliminate redundancy among models by choosing dissimilar representatives

Contrasts of models Cont.

- Isn’t this just a dissimilarity measure ? Like Euclidean distance ?
  - Similar, but operating on more complex objects, not just vectors
- Difficulties are
  - For rule based classifiers, can’t just report on number of different rules

Clustering comparison

- Popular clustering comparison measures
  - Rand index and Jaccard index
    - Measure the proportion of point pairs on which the two clusterings agree
  - Mutual information
    - How much information one clustering gives about the other
  - Clustering error
    - Classification error metric

Clustering Comparison Measures

- Nearly all techniques use a ‘Confusion Matrix’ of two clusterings. Example : Let C = {c_1, c_2, c_3) and C’ = {c'_1, c'_2, c'_3}

m	c_1	c_2	c_3
c'_1	5	14	1
c'_2	10	2	8
c'_3	8	7	5

$$m_{ij} = | c_i \cap c'_j |$$

Pair counting

- Considers the number of points on which two clusterings agree or disagree. Each pair falls into one of four categories
  - N_{11} – contains the pairs of points which are in the same cluster both in C and C'
  - N_{00} – contains the pairs of points which are not in the same cluster in both C and C'
  - N_{10} – contains the pairs of points which are in the same cluster in C but not in C'
  - N_{01} – contains the pairs of points which are in the same cluster in C' but not in C
  - N - total number of pairs of points

Pair Counting

- Two popular indexes - Rand and Jaccard
  - Rand(C,C')= $\frac{N_{11} + N_{00}}{N}$
  - Jaccard(C,C')= $\frac{N_{11}}{N_{11} + N_{01} + N_{10}}$

Clustering Error Metric (Classification Error Metric)

Is an injective mapping of C={1,…,K} into C'={1…,K'}. Need to find *maximum* intersection for all possible mappings.

m	c_1	c_2	c_3
c'_1	5	14	1
c'_2	10	2	8
c'_3	8	7	5

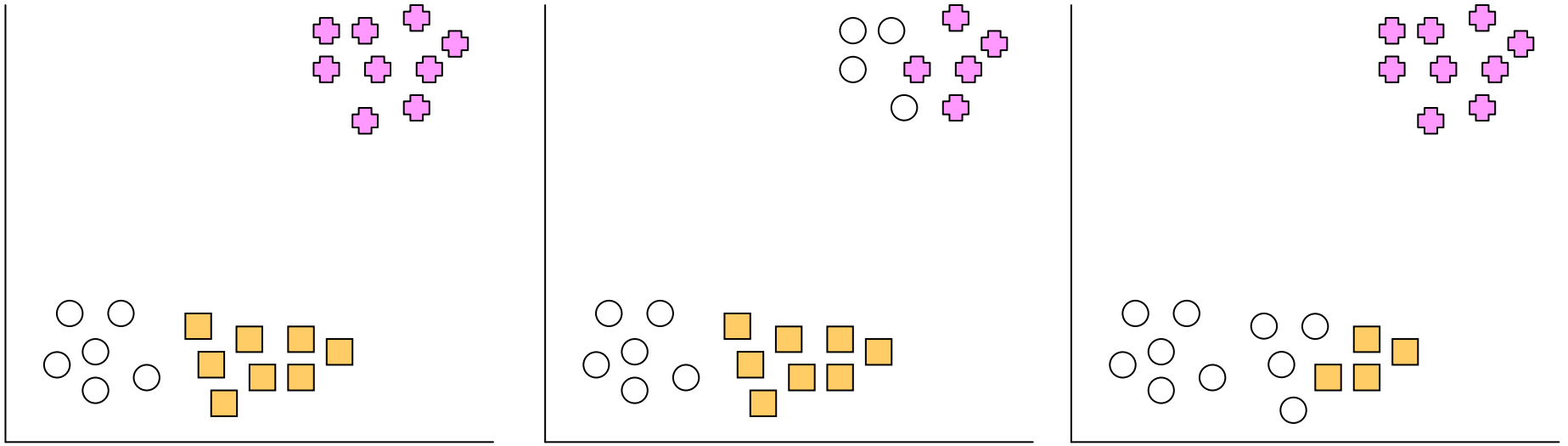
Best match is
$\{c_2, c'_1\}$, $\{c_1, c'_2\}$,
$\{c_3, c'_3\}$

Clustering error=
(14+10+5)/60=0.483

Clustering Comparison Difficulties

Reference

Which most similar to clustering (a)?
Rand(a,b)=Rand(a,c)
Jaccard(a,b)=Jaccard(a,c) !



(a) (b) (c)

Comparing datasets via induced models

- Give two datasets, we may compare their difference, by considering the difference or deviation between the models that can be induced from them
- Models here can refer to decision trees, frequent itemsets, emerging patterns, etc
- May also compare an old model to a new dataset
  - How much does it misrepresent ?

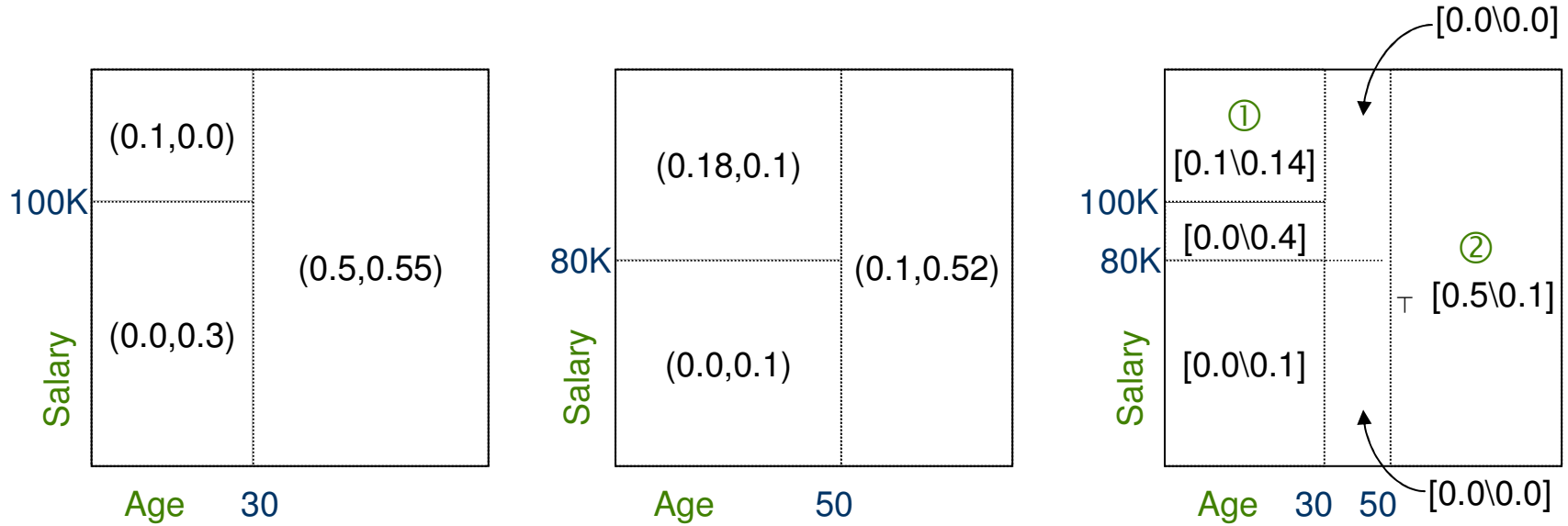
The *FOCUS* Framework [Ganti et al 99]

- Develops a single measure for quantifying the difference between the interesting characteristics in each dataset.
- Key Idea: ``A model has a structural component that identifies interesting regions of the attribute space … each such region summarized by one (or several) measure(s)
- Difference between two classifiers is measured by amount of work needed to change them into some common specialization

Focus Framework Cont.

- For comparing two models, divide the models each into regions and then compare the regions individually
  - For a decision tree, compare leaf nodes of each model
  - Aggregate the pairwise differences between each of the regions

Decision tree example [Taken from Ganti et 99]



T1:D1 **T2:D2** **T3: GCR of T1 and T2**

Difference(D1,D2)=|0.5-0.1|+|0.4-0.3|+|0.1-0.5|+|0.25-0.05|+|0.05-0.2| =1.125

Correspondence Tracing of Changes [Wang et al 03]

- Correspondence tracing aims to make change between the two models understandable by explicitly describing changes and then ranking them

Correspondence Tracing Example

[Taken from Wang et al 03]

- Consider old and new rule based classifiers
- Old
  - O1: If A4=1 then C3 [0,2,7,9,13,15,17]
  - O2: If A3=1 and A4=2 then C2 [1,4,6,10,12,16]
  - O3: If A3=2 and A4=2 then C1 [3,5,8,11,14]
- New
  - N1: If A3=1 and A4=1 then C4 [0,9,15]
  - N2: If A3=1 and A4=2 then C2 [1,4,6,10,12,16]
  - N3: If A3=2 and A4=1 then C2 [2,7,13,17]
  - N4: If A3=2 and A4=2 then C1 [3,5,8,11,14]

Correspondence Example cont.

- Rules N1 and N3 classify the examples that were classified by rule O1. So the changes for the sub population covered by O1 can be described as

  <O1,N1> and <O1,N3>

Changes <O2,N2> and <O3,N4> are trivial because the old and new rules are identical.

Rule Accuracy Increase.

- The quantitative change Q of <O,N> is the estimated accuracy increase (+ or -) due to the change from O to N.
- Changes are ranked according to quantitative change Q and then presented to the user

Common themes for contrast mining

- Different representations
  - Minimality is the most common
  - Support/ratio constraints most popular, though not necessarily the best
  - Conjunctions most popular for relational case
- Large output size

Recommendations to Practitioners

- Some important points are
  - Contrast patterns can capture distinguishing patterns between classes
  - Contrast patterns can be used to build high quality classifiers
  - Contrast patterns can capture useful patterns for detecting/treating diseases

Open Problems in Contrast Data Mining

- How to meaningfully assess quality of contrasts, especially for non-relational data.
  - Little work on contrasts for mixed forms of data
- How to explain the semantics of contrasts
- Highly expressive contrasts (first order ..)
- Develop new ways to build contrast based classifiers
- Discovery of contrasts in massive datasets.
  - Efficiently mine contrasts when there are thousand of attributes
  - Efficient mining of top-k contrast patterns
  - Are there meaningful approximations (e.g. sampling) ?

Summary

- We have given a wide survey of contrast mining. It should now be clearer
  - Why contrast data mining is important and when it can be used
  - How it can be used for very powerful classifiers
  - What algorithms can be used for contrast data mining

Acknowledgements

- We are grateful to the following people for their helpful comments or materials for this tutorial
  - Eric Bae
  - Jiawei Han
  - Xiaonan Ji
  - Jinyan Li
  - Elsa Loekito
  - Limsoon Wong
  - …..

Bibliography

This bibliography contains three sections but not necessarily exhaustive:

- Mining of Emerging Patterns, Change Patterns, Contrast/Difference Patterns
- Emerging/Contrast Pattern Based Classification
- Other Applications of Emerging Patterns

Bibliography (Mining of Emerging Patterns, Change Patterns, Contrast/Difference Patterns)

- Arunasalam, Bavani and Chawla, Sanjay and Sun, Pei. Striking Two Birds with One Stone: Simultaneous Mining of Positive and Negative Spatial Patterns. In Proceedings of the Fifth SIAM International Conference on Data Mining, April 21-23, pp, Newport Beach, CA, USA, SIAM 2005
- Bavani Arunasalam, Sanjay Chawla: CCCS: a top-down associative classifier for imbalanced class distribution. KDD 2006: 517-522
- Eric Bae, James Bailey, Guozhu Dong: Clustering Similarity Comparison Using Density Profiles. Australian Conference on Artificial Intelligence 2006: 342-351
- James Bailey, Thomas Manoukian, Kotagiri Ramamohanarao: Fast Algorithms for Mining Emerging Patterns. PKDD 2002: 39-50.
- J. Bailey and T. Manoukian and K. Ramamohanarao: A Fast Algorithm for Computing Hypergraph Transversals and its Application in Mining Emerging Patterns. Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM). Pages 485-488. Florida, USA, November 2003.
- Stephen D. Bay, Michael J. Pazzani: Detecting Change in Categorical Data: Mining Contrast Sets. KDD 1999: 302-306.
- Stephen D. Bay, Michael J. Pazzani: Detecting Group Differences: Mining Contrast Sets. Data Min. Knowl. Discov. 5(3): 213-246 (2001)
- Cristian Bucila, Johannes Gehrke, Daniel Kifer, Walker M. White: DualMiner: A Dual-Pruning Algorithm for Itemsets with Constraints. Data Min. Knowl. Discov. 7(3): 241-272 (2003)
- Yandong Cai, Nick Cercone, Jiawei Han: An Attribute-Oriented Approach for Learning Classification Rules from Relational Databases. ICDE 1990: 281-288
- Sarah Chan, Ben Kao, Chi Lap Yip, Michael Tang: Mining Emerging Substrings. DASFAA 2003.
- Yixin Chen, Guozhu Dong, Jiawei Han, Jian Pei, Benjamin W. Wah, Jianyong Wang: Online Analytical Processing Stream Data: Is It Feasible? DMKD 2002
- Chen Chen, Xifeng Yan, Philip S. Yu, Jiawei Han, Dong-Qing Zhang, Xiaohui Gu: Towards Graph Containment Search and Indexing. VLDB 2007: 926-937

Bibliography (Mining of Emerging Patterns, Change Patterns, Contrast/Difference Patterns)

- Graham Cormode, S. Muthukrishnan: What's new: finding significant differences in network data streams. IEEE/ACM Trans. Netw. 13(6): 1219-1232 (2005)
- Luc De Raedt, Albrecht Zimmermann: Constraint-Based Pattern Set Mining. SDM 2007
- Luc De Raedt: Towards Query Evaluation in Inductive Databases Using Version Spaces. Database Support for Data Mining Applications 2004: 117-134
- Luc De Raedt, Stefan Kramer: The Levelwise Version Space Algorithm and its Application to Molecular Fragment Finding. IJCAI 2001: 853-862
- Guozhu Dong, Jinyan Li: Efficient Mining of Emerging Patterns: Discovering Trends and Differences. KDD 1999: 43-52.
- Guozhu Dong, Jinyan Li: Mining border descriptions of emerging patterns from dataset pairs. Knowl. Inf. Syst. 8(2): 178-202 (2005).
- Dong, G. and Han, J. and Lakshmanan, L.V.S. and Pei, J. and Wang, H. and Yu, P.S. Online Mining of Changes from Data Streams: Research Problems and Preliminary Results, Proceedings of the 2003 ACM SIGMOD Workshop on Management and Processing of Data Streams, 2003
- Guozhu Dong, Jiawei Han, Joyce M. W. Lam, Jian Pei, Ke Wang, Wei Zou: Mining Constrained Gradients in Large Databases. IEEE Trans. Knowl. Data Eng. 16(8): 922-938 (2004).
- Johannes Fischer, Volker Heun, Stefan Kramer: Optimal String Mining Under Frequency Constraints. PKDD 2006: 139-150
- Venkatesh Ganti, Johannes Gehrke, Raghu Ramakrishnan: A Framework for Measuring Changes in Data Characteristics. PODS 1999: 126-137
- Venkatesh Ganti, Johannes Gehrke, Raghu Ramakrishnan, Wei-Yin Loh: A Framework for Measuring Differences in Data Characteristics. J. Comput. Syst. Sci. 64(3): 542-578 (2002)
- Garriga, G.C. and Kralj, P. and Lavrac, N. Closed Sets for Labeled Data?, PKDD, 2006
- Hilderman, R.J. and Peckham, T. A Statistically Sound Alternative Approach to Mining Contrast Sets, Proceedings of the 4th Australasian Data Mining Conference, 2005 (pp157-172)

Bibliography (Mining of Emerging Patterns, Change Patterns, Contrast/Difference Patterns)

- Hui-jing Huang, Yongsong Qin, Xiaofeng Zhu, Jilian Zhang, and Shichao Zhang. Difference Detection Between Two Contrast Sets. Proceedings of the 8th International Conference on Data Warehousing and Knowledge Discovery (DaWak), 2006.
- Imberman, S.P. and Tansel, A.U. and Pacuit, E. An Efficient Method For Finding Emerging Frequent Itemsets, 3rd International Workshop on Mining Temporal and Sequential Data, pp112--121, 2004
- Tomasz Imielinski, Leonid Khachiyan, Amin Abdulghani: Cubegrades: Generalizing Association Rules. Data Min. Knowl. Discov. 6(3): 219-257 (2002)
- Inakoshi, H. and Ando, T. and Sato, A. and Okamoto, S. Discovery of emerging patterns from nearest neighbors, International Conference on Machine Learning and Cybernetics, 2002.
- Xiaonan Ji, James Bailey, Guozhu Dong: Mining Minimal Distinguishing Subsequence Patterns with Gap Constraints. ICDM 2005: 194-201.
- Xiaonan Ji, James Bailey, Guozhu Dong: Mining Minimal Distinguishing Subsequence Patterns with Gap Constraints. Knowl. Inf. Syst. 11(3): 259--286 (2007).
- Daniel Kifer, Shai Ben-David, Johannes Gehrke: Detecting Change in Data Streams. VLDB 2004: 180-191
- P Kralj, N Lavrac, D Gamberger, A Krstacic. Contrast Set Mining for Distinguishing Between Similar Diseases. LNCS Volume 4594, 2007.
- Sau Dan Lee, Luc De Raedt: An Efficient Algorithm for Mining String Databases Under Constraints. KDID 2004: 108-129
- Haiquan Li, Jinyan Li, Limsoon Wong, Mengling Feng, Yap-Peng Tan: Relative risk and odds ratio: a data mining perspective. PODS 2005: 368-377
- Jinyan Li, Guimei Liu and Limsoon Wong. Mining Statistically Important Equivalence Classes and delta-Discriminative Emerging Patterns. KDD 2007.
- Jinyan Li, Thomas Manoukian, Guozhu Dong, Kotagiri Ramamohanarao: Incremental Maintenance on the Border of the Space of Emerging Patterns. Data Min. Knowl. Discov. 9(1): 89-116 (2004).

Bibliography (Mining of Emerging Patterns, Change Patterns, Contrast/Difference Patterns)

- Jinyan Li and Qiang Yang. Strong Compound-Risk Factors: Efficient Discovery through Emerging Patterns and Contrast Sets. IEEE Transactions on Information Technology in Biomedicine. To appear.
- Lin, J. and Keogh, E. Group SAX: Extending the Notion of Contrast Sets to Time Series and Multimedia Data. Proceedings of the 10th european conference on principles and practice of knowledge discovery in databases. Berlin, Germany, September, 2006.
- Bing Liu, Ke Wang, Lai-Fun Mun, Xin-Zhi Qi: Using Decision Tree Induction for Discovering Holes in Data. PRICAI 1998: 182-193
- Bing Liu, Liang-Ping Ku, Wynne Hsu: Discovering Interesting Holes in Data. IJCAI (2) 1997: 930-935
- Bing Liu, Wynne Hsu, Yiming Ma: Discovering the set of fundamental rule changes. KDD 2001: 335-340.
- Elsa Loekito, James Bailey: Fast Mining of High Dimensional Expressive Contrast Patterns Using Zero-suppressed Binary Decision Diagrams. KDD 2006: 307-316.
- Yu Meng, Margaret H. Dunham: Efficient Mining of Emerging Events in a Dynamic Spatiotemporal Environment. PAKDD 2006: 750-754
- Tom M. Mitchell: Version Spaces: A Candidate Elimination Approach to Rule Learning. IJCAI 1977: 305-310
- Amit Satsangi, Osmar R. Zaiane, Contrasting the Contrast Sets: An Alternative Approach, Eleventh International Database Engineering and Applications Symposium (IDEAS 2007), Banff, Canada, September 6-8, 2007
- Michele Sebag: Delaying the Choice of Bias: A Disjunctive Version Space Approach. ICML 1996: 444-452
- Michele Sebag: Using Constraints to Building Version Spaces. ECML 1994: 257-271
- Arnaud Soulet, Bruno Crémilleux, François Rioult: Condensed Representation of EPs and Patterns Quantified by Frequency-Based Measures. KDID 2004: 173-190
- Pawel Terlecki, Krzysztof Walczak: On the relation between rough set reducts and jumping emerging patterns. Inf. Sci. 177(1): 74-83 (2007).

Bibliography (Mining of Emerging Patterns, Change Patterns, Contrast/Difference Patterns)

- Roger Ming Hieng Ting, James Bailey: Mining Minimal Contrast Subgraph Patterns. SDM 2006.
- V. S. Tseng, C. J. Chu, and Tyne Liang, An Efficient Method for Mining Temporal Emerging Itemsets From Data Streams, International Computer Symposium, Workshop on Software Engineering, Databases and Knowledge Discovery, 2006
- J. Vreeken, M. van Leeuwen, A. Siebes: Characterising the Difference. KDD 2007.
- Haixun Wang, Wei Fan, Philip S. Yu, Jiawei Han: Mining concept-drifting data streams using ensemble classifiers. KDD 2003: 226-235
- Peng Wang, Haixun Wang, Xiaochen Wu, Wei Wang, Baile Shi: On Reducing Classifier Granularity in Mining Concept-Drifting Data Streams. ICDM 2005: 474-481
- Lusheng Wang, Hao Zhao, Guozhu Dong, Jianping Li: On the complexity of finding emerging patterns. Theor. Comput. Sci. 335(1): 15-27 (2005).
- Ke Wang, Senqiang Zhou, Ada Wai-Chee Fu, Jeffrey Xu Yu: Mining Changes of Classification by Correspondence Tracing. SDM 2003.
- Geoffrey I. Webb: Discovering Significant Patterns. Machine Learning 68(1): 1-33 (2007)
- Geoffrey I. Webb, Songmao Zhang: K-Optimal Rule Discovery. Data Min. Knowl. Discov. 10(1): 39-79 (2005)
- Geoffrey I. Webb, Shane M. Butler, Douglas A. Newlands: On detecting differences between groups. KDD 2003: 256-265.
- Xiuzhen Zhang, Guozhu Dong, Kotagiri Ramamohanarao: Exploring constraints to efficiently mine emerging patterns from large high-dimensional datasets. KDD 2000: 310-314.
- Lizhuang Zhao, Mohammed J. Zaki, Naren Ramakrishnan: BLOSOM: a framework for mining arbitrary boolean expressions. KDD 2006: 827-832

Bibliography (Emerging/Contrast Pattern Based Classification)

- Hamad Alhammady, Kotagiri Ramamohanarao: The Application of Emerging Patterns for Improving the Quality of Rare-Class Classification. PAKDD 2004: 207-211
- Hamad Alhammady, Kotagiri Ramamohanarao: Using Emerging Patterns and Decision Trees in Rare-Class Classification. ICDM 2004: 315-318
- Hamad Alhammady, Kotagiri Ramamohanarao: Expanding the Training Data Space Using Emerging Patterns and Genetic Methods. SDM 2005
- Hamad Alhammady, Kotagiri Ramamohanarao: Using Emerging Patterns to Construct Weighted Decision Trees. IEEE Trans. Knowl. Data Eng.
- 18(7): 865-876 (2006).
- Hamad Alhammady, Kotagiri Ramamohanarao: Mining Emerging Patterns and Classification in Data Streams. Web Intelligence 2005: 272-275
- James Bailey, Thomas Manoukian, Kotagiri Ramamohanarao: Classification Using Constrained Emerging Patterns. WAIM 2003: 226-237
- Guozhu Dong, Xiuzhen Zhang, Limsoon Wong, Jinyan Li: CAEP: Classification by Aggregating Emerging Patterns. Discovery Science
- 1999: 30-42.
- Hongjian Fan, Kotagiri Ramamohanarao: An Efficient Single-Scan Algorithm for Mining Essential Jumping Emerging Patterns for Classification. PAKDD 2002: 456-462
- Hongjian Fan, Kotagiri Ramamohanarao: Efficiently Mining Interesting Emerging Patterns. WAIM 2003: 189-201
- Hongjian Fan, Kotagiri Ramamohanarao: Noise Tolerant Classification by Chi Emerging Patterns. PAKDD 2004: 201-206
- Hongjian Fan, Ming Fan, Kotagiri Ramamohanarao, Mengxu Liu: Further Improving Emerging Pattern Based Classifiers Via Bagging. PAKDD 2006: 91-96
- Hongjian Fan, Kotagiri Ramamohanarao: A weighting scheme based on emerging patterns for weighted support vector machines. GrC 2005: 435-440

Bibliography (Emerging/Contrast Pattern Based Classification)

- Hongjian Fan, Kotagiri Ramamohanarao: Fast Discovery and the Generalization of Strong Jumping Emerging Patterns for Building Compact and Accurate Classifiers. IEEE Trans. Knowl. Data Eng. 18(6): 721-737 (2006)
- Jinyan Li, Guozhu Dong, Kotagiri Ramamohanarao: Instance-Based Classification by Emerging Patterns. PKDD 2000: 191-200
- Jinyan Li, Guozhu Dong, Kotagiri Ramamohanarao: Making Use of the Most Expressive Jumping Emerging Patterns for Classification. PAKDD 2000: 220-232
- Jinyan Li, Guozhu Dong, Kotagiri Ramamohanarao: Making Use of the Most Expressive Jumping Emerging Patterns for Classification. Knowl. Inf. Syst. 3(2): 131-145 (2001)
- Jinyan Li, Kotagiri Ramamohanarao, Guozhu Dong: Emerging Patterns and Classification. ASIAN 2000: 15-32
- Jinyan Li, Guozhu Dong, Kotagiri Ramamohanarao, Limsoon Wong: DeEPs: A New Instance-Based Lazy Discovery and Classification System. Machine Learning 54(2): 99-124 (2004).
- Wenmin Li, Jiawei Han, Jian Pei: CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules. ICDM 2001: 369-376
- Jinyan Li, Kotagiri Ramamohanarao, Guozhu Dong: Combining the Strength of Pattern Frequency and Distance for Classification. PAKDD 2001: 455-466
- Bing Liu, Wynne Hsu, Yiming Ma: Integrating Classification and Association Rule Mining. KDD 1998: 80-86
- Kotagiri Ramamohanarao, James Bailey: Discovery of Emerging Patterns and Their Use in Classification. Australian Conference on Artificial Intelligence 2003: 1-12
- Ramamohanarao, K. and Bailey, J. and Fan, H. Efficient Mining of Contrast Patterns and Their Applications to Classification, Third International Conference on Intelligent Sensing and Information Processing, 2005 (39--47).
- Ramamohanarao, K. and Fan, H. Patterns Based Classifiers, World Wide Web 2007: 10(71--83).
- Qun Sun, Xiuzhen Zhang, Kotagiri Ramamohanarao: Noise Tolerance of EP-Based Classifiers. Australian Conference on Artificial Intelligence 2003: 796-806

Bibliography (Emerging/Contrast Pattern Based Classification)

- Xiaoxin Yin, Jiawei Han: CPAR: Classification based on Predictive Association Rules. SDM 2003
- Xiuzhen Zhang, Guozhu Dong, Kotagiri Ramamohanarao: Information-Based Classification by Aggregating Emerging Patterns. IDEAL 2000: 48-53
- Xiuzhen Zhang, Guozhu Dong, Kotagiri Ramamohanarao: Building Behaviour Knowledge Space to Make Classification Decision. PAKDD 2001: 488-494
- Zhou Wang, Hongjian Fan, Kotagiri Ramamohanarao: Exploiting Maximal Emerging Patterns for Classification. Australian Conference on Artificial Intelligence 2004: 1062-1068

Bibliography (Other Applications of Emerging Patterns)

- Anne-Laure Boulesteix, Gerhard Tutz, Korbinian Strimmer: A CART-based approach to discover emerging patterns in microarray data. Bioinformatics 19(18): 2465-2472 (2003).
- Lijun Chen, Guozhu Dong: Masquerader Detection Using OCLEP: One Class Classification Using Length Statistics of Emerging Patterns. Proceedings of International Workshop on INformation Processing over Evolving Networks (WINPEN), 2006.
- Guozhu Dong, Kaustubh Deshpande: Efficient Mining of Niches and Set Routines. PAKDD 2001: 234-246
- Grandinetti, W.M. and Chesnevar, C.I. and Falappa, M.A. Enhanced Approximation of the Emerging Pattern Space using an Incremental Approach, Proceedings of VII Workshop of Researchers in Computer Sciences, Argentine, pp263--267, 2005
- Jinyan Li, Huiqing Liu, See-Kiong Ng, Limsoon Wong. Discovery of Significant Rules for Classifying Cancer Diagnosis Data . Bioinformatics. 19 (suppl. 2): ii93-ii102. (This paper was also presented in the 2003 European Conference on Computational Biology, Paris, France, September 26-30.)
- Jinyan Li, Huiqing Liu, James R. Downing, Allen Eng-Juh Yeoh, Limsoon Wong. Simple Rules Underlying Gene Expression Profiles of More than Six Subtypes of Acute Lymphoblastic Leukemia (ALL) Patients. Bioinformatics. 19:71--78, 2003.
- Jinyan Li, Limsoon Wong: Emerging patterns and gene expression data. Genome Informatics, 2001:12(3--13).
- Jinyan Li, Limsoon Wong: Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns. Bioinformatics 18(5): 725-734 (2002)
- Jinyan Li, Limsoon Wong. Geography of Differences Between Two Classes of Data. Proceedings 6th European Conference on Principles of Data Mining and Knowledge Discovery, pages 325--337, Helsinki, Finland, August 2002.
- Jinyan Li and Limsoon Wong. Structural Geography of the space of emerging patterns. Intelligent Data Analysis (IDA): An International Journal, Volume 9, pages 567-588, November 2005.
- Jinyan Li, Xiuzhen Zhang, Guozhu Dong, Kotagiri Ramamohanarao, Qun Sun: Efficient Mining of High Confidience Association Rules without Support Thresholds. PKDD 1999: 406-411

Bibliography (Other Applications of Emerging Patterns)

- Shihong Mao, Guozhu Dong: Discovery of Highly Differentiative Gene Groups from Microarray Gene Expression Data Using the Gene Club Approach. J. Bioinformatics and Computational Biology 3(6): 1263-1280 (2005).
- Podraza, R. and Tomaszewski, K. KTDA: Emerging Patterns Based Data Analysis System, Proceedings of XXI Fall Meeting of Polish Information Processing Society, pp213--221, 2005
- Rioult, F. Mining strong emerging patterns in wide SAGE data, Proceedings of the ECML/PKDD Discovery Challenge Workshop, Pisa, Italy, pp127--138, 2004
- Eng-Juh Yeoh, Mary E. Ross, Sheila A. Shurtleff, W. Kent William, Divyen Patel, Rami Mahfouz, Fred G. Behm, Susana C. Raimondi, Mary V. Reilling, Anami Patel, Cheng Cheng, Dario Campana, Dawn Wilkins, Xiaodong Zhou, Jinyan Li, Huiqing Liu, Chin-Hon Pui, William E. Evans, Clayton Naeve, Limsoon Wong, James R. Downing. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. Cancer Cell, 1:133--143, March 2002.
- Yoon, H.S. and Lee, S.H. and Kim, J.H. Application of Emerging Patterns for Multi-source Bio-Data Classification and Analysis, LECTURE NOTES IN COMPUTER SCIENCE Vol 3610, 2005.
- Yu, L.T.H. and Chung, F. and Chan, S.C.F. and Yuen, S.M.C. Using emerging pattern based projected clustering and gene expression data for cancer detection, Proceedings of the second conference on Asia-Pacific bioinformatics, pp75--84, 2004.
- Zhang, X. and Dong, G. and Wong, L. Using CAEP to predict translation initiation sites from genomic DNA sequences, TR2001/22, CSSE, Univ. of Melbourne, 2001.