Declarative data analysis

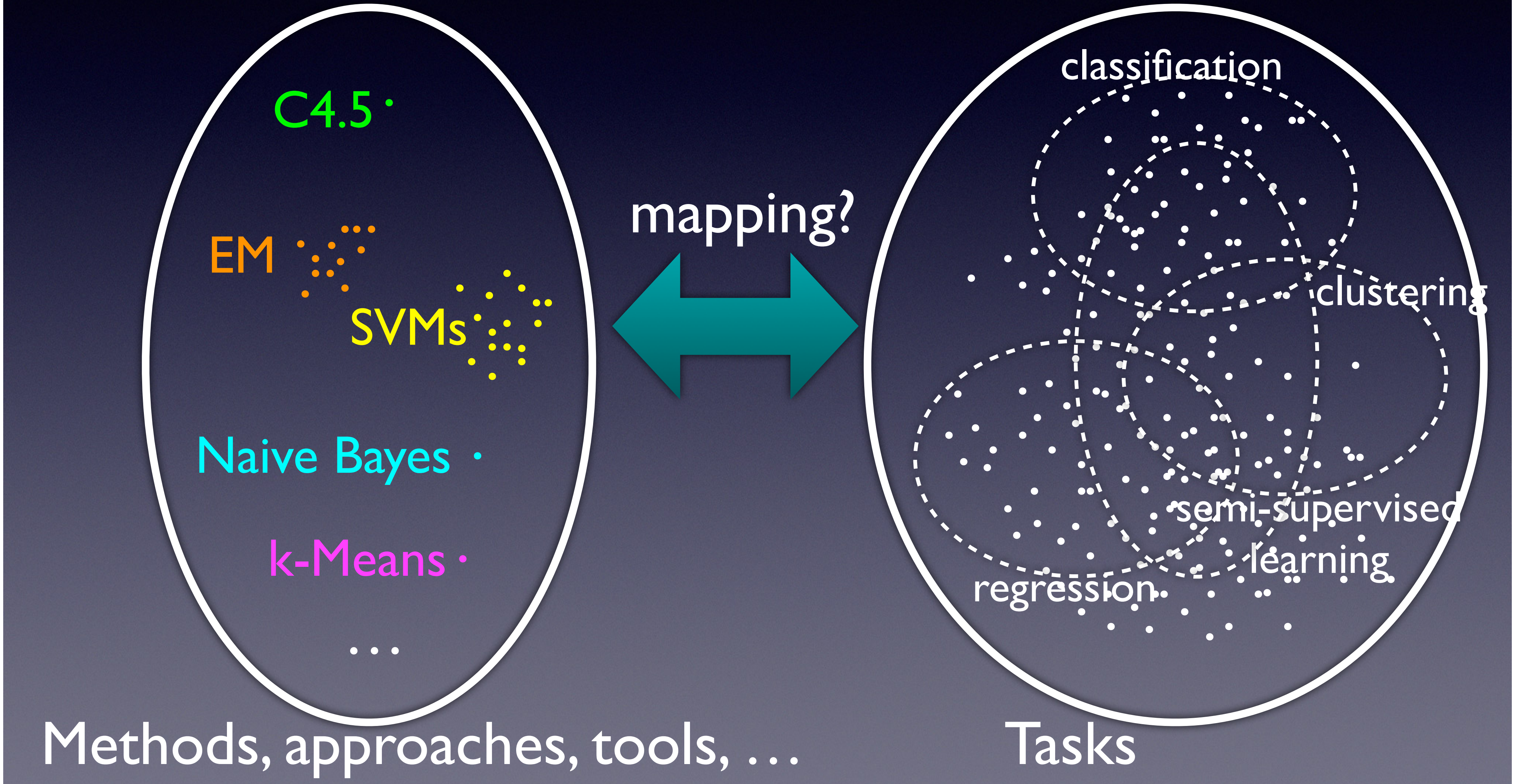
Hendrik Blockeel

KU Leuven

Motivation

- Data analysis:
  - studied in machine learning, data mining, statistics
  - Thousands of tools, methods, algorithms, …
  - Millions of (slightly) different kinds of tasks
- How can a data analyst choose optimally?

Tasks & methods



Variety in tasks

- Categories: Classification, regression, clustering, association rules, reinforcement learning, …
- Within each category:
  - semi-supervised; multi-label; multi-instance; … classification
  - learning from i.i.d. data, trees, sequences, graphs, …
  - transfer learning
  - different target criteria (e.g. for clustering)
  - exploiting background knowledge
  - constraints imposed on solutions
  - …

Variety in tools

- E.g., classification: decision trees, rules, random forests, SVM, Naive Bayes, logistic regression, …
- E.g., clustering: k-means, EM, single linkage, spectral clustering, …
- They all have their own bias
- Which one to use for a particular task? How to set the parameters?

- The best way to address this variety of tasks is to make it possible for the user to describe the task, not the approach
- This is the basic mantra of declarative programming

Compare to SQL

- SQL was a huge leap forward for databases
- Before SQL: program the retrieval procedure yourself
- With SQL: formulate the question in domain terminology; database system determines optimal execution strategy
- SQL made retrieval easier and more efficient
- Data mining is still at the "pre-SQL" stage

Motivation, part 2: *correctness*

- It is easy to use data mining tools incorrectly, or interpret their results incorrectly
- This holds even for basic statistical methods!

Experimental evaluation in machine learning

- Researchers propose new methods, and experimentally evaluate them
- Very often, statistical significance tests are used to show "significant" improvements
- These tests are often used incorrectly
  - See, e.g., Dietterich 1998; Demsar 2006; …
  - The more advanced statistical tests become, the less users understand them, and the higher the risk of mistakes
  - E.g., independence assumptions often violated

Example: cross-validation

- Standard deviations reported in a cross-validation = ?
  - stdev of individual fold estimates?
  - deviation of estimated accuracy from true accuracy?
- Bengio & Grandvalet, 2004: *no unbiased estimate of variance of CV*
- So, whatever these stdevs are, they are not the ones we want
- Hence, P-values, significance tests, … make no sense!

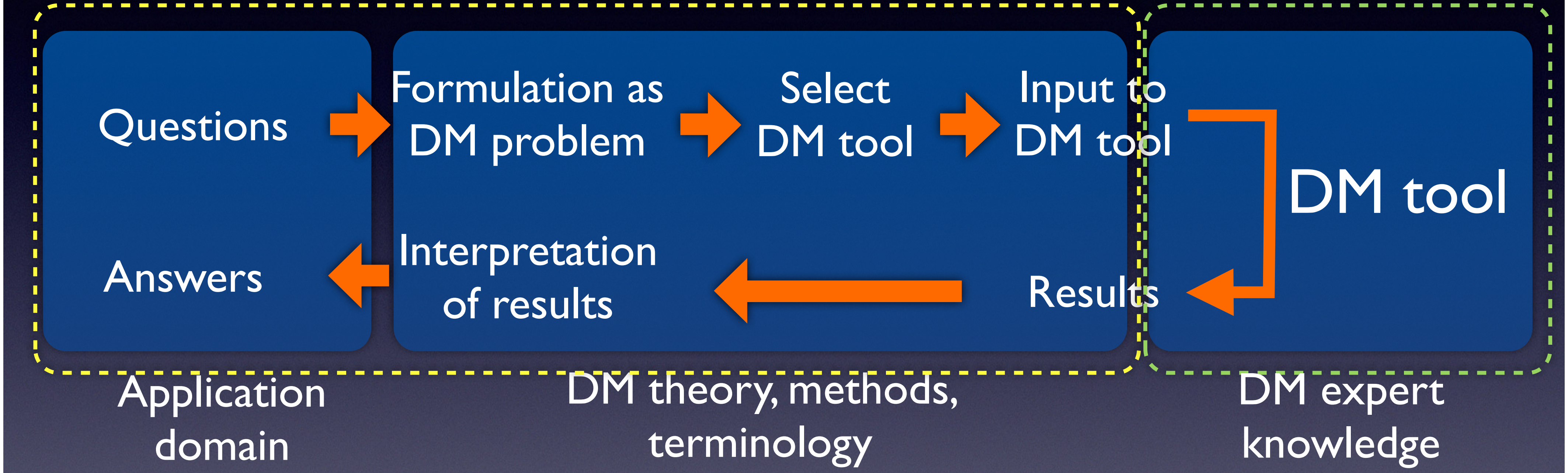
	Method A	**Method B**	**Method C**
Dataset 1	**0.86 (0.02)**	0.83 (0.01)	0.82 (0.01)
Dataset 2	0.85 (0.01)	**0.91 (0.01)**	0.82 (0.03)
...			

← acc (stdev)

Statistics is tricky

- There are many subtle issues in statistics
- Personal opinion: We should not leave computation & interpretation of statistics to the user
- Ideally, build it into the system

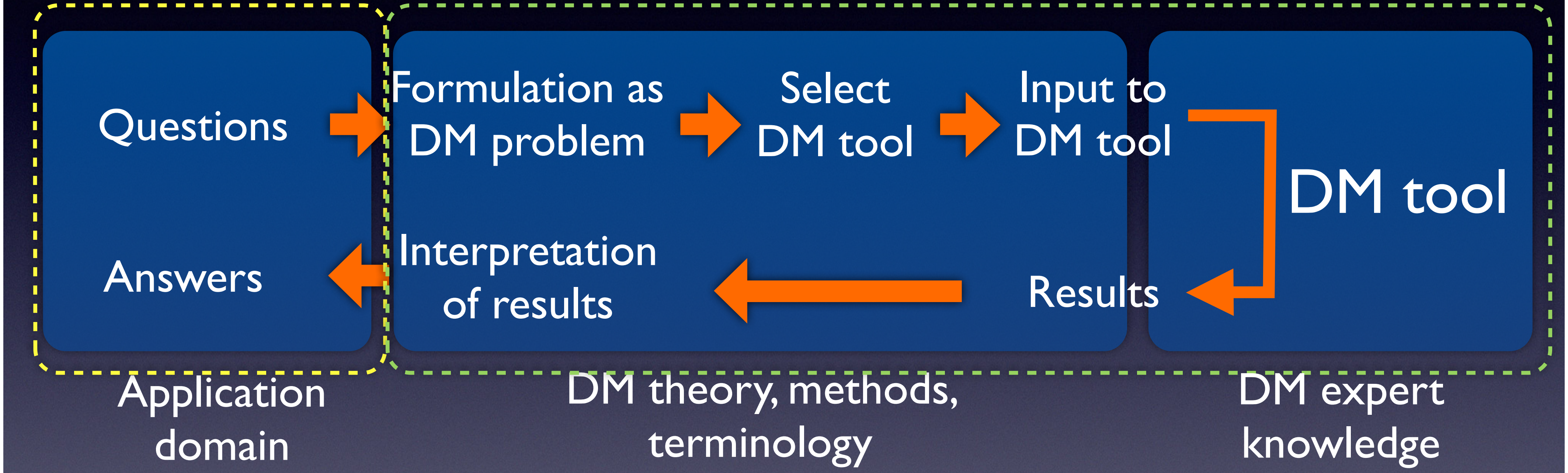
Data analysis as it is now



user's responsibility

system's responsibility

Data analysis as it should be



user's responsibility

system's responsibility

Steps towards declarative data analysis

- Relevant fields:
  - Inductive databases
  - Query languages for data mining
  - Modeling languages for data mining (2010-)
  - Constraint-based data mining
  - Meta-learning
  - Evaluation procedures
  - …

This talk

- This talk: some illustrations of
  - declarative query languages for data mining
  - declarative modeling languages for DM
  - declarative statistical inference
  - subleties in interpretation of DM results

A declarative language for clustering

- An example of integrating “clustering queries” into database languages
- Ongoing work

A. Adam, H. Blockeel, S. Govers, A. Aertsen (2013). *SCCQL: A constraint-based clustering system.* ECMLPKDD 2013 demo. Proc. of ECMLPKDD 2013 part 3: 681-684.

SCCQL

Rel.
DB

Id	Mutant	LengthMean	WidthMean
1	0		
2	0		
3	0		
4	1		
5	1		
6	1		

```
CLUSTER LengthMean, WidthMean
FROM (SELECT c.Id, l.Mutant, AVG(s.Length) AS LengthMean,
                        AVG(s.Width) AS WidthMean
          FROM stateovertime s, cell c, lineage l
          WHERE l.ExperimentId=5 AND c.LineageId = l.Id AND s.CellId = c.Id
          GROUP BY c.id) AS data
WITH SOFT MUST LINK WHERE data.Mutant=0 BY Mutant
```

Subquery defines the data to be clustered.

Cluster according to mean length & width, using as soft constraint that all “Mutant 0” should be in one cluster.

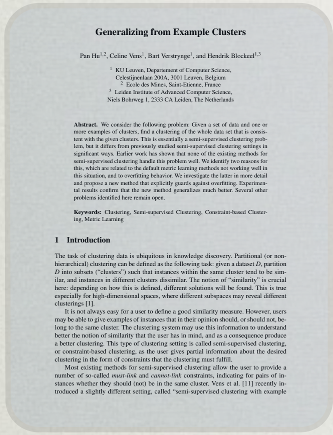
Constraint-based clustering

- Difficult for user: choose clustering algorithm, distance metric, parameters
- Often easier: show pairs of instances that "must/cannot link", or show example clusters
- This motivates *constraint-based clustering*
    - Pairwise constraints: multiple approaches
    - Whole clusters as examples

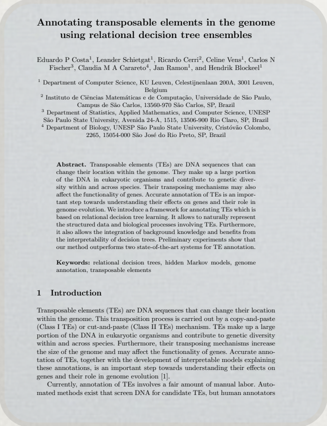
Pan Hu, Celine Vens, Bart Verstrynge, Hendrik Blockeel. *Generalizing from Example Clusters*. Discovery Science 2013: 64-78

(Slide by Celine Vens)

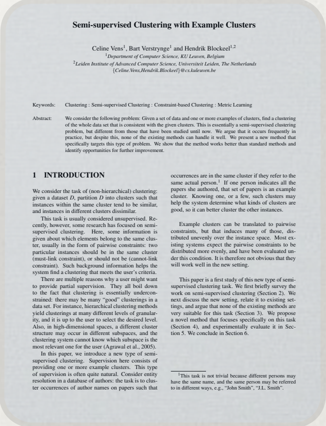
Example application: Entity resolution


Generalizing from Example Clusters

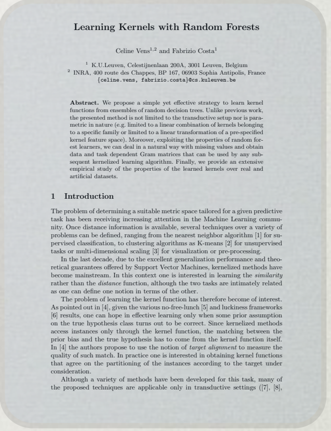
Robert Smith


Annotating transposable elements in the genome using relational decision tree ensembles

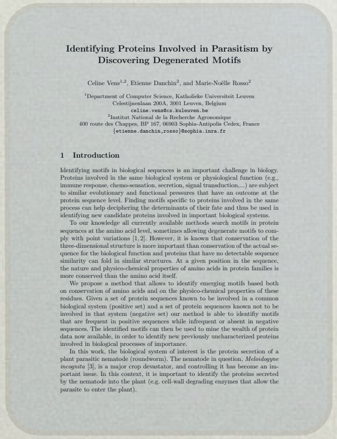
Bob Smith


Semi-supervised Clustering with Example Clusters

R. W. Smith


Learning Kernels with Random Forests

Robert Smith

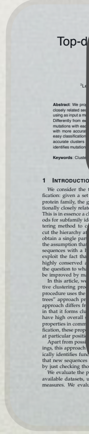

Identifying Proteins Involved in Parasitism by Discovering Degenerated Motifs

Robert L. Smith

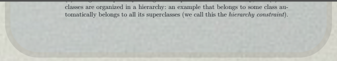
BIOINFORMATICS

Identifying Discriminative Classification Based Motifs in Biological Sequences

B.W. Smith


Top-d

Robert L. W. Smith

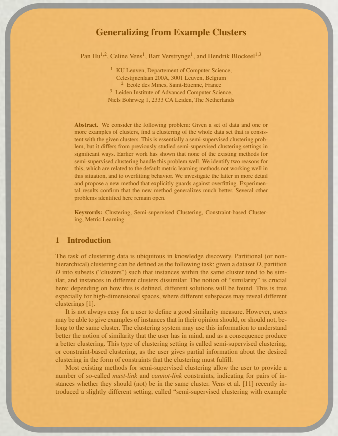

Bob L. Smith

Determining which authors are the same = clustering author occurrences

Q: How to define similarity? Co-authors, keywords, ...? What works best?

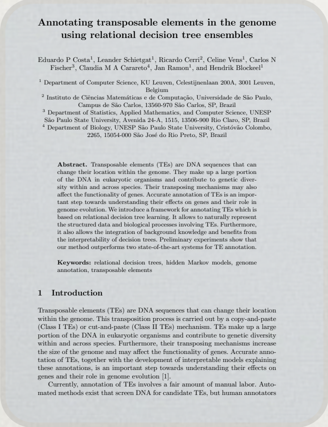
(Slide by Celine Vens)

Entity resolution


Generalizing from Example Clusters

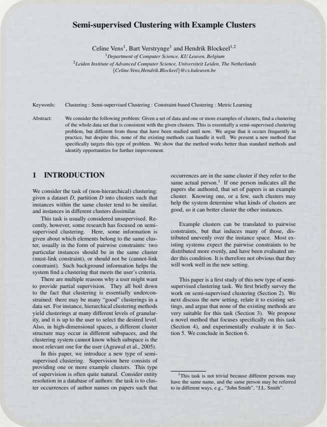
Pan Hu, Celine Vens, Bart Verstrynge, and Hendrik Blockeel

Robert Smith


Annotating transposable elements in the genome using relational decision tree ensembles

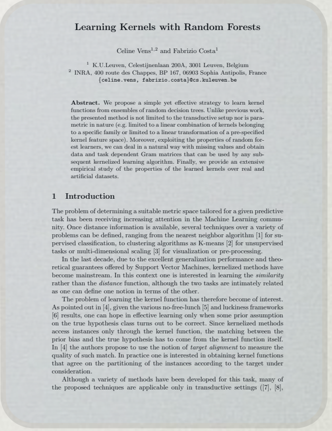
Eduardo P Costa, Leander Schietgat, Ricardo Cerri, Celine Vens, Carlos N Fischer, Claudia M A Carareto, Jan Ramon, and Hendrik Blockeel

Bob Smith


Semi-supervised Clustering with Example Clusters

Celine Vens, Bart Verstrynge and Hendrik Blockeel

R. W. Smith


Learning Kernels with Random Forests

Celine Vens and Fabrizio Costa

Robert Smith

Identifying Proteins Involved in Parasitism by Discovering Degenerated Motifs

Celine Vens, Etienne Danchin, and Marie-Noëlle Rosso

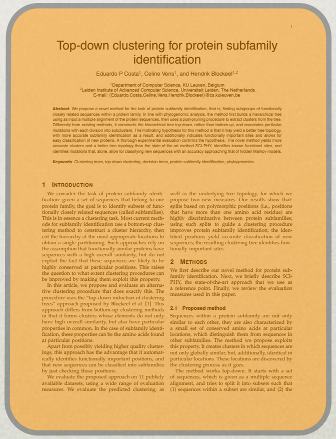
Robert L. Smith

BIOINFORMATICS

Identifying Discriminative Classification Based Motifs in Biological Sequences

Celine Vens, Marie-Noëlle Rosso and Etienne G.J. Danchin

B.W. Smith


Top-down clustering for protein subfamily identification

Eduardo P Costa, Celine Vens, and Hendrik Blockeel

Robert L. W. Smith

Decision Trees for Hierarchical Multi-label Classification

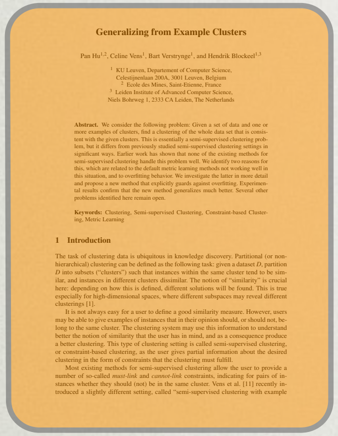
Celine Vens, Jan Struyf, Leander Schietgat, Sašo Džeroski, and Hendrik Blockeel

Bob L. Smith

Example clusters are easy to provide
(complete publication list of one author)

(Slide by Celine Vens)

Entity resolution


Generalizing from Example Clusters

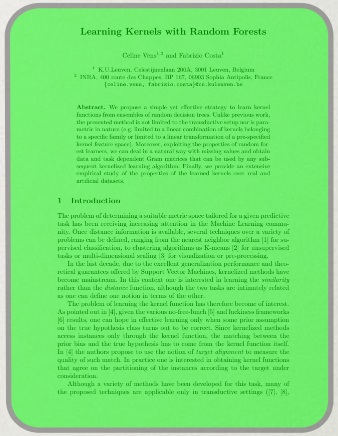
Robert Smith

Annotating transposable elements in the genome using relational decision tree ensembles

Bob Smith

Semi-supervised Clustering with Example Clusters

R. W. Smith


Learning Kernels with Random Forests

Robert Smith

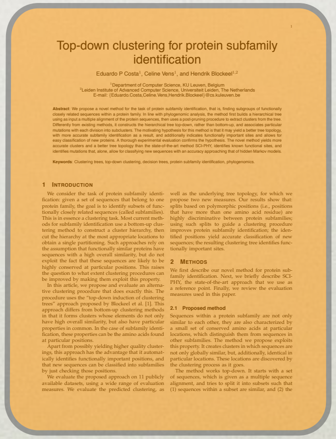
Identifying Proteins Involved in Parasitism by Discovering Degenerated Motifs

Robert L. Smith

BIOINFORMATICS

Identifying Discriminative Classification Based Motifs in Biological Sequences

B.W. Smith


Top-down clustering for protein subfamily identification

Robert L. W. Smith

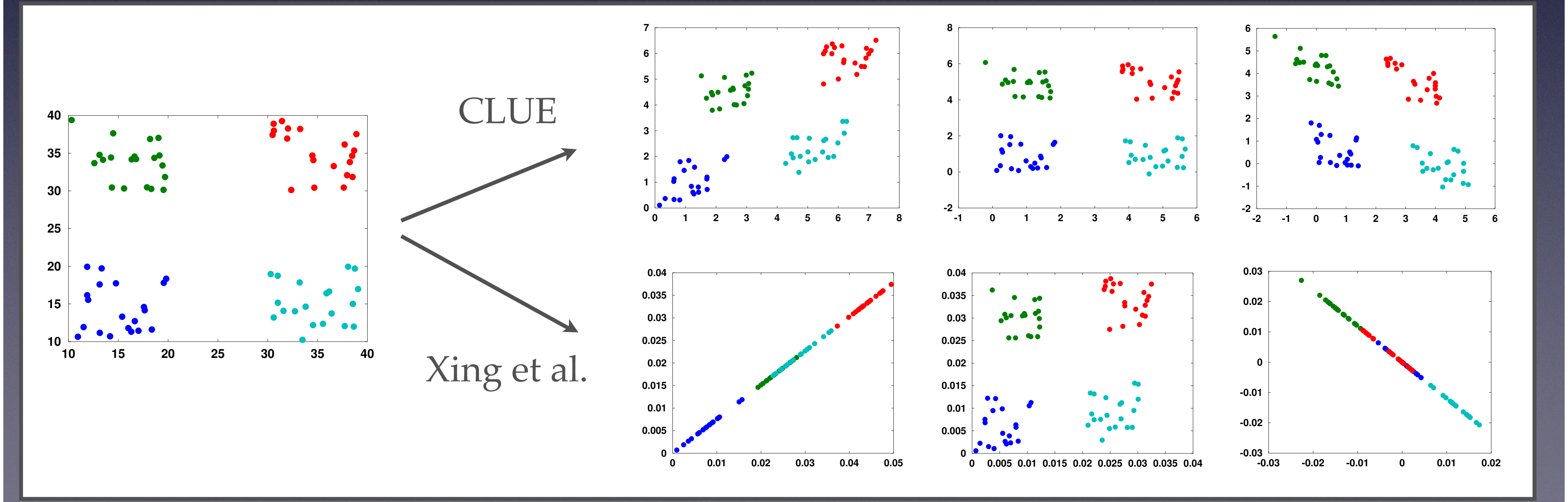
Decision Trees for Hierarchical Multi-label Classification

Bob L. Smith

Rest of the data clustered "in a similar way"

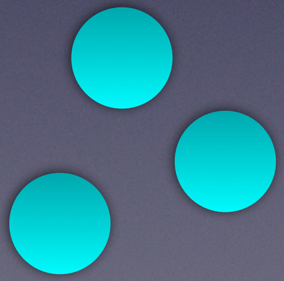
Generalizing from example clusters

- Convert example cluster to pairwise constraints?
- Problem: high concentration of constraints in one part of the space (see Hu et al., DS 2013)

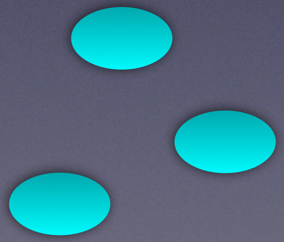


Choosing the clustering approach

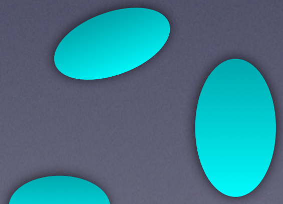
- Most work on constraint-based clustering adapts one approach to incorporate constraints
- But different approaches have very different biases!
- Use constraints to select the most suitable clustering approach?
- Ongoing work (A. Adam et al.)



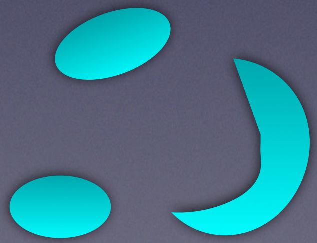
k-means



k-means +
metric learning



EM



Density-based

Modeling languages for data mining

- IDP3: a system for knowledge representation and inference
- Can be used for modeling and solving data mining tasks
- Case study: Analysis of written traditions

M. Bruynooghe, H. Blockeel, B. Bogaerts, B. De Cat, S. De Pooter, J. Jansen, A. Labarre, J. Ramon, M. Denecker, S. Verwer. *Predicate logic as a modeling language: Modeling and solving some machine learning and data mining problems with IDP3*. Theory and Practice of Logic Programming, 2014 (Accepted)

IDP3

- An environment for knowledge-based programming (Wittocx et al. 2008)
- Combines imperative and declarative elements
  - declarative objects: vocabularies, theories, structures
  - (predefined) procedures to
    - create and manipulate these objects
    - perform inference on them (model expansion, ...)
- Includes a state-of-the-art model generator (ref. ASP competition)
- Uses an extension of first order logic (integers, …)

Example:
find frequent itemsets

```
vocabulary FrequentItemsetMiningVoc {
  type Transaction
  type Item
  Freq: int
  Includes(Transaction,Item)
  FrequentItemset(Item)
}
theory FrequentItemsetMiningTh: FrequentItemsetMiningVoc {
  #{t: !i: FrequentItemset(i) => Includes(t,i) } >= Freq.
}
structure Input : FrequentItemsetMiningVoc {
  Freq = 7 // threshold for frequent itemsets
  Transaction = { t1; … ; tn } // n transactions
  Item = {i1 ; … ;  im }         // m items
  Includes = {t1,i2; t1,i7; …} // items of transactions
}
```

FrequentItemset represents a set of items

#{t: FrequentItemset ⊆ t} >= Freq.

Using a vanilla solver for data mining

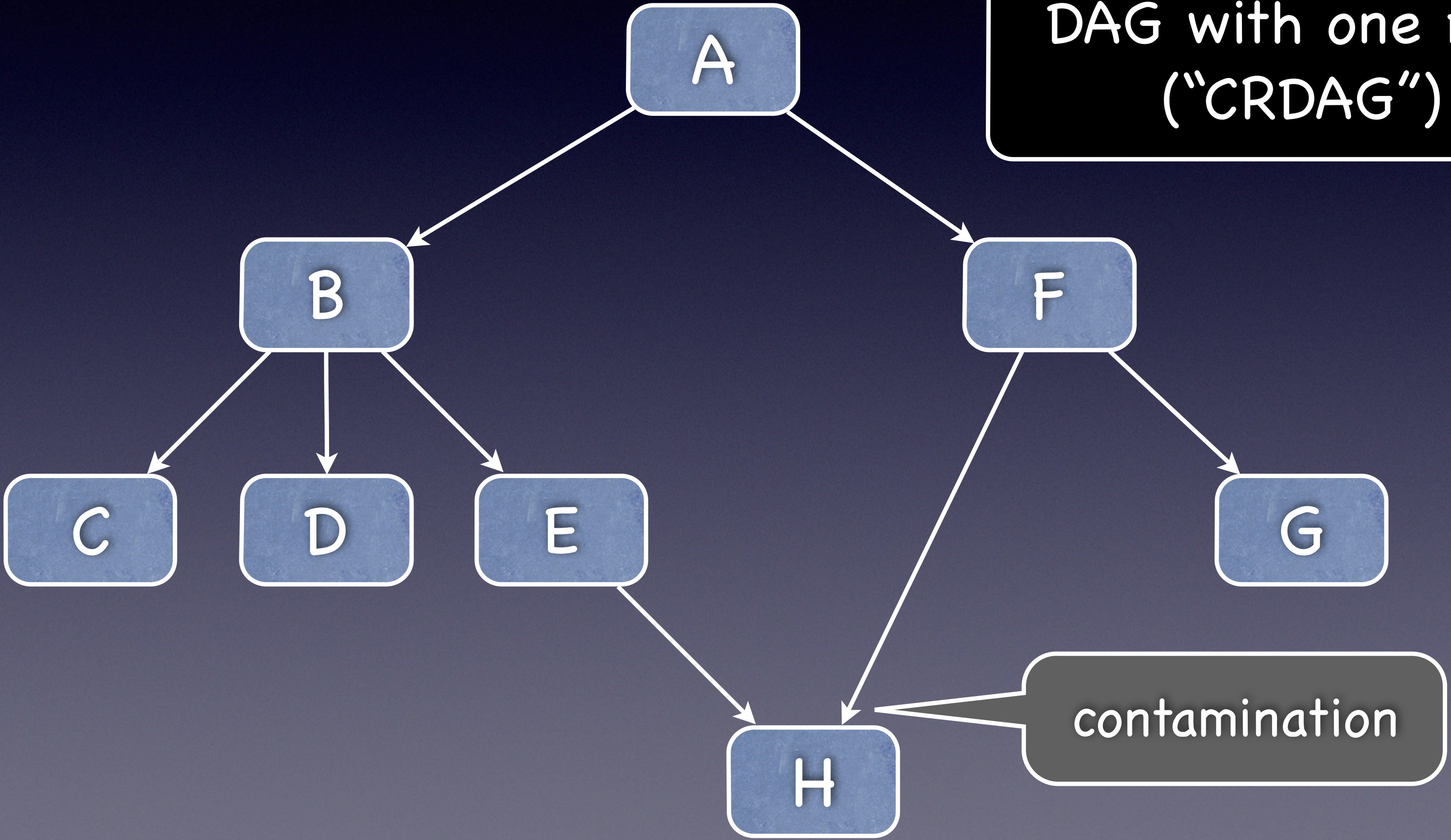
- Will this work? Can a declarative modeling approach be as efficient as a custom-made data mining algorithm (e.g.,Apriori)?
- With current constraint solving technology: yes. Plus, can easy model variants of problems for which no standard algorithm exists!

T. Guns, S. Nijssen, L. De Raedt. *Itemset mining:A constraint programming perspective.* Artificial Intelligence 175(12-13): 1951-1983 (2011)

Stemmatology

- Subfield of philology
- Monks copied manuscripts manually, made changes -> “evolution” of the story
- Study relationships between surviving variants of the story (e.g., to reconstruct a lost original)
- Stemma = “family tree” of a set of manuscripts
- Somewhat similar to **phylogenetic trees** in bioinformatics
  - but there are some differences...
  - solutions specific to stemmatology are needed

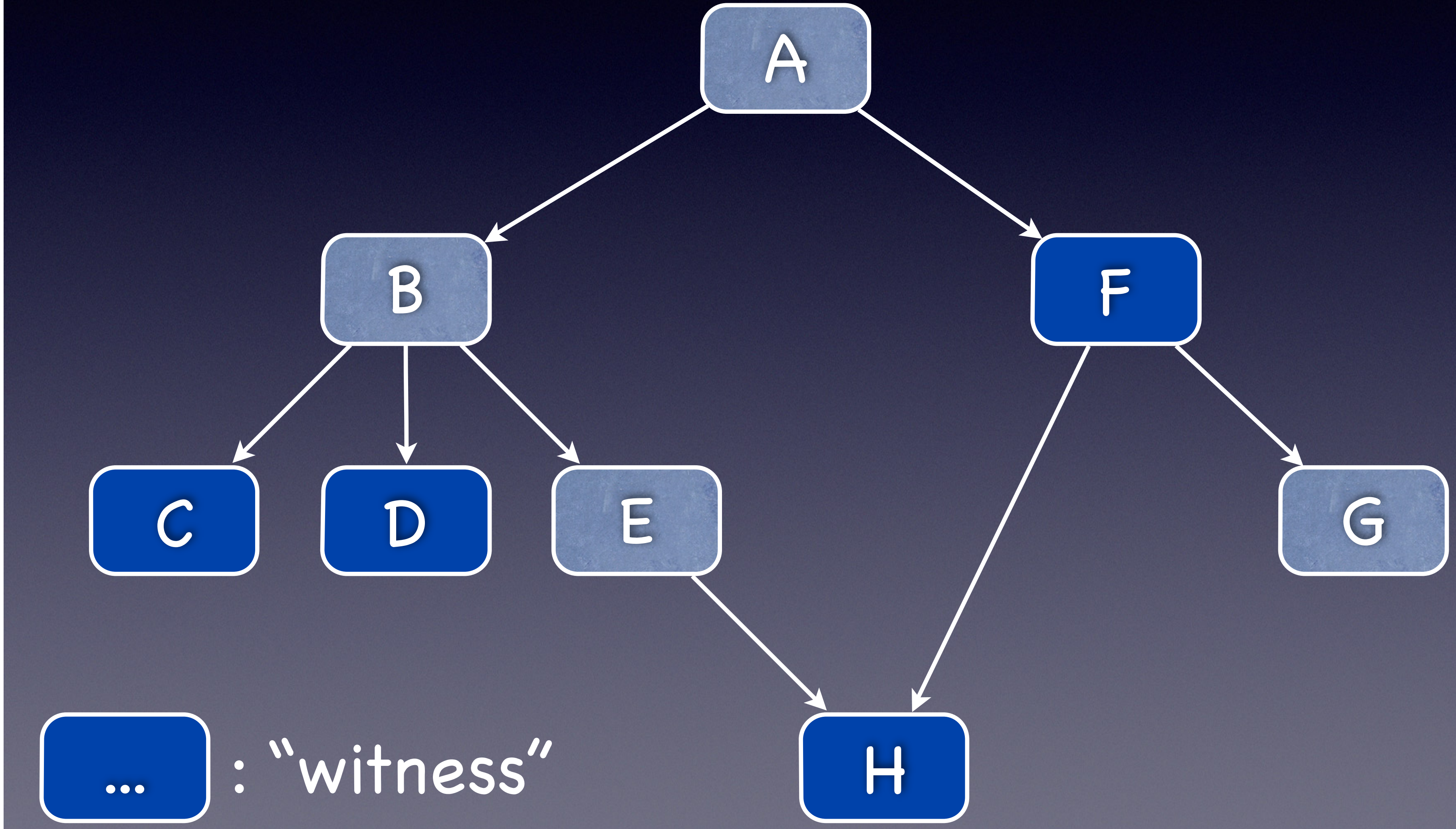
Stemma



stemma = connected DAG with one root ("CRDAG")

contamination

Stemma with witnesses



: "witness"

The data

- A set of manuscripts, which differ in particular places
- Each manuscript is described by a fixed set of attributes
- Each attribute indicates for a particular position which variant occurs there

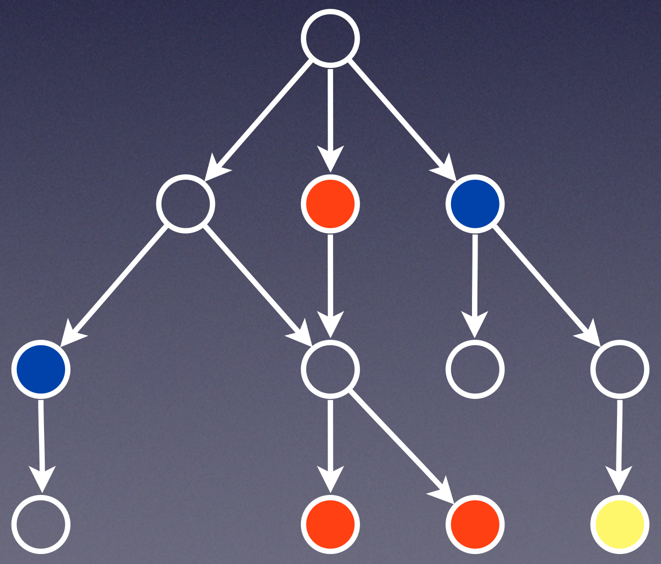
	P1	P2	P3	...
text1	has	Fred	"no", he said	...
text2	had	he	he said no	...
text3	has	he	"never", he said	...

The task

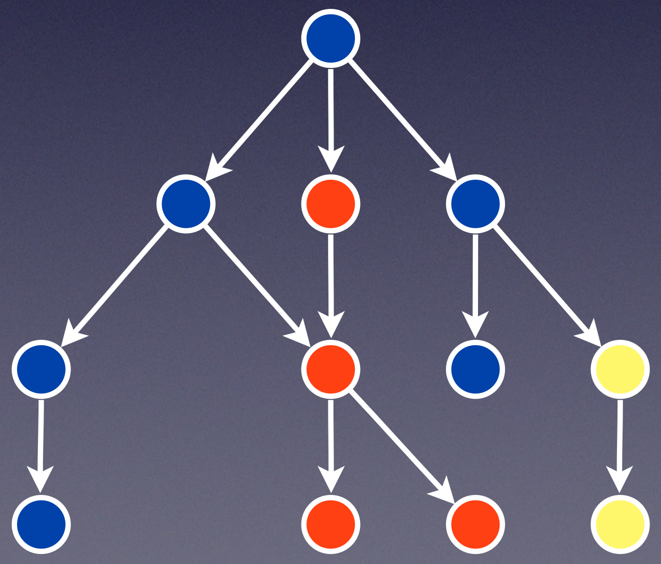
- The classical task: given the data, hypothesize a stemma (cf. phylogenetic tree construction)
- But this is not the only task scholars are interested in
- Here: *Given a stemma and a particular position with multiple variants, is it possible that each variant originated just once? (and if yes, where did it originate?)*

DAG formulation

- In a CRDAG with some groups of nodes defined, complete the groups such that each group forms a CRDAG itself



given



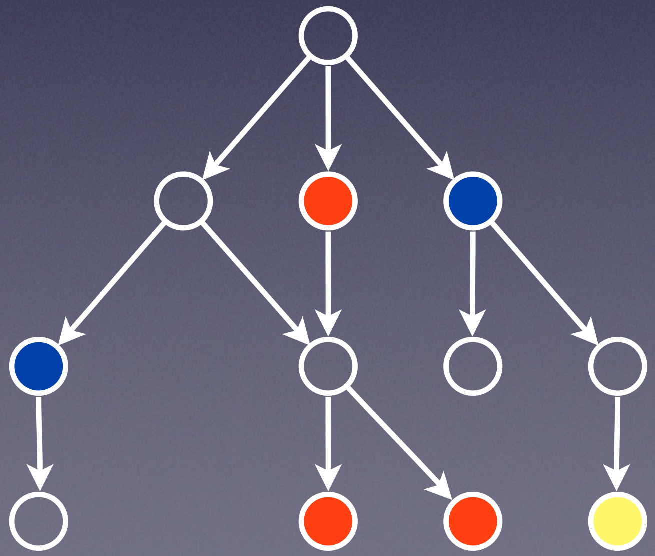
solution

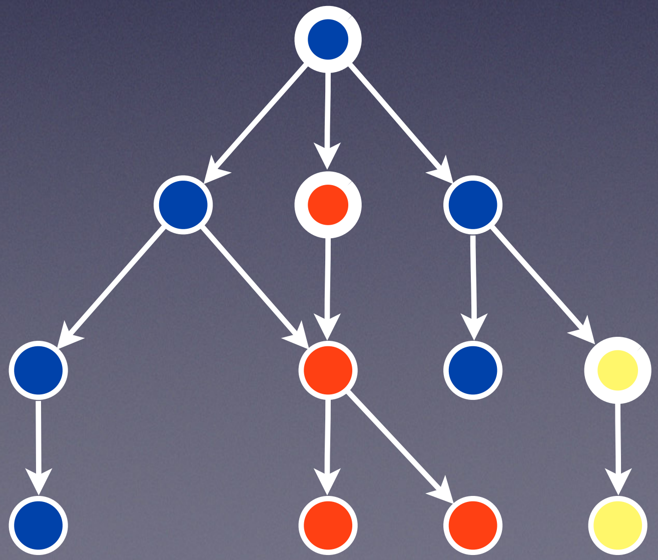
How to solve?

- This is a data analysis question for which no existing method can be readily used - so the data analyst wrote a program herself
- Several versions written; all but the last one found incorrect on at least one case
- “I haven’t been able to find any case where my latest algorithm won’t work - but I can’t prove it’s correct either.” (370 lines of Perl code, excluding graph handling libraries, excluding I/O etc.)
- So we tried a declarative approach

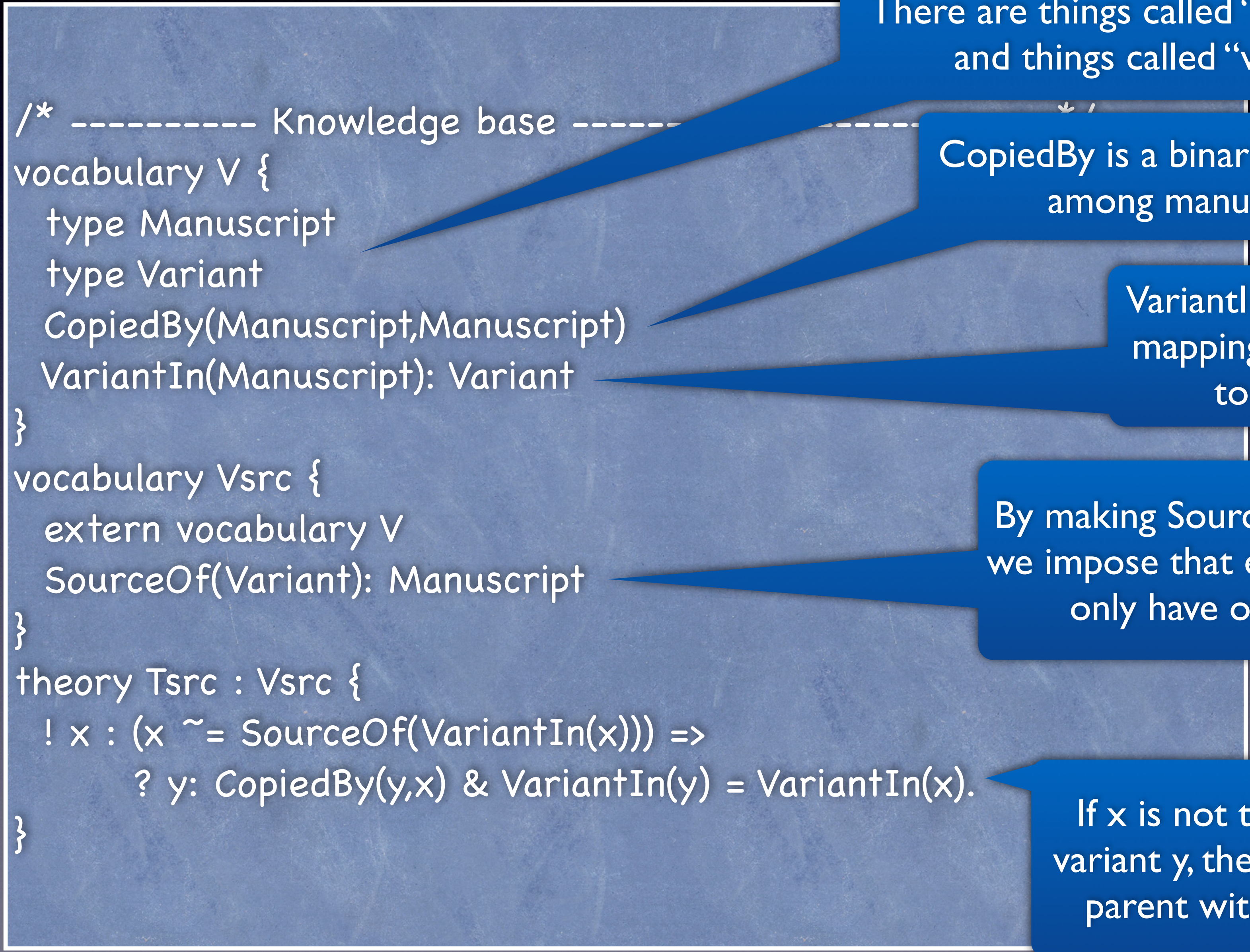
Terminology

- A source of a variant = document where the variant first occurred (= parents do not have that variant)
- Problem reduces to: "given a partially labeled DAG, can you complete the labeling such that each label has only one source?"





IDP formulation



```
/* ---------- Knowledge base ------        ------  */
vocabulary V {
   type Manuscript
   type Variant
   CopiedBy(Manuscript,Manuscript)
  VariantIn(Manuscript): Variant
}
vocabulary Vsrc {
   extern vocabulary V
   SourceOf(Variant): Manuscript
}
theory Tsrc : Vsrc {
  ! x : (x ~= SourceOf(VariantIn(x))) =>
         ? y: CopiedBy(y,x) & VariantIn(y) = VariantIn(x).
}
```

There are things called “manuscripts” and things called “variants”

CopiedBy is a binary relationship among manuscripts

VariantIn is a function mapping manuscripts to variants

By making SourceOf a function, we impose that each variant can only have one source.

If x is not the source of a variant y, then x must have a parent with that variant.

IDP formulation

```
/* --------- Check whether sample fits stemma -------- */
procedure check(sample) {
  idpintern.setvocabulary(sample,Vsrc)
  return sat(Tsrc,sample)
}
```

Checking whether a solution exists =
checking satisfiability of the theory for the given data

IDP formulation

```
procedure main() {
  process("besoin")
  process("parzival")
  process("florilegium")
  process("sermon158")
  process("heinrichi")
}
/* ---------- Procedures for processing -------------- */
procedure process(name) {
  io.write("Processing ",name,".\n")
  local path = "data/"
  local stemmafilename = path..name..".dot"
  local samplefilename = path..name..".json"
  processFiles(stemmafilename,samplefilename)
}
procedure processFiles(stemmafilename,samplefilename) {
  local stemma,nbnodes,nbedges = readStemma(stemmafilename)
  io.write("Stemma has ",nbnodes," nodes and ",nbedges, " edges.\n")
  local nbp,nbs,time = processSamples(stemma,samplefilename)
  io.write("Found ",nbp," positive out of ",nbs," groupings ")
  io.write("in ",time," sec.\n")
}
procedure readStemma(stemmafilename) { ... }
procedure processSamples(stemma,samplefilename) { ... }
```

creates structures

Results

- Tested on five datasets: same results as earlier procedural implementation, and slightly faster
- Easier to write, and provably correct !
- The original implementation turned out to be incorrect. (First suspicions arose when we noticed the problem was NP-complete, and the algorithm polynomial.)

Further steps...

- Many problems were not satisfiable (stemma + observed variants contradict one-source hypothesis)
- So, what's the minimal number of sources needed to explain the observations for a particular stemma & attribute?

IDP formulation

```
vocabulary V { ... }
vocabulary Vms {
  extern vocabulary V
  IsSource(Manuscript)
}
theory Tms : Vms {
  { !x: IsSource(x) <- ~?y: CopiedBy(y,x) & VariantIn(y)=VariantIn(x). }
}
term NbOfSources : Vms {
  #{x:IsSource(x)}
}
procedure minSources(sample) {
  idpintern.setvocabulary(sample,Vms)
  return minimize(Tms, sample, NbOfSources)[1]
}
```

Now, we allow multiple sources per variant (restriction “one source per variant” is gone)

x is a source if (and only if) it does not have a parent with the same variant.

NbOfSources is the number of x for which IsSource(x) is true

Complete the theory so that NbOfSources is minimal

Results

- With limited changes to the declarative specification, this problem gets solved in seconds
- Adapting the procedural program would not be trivial

```
Processing besoin.
Stemma has 13 nodes and 13 edges.
IsSource = { T2; U }
IsSource = { C; T2 }
IsSource = { D; J; L; M; T2; U; V }
...
IsSource = { B; F; J; T2 }
Minimized for 44 groupings in 0 sec.
```

IDP3 for Data Analysis

- We experimented with multiple other tasks
- We consistently found those tasks relatively easy to define, and the correctness of their description easily checked
- In the one case where we could compare with a procedural solution, the declarative solver was as fast as the tailor-made program

Declarative Data Analysis

- Some data analysis tasks do not fit existing systems
- Writing a program that correctly addresses the task can be challenging
- Declarative modeling languages can be an *easy, flexible and efficient* solution for such data analysis tasks

Declarative experimentation

- Basic idea:
  - Ask a question about some population
  - Let the system answer it
- System may
  - use an existing database that is a sample from the population
  - collect more data if the existing database is insufficient
- From user's point of view:
  - Query the *population* instead of the database itself
  - Choice of statistical methodology & interpretation of outcome are moved into the system

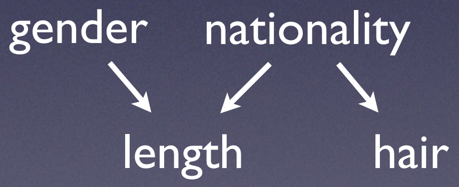
Example

```
ESTIMATE MEAN length
FROM employee
WHERE gender=‘male’ AND nationality= ‘Swedish’ AND haircolor= ‘red’
ENSURING CONF=0.95 AND WIDTH <= 5
```

- population mean, not DB mean
- if not enough data, collect more

What if a qualitative model of the population is given?

Can simplify query using the model (more data available)



```
ESTIMATE MEAN length
FROM employee
WHERE gender=‘male’ AND nationality= ‘Swedish’
ENSURING CONF=0.95 AND WIDTH <= 5
```

G.Vanwinckelen & H. Blockeel. *A query language for statistical inference.* ECMLPKDD-2013 Workshop on languages for ML and DM.

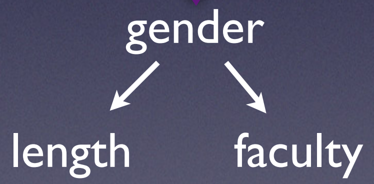
+ ongoing work

Example

```
ESTIMATE MEAN length
FROM student
WHERE faculty=‘engineering’
ENSURING CONF=0.95 AND WIDTH <= 5
```

Say, not enough measurements of “length” among eng. students…

… but we have this qualitative model of the population…



… and we observe: 90% male and 10% female among engineers…

… and we have lots of length measurements for other students

Mean length can be estimated as :
0.9*(MEAN length FROM student WHERE gender=‘male’)
+ 0.1*(MEAN length FROM student WHERE gender=‘female’)

Hypothesis tests

- Instead of estimation, consider hypothesis tests
- Ideally:
  - the *hypothesis* is formulated
  - the system chooses an appropriate statistical test (= assumptions not violated by the data)
  - the system tells us what we can conclude about the hypothesis
- This relieves the user from having to know many hypothesis tests, their interpretation, their correct usage, …

Finding “action rules”

- Say, you want to sell more cigarettes
- But you’re not allowed to promote tobacco directly
- Perhaps you can promote something else, hoping that it will indirectly increase the sales of tobacco?
- *Action rule mining*: given some desired outcome, learn rules that tell you what to do to achieve that outcome

Association rules

- Association rules:“people who bought … also bought …”
- Lots of research on finding such rules
- Can you use them for action rule mining? E.g.: if X and Y are often bought together, promote X to sell more Y?

Example

- Association rule:

IF bread & cheese THEN wine (14%)

- Suppose wine is bought by 6% of total population, but 14% of B&C subpopulation; then this rule tells us: *people who buy bread & cheese are more likely to buy wine*
- So can we sell more wine by promoting cheese?

Incorrect causal interpretations

- Association rules do not necessarily indicate causal relationships!
- Much work on action rules assumes that association rules indicate causal relationships
- Similar problem with “What-if analysis” in predictive modeling
  - “If we increase the value assigned to input variable X_4, our model predicts a lower Y”
  - Danger of causal interpretation: “our model says that if we increase X_4, Y will decrease”, rather than “if X_4 had been higher, Y would likely have been lower”
- “Correlation ≠ causation”: the eternal pitfall !

Setting: "cost-effective action mining"

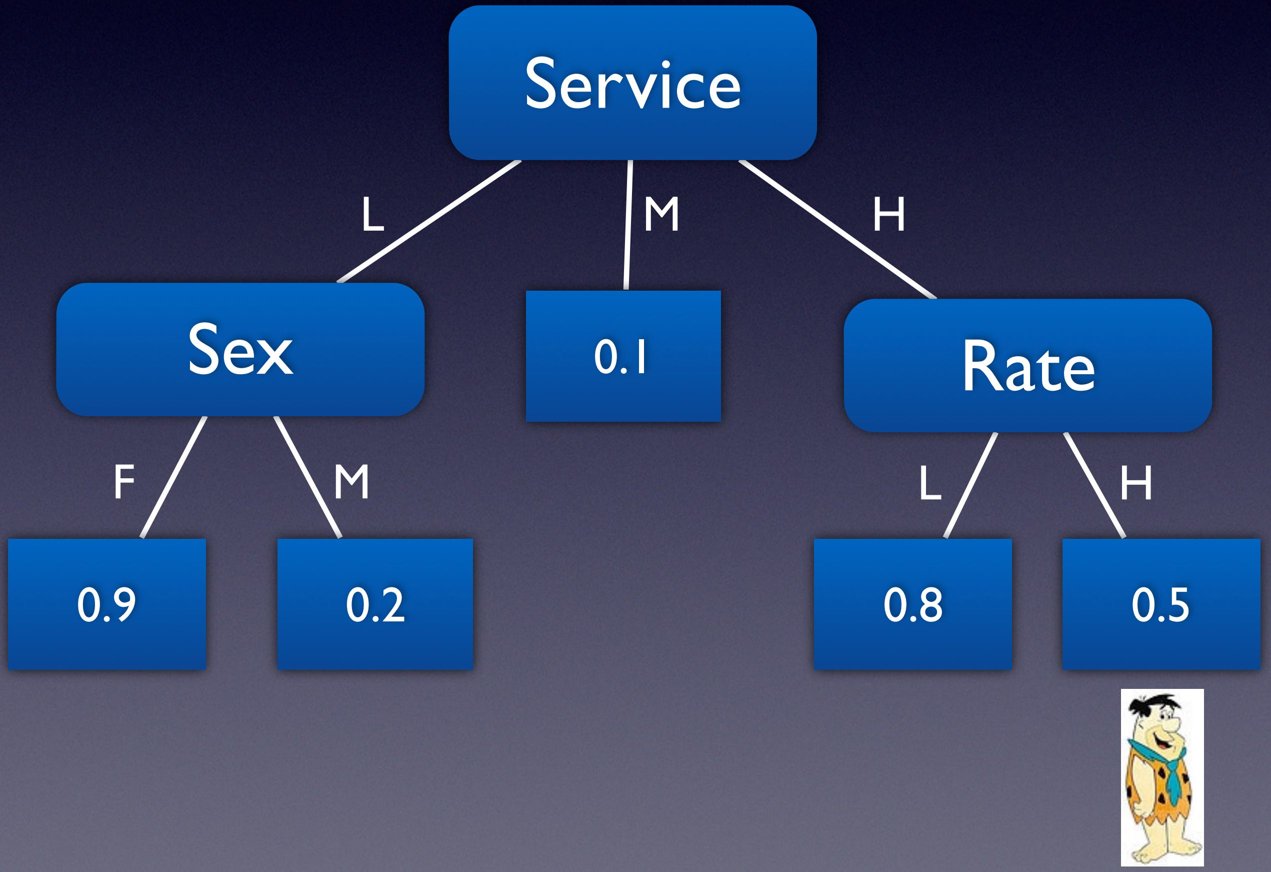
- We are given:
    - A set of attributes A_i with domains D_i, and cost functions C_i: $D_i \times D_i \rightarrow \mathbb{R}$
    - A "target attribute" T with domain D_T and profit function P: V $\rightarrow \mathbb{R}$
- An <u>action</u> A is a set of externally induced changes $a_i \rightarrow a_i'$ of attribute values ("interventions")
- The <u>cost</u> of an action is the sum of the costs of the changes: $C(A) = \sum_{(ai \rightarrow ai') \in A} C_i(a_i, a_i')$

- Changing one attribute may have an effect on other attributes or on the target
- Let *t* be the original (pre-action) value of the target, and *t'* the new value
- The profit of an action A is P(*t'*)-P(*t*)
- The net profit of A is NP(A)=P(*t'*)-P(*t*)-C(A)
  - this assumes *t'* is known
- The expected net profit of A is ENP(A)=*E*(P(*t'*))-P(*t*)-C(A)
  - *t'* not known

Action (rule) mining

- Given the C_i and P functions and a dataset D $\subseteq D_1 \times \ldots \times D_n \times D_T$
- Find:
  - For a given instance x, the action with highest ENP [“action mining”, transductive]
  - A set of rules that predict for any instance x the action with highest ENP [“action rule mining”, inductive]

Is it straightforward?



Fred has high service level, high rate; can we make him more loyal?

(inspired by Yang et al., ICDM 2003)

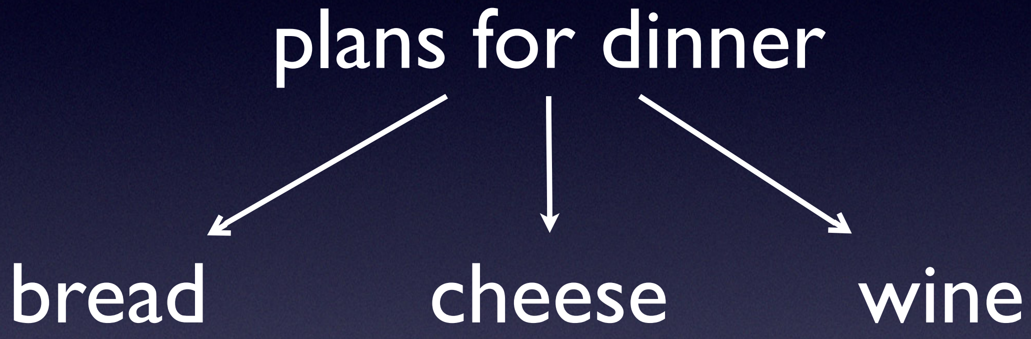
Is it straightforward?

IF bread & cheese THEN wine

- Suppose many people buy bread, but few buy cheese; and we want to sell more wine (high profit). Can we achieve that by giving them cheese for free?

It is not straightforward

- The real question is: will changing a value **cause** the target value to change?
- Causal information is necessary!
- Existing methods implicitly assume
  - each A_i causally affects T
  - no A_i causally affects any A_j, $j \neq i$



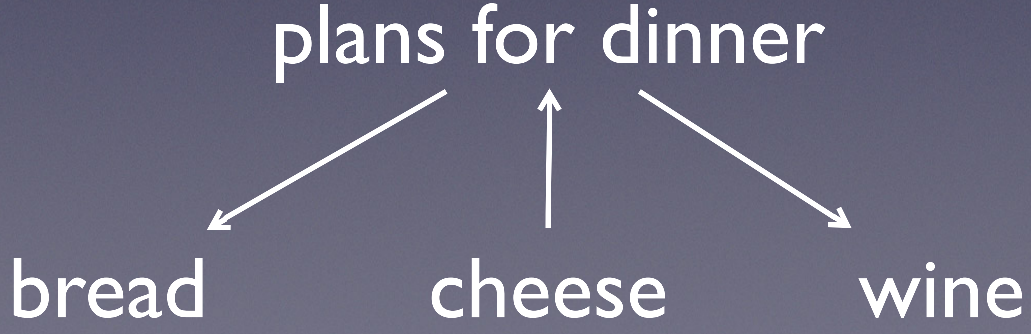
plans for dinner

bread

cheese

wine

Setting 1:
dinner plans affect
bought products



plans for dinner

bread

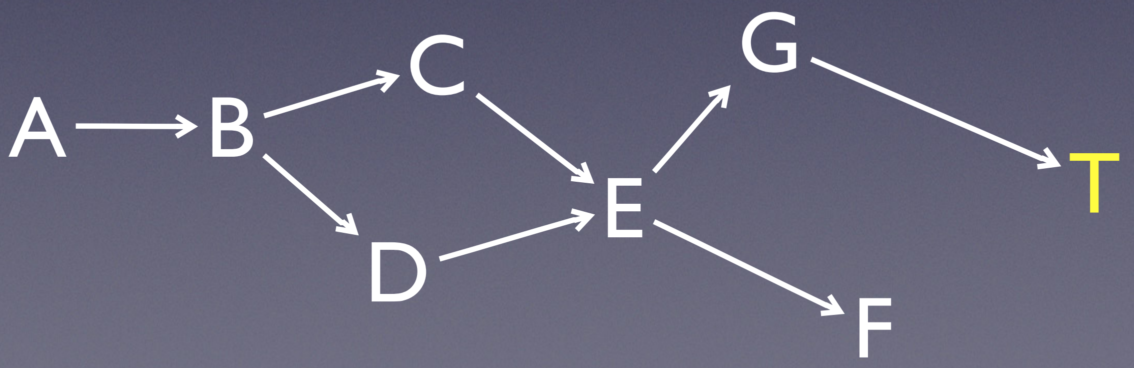
cheese

wine

Setting 2:
promotion affects
dinner plans

Incorporating causal information

- Causal information can be represented as a causal network
- Case 1: causal network is available
- Case 2: causal network is not available



Case 1: CREAM

- “Causal-Relationships-based Economical Action Mining” (CREAM)
- Given a causal network, and an action A, we can compute ENP(A) (standard inference)
- Find the action that maximizes ENP
- CREAM uses a straightforward approach: try many different actions, see how they affect target

P. Shamsinejadbabaki, M. Saraee, H. Blockeel: *Causality-based cost-effective action mining*. Intelligent Data Analysis 17(6): 1075-1091 (2013)

Case 2: no causal information

- CREAM assume a causal network is given
- Often, this is not the case
- Can we *learn* the causal network from the data?
  - Classic view in statistics: only from experimental studies, not from observational ones (correlation ≠ causation)
  - Pearl (1990-...): In some cases (and under mild assumptions), we *can* determine causal relationships from observations!
  - Recent results (Schölkopf et al., 2010-…) broaden the conditions under which causality can be determined

Inferring causation: the basic idea

Suppose there is evidence that A and B are directly dependent, and B and C too, but no direct connection between A and C (could be based on pre-existing knowledge, or observations of dependencies)

$A — B — C$

No direct link between A and C; all information flow goes through B

$A \rightarrow B \rightarrow C$

$A \leftarrow B \leftarrow C$

$A \leftarrow B \rightarrow C$

$A \rightarrow B \leftarrow C$

4 different causal connections possible

Inferring causation: the basic idea

Find a number of cases with the same value for B...

$A \rightarrow B \rightarrow C$

- A and C correlate
- Fixing B removes correlation

$A \leftarrow B \leftarrow C$

- A and C correlate
- Fixing B removes correlation

$A \leftarrow B \rightarrow C$

- A and C correlate
- Fixing B removes correlation

$A \rightarrow B \leftarrow C$

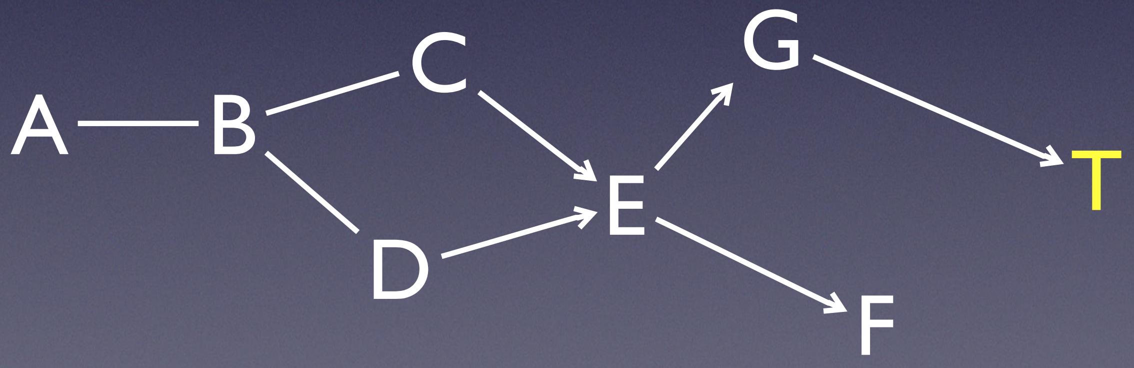
- A and C *do not* correlate
- Fixing B introduces correlation

Causality among 2 variables

- Even among 2 variables, causality can be determined if noise is present (intuitively, the noise is "the third variable")
- Series of recent work by Max Planck, Tübingen (Schölkopf, Janzing, …)

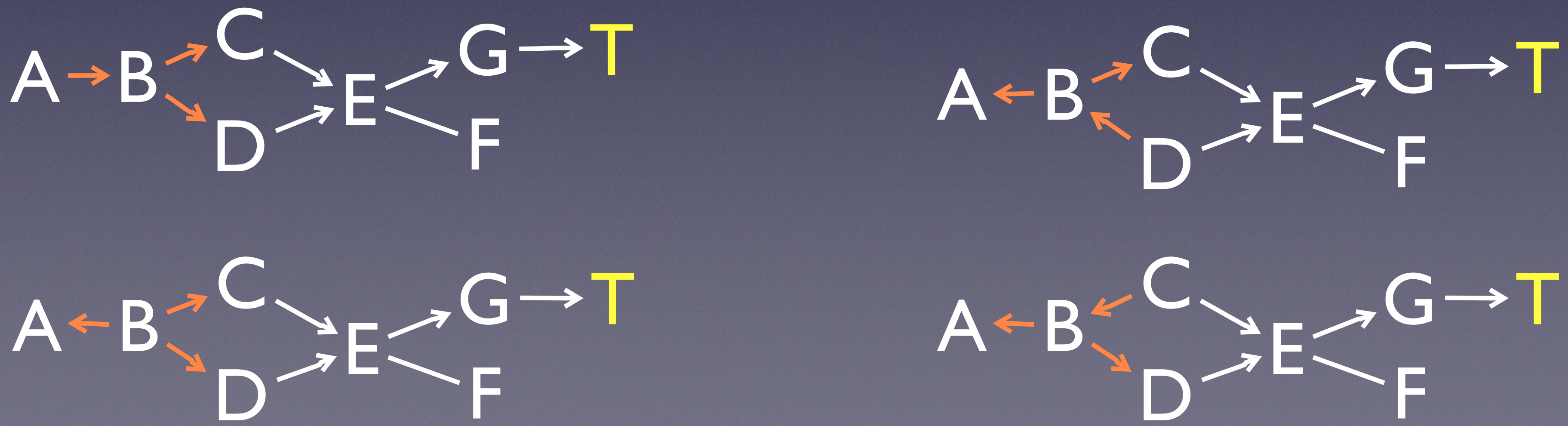
Partial causal networks

- For *some* edges in a network, the direction can be determined; for others it cannot
- This gives only partial causal information



What is the effect of A on T?

- The question cannot be answered with certainty: not enough information
- (Ugly) solution: make different guesses of the complete network, perform inference in these, combine results.



ICE-CREAM

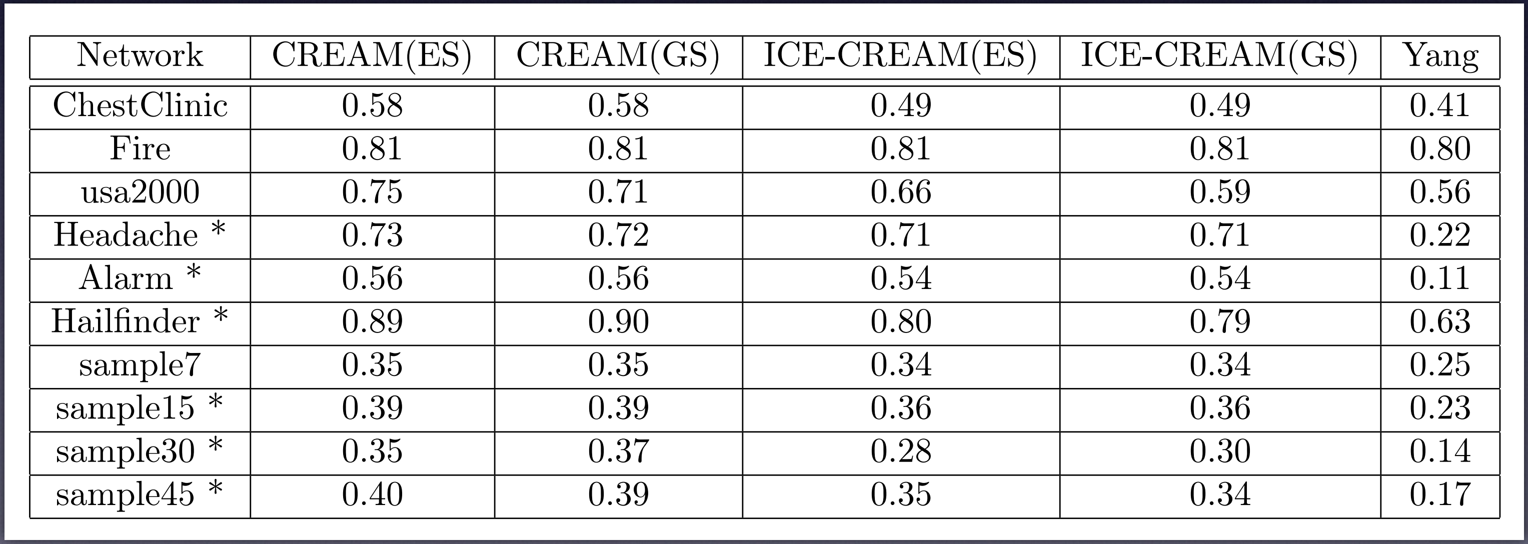
- “IC-enabled CREAM”
- Run IC (“Inductive Causation”, Verma & Pearl, 1991) to derive a partial causal network
- For any action A, estimate ENP(A) as follows:
    - repeat n times:
        - create a random complete network CN consistent with the partial one
        - compute ENP for CN using CREAM
    - return the average of all ENPs thus computed

Experiments

- Experiments on some “real” (pre-existing) and artificial (created for this purpose) datasets
- For all these datasets, we know the real causal model
- Thus, we can compare:
  - methods that ignore causality (e.g., Yang et al.’s)
  - methods that use the causal network (CREAM)
  - methods that use the estimated, partial causal network (ICE-CREAM)

Results

Average ENP of actions suggested by the method:



Network	CREAM(ES)	CREAM(GS)	ICE-CREAM(ES)	ICE-CREAM(GS)	Yang
ChestClinic	0.58	0.58	0.49	0.49	0.41
Fire	0.81	0.81	0.81	0.81	0.80
usa2000	0.75	0.71	0.66	0.59	0.56
Headache *	0.73	0.72	0.71	0.71	0.22
Alarm *	0.56	0.56	0.54	0.54	0.11
Hailfinder *	0.89	0.90	0.80	0.79	0.63
sample7	0.35	0.35	0.34	0.34	0.25
sample15 *	0.39	0.39	0.36	0.36	0.23
sample30 *	0.35	0.37	0.28	0.30	0.14
sample45 *	0.40	0.39	0.35	0.34	0.17

Causality & action rule mining

- Traditional methods for action rule mining make strong assumptions about causality
- Trying to determine the actual causal relationships (IC) and taking these into account (CREAM) gives better results

- Overall conclusion: be cautious with causal interpretation of predictive models
- Declarative data mining could guard against this, if a “causality-aware” language is used

Conclusions

- “Declarative data mining” has the potential of *making data analysis easier, more efficient, more accurate and less error-prone*
- Research on inductive databases, constraint-based data mining, meta-learning, declarative knowledge representation is highly relevant for achieving this goal

Many thanks to…

- Antoine Adam
- Tara Andrews
- Sander Beckers
- Maurice Bruynooghe
- Marc Denecker
- Luc De Raedt
- Tias Guns
- Pan Hu
- Caroline Macé
- Wannes Meert
- Siegfried Nijssen
- Pirooz Shamsinejadbabaki
- Joaquin Vanschoren
- Gitte Vanwickelen
- Joost Vennekens
- Celine Vens