

# Declarative data analysis

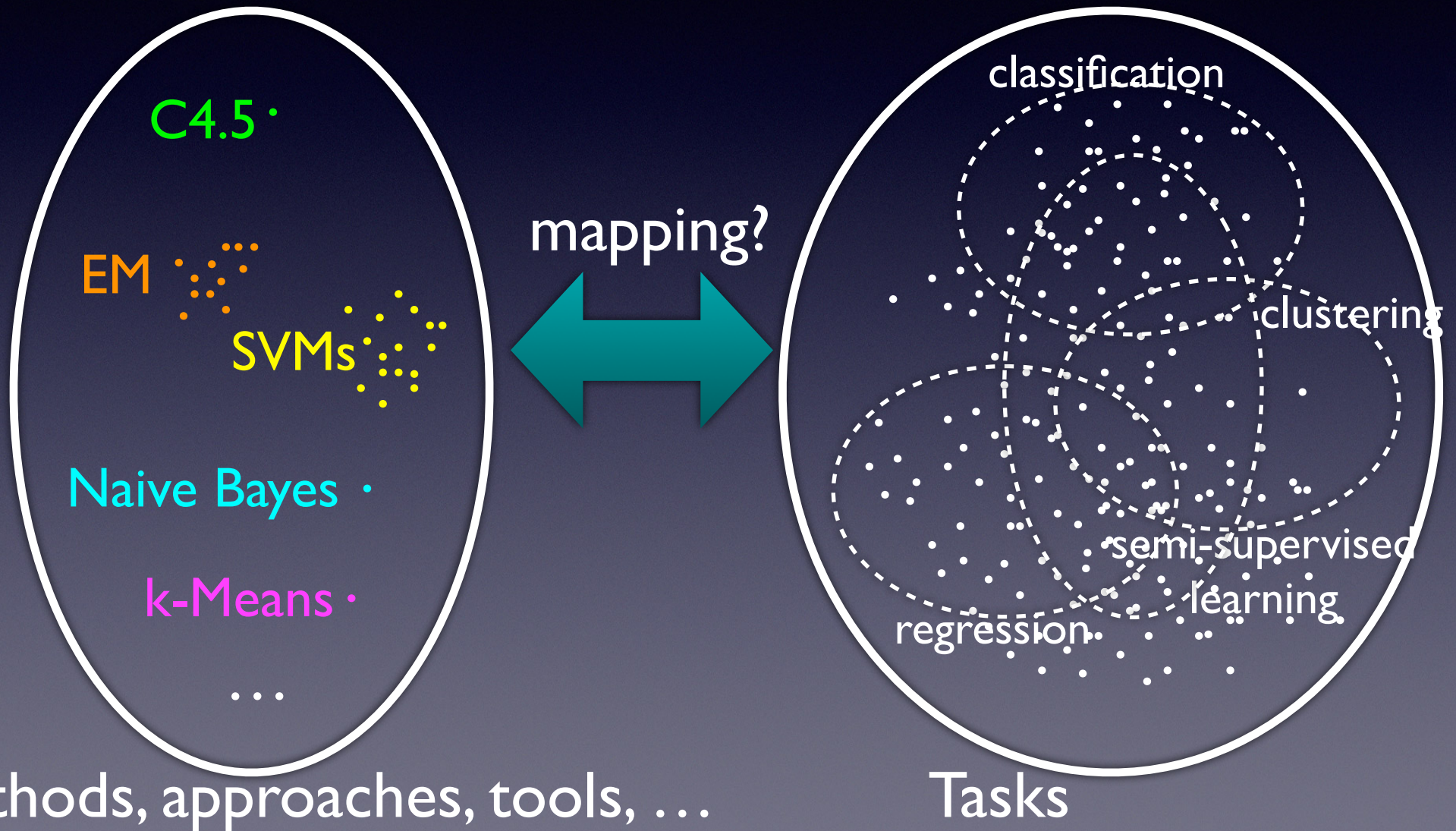
Hendrik Blockeel

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!

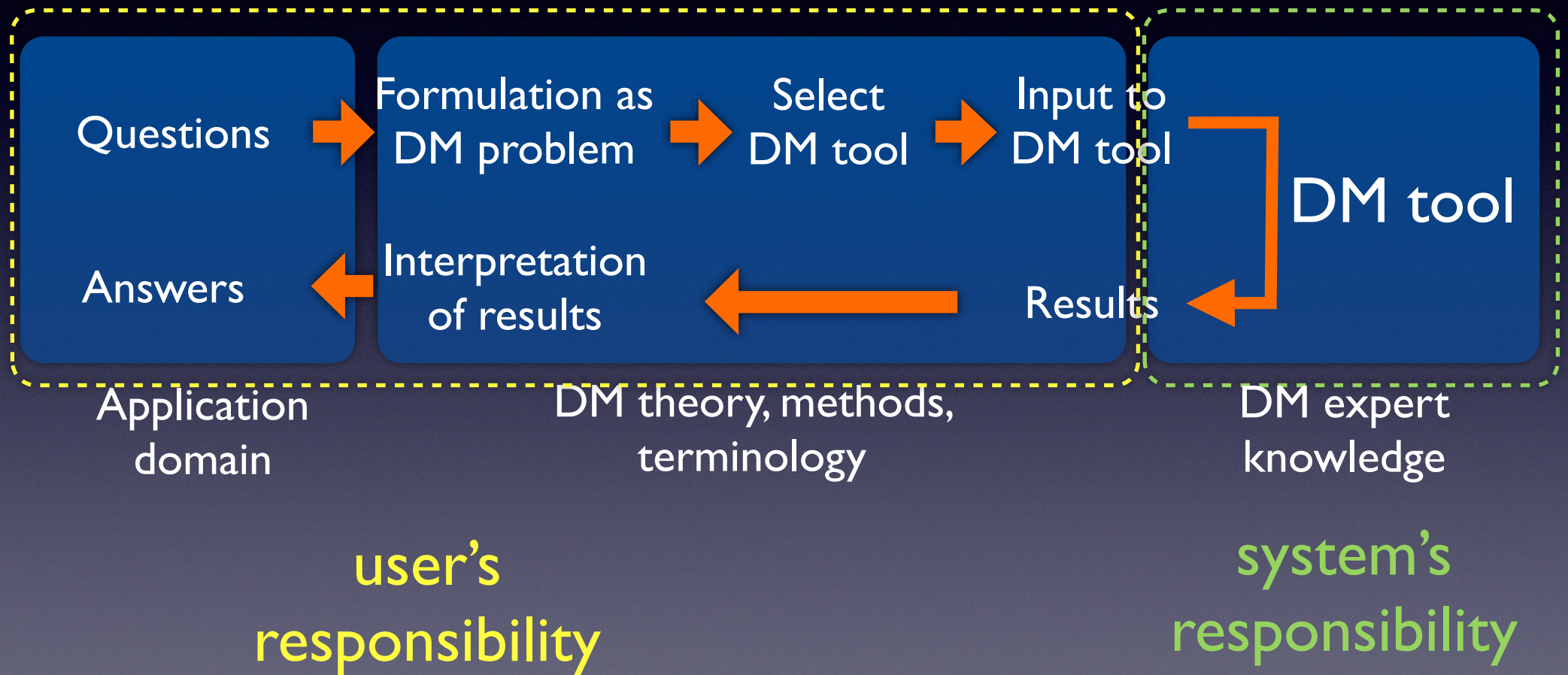
	Method A	Method B	Method C
Dataset 1	0.86 (0.02)	0.83 (0.01)	0.82 (0.01)
Dataset 2	0.85 (0.01)	<b>0.91 (0.01)</b>	0.82 (0.03)
...			

← acc (stdev)

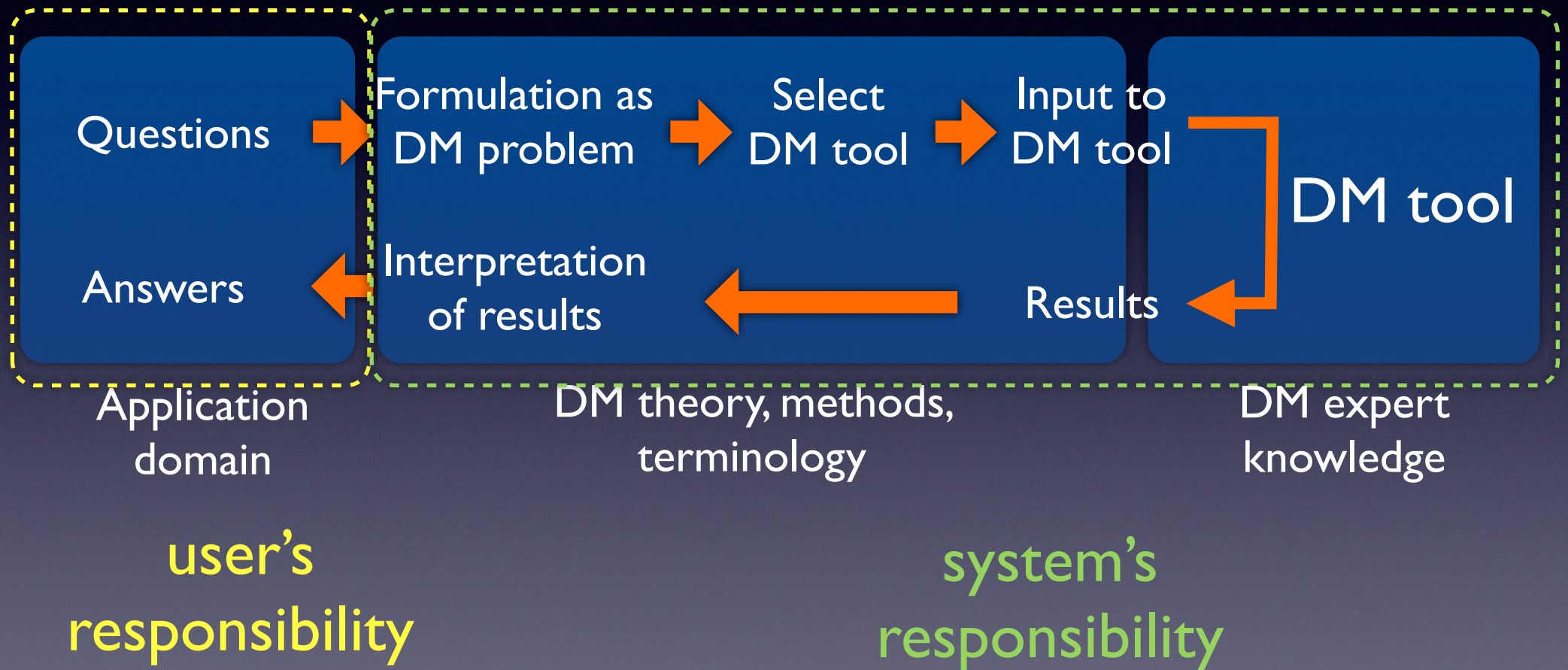
# Statistics is tricky

- 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



# Data analysis as it should be



# 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
  - subtleties in interpretation of DM results

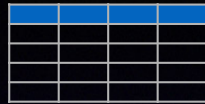
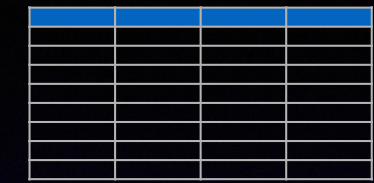
# A declarative language for clustering

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

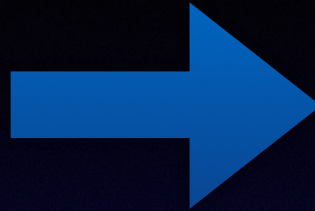
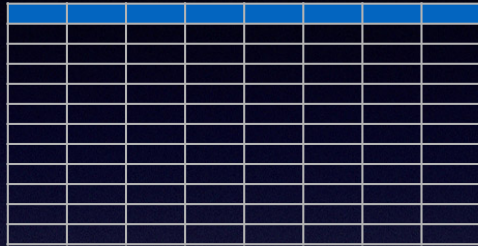
A.Adam, H. Blockeel, S. Govers, A.Aertsen (2013). *SCCQL: A constraint-based clustering system*. ECMLPKDD 2013 demo. Proc. of ECMLPKDD 2013 part 3: 681-684.



# 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 statevertime 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

# EXAMPLE APPLICATION:

# ENTITY RESOLUTION

### Generalizing from Example Clusters

Pat Heu<sup>1,2</sup>, Celine Vens<sup>1</sup>, Bart Verwey<sup>1,2</sup>, and Hendrik Blockeel<sup>1,2</sup>

<sup>1</sup> KU Leuven, Department of Computer Science, Celestijnenlaan 200A, 3000 Leuven, Belgium  
<sup>2</sup> Ghent Institute of Artificial Intelligence Research, Ghent University, Coupure links 653, 9000 Ghent, Belgium

**Abstract.** We consider the following problem: Given a set of data and one or more examples of clusters, find a clustering of the whole data that is consistent with the given clusters. This is essentially a semi-supervised clustering problem. In this paper we present a novel semi-supervised clustering setting in which we are given a set of clusters and a set of data. We identify two classes for the data, which are related to the given clusters. We show that this problem can be solved by a semi-supervised clustering method. We identify two classes for the data, which are related to the given clusters. We show that this problem can be solved by a semi-supervised clustering method. We identify two classes for the data, which are related to the given clusters. We show that this problem can be solved by a semi-supervised clustering method.

**Keywords:** Clustering, Semi-supervised Clustering, Consensus-based Clustering, Machine Learning

#### 1 Introduction

The task of clustering data is ubiquitous in knowledge discovery. Partitioned (or semi-supervised) clustering can be defined as the following task: given a dataset  $D$  and a partition  $D$  into subsets ( $C$  clusters) such that instances within the same cluster tend to be similar, and instances in different clusters dissimilar. The notion of "similarity" is crucial here depending on how this is defined, different solutions will be found. This is true especially for high-dimensional spaces, where different solutions may reveal different clusterings [1].

It is not always easy for a user to define a good similarity measure. However, users may be able to give examples of instances that in their opinion should, or should not, belong to the same cluster. The clustering one may want to be found should be understood here in the sense of similarity that the user has in mind, and as a consequence produce a clustering that is similar to the type of clustering originally intended. This is called semi-supervised clustering, as the user gives partial information about the desired clustering of the data and constraints that the clustering must satisfy.

Most existing methods for semi-supervised clustering allow the user to provide a number of so-called seed and outlier-like constraints, indicating for each instance whether they should (or not) be in the same cluster. Vens et al. [11] recently introduced a slightly different setting, called "semi-supervised clustering with example

Robert Smith

### Identifying Proteins Involved in Parasitism by Discovering Degenerated Motifs

Celine Vens<sup>1,2</sup>, Edoardo D'Amico<sup>1</sup>, and Martinus Thunnissen<sup>1</sup>

<sup>1</sup> Department of Computer Science, Ghent University, Coupure links 653, 9000 Ghent, Belgium  
<sup>2</sup> Ghent Institute of Artificial Intelligence Research, Ghent University, Coupure links 653, 9000 Ghent, Belgium

**Abstract.** We propose a method to identify degenerated motifs based on a comparison of amino acids and on the physico-chemical properties of those residues. Given a set of protein sequences known to be involved in a certain biological system (identified motifs) and a set of protein sequences known not to be involved in that system (negative motifs), our method is able to identify motifs that are frequent in positive sequences while infrequent or absent in negative sequences. The identified motifs can then be used to identify motifs that are frequent in positive sequences while infrequent or absent in negative sequences. The identified motifs can then be used to identify motifs that are frequent in positive sequences while infrequent or absent in negative sequences.

**Keywords:** Clustering, Semi-supervised Clustering, Consensus-based Clustering, Machine Learning

#### 1 Introduction

Identifying motifs in biological sequences is an important challenge in biology. Proteins involved in the same biological system or physiological function (e.g., immune response, chemokinesis, secretion, signal transduction, ...) are subject to similar evolutionary and functional processes that can be observed in the protein sequence level. Finding motifs specific to proteins involved in the same process can help deciphering the determination of their function and thus be used in identifying new candidate proteins involved in important biological systems.

To our knowledge, all currently available methods search motifs in protein sequences at the amino acid level. However, when comparing motifs to sequences with high variability (e.g., proteins, ...), a better choice is to compare three-dimensional structures. In order to capture the three-dimensional structure of a protein, it is important to consider the actual sequence of the amino acid residues and not just the amino acid side chains. This is especially true for motifs that are highly conserved in the sequence, the amino acid side chains are highly conserved and the amino acid side chains are highly conserved in the sequence.

We propose a method that allows to identify conserved motifs based on a comparison of amino acids and on the physico-chemical properties of those residues. Given a set of protein sequences known to be involved in a certain biological system (identified motifs) and a set of protein sequences known not to be involved in that system (negative motifs), our method is able to identify motifs that are frequent in positive sequences while infrequent or absent in negative sequences. The identified motifs can then be used to identify motifs that are frequent in positive sequences while infrequent or absent in negative sequences.

Robert L. Smith

### Annotating transposable elements in the genome using relational decision tree ensembles

Edoardo P. Costa<sup>1</sup>, Leandro Schenker<sup>1</sup>, Ricardo Cerri<sup>1</sup>, Celine Vens<sup>1</sup>, Carlos N. Freitas<sup>1</sup>, Claudio M. A. Correia<sup>1</sup>, Jan Kramler<sup>2</sup>, and Hendrik Blockeel<sup>1</sup>

<sup>1</sup> Department of Computer Science, KU Leuven, Celestijnenlaan 200A, 3000 Leuven, Belgium  
<sup>2</sup> Instituto de Ciências Matemáticas e Computação, Universidade de São Paulo, Caixa de São Carlos, 13069-700 São Carlos, SP, Brazil

**Abstract.** Transposable elements (TEs) are DNA sequences that can change their location within the genome. They make up a large portion of the DNA in eukaryotic organisms and contribute to genetic diversity within and across species. Their transposing mechanisms may also affect the functionality of genes. Accurate annotation of TEs is an important step towards understanding their effects on genes and their role in genome evolution. We introduce a framework for annotating TEs which is based on relational decision tree learning. It allows to naturally represent the structural data and biological processes involving TEs. Furthermore, it also allows the integration of background knowledge and benefits from the interpretability of decision trees. Preliminary experiments show that our method outperforms state-of-the-art systems for TE annotation.

**Keywords:** relational decision trees, hidden Markov models, genome annotation, transposable elements

#### 1 Introduction

Transposable elements (TEs) are DNA sequences that can change their location within the genome. They make up a large portion of the DNA in eukaryotic organisms and contribute to genetic diversity within and across species. Furthermore, their transposing mechanisms increase the size of the genome and affect the functionality of genes. Accurate annotation of TEs, together with the development of integrable models explaining these annotations, is an important step towards understanding their effects on genes and their role in genome evolution [1].

Currently, annotation of TEs involves a lot of manual work. Automated methods exist that search DNA & candidate TEs, but human annotations

Bob Smith

### Identifying Discriminative Classification Based Motifs in Biological Sequences

Celine Vens<sup>1,2</sup>, Marie-Noëlle Rossé<sup>1</sup>, and Edoardo P. Costa<sup>1</sup>

<sup>1</sup> Department of Computer Science, Ghent University, Coupure links 653, 9000 Ghent, Belgium  
<sup>2</sup> Ghent Institute of Artificial Intelligence Research, Ghent University, Coupure links 653, 9000 Ghent, Belgium

**Abstract.** We propose a method to identify discriminative motifs based on a comparison of amino acids and on the physico-chemical properties of those residues. Given a set of protein sequences known to be involved in a certain biological system (identified motifs) and a set of protein sequences known not to be involved in that system (negative motifs), our method is able to identify motifs that are frequent in positive sequences while infrequent or absent in negative sequences. The identified motifs can then be used to identify motifs that are frequent in positive sequences while infrequent or absent in negative sequences.

**Keywords:** Clustering, Semi-supervised Clustering, Consensus-based Clustering, Machine Learning

#### 1 Introduction

Identifying motifs in biological sequences is an important challenge in biology. Proteins involved in the same biological system or physiological function (e.g., immune response, chemokinesis, secretion, signal transduction, ...) are subject to similar evolutionary and functional processes that can be observed in the protein sequence level. Finding motifs specific to proteins involved in the same process can help deciphering the determination of their function and thus be used in identifying new candidate proteins involved in important biological systems.

To our knowledge, all currently available methods search motifs in protein sequences at the amino acid level. However, when comparing motifs to sequences with high variability (e.g., proteins, ...), a better choice is to compare three-dimensional structures. In order to capture the three-dimensional structure of a protein, it is important to consider the actual sequence of the amino acid residues and not just the amino acid side chains. This is especially true for motifs that are highly conserved in the sequence, the amino acid side chains are highly conserved and the amino acid side chains are highly conserved in the sequence.

We propose a method that allows to identify discriminative motifs based on a comparison of amino acids and on the physico-chemical properties of those residues. Given a set of protein sequences known to be involved in a certain biological system (identified motifs) and a set of protein sequences known not to be involved in that system (negative motifs), our method is able to identify motifs that are frequent in positive sequences while infrequent or absent in negative sequences. The identified motifs can then be used to identify motifs that are frequent in positive sequences while infrequent or absent in negative sequences.

B.W. Smith

### Semi-supervised Clustering with Example Clusters

Celine Vens<sup>1,2</sup>, Bart Verwey<sup>1,2</sup>, and Hendrik Blockeel<sup>1,2</sup>

<sup>1</sup> Department of Computer Science, KU Leuven, Belgium  
<sup>2</sup> Ghent Institute of Artificial Intelligence Research, Ghent University, Coupure links 653, 9000 Ghent, Belgium

**Abstract.** We consider the following problem: Given a set of data and one or more examples of clusters, find a clustering of the whole data that is consistent with the given clusters. This is essentially a semi-supervised clustering problem. In this paper we present a novel semi-supervised clustering setting in which we are given a set of clusters and a set of data. We identify two classes for the data, which are related to the given clusters. We show that this problem can be solved by a semi-supervised clustering method. We identify two classes for the data, which are related to the given clusters. We show that this problem can be solved by a semi-supervised clustering method.

**Keywords:** Clustering, Semi-supervised Clustering, Consensus-based Clustering, Machine Learning

#### 1 INTRODUCTION

The task of clustering data is ubiquitous in knowledge discovery. Partitioned (or semi-supervised) clustering can be defined as the following task: given a dataset  $D$  and a partition  $D$  into subsets ( $C$  clusters) such that instances within the same cluster tend to be similar, and instances in different clusters dissimilar. The notion of "similarity" is crucial here depending on how this is defined, different solutions will be found. This is true especially for high-dimensional spaces, where different solutions may reveal different clusterings [1].

It is not always easy for a user to define a good similarity measure. However, users may be able to give examples of instances that in their opinion should, or should not, belong to the same cluster. The clustering one may want to be found should be understood here in the sense of similarity that the user has in mind, and as a consequence produce a clustering that is similar to the type of clustering originally intended. This is called semi-supervised clustering, as the user gives partial information about the desired clustering of the data and constraints that the clustering must satisfy.

Most existing methods for semi-supervised clustering allow the user to provide a number of so-called seed and outlier-like constraints, indicating for each instance whether they should (or not) be in the same cluster. Vens et al. [11] recently introduced a slightly different setting, called "semi-supervised clustering with example

R. W. Smith

### Identifying Discriminative Classification Based Motifs in Biological Sequences

Celine Vens<sup>1,2</sup>, Marie-Noëlle Rossé<sup>1</sup>, and Edoardo P. Costa<sup>1</sup>

<sup>1</sup> Department of Computer Science, Ghent University, Coupure links 653, 9000 Ghent, Belgium  
<sup>2</sup> Ghent Institute of Artificial Intelligence Research, Ghent University, Coupure links 653, 9000 Ghent, Belgium

**Abstract.** We propose a method to identify discriminative motifs based on a comparison of amino acids and on the physico-chemical properties of those residues. Given a set of protein sequences known to be involved in a certain biological system (identified motifs) and a set of protein sequences known not to be involved in that system (negative motifs), our method is able to identify motifs that are frequent in positive sequences while infrequent or absent in negative sequences. The identified motifs can then be used to identify motifs that are frequent in positive sequences while infrequent or absent in negative sequences.

**Keywords:** Clustering, Semi-supervised Clustering, Consensus-based Clustering, Machine Learning

#### 1 INTRODUCTION

Identifying motifs in biological sequences is an important challenge in biology. Proteins involved in the same biological system or physiological function (e.g., immune response, chemokinesis, secretion, signal transduction, ...) are subject to similar evolutionary and functional processes that can be observed in the protein sequence level. Finding motifs specific to proteins involved in the same process can help deciphering the determination of their function and thus be used in identifying new candidate proteins involved in important biological systems.

To our knowledge, all currently available methods search motifs in protein sequences at the amino acid level. However, when comparing motifs to sequences with high variability (e.g., proteins, ...), a better choice is to compare three-dimensional structures. In order to capture the three-dimensional structure of a protein, it is important to consider the actual sequence of the amino acid residues and not just the amino acid side chains. This is especially true for motifs that are highly conserved in the sequence, the amino acid side chains are highly conserved and the amino acid side chains are highly conserved in the sequence.

We propose a method that allows to identify discriminative motifs based on a comparison of amino acids and on the physico-chemical properties of those residues. Given a set of protein sequences known to be involved in a certain biological system (identified motifs) and a set of protein sequences known not to be involved in that system (negative motifs), our method is able to identify motifs that are frequent in positive sequences while infrequent or absent in negative sequences. The identified motifs can then be used to identify motifs that are frequent in positive sequences while infrequent or absent in negative sequences.

Robert L. W. Smith

### Learning Kernels with Random Forests

Celine Vens<sup>1,2</sup> and Fabrizio Costa<sup>1</sup>

<sup>1</sup> KU Leuven, Department of Computer Science, Celestijnenlaan 200A, 3000 Leuven, Belgium  
<sup>2</sup> Ghent Institute of Artificial Intelligence Research, Ghent University, Coupure links 653, 9000 Ghent, Belgium

**Abstract.** We propose a method to learn effective kernels for semi-supervised clustering. We propose a method to learn effective kernels for semi-supervised clustering. We propose a method to learn effective kernels for semi-supervised clustering. We propose a method to learn effective kernels for semi-supervised clustering.

**Keywords:** Clustering, Semi-supervised Clustering, Consensus-based Clustering, Machine Learning

#### 1 Introduction

The problem of determining a suitable metric space tailored for a given predictive task has been receiving increasing attention in the Machine Learning community. Once distance information is available, several techniques exist for a variety of problems can be defined, ranging from the nearest neighbor algorithm [1] for supervised classification, to clustering algorithms as K-means [2] for unsupervised tasks or multi-dimensional scaling [3] for visualization or pre-processing.

In the last decade, due to the increasing generalization performance and theoretical guarantees offered by Support Vector Machines, kernel-based methods have become mainstream. In this context one is interested in learning the similarity rather than the distance function, although the two tasks are intimately related or may even define one another in terms of the other.

The problem of learning the kernel function has therefore become of interest. As pointed out in [4], given the nature of the task [5] and the nature of the data [6], one can hope to learn effective kernels only when some prior information on the true hypothesis class seems not to be correct. Since kernel-based methods access instances only through the kernel function, the methods between the prior bias and the true hypothesis class can be learned from the kernel function itself. In [4] the authors propose to use the notion of target alignment to measure the quality of such metrics. Its practice is motivated in obtaining kernel functions that agree on the partitioning of the instances according to the target under consideration.

Although a variety of methods have been developed for this task, many of the proposed techniques are applicable only in transductive settings [7], [8].

Robert Smith

### Identifying Discriminative Classification Based Motifs in Biological Sequences

Celine Vens<sup>1,2</sup>, Marie-Noëlle Rossé<sup>1</sup>, and Edoardo P. Costa<sup>1</sup>

<sup>1</sup> Department of Computer Science, Ghent University, Coupure links 653, 9000 Ghent, Belgium  
<sup>2</sup> Ghent Institute of Artificial Intelligence Research, Ghent University, Coupure links 653, 9000 Ghent, Belgium

**Abstract.** We propose a method to identify discriminative motifs based on a comparison of amino acids and on the physico-chemical properties of those residues. Given a set of protein sequences known to be involved in a certain biological system (identified motifs) and a set of protein sequences known not to be involved in that system (negative motifs), our method is able to identify motifs that are frequent in positive sequences while infrequent or absent in negative sequences. The identified motifs can then be used to identify motifs that are frequent in positive sequences while infrequent or absent in negative sequences.

**Keywords:** Clustering, Semi-supervised Clustering, Consensus-based Clustering, Machine Learning

#### 1 Introduction

Identifying motifs in biological sequences is an important challenge in biology. Proteins involved in the same biological system or physiological function (e.g., immune response, chemokinesis, secretion, signal transduction, ...) are subject to similar evolutionary and functional processes that can be observed in the protein sequence level. Finding motifs specific to proteins involved in the same process can help deciphering the determination of their function and thus be used in identifying new candidate proteins involved in important biological systems.

To our knowledge, all currently available methods search motifs in protein sequences at the amino acid level. However, when comparing motifs to sequences with high variability (e.g., proteins, ...), a better choice is to compare three-dimensional structures. In order to capture the three-dimensional structure of a protein, it is important to consider the actual sequence of the amino acid residues and not just the amino acid side chains. This is especially true for motifs that are highly conserved in the sequence, the amino acid side chains are highly conserved and the amino acid side chains are highly conserved in the sequence.

We propose a method that allows to identify discriminative motifs based on a comparison of amino acids and on the physico-chemical properties of those residues. Given a set of protein sequences known to be involved in a certain biological system (identified motifs) and a set of protein sequences known not to be involved in that system (negative motifs), our method is able to identify motifs that are frequent in positive sequences while infrequent or absent in negative sequences. The identified motifs can then be used to identify motifs that are frequent in positive sequences while infrequent or absent in negative sequences.

Bob L. Smith

Top-down

## Determining which authors are the same = clustering author occurrences

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

Bottom-up

Robert L. Smith

B.W. Smith

Robert L. W. Smith

Bob L. Smith

# ENTITY RESOLUTION

### Generalizing from Example Clusters

Paal Haug<sup>1,2</sup>, Celine Vens<sup>3</sup>, Bart Vervaeke<sup>3</sup>, and Hendrik Blockeel<sup>1,2</sup>

<sup>1</sup> KU Leuven, Department of Computer Science, Charelavenstraat 200A, 3000 Leuven, Belgium  
<sup>2</sup> Ghent University, Ghent University, Coupure links 653, Ghent, Belgium  
<sup>3</sup> Ghent University, Ghent University, Coupure links 653, Ghent, Belgium

**Abstract.** We consider the following problem: Given a set of data and one or more clusters of elements, find a clustering of the whole data set that is consistent with the given clusters. This is naturally a non-supervised clustering problem, but it differs from previously studied non-supervised clustering settings in significant ways. Unlike work that focuses on the task of clustering without the non-supervised clustering handle, this problem with We identify two reasons for this, which we tried to solve using existing methods, but we failed to do so. In this situation, and to overcome it, we investigate the latter in more detail and propose a new method that explicitly pushes against existing. Experimental results confirm that the new method generalizes much better. Several other problem variations have been proposed.

**Keywords.** Clustering, Semi-supervised Clustering, Consensus-based Clustering, Meta Learning

#### 1 Introduction

The task of clustering data is ubiquitous in knowledge discovery. Partitioned (or semi-supervised) clustering can be defined as the following task: given a dataset  $D$  partitioned in  $m$  non-overlapping clusters  $C_1, \dots, C_m$  and a set of instances  $S$  within the same cluster used to be similar, and instances in different clusters dissimilar. The notion of "similarity" is usually being depending on how this is defined, different solutions will be found. This is true especially for high-dimensional spaces, where different solutions may reveal different clusterings [1].

It is often easy for a user to define a good similarity measure. However, many may be able to give examples of instances that their system should, or should not, belong to the same cluster. The clustering system may use this information to understand better the notion of similarity that the user has in mind, and as a consequence produce a clustering of the data type. This type of clustering is called semi-supervised clustering, or consensus-based clustering, as the user gives partial information about the desired clustering of the data, but does not specify the similarity function used.

Most existing methods for semi-supervised clustering allow the user to provide a number of seed clusters and/or seed instances, indicating that they are of instances, whereas they should (not) be in the same cluster. Vens et al. [11] recently introduced a slightly different setting, called "semi-supervised clustering with example

Robert Smith

### Identifying Proteins Involved in Parasitism by Discovering Degenerated Motifs

Celine Vens<sup>1,2</sup>, Etienne Decaeste<sup>1</sup>, and Martin Naillie Broom<sup>3</sup>

<sup>1</sup>Department of Computer Science, Katholieke Universiteit Leuven, Charelavenstraat 200A, 3000 Leuven, Belgium  
<sup>2</sup> Ghent University, Ghent University, Coupure links 653, Ghent, Belgium  
<sup>3</sup>Department of Microbiology and Immunology, University of Alberta, Edmonton, Alberta, Canada

**Abstract.** Parasitism is a complex biological process that involves many different proteins. Identifying these proteins is a challenging task. In this paper, we propose a method for identifying proteins involved in parasitism by discovering degenerated motifs. This method is based on the discovery of motifs that are conserved across different species, but that have degenerated in the parasite. We show that this method is effective in identifying proteins involved in parasitism, and that it can be used to identify new proteins involved in parasitism.

**Keywords.** Protein identification, Parasitism, Motif discovery, Degenerated motifs

#### 1 Introduction

Identifying motifs in biological systems is an important challenge in biology. Proteins involved in the same biological system or physiological function (e.g., immune response, photosynthesis, signal transduction, ...), are subject to similar evolutionary and functional pressures that have as outcome that they are conserved in the same biological system. This conservation can be used to identify motifs that are conserved across different species, but that have degenerated in the parasite. We show that this method is effective in identifying proteins involved in parasitism, and that it can be used to identify new proteins involved in parasitism.

Robert L. Smith

### Annotating transposable elements in the genome using relational decision tree ensembles

Edouard P. Coste<sup>1</sup>, Leandro Schikapel<sup>1</sup>, Ricardo Carré<sup>1</sup>, Celine Vens<sup>2</sup>, Carlos N. Fariñas<sup>1</sup>, Claudia M. A. Gonçalves<sup>1</sup>, Jan Rasmussen<sup>3</sup>, and Hendrik Blockeel<sup>1,2</sup>

<sup>1</sup> Department of Computer Science, KU Leuven, Charelavenstraat 200A, 3000 Leuven, Belgium  
<sup>2</sup> Institute of Computer Mathematics and Computer Science, University of São Paulo, São Carlos, São Carlos, 13506-970 São Carlos, SP, Brazil  
<sup>3</sup> Department of Statistics, Applied Mathematics, and Computer Science, UNESP São Paulo State University, Avenida 24-A, 1513, 13066-900 São Carlos, SP, Brazil  
<sup>4</sup> Department of Biology, UNESP São Paulo State University, Charelavenstraat 200A, 3000 Leuven, Belgium

**Abstract.** Transposable elements (TEs) are DNA sequences that can change their location within the genome. They make up a large portion of the DNA in eukaryotic organisms and contribute to genetic diversity within and across species. Their transposing mechanisms may also affect the functionality of genes. Accurate annotation of TEs is important to study their evolutionary history and to understand their role in genome evolution. We introduce a framework for annotating TEs which is based on relational decision tree learning. It allows to naturally represent the structural and biological properties of TEs. Furthermore, it also allows the integration of background knowledge and benefits from the interpretability of decision trees. Preliminary experiments show that our method outperforms two state-of-the-art systems for TE annotation.

**Keywords.** relational decision trees, hidden Markov models, genome annotation, transposable elements

#### 1 Introduction

Transposable elements (TEs) are DNA sequences that can change their location within the genome. They make up a large portion of the DNA in eukaryotic organisms and contribute to genetic diversity within and across species. Furthermore, their transposing mechanisms increase the size of the genome and affect the functionality of genes. Accurate annotation of TEs is important to study their evolutionary history and to understand their role in genome evolution. We introduce a framework for annotating TEs which is based on relational decision tree learning. It allows to naturally represent the structural and biological properties of TEs. Furthermore, it also allows the integration of background knowledge and benefits from the interpretability of decision trees. Preliminary experiments show that our method outperforms two state-of-the-art systems for TE annotation.

Bob Smith

### Top-down clustering for protein subfamily identification

Edouard P. Coste<sup>1</sup>, Celine Vens<sup>2</sup>, and Hendrik Blockeel<sup>1,2</sup>

<sup>1</sup>Department of Computer Science, KU Leuven, Belgium  
<sup>2</sup>Institute of Computer Mathematics and Computer Science, University of São Paulo, São Carlos, São Carlos, 13506-970 São Carlos, SP, Brazil

**Abstract.** Protein subfamily identification is a challenging task. In this paper, we propose a method for identifying protein subfamilies by top-down clustering. This method is based on the discovery of motifs that are conserved across different species, but that have degenerated in the parasite. We show that this method is effective in identifying protein subfamilies, and that it can be used to identify new protein subfamilies.

**Keywords.** Protein subfamily identification, Clustering, Motif discovery

#### 1 Introduction

Protein subfamily identification is a challenging task. In this paper, we propose a method for identifying protein subfamilies by top-down clustering. This method is based on the discovery of motifs that are conserved across different species, but that have degenerated in the parasite. We show that this method is effective in identifying protein subfamilies, and that it can be used to identify new protein subfamilies.

R. W. Smith

### Decision Trees for Hierarchical Multi-label Classification

Celine Vens<sup>1</sup>, Jan Rasmussen<sup>2</sup>, and Hendrik Blockeel<sup>1</sup>

<sup>1</sup>Department of Computer Science, Katholieke Universiteit Leuven, Charelavenstraat 200A, 3000 Leuven, Belgium  
<sup>2</sup>Department of Knowledge Technology, Jozef Stefan Institute, Jamnikova 39, 1000 Ljubljana, Slovenia

**Abstract.** Hierarchical multi-label classification (HMLC) is a variant of classification where instances may belong to multiple classes at the same time and these classes are organized in a hierarchy. This article presents several approaches to the problem of decision trees for HMLC, as well as an empirical study of their use in functional genomics. We compare learning a single HMLC tree (which makes predictions for all classes together) to two approaches that learn a set of regular classification trees (one for each class). The first approach defines an independent single-label classification task for each class. The second approach defines a regularized multi-label classification task for each class. We show that the second approach is more effective than the first approach in terms of accuracy and interpretability. We also show that the second approach is more effective than the first approach in terms of accuracy and interpretability.

**Keywords.** Hierarchical multi-label classification, Decision trees, Functional genomics

#### 1 Introduction

Hierarchical multi-label classification (HMLC) is a variant of classification where instances may belong to multiple classes at the same time and these classes are organized in a hierarchy. This article presents several approaches to the problem of decision trees for HMLC, as well as an empirical study of their use in functional genomics. We compare learning a single HMLC tree (which makes predictions for all classes together) to two approaches that learn a set of regular classification trees (one for each class). The first approach defines an independent single-label classification task for each class. The second approach defines a regularized multi-label classification task for each class. We show that the second approach is more effective than the first approach in terms of accuracy and interpretability. We also show that the second approach is more effective than the first approach in terms of accuracy and interpretability.

Robert L. W. Smith

### Semi-supervised Clustering with Example Clusters

Celine Vens<sup>1,2</sup>, Bart Vervaeke<sup>3</sup>, and Hendrik Blockeel<sup>1,2</sup>

<sup>1</sup>Department of Computer Science, KU Leuven, Belgium  
<sup>2</sup>Institute of Computer Mathematics and Computer Science, University of São Paulo, São Carlos, São Carlos, 13506-970 São Carlos, SP, Brazil  
<sup>3</sup>Department of Statistics, Applied Mathematics, and Computer Science, UNESP São Paulo State University, Avenida 24-A, 1513, 13066-900 São Carlos, SP, Brazil

**Abstract.** We consider the following problem: Given a set of data and one or more clusters of elements, find a clustering of the whole data set that is consistent with the given clusters. This is naturally a non-supervised clustering problem, but it differs from previously studied non-supervised clustering settings in significant ways. Unlike work that focuses on the task of clustering without the non-supervised clustering handle, this problem with We identify two reasons for this, which we tried to solve using existing methods, but we failed to do so. In this situation, and to overcome it, we investigate the latter in more detail and propose a new method that explicitly pushes against existing. Experimental results confirm that the new method generalizes much better. Several other problem variations have been proposed.

**Keywords.** Clustering, Semi-supervised Clustering, Consensus-based Clustering, Meta Learning

#### 1 Introduction

The task of clustering data is ubiquitous in knowledge discovery. Partitioned (or semi-supervised) clustering can be defined as the following task: given a dataset  $D$  partitioned in  $m$  non-overlapping clusters  $C_1, \dots, C_m$  and a set of instances  $S$  within the same cluster used to be similar, and instances in different clusters dissimilar. The notion of "similarity" is usually being depending on how this is defined, different solutions will be found. This is true especially for high-dimensional spaces, where different solutions may reveal different clusterings [1].

Robert Smith

### Learning Kernels with Random Forests

Celine Vens<sup>1,2</sup> and Fabrizio Costa<sup>3</sup>

<sup>1</sup> KU Leuven, Charelavenstraat 200A, 3000 Leuven, Belgium  
<sup>2</sup> Ghent University, Ghent University, Coupure links 653, Ghent, Belgium  
<sup>3</sup> Ghent University, Ghent University, Coupure links 653, Ghent, Belgium

**Abstract.** We propose a simple and effective strategy to learn kernel functions from ensembles of random decision trees. Unlike previous work, the proposed method is not limited to the classification setting, as it is possible to learn a kernel from a forest of random decision trees. We show that this method is effective in learning kernels, and that it can be used to identify new kernels.

**Keywords.** Kernel learning, Random forests, Decision trees

#### 1 Introduction

The problem of determining a suitable metric (kernel) for a given problem has been receiving increasing attention in the Machine Learning community. One standard technique to learn a kernel is to use a set of positive semi-definite (PSD) matrices, representing the pairwise similarities between the data points. This is often done by using a set of random decision trees, which are used to learn a kernel from a forest of random decision trees. We show that this method is effective in learning kernels, and that it can be used to identify new kernels.

Bob L. Smith

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

# ENTITY RESOLUTION

### Generalizing from Example Clusters

Paol Van<sup>1,2</sup>, Celine Vens<sup>1</sup>, Bart Verschuyl<sup>1</sup>, and Hendrik Blockeel<sup>1,3,4</sup>

<sup>1</sup> KU Leuven, Department of Computer Science, Charelathouwen 200A, 3001 Leuven, Belgium  
<sup>2</sup> Ecole des Mines de Douai, Douai, France  
<sup>3</sup> Ghent University of Applied Sciences Ghent College of Business, Nuth Building 1,2513 CA Leuven, The Netherlands

**Abstract.** We consider the following problem: Given a set of data and one or more clusters of clusters, find a clustering of the whole data set that is consistent with the given clusters. This is naturally a non-supervised clustering problem, but it differs from previously studied non-supervised clustering settings in significant ways. First, we do not know the class labels of the clusters within the non-supervised clustering handle this problem well. We identify two reasons for this, which we tried to solve using various methods, but we could not find this solution, and in certain heuristics. We investigate the latter in more detail and propose a new method that explicitly pushes against overfitting. Empirical results confirm that the new method generalizes much better. Several other problems identified here remain open.

**Keywords.** Clustering, Semi-supervised Clustering, Consistent-based Clustering, Metric Learning

#### 1 Introduction

The task of clustering data is ubiquitous in knowledge discovery. Particular for non-hereditary clustering can be defined as the following task: given a dataset  $D$  partitioned into  $m$  clusters ( $m > 1$ ) and  $n$  instances within the same cluster need to be similar, and instances in different clusters dissimilar. The notion of "similarity" is crucial here depending on how this is defined, different solutions will be found. This is true especially for high-dimensional spaces, where different solutions may reveal different clusterings [1].

It is not always easy for a user to define a good similarity measure. However, even when we are able to give examples of instances that in their opinion should, or should not, belong to the same cluster. The clustering criterion may then be automatically understood based on the notion of similarity that the user has in mind, and as a consequence produce a clustering of the data that is similar to the type of clustering that the user would expect, or even better clustering, as the user gives partial information about the desired clustering by giving clusters that the user has in mind.

Most existing methods for semi-supervised clustering allow the user to provide a number of seed clusters and/or cluster labels, indicating the type of instances that they should (not) be in the same cluster. Vens et al. [11] recently introduced a slightly different setting, called "semi-supervised clustering with example clusters".

Robert Smith

### Annotating transposable elements in the genome using relational decision tree ensembles

Edmaris P. Costa<sup>1</sup>, Leandro Schepers<sup>1</sup>, Ricardo Cruz<sup>1</sup>, Celine Vens<sup>1</sup>, Carlos N. Farias<sup>1</sup>, Claudia M.A. Gonçalves<sup>1</sup>, Jan Rimmé<sup>2</sup>, and Hendrik Blockeel<sup>1</sup>

<sup>1</sup> Department of Computer Science, KU Leuven, Charelathouwen 200A, 3001 Leuven, Belgium  
<sup>2</sup> Institute of Clinical Mathematics & Computational, Universidade de São Paulo, Caixa de São Carlos, 13060-970 São Carlos, SP, Brazil  
<sup>3</sup> Department of Statistics, Applied Mathematics, and Computer Science, UNESP São Paulo State University, Avenida 24-A, 1515, 13060-900 São Carlos, SP, Brazil  
<sup>4</sup> Department of Statistics, UNESP São Paulo State University, Campinas, Campinas, Campinas, 13081-087 São Carlos, SP, Brazil

**Abstract.** Transposable elements (TEs) are DNA sequences that can change their location within the genome. They make up a large portion of the DNA in eukaryotic organisms and contribute to genetic diversity within and across species. Their transposing mechanisms may also affect the functionality of genes. Accurate annotation of TEs is important to any genome analysis. We investigate a framework for annotating TEs which uses relational decision tree learning. It allows to naturally represent the structural data and biological processes involving TEs. Furthermore, it also allows the integration of background knowledge and benefits from the interpretability of decision trees. Preliminary experiments show that our method outperforms two state-of-the-art systems for TE annotation.

**Keywords.** relational decision trees, hidden Markov models, genome annotation, transposable elements

#### 1 Introduction

Transposable elements (TEs) are DNA sequences that can change their location within the genome. They make up a large portion of the DNA in eukaryotic organisms and contribute to genetic diversity within and across species. Furthermore, their transposing mechanisms increase the size of the genome over time and affect the functionality of genes. Accurate annotation of TEs, together with the development of transferable models explaining these annotations, is an important step towards understanding their effects on genes and their role in genome evolution [1].

Currently, annotation of TEs involves a fair amount of manual labor. Automated methods exist that process DNA & candidate TEs, but human annotations

Bob Smith

### Semi-supervised Clustering with Example Clusters

Celine Vens<sup>1,2</sup>, Bart Verschuyl<sup>1</sup>, and Hendrik Blockeel<sup>1,3</sup>

<sup>1</sup> Department of Computer Science, KU Leuven, Belgium  
<sup>2</sup> Ecole des Mines de Douai, Douai, France  
<sup>3</sup> Ghent University of Applied Sciences Ghent College of Business, Nuth Building 1,2513 CA Leuven, The Netherlands

**Keywords.** Clustering, Semi-supervised Clustering, Consistent-based Clustering, Metric Learning

**Abstract.** We consider the following problem: Given a set of data and one or more clusters of clusters, find a clustering of the whole data set that is consistent with the given clusters. This is naturally a non-supervised clustering problem, but it differs from previously studied non-supervised clustering settings in significant ways. First, we do not know the class labels of the clusters within the non-supervised clustering handle this problem well. We identify two reasons for this, which we tried to solve using various methods, but we could not find this solution, and in certain heuristics. We investigate the latter in more detail and propose a new method that explicitly pushes against overfitting. Empirical results confirm that the new method generalizes much better. Several other problems identified here remain open.

#### 1 INTRODUCTION

We consider the task of semi-supervised clustering. The problem is to partition a dataset into  $m$  clusters ( $m > 1$ ) and  $n$  instances within the same cluster need to be similar, and instances in different clusters dissimilar. The notion of "similarity" is crucial here depending on how this is defined, different solutions will be found. This is true especially for high-dimensional spaces, where different solutions may reveal different clusterings [1].

It is not always easy for a user to define a good similarity measure. However, even when we are able to give examples of instances that in their opinion should, or should not, belong to the same cluster. The clustering criterion may then be automatically understood based on the notion of similarity that the user has in mind, and as a consequence produce a clustering of the data that is similar to the type of clustering that the user would expect, or even better clustering, as the user gives partial information about the desired clustering by giving clusters that the user has in mind.

Most existing methods for semi-supervised clustering allow the user to provide a number of seed clusters and/or cluster labels, indicating the type of instances that they should (not) be in the same cluster. Vens et al. [11] recently introduced a slightly different setting, called "semi-supervised clustering with example clusters".

R. W. Smith

### Learning Kernels with Random Forests

Celine Vens<sup>1,2</sup> and Frédéric Cresto<sup>1,3</sup>

<sup>1</sup> KU Leuven, Charelathouwen 200A, 3001 Leuven, Belgium  
<sup>2</sup> INRIA, 400 route des Chappes, BP 101, 08001 Sillans Ardennes, France  
<sup>3</sup> Ecole des Mines de Douai, Douai, France

**Abstract.** We propose a simple and effective strategy to learn kernel functions from examples of random decision trees. Unlike previous work, the proposed method is not limited to the construction of a single  $k$ -positive kernel function (e.g., limited to a linear combination of kernels belonging to a specific family) but allows for a broad transformation of a pre-specified kernel function space. Moreover, exploiting the properties of random forests, we are able to build a kernel with learning rules and criteria data and task dependent. Such matrices can be used by any well-known kernel-based learning algorithms. Finally, we provide an extensive empirical study of the properties of the learned kernels based on real and artificial datasets.

#### 1 Introduction

The problem of determining a suitable metric space tailored for a given predictive task has been receiving increasing attention in the Machine Learning community. One direction followed by available, centered techniques uses a variety of procedures can be defined, ranging from the nearest neighbor algorithm [1] for supervised classification, to clustering algorithms like  $k$ -means [2] for unsupervised tasks or multi-dimensional scaling [3] for visualization or preprocessing.

In the last decade, due to the recent progress in performance and theoretical guarantees offered by Support Vector Machines, kernel-based methods have become mainstream. In fact, many tasks can be formulated in learning this way, rather than the distance function, although the two tasks are intrinsically related or may even share some features in some of them.

The problem of learning the kernel function has therefore become of interest in its own right. It gives the user more freedom [4] and facilitates methods [5] results, one can hope to improve learning only when using prior information on the task (partially known) rather than to be generic. Other kernel-based methods seem to be more effective only through the learned function, the methods based on the prior bias and the prior information can be seen from the kernel function [6]. In [4] the authors propose to use the notion of logical adaptation to measure the quality of the kernel function. In practice, a suitable method for learning kernels that agree on the partitioning of the instances according to the target variable.

Although a variety of methods have been developed for this task, many of the proposed techniques are applicable only to transductive settings [7], [8].

Robert Smith

### Identifying Proteins Involved in Parasitism by Discovering Degenerated Motifs

Celine Vens<sup>1,2</sup>, Hsuan-Daun<sup>3</sup>, and Martin-Naila Bamer<sup>1</sup>

<sup>1</sup> Department of Computer Science, KU Leuven, Belgium  
<sup>2</sup> Ghent University of Applied Sciences Ghent College of Business, Nuth Building 1,2513 CA Leuven, The Netherlands  
<sup>3</sup> Institute of Parasitology, University of Würzburg, Würzburg, Germany

**Abstract.** Parasitism is an important biological process. Identifying proteins involved in this process is crucial for understanding the host-parasite interaction. In this paper, we propose a method for identifying proteins involved in parasitism by discovering degenerated motifs. We use a combination of machine learning and biological domain knowledge to identify proteins involved in parasitism. Our method is based on the discovery of degenerated motifs, which are motifs that have been modified over time. We use a combination of machine learning and biological domain knowledge to identify proteins involved in parasitism. Our method is based on the discovery of degenerated motifs, which are motifs that have been modified over time.

#### 1 Introduction

Identifying motifs in biological systems is an important challenge in biology. Parasitism involves the interaction between a parasite and its host, which is a complex process. In this paper, we propose a method for identifying proteins involved in parasitism by discovering degenerated motifs. We use a combination of machine learning and biological domain knowledge to identify proteins involved in parasitism. Our method is based on the discovery of degenerated motifs, which are motifs that have been modified over time.

Robert L. Smith

### Identifying Discriminative Classification Based Motifs in Biological Sequences

Celine Vens<sup>1,2</sup>, Marie-Nolée Rosso<sup>1</sup>, and Elvira G.J. Carachi<sup>1</sup>

<sup>1</sup> Ghent University of Applied Sciences Ghent College of Business, Nuth Building 1,2513 CA Leuven, Belgium  
<sup>2</sup> Department of Computer Science, KU Leuven, Belgium  
<sup>3</sup> Institute of Parasitology, University of Würzburg, Würzburg, Germany

**Abstract.** Identifying discriminative motifs in biological sequences is a crucial task for understanding the underlying biological processes. In this paper, we propose a method for identifying discriminative motifs in biological sequences. We use a combination of machine learning and biological domain knowledge to identify discriminative motifs. Our method is based on the discovery of discriminative motifs, which are motifs that have been modified over time.

#### 1 INTRODUCTION

Identifying motifs in biological sequences is a crucial task for understanding the underlying biological processes. In this paper, we propose a method for identifying discriminative motifs in biological sequences. We use a combination of machine learning and biological domain knowledge to identify discriminative motifs. Our method is based on the discovery of discriminative motifs, which are motifs that have been modified over time.

B.W. Smith

### Top-down clustering for protein subfamily identification

Edmaris P. Costa<sup>1</sup>, Celine Vens<sup>1</sup>, and Hendrik Blockeel<sup>1</sup>

<sup>1</sup> Department of Computer Science, KU Leuven, Belgium  
<sup>2</sup> Ghent University of Applied Sciences Ghent College of Business, Nuth Building 1,2513 CA Leuven, Belgium

**Abstract.** Protein subfamily identification is a crucial task for understanding the underlying biological processes. In this paper, we propose a method for identifying protein subfamilies using top-down clustering. We use a combination of machine learning and biological domain knowledge to identify protein subfamilies. Our method is based on the discovery of protein subfamilies, which are subfamilies that have been modified over time.

#### 1 INTRODUCTION

Protein subfamily identification is a crucial task for understanding the underlying biological processes. In this paper, we propose a method for identifying protein subfamilies using top-down clustering. We use a combination of machine learning and biological domain knowledge to identify protein subfamilies. Our method is based on the discovery of protein subfamilies, which are subfamilies that have been modified over time.

Robert L. W. Smith

### Decision Trees for Hierarchical Multi-label Classification

Celine Vens<sup>1,2</sup>, Jan Rimmé<sup>3</sup>, and Hendrik Blockeel<sup>1,4</sup>

<sup>1</sup> Department of Computer Science, KU Leuven, Belgium  
<sup>2</sup> Ghent University of Applied Sciences Ghent College of Business, Nuth Building 1,2513 CA Leuven, Belgium  
<sup>3</sup> Institute of Parasitology, University of Würzburg, Würzburg, Germany  
<sup>4</sup> Institute of Clinical Mathematics & Computational, Universidade de São Paulo, Caixa de São Carlos, 13060-970 São Carlos, SP, Brazil

**Abstract.** Hierarchical multi-label classification (HMLC) is a natural extension of multi-label classification. In this paper, we propose a method for identifying protein subfamilies using top-down clustering. We use a combination of machine learning and biological domain knowledge to identify protein subfamilies. Our method is based on the discovery of protein subfamilies, which are subfamilies that have been modified over time.

#### 1 Introduction

Hierarchical multi-label classification (HMLC) is a natural extension of multi-label classification. In this paper, we propose a method for identifying protein subfamilies using top-down clustering. We use a combination of machine learning and biological domain knowledge to identify protein subfamilies. Our method is based on the discovery of protein subfamilies, which are subfamilies that have been modified over time.

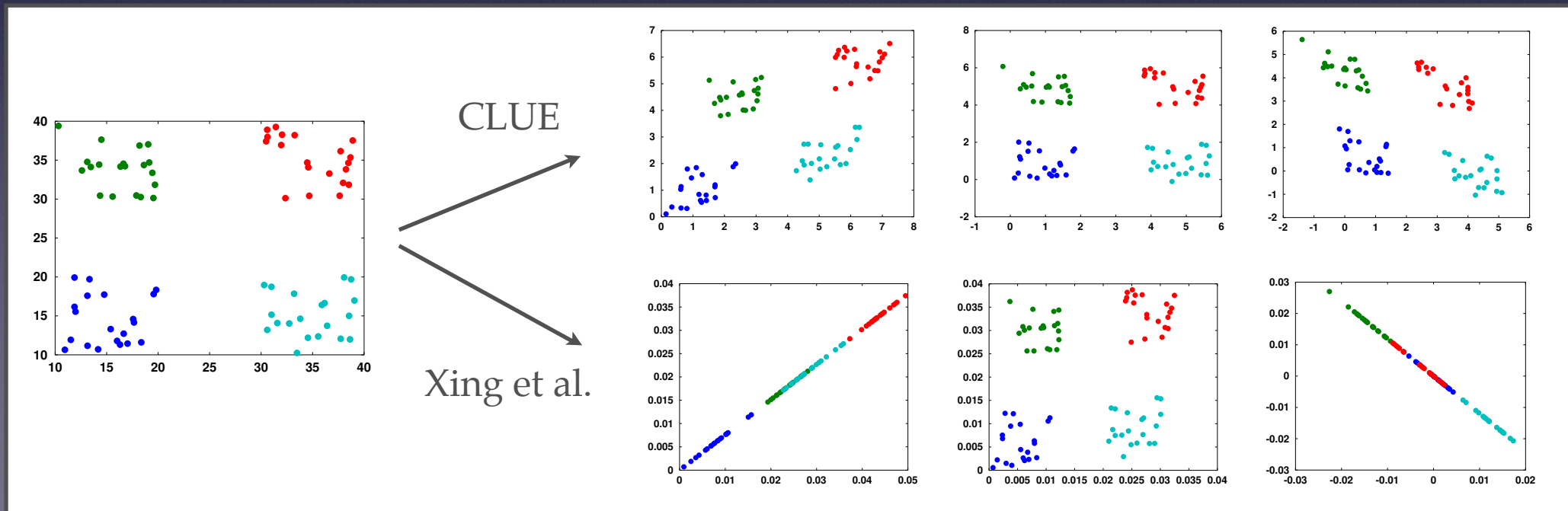
Bob L. Smith



Rest if the data clustered "in a similar way"

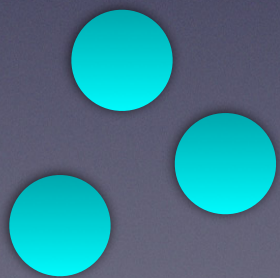
# Generalizing from example clusters

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

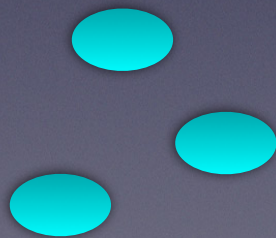


# Choosing the clustering approach

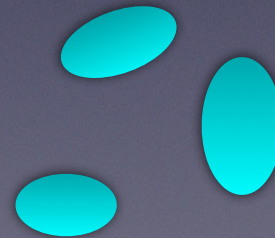
- Most work on constraint-based clustering adapts one approach to incorporate constraints
- But different approaches have very different biases!
- Use constraints to select the most suitable clustering approach?
- Ongoing work (A.Adam et al.)



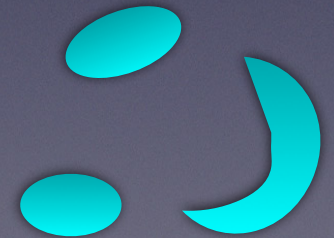
k-means



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

$\#{t: \text{FrequentItemset} \subseteq t}$   
 $\geq \text{Freq.}$

# 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)

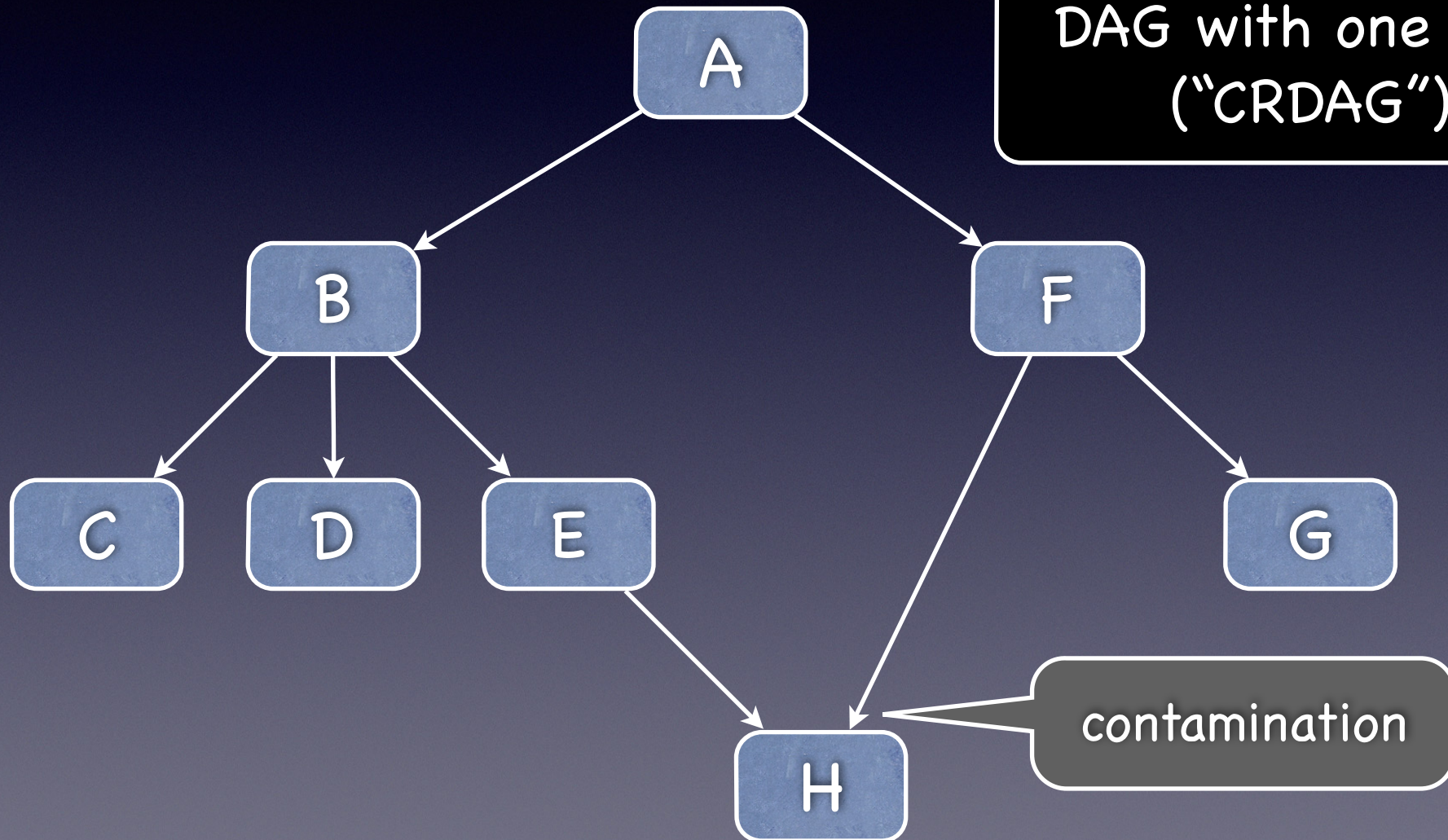
# Stemmatology



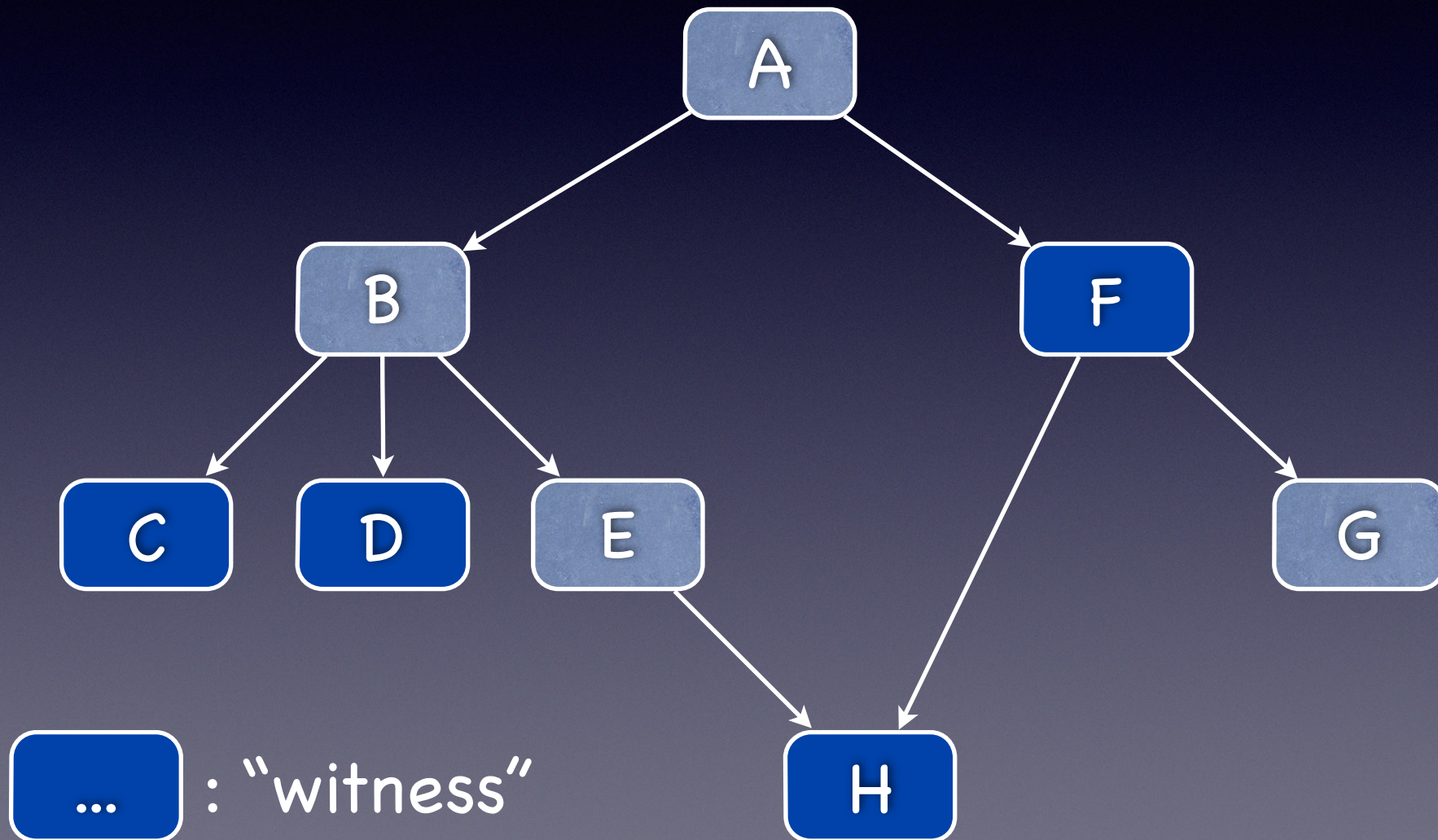
- Subfield of philology
- Monks copied manuscripts manually, made changes -> “evolution” of the story
- Study relationships between surviving variants of the story (e.g., to reconstruct a lost original)
- Stemma = “family tree” of a set of manuscripts
- Somewhat similar to phylogenetic trees in bioinformatics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

# Stemma

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



# Stemma with witnesses



# 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	...

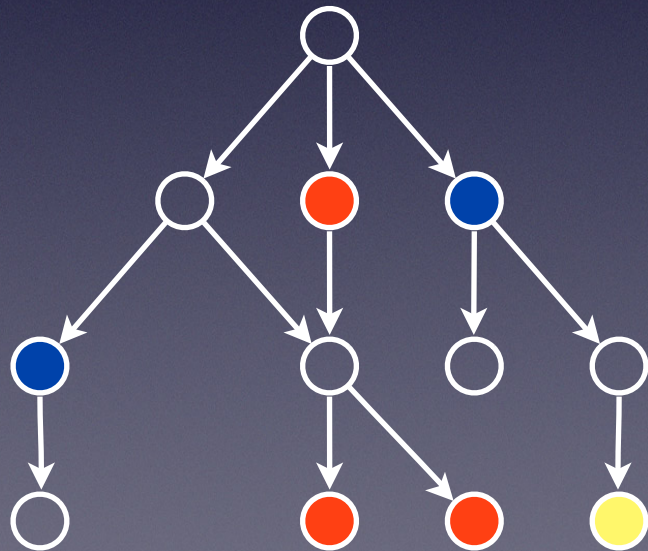
# The task

- 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?)*

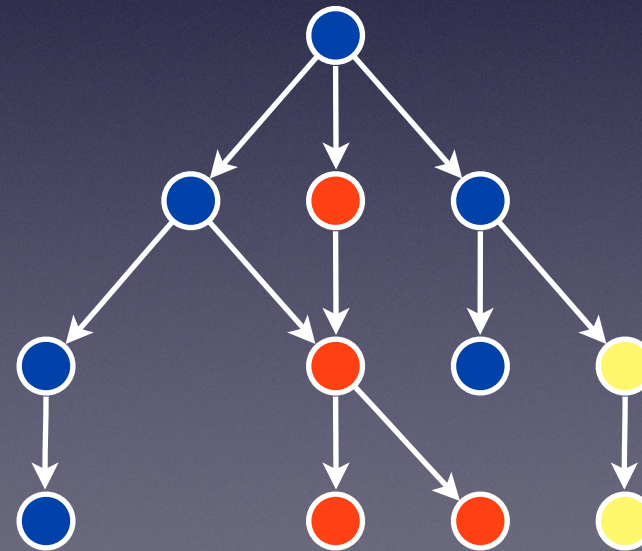


# DAG formulation

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



given



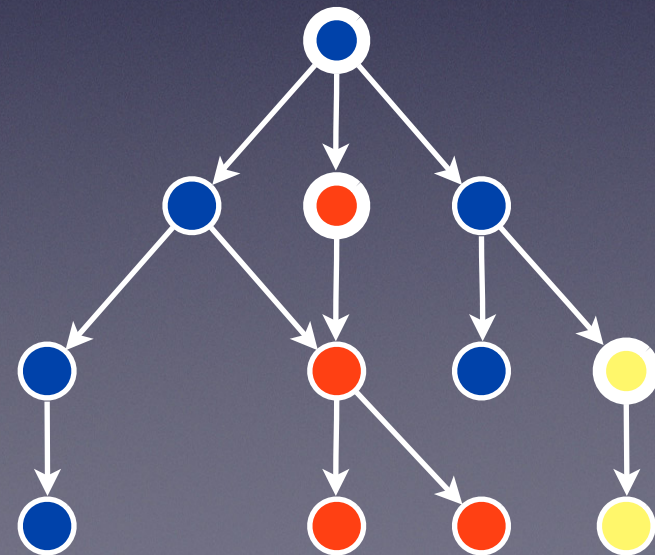
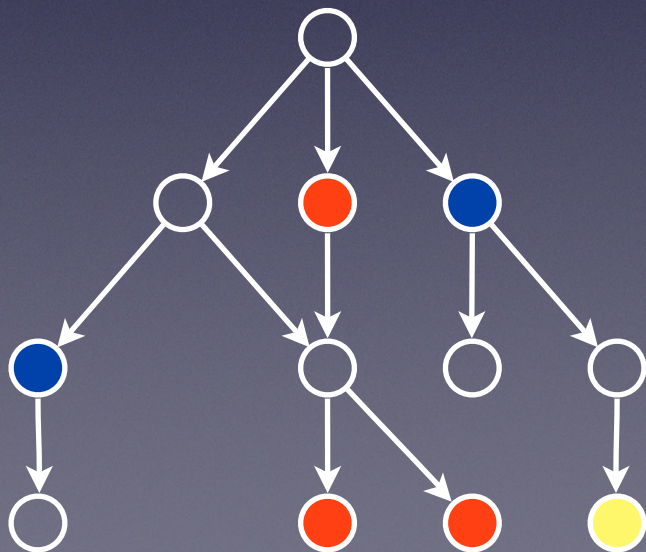
solution

# 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

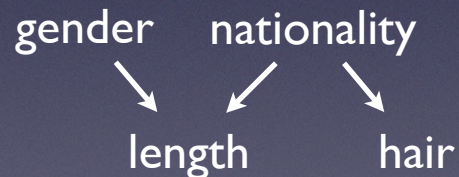
# Example

```
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
```

# Example

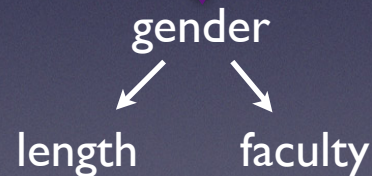
ESTIMATE MEAN length  
FROM student  
WHERE faculty='engineering'  
ENSURING CONF=0.95 AND WIDTH <= 5

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 :

$$0.9 * (\text{MEAN length FROM student WHERE gender='male'}) + 0.1 * (\text{MEAN length FROM student WHERE gender='female'})$$

# 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  $\neq$  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 action  $A$  is a set of externally induced changes  $a_i \rightarrow a_i'$  of attribute values (“interventions”)
- The cost of an action is the sum of the costs of the changes:  $C(A) = \sum_{(a_i \rightarrow a_i') \in A} C_i(a_i, a_i')$

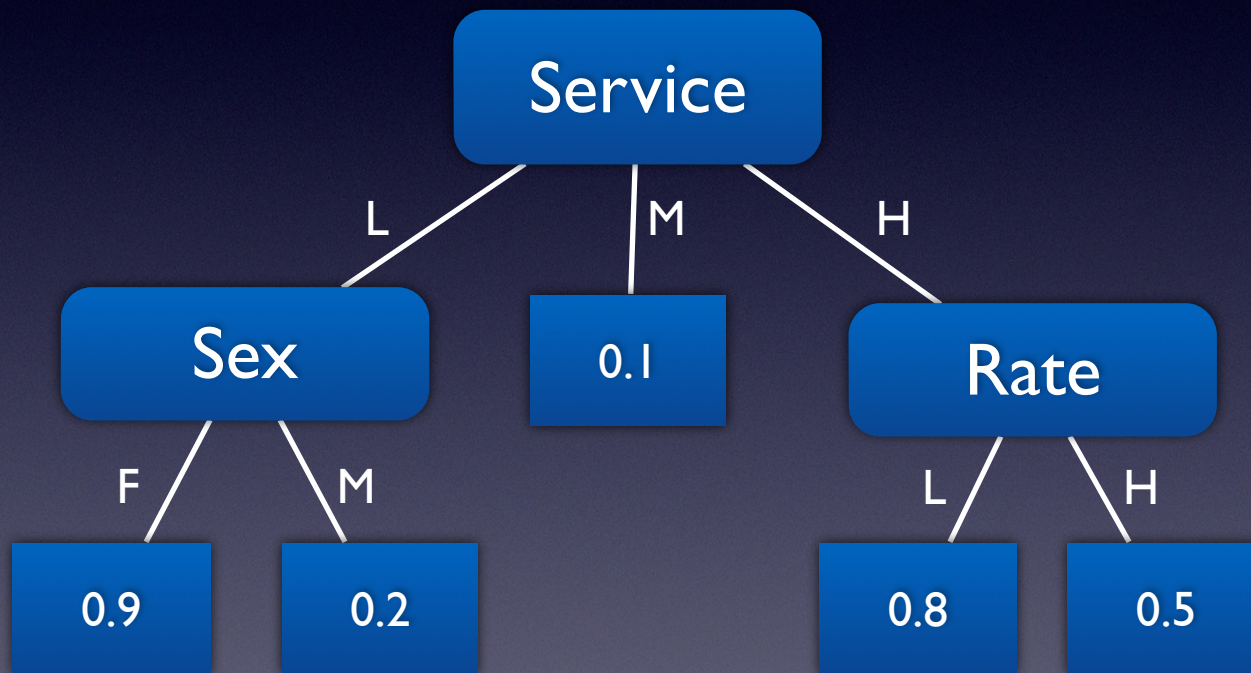
- Changing one attribute may have an effect on other attributes or on the target
- Let  $t$  be the original (pre-action) value of the target, and  $t'$  the new value
- The profit of an action  $A$  is  $P(t')-P(t)$
- The net profit of  $A$  is  $NP(A)=P(t')-P(t)-C(A)$ 
  - this assumes  $t'$  is known
- 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  $D \subseteq D_1 \times \dots \times D_n \times D_T$
- 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)



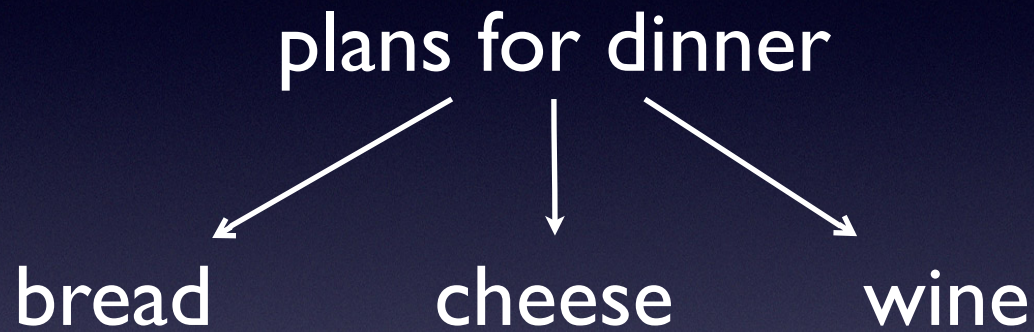
# Is it straightforward?

*IF bread & cheese THEN wine*

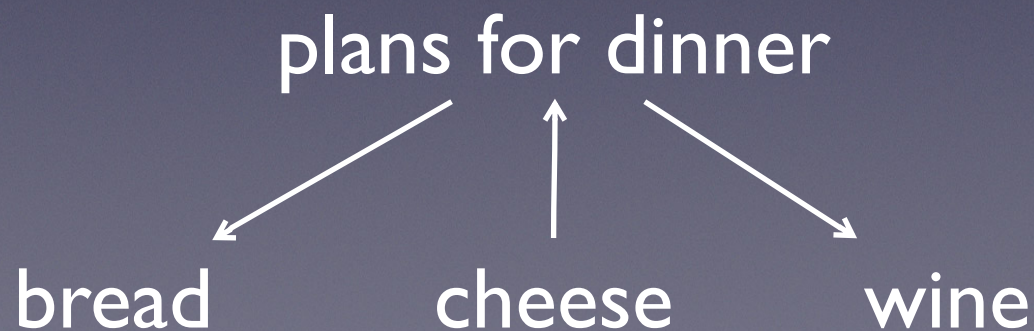
- Suppose many people buy bread, but few buy cheese; and we want to sell more wine (high profit). Can we achieve that by giving them cheese for free?

# 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  $A_j, j \neq i$



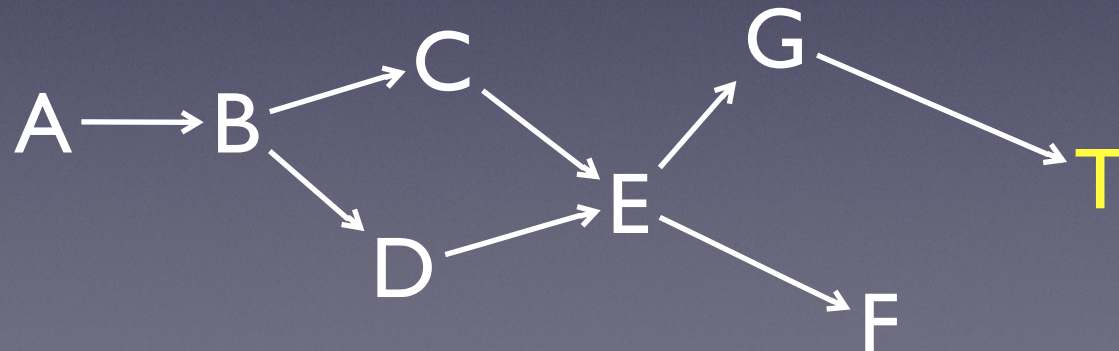
Setting 1:  
dinner plans affect  
bought products



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 I: 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  $\neq$  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 → B → C

A ← B ← C

A ← B → C

A → B ← 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

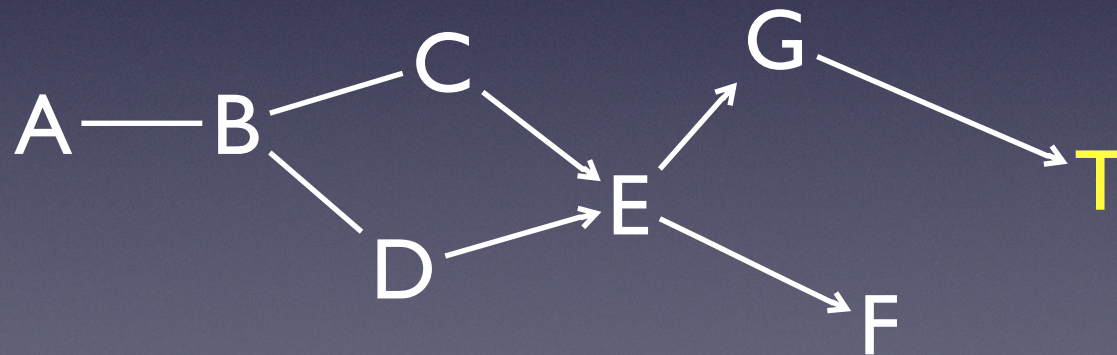


# Causality among 2 variables

- 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, ...)

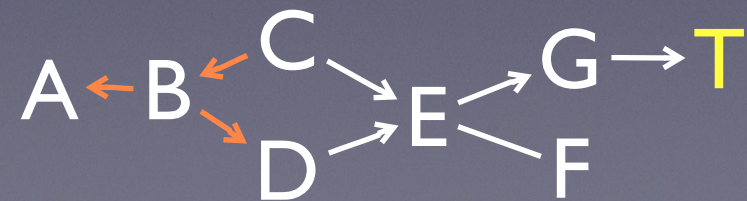
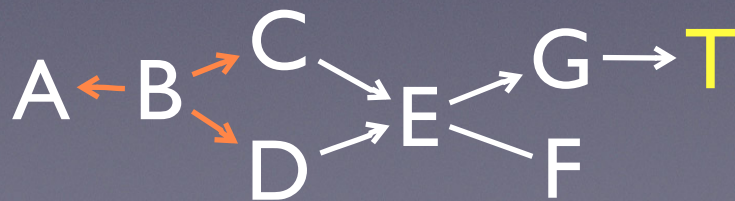
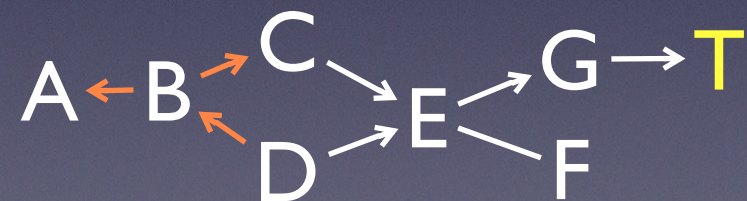
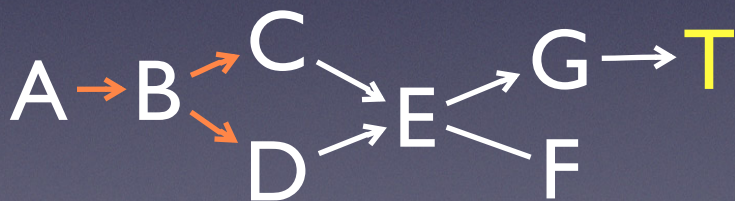
# Partial causal networks

- 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