KU LEUVEN


# Interactive Pattern Mining

## MINE, INTERACT, LEARN, REPEAT

Vladimir Dzyuba

DTAI Seminar

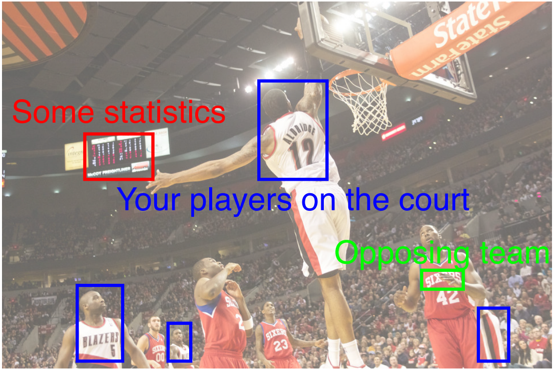
October 6, 2014

# Outline

1. Introduction
   - Exploratory data analysis (EDA)
   - Pattern mining
   - Why interactive?
2. Our approach to interactive pattern mining
3. Conclusions & future work

# Exploratory data analysis: Case study

Imagine you are a basketball coach
You have the data about your team's games



# Exploratory data analysis: Case study



Imagine you are a basketball coach
You have the data about your team's games

| $aldridge$ | $felton$ | $opponent$ | $def\_reb$ | $opp\_def\_reb$ | $off\_rtg$ |
|---|---|---|---|---|---|
| $In$ | $In$ | $PHI$ | $High$ | $Low$ | $High$ |
| $In$ | $In$ | $PHI$ | $High$ | $High$ | $Low$ |
| $In$ | $Out$ | $PHI$ | $High$ | $Low$ | $High$ |
| $Out$ | $In$ | $PHI$ | $Low$ | $High$ | $High$ |
| | | | $\ldots$ | | |
| $Out$ | $Out$ | $PHI$ | $Low$ | $High$ | $Low$ |

# Pattern mining



**Discovering *and describing* local structure in data**

## Given

- Dataset $\mathcal{D}$
- Pattern language $\mathcal{L}$
  *How to describe the structure*
- Interestingness predicate $q$
  *Which structure we are interested in*

## Find

Patterns $p \in \mathcal{L}$ such that $q\left(p\right)$ is true

All *interesting* patterns

# Itemset mining



**Discovering and describing local structure *in binary data***

## Data

Sets of items (*binary matrix*)

## Pattern language

Sets of items

## Interestingness predicate

$Frequency(p) > \theta$

Frequently co-occuring combinations of items

# Itemsets in basketball data



## Players often playing together

| $aldridge$ | $crawford$ | $felton$ | $hickson$ | $matthews$ |
|---|---|---|---|---|
| $In$ | $In$ | $In$ | $Out$ | $Out$ |
| $In$ | $Out$ | $In$ | $Out$ | $Out$ |
| $In$ | $Out$ | $Out$ | $In$ | $Out$ |
| $In$ | $Out$ | $In$ | $Out$ | $In$ |
| $In$ | $In$ | $In$ | $In$ | $Out$ |

# Itemsets in basketball data



## Players often playing together

| $aldridge$ | $crawford$ | $felton$ | $hickson$ | $matthews$ |
|---|---|---|---|---|
| $In$ | $In$ | $In$ | $Out$ | $Out$ |
| $In$ | $Out$ | $In$ | $Out$ | $Out$ |
| $In$ | $Out$ | $Out$ | $In$ | $Out$ |
| $In$ | $Out$ | $In$ | $Out$ | $In$ |
| $In$ | $In$ | $In$ | $In$ | $Out$ |

# Itemsets in basketball data



## Players often playing together

| $aldridge$ | $crawford$ | $felton$ | $hickson$ | $matthews$ |
|---|---|---|---|---|
| $In$ | $In$ | $In$ | | |
| $In$ | | $In$ | | |
| $In$ | | | $In$ | |
| $In$ | | $In$ | | $In$ |
| $In$ | $In$ | $In$ | $In$ | |

# Itemsets in basketball data



## Players often playing together

| $aldridge$ | $crawford$ | $felton$ | $hickson$ | $matthews$ |
|---|---|---|---|---|
| $In$ | $In$ | $In$ | | |
| $In$ | | $In$ | | |
| $In$ | | | $In$ | |
| $In$ | | $In$ | | $In$ |
| $In$ | $In$ | $In$ | $In$ | |

$$\{aldridge,\ felton\} = 4/5 = 80\%$$

$$\{aldridge,\ hickson\} = 2/5 = 40\%$$

$$\{aldridge,\ felton,\ hickson\} = 1/5 = 20\%$$

# Subgroup discovery 

**Discovering and describing local structure *in labelled data***

## Data

Single table, binary *target attribute* $A_T$

## Pattern language

Logical predicates

## Interestingness predicate

For example, $WRAcc\,(p) \approx |p| \times (|p^+| - |\mathcal{D}^+|)$

Differences between positives and negatives

# Subgroup discovery in basketball data



## Target is *Offensive Rating* – it should be *High*

| $aldridge$ | $crawford$ | $felton$ | $hickson$ | $matthews$ | $off\_rtg$ |
|---|---|---|---|---|---|
| $In$ | $In$ | $In$ | $Out$ | $Out$ | $High$ |
| $In$ | $Out$ | $In$ | $Out$ | $Out$ | $Low$ |
| $In$ | $Out$ | $Out$ | $In$ | $Out$ | $High$ |
| $In$ | $Out$ | $In$ | $Out$ | $In$ | $Low$ |
| $In$ | $In$ | $In$ | $In$ | $Out$ | $High$ |

$$|p| \quad \times \quad (|p^+| - |\mathcal{D}^+|)$$

$$\{matthews = Out\} = \quad 4/5 \quad \times \quad (3/4 - 3/5) = 0.09$$

$$\{matthews = Out = \quad 2/5 \quad \times \quad (2/2 - 3/5) = 0.16$$
$$\wedge\ hickson = In\}$$

# EDA is an informal task



**Tell me *something interesting* about my data**

Different from:

- Machine learning
  *We are not predicting the target*
- Hypothesis testing
  *We do not know what we are looking for*

Automated *generation* of hypotheses that the data would support

# …In the real world



Top $5$ subgroups discovered by a state-of-the-art algorithm

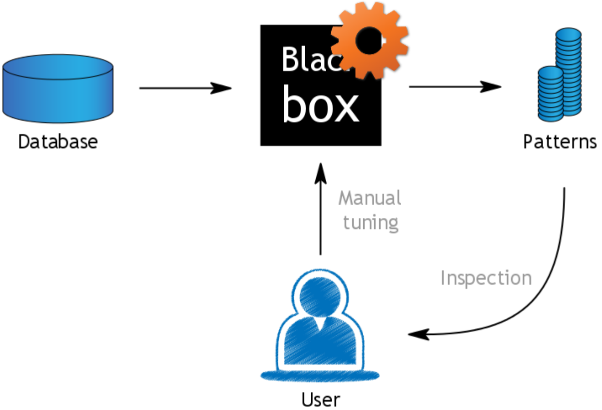
| **Description** | **Size** | **WRAcc** |
|---|---|---|
| $opp\_def\_reb = Low \wedge opponent \neq ATL \wedge thabeet = Out$ | $219$ | $0.0692$ |
| $opp\_def\_reb = Low \wedge opponent \neq ATL$ | $222$ | $0.0689$ |
| $opp\_def\_reb = Low \wedge opponent \neq ATL \wedge ajohnson = Out$ | $222$ | $0.0689$ |
| $opp\_def\_reb = Low \wedge opponent \neq PHI \wedge thabeet = Out$ | $225$ | $0.0685$ |
| $opp\_def\_reb = Low \wedge opponent \neq PHI$ | $228$ | $0.0682$ |

- Common knowledge is re-discovered
- Descriptions are uninteresting (absence of reserve players)

# What can we do with the algorithms?



## Tune parameters or write a new one



# What should we (ideally) be able to do?



**In EDA, the user is "the success criterion"**

Let the user say something like what we've just said:

- This is interesting
- This is boring
- Show me something else…

…and use this to improve the results

# Outline

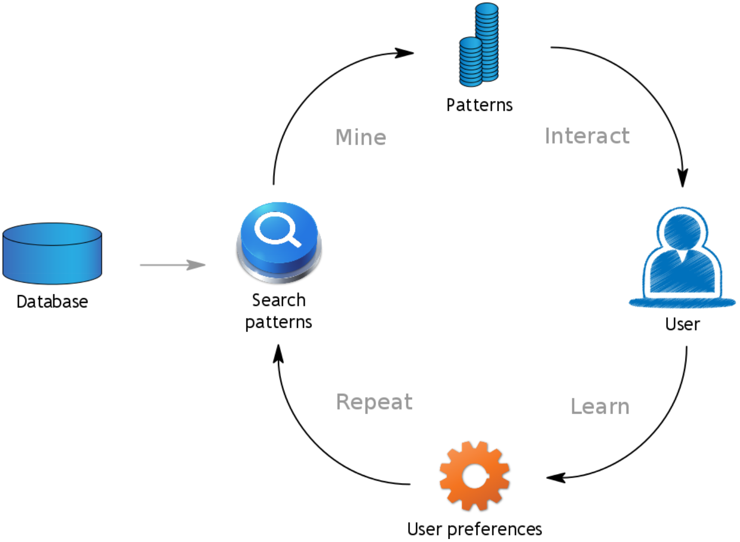
1 Introduction

2 Our approach to interactive pattern mining
- General framework
- Interactive subgroup discovery
- Learning to rank patterns

3 Conclusions & future work

# Mine, Interact, Learn, Repeat

## As opposed to *Big modelling up front*



# What do we need?



## Research questions

| | |
|---|---|
| Mine | How to mine patterns |
| Interact | What to show to the user<br>What the user can do with it |
| Learn | How to intepret what the user does |
| Repeat | How to mine patterns, *given the user actions*<br>When to stop |

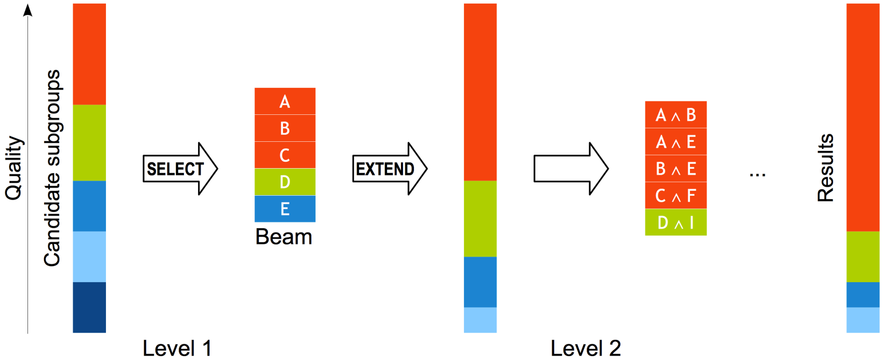
# Outline

# How subgroup discovery works



## We have used DSSD algorithm

It is based on *levelwise beam search*

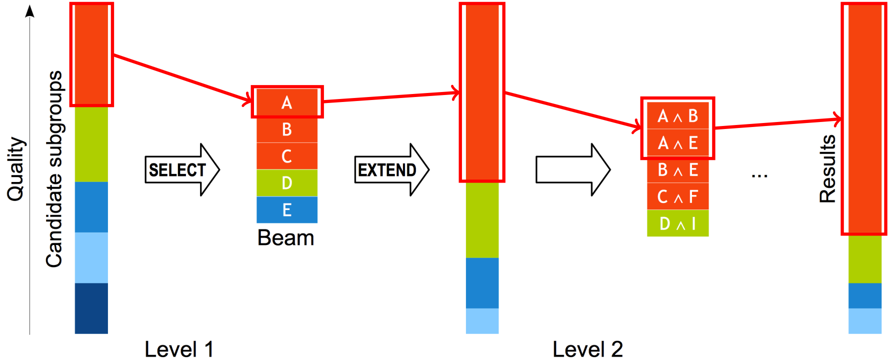


# Is there a problem?



## High-quality *conditions* can dominate beams

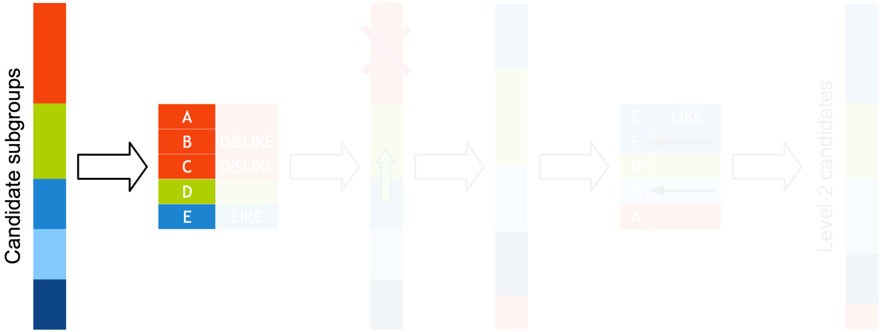
Recall $opp\_def\_reb = Low$ from our example



# Interactive subgroup discovery: Workflow



## Generate level-1 candidate and select a beam, just like in DSSD...

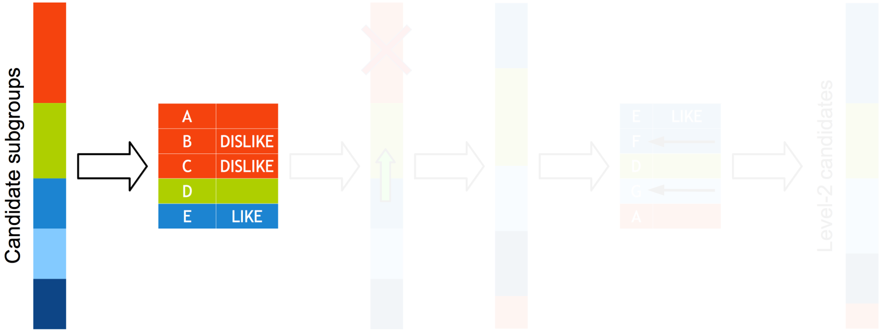



Dzyuba, van Leeuwen. *Interactive Discovery of Interesting Subgroup Sets*. Proceedings of IDA, 2013.

# Interactive subgroup discovery: Workflow



## ...Only now the user can interact with it: 👍 / 👎

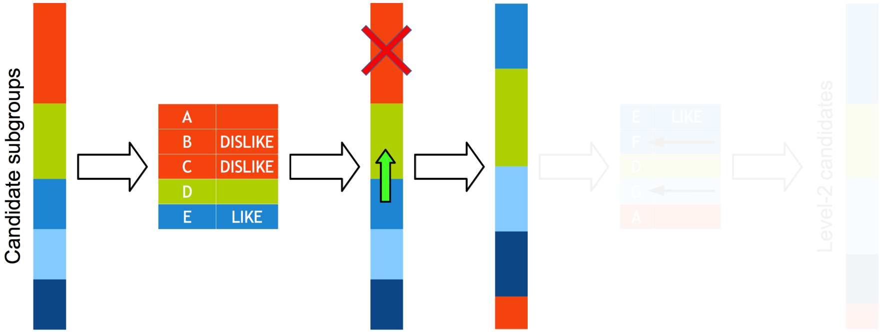



Dzyuba, van Leeuwen. *Interactive Discovery of Interesting Subgroup Sets*. Proceedings of IDA, 2013.

# Interactive subgroup discovery: Workflow



## Candidate subgroups are re-ranked, based on the feedback

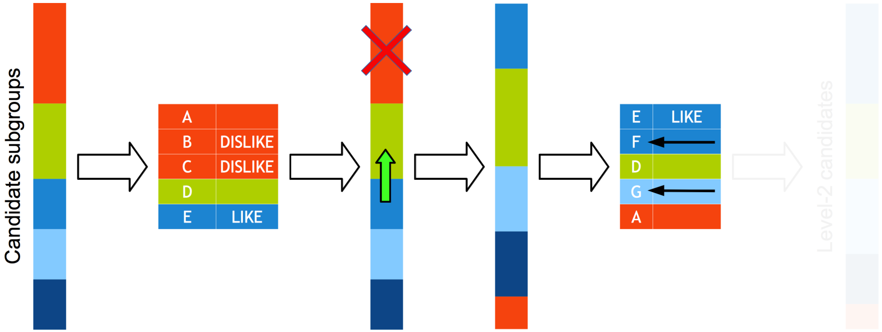



Dzyuba, van Leeuwen. *Interactive Discovery of Interesting Subgroup Sets.* Proceedings of IDA, 2013.

# Interactive subgroup discovery: Workflow



## Disliked subgroups are replaced

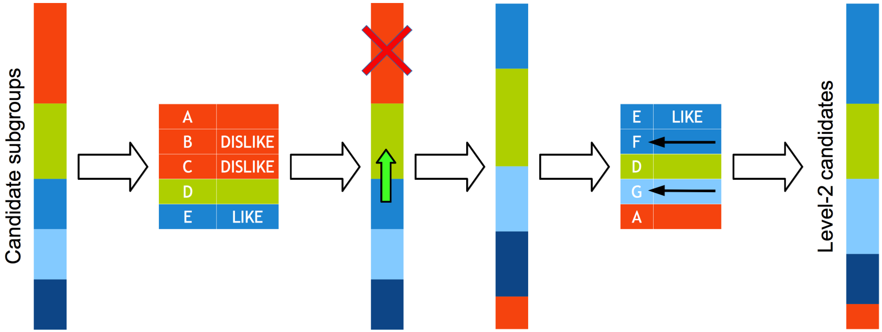



Dzyuba, van Leeuwen. *Interactive Discovery of Interesting Subgroup Sets*. Proceedings of IDA, 2013.

# Interactive subgroup discovery: Workflow



## Level-2 candidates are generated, taking feedback into account




Dzyuba, van Leeuwen. *Interactive Discovery of Interesting Subgroup Sets*. Proceedings of IDA, 2013.

# Re-ranking candidate subgroups



Re-weighing objective quality:

$$WRAcc'(p) = WRAcc(p) \times \frac{Similarity(p, \text{ liked subgroups})}{Similarity(p, \text{ disliked subgroups})}$$

- Description similarity
- Cover similarity

Dzyuba, van Leeuwen. *Interactive Discovery of Interesting Subgroup Sets*. Proceedings of IDA, 2013.

# Interactive subgroup discovery



| | |
|---|---|
| Mine | Generate candidate subgroups of length $L$ |
| Interact | Select a diverse beam<br>*Like* or *Dislike* subgroups in the beam |
| Learn | Discard disliked subgroups<br>Re-rank the candidates |
| Repeat | Re-select the beam<br>*or*<br>Generate candidate subgroups of length $L+1$ |

Dzyuba, van Leeuwen. *Interactive Discovery of Interesting Subgroup Sets*. Proceedings of IDA, 2013.

# We let a basketball journalist use it



## He was able to discover more interesting subgroups

| Description | Size | WRAcc |
|---|---:|---:|
| $crawford = Out \wedge matthews = In$ | 290 | 0.0328 |
| $hickson = In$ | 143 | 0.0219 |
| $crawford = Out \wedge hickson = In$ | 63 | 0.0211 |
| $matthews = In \wedge hickson = In$ | 99 | 0.0163 |
| $matthews = In \wedge pace < 88.518$ | 303 | 0.0221 |

- User liked 7 and disliked 11 subgroups
- Descriptions are more interesting (key players)
- However, objective quality is lower

# Is this evidence sufficient?



**Evaluating interactive algorithms is a major issue**

One case study with one user is,
of course, anecdotal evidence

Proper evaluation requires more:

- Datasets
- Users
- Algorithms
- Parameter settings
- …

# Automated user emulation



**We need a plausible hypothesis regarding human user behaviour**

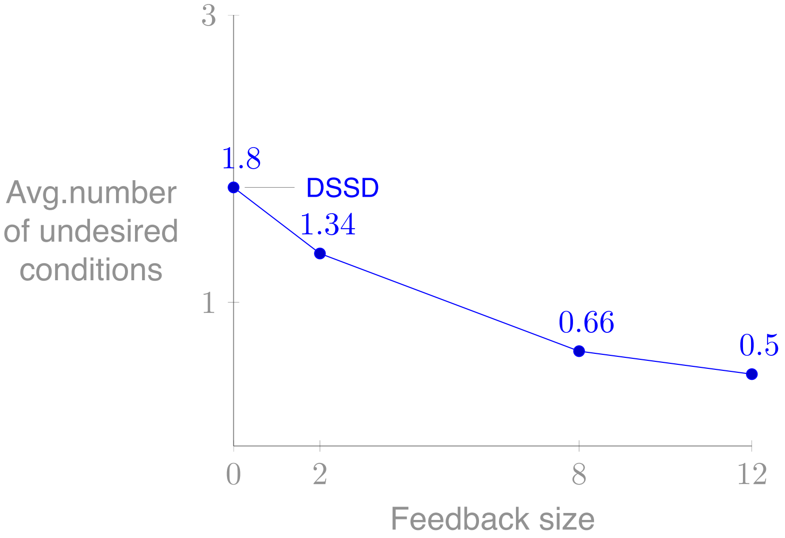
Motivated by basketball example:

- High-quality subgroups correspond to common knowledge
- Select a few top subgroups as *background knowledge*
- Dislike subgroups similar to these *well-known* ones

## Little effort is required to eliminate undesired conditions



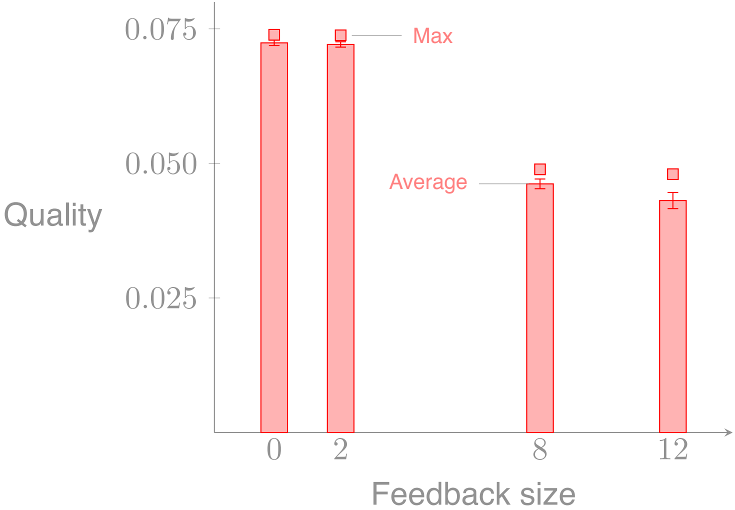
### We used description similarity



# Objective quality remains reasonably high



## Algorithm discovers *other* dependencies

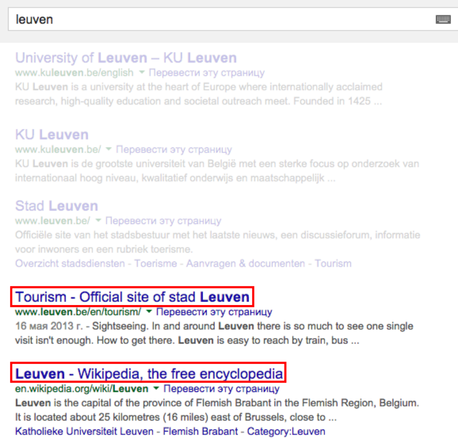


# Re-weighing quality is quite ad-hoc



## Can we do better?

Similar problem occurs in search engines:




**Preference learning**

Also known as *object ranking* or *learning to rank*

# Outline

# Preference learning by example



## Input data and training

Ordered feedback: $\{p_1, p_2, p_3\} \Rightarrow p_3 \succ p_1 \succ p_2$

We use an off-the-shelf learner, Ranking SVM

| | Features | | | |
|---|---|---|---|---|
| $p_3$ | $1$ | $1$ | $1$ | $1$ |
| $p_1$ | $1$ | $1$ | $0$ | $0$ |
| $p_2$ | $1$ | $1$ | $0$ | $1$ |
| Learned weights | $0.0000$ | $0.0000$ | $0.0050$ | $-0.0025$ |

# Preference learning by example

## Using learned ranking functions

Rank new bunch of objects *knowing their features*:
$\{p_1, p_3, p_4, p_5, p_6\}$

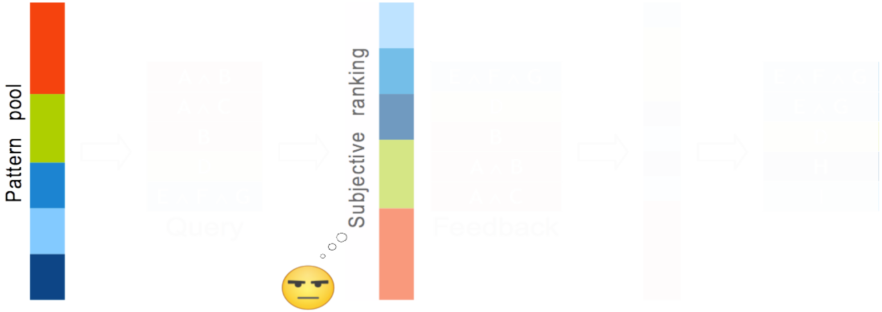
| | Features | | | | $\hat{R}$ |
|---|---|---|---|---|---|
| $p_3$ | 1 | 1 | 1 | 1 | 0.025 |
| $p_1$ | 1 | 1 | 0 | 0 | 0.000 |
| $p_4$ | 0 | 1 | 0 | 1 | −0.025 |
| $p_5$ | 0 | 0 | 1 | 0 | 0.050 |
| $p_6$ | 0 | 0 | 0 | 0 | 0.000 |

$$p_5 \succ p_3 \succ \{p_1, p_6\} \succ p_4$$

# Preference learning for ranking patterns



## User has an *implicit* subjective ranking of patterns

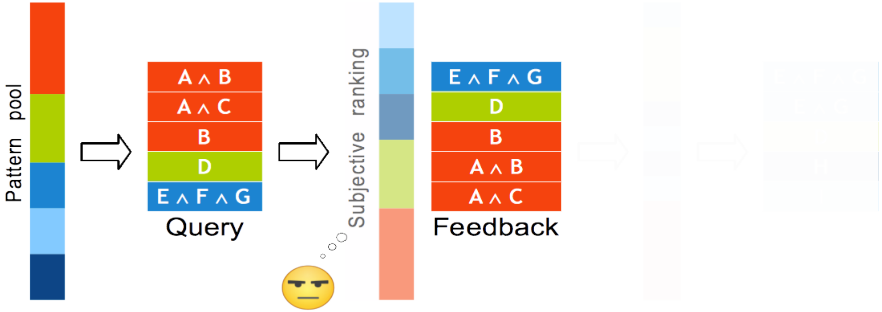



Dzyuba, van Leeuwen, Nijssen, De Raedt. *Active Preference Learning for Ranking Patterns*. Proceedings of ICTAI, 2013.

# Preference learning for ranking patterns



## User ranks small sets of patterns

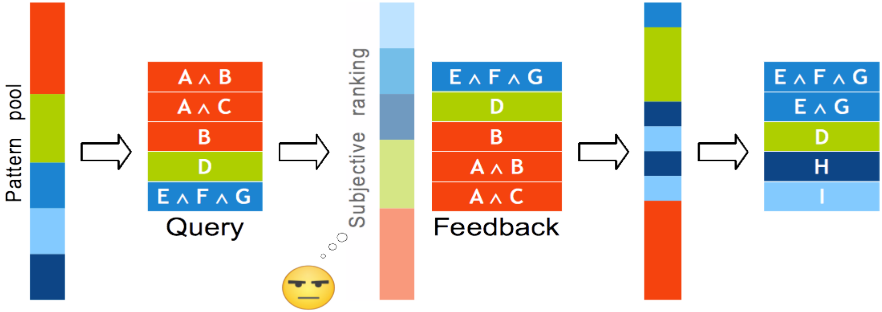



Dzyuba, van Leeuwen, Nijssen, De Raedt. *Active Preference Learning for Ranking Patterns*. Proceedings of ICTAI, 2013.

# Preference learning for ranking patterns



## User preferences w.r.t. pattern are learned from feedback




Dzyuba, van Leeuwen, Nijssen, De Raedt. *Active Preference Learning for Ranking Patterns*. Proceedings of ICTAI, 2013.

# Active preference learning for ranking patterns 

| | |
|---|---|
| Mine | Mine a pool of patterns |
| Interact | Select a small subset of the pool<br>Let user re-order it |
| Learn | Learn a pattern ranking function |
| Repeat | Stop after a number of iterations<br>Mine new patterns using the learned ranking function |

Dzyuba, van Leeuwen, Nijssen, De Raedt. *Active Preference Learning for Ranking Patterns*. Proceedings of ICTAI, 2013.

# Ordered feedback requires considerable effort



## Computing $\binom{n}{2}$ pairwise preferences

$$p_3 \succ p_1 \succ p_2 = \{p_3 \succ p_1,\ p_3 \succ p_2,\ p_1 \succ p_2\}$$

$$Effort\,(n = 3) = 3$$
$$Effort\,(n = 5) = 10$$
$$Effort\,(n = 10) = 45$$

We want to minimize the user effort: *Active learning*

1. Inspired by Information Retrieval
2. Specific to Ranking SVM learner

# User emulation for active preference learning 

## Application to subgroup discovery

Target pattern ranking by a *complex* objective quality measure:

$$\chi^2(P) = \sum_{c \in \{-,+\}} \frac{(|P| \cdot (|P^c| - |\mathcal{D}^c|))^2}{|P| \cdot |\mathcal{D}^c|} + \frac{(|P| \cdot (|P^c| - |\mathcal{D}^c|))^2}{(|\mathcal{D}| - |P|) \cdot |\mathcal{D}^c|}$$

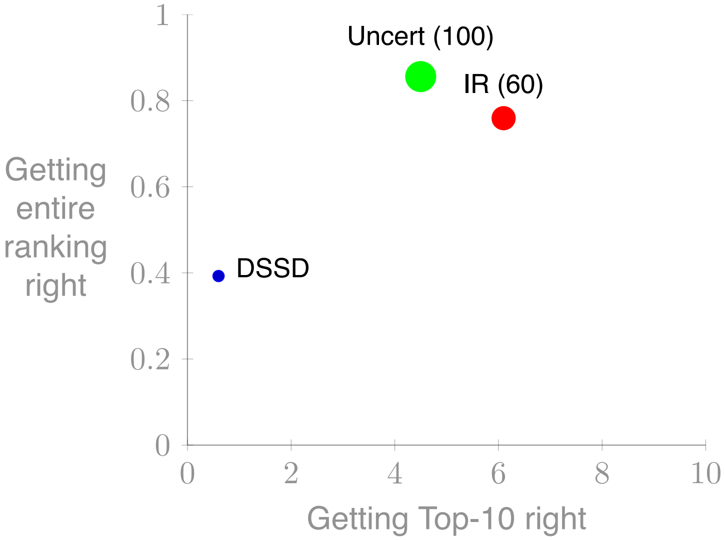
Pattern pools are substantially different, by intention:

$$Coverage(P) = \frac{|P|}{|\mathcal{D}|}, Sensitivity(P) = \frac{|P^+|}{|\mathcal{D}^+|}, Specificity(P) = 1 - \frac{|P^-|}{|\mathcal{D}^-|}$$

# Accurate rankings are learned



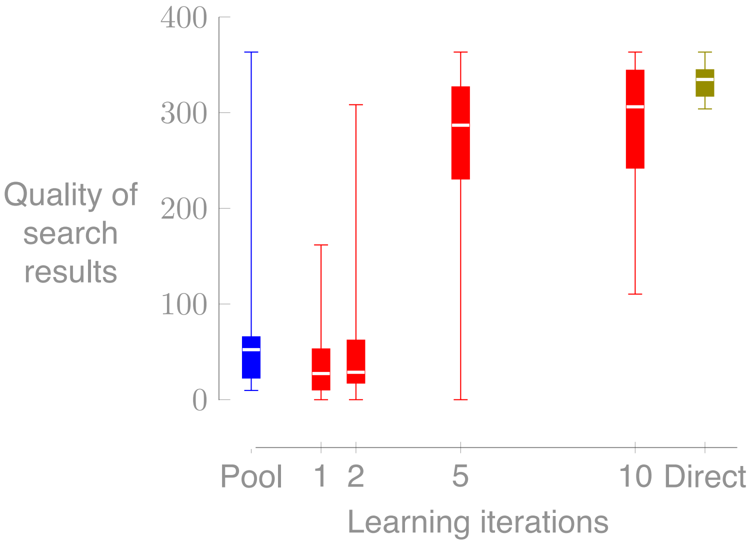
## Active learning helps reduce the effort



# Learned ranking function generalise well



## They can be used as search heuristics



# Outline

# Future work

## Better algorithms and models CS perspective

- Versatile user models
- Principled search with learned models
- Convergence and stopping criteria

## Better interaction HCI perspective

- Natural feedback formats
- Pattern visualisation

# Take-away message

**Machine learning for interactive pattern mining**

| | | |
|---|---|---|
| | MINE | patterns |
| Let user | INTERACT | with them |
| | LEARN | what the user is interested in |
| | REPEAT | until something interesting is found |

# Interactive Pattern Mining

Joint work with:

- Dr.Matthijs van Leeuwen
- Dr.Siegfried Nijssen
- Prof.Luc De Raedt

Thank you for your attention!

May I answer any questions?