Conditional Ranking on Relational Data: Theory and Applications

Willem Waegeman
(Ghent University, Belgium)

POSSIBILISTIC INFORMATION:
A Tutorial
George J. Klir
State University of New York (SUNY)
Binghamton, New York 13902, USA
gklir@binghamton.edu

Willem Waegeman (DAMOL)
Conditional Ranking on Relational Data
January 10, 2013 1 / 1
Conditional Ranking on Relational Data: Theory and Applications

Willem Waegeman
Research Unit Knowledge-based Systems (KERMIT)
Department of Mathematical Modelling, Statistics and Bioinformatics
Ghent University, Belgium

Making Sense of Big Data

Olomouc, Czech Republic, January 10th 2013
The Netflix 2006 prize: a $1,000,000 reward for the research group that could improve the Netflix recommendation engine with 10%
After more than three years, the Netflix prize was won in 2009 by BellKor’s Pragmatic Chaos Team.
Outline

• Introduction
• **Learning dyadic relations**
• Conditional ranking
• Matrix factorization
• Learning monadic relations
• Conclusions and further perspectives
How does the data look like?

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>3</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Known rating**

**Unknown rating**

**BASIC SETTING**

- COLLABORATIVE FILTERING
- USING ONLY THE LABEL MATRIX OF USERS AND ITEMS
How does the data look like?

**Users**
- **User feature 1:** age
- **User feature 2:** gender

**Products / Items**
- **Item feature 1:** comedy?
- **Item feature 2:** action?

**Known rating**
- 5
- 5
- 4
- 3
- 1
- 2
- 4
- 3

**Unknown rating**
- 0
- 0
- 0
- 0
- 1
- 1
- 1
- 0

**Known rating**
- 1
- 2
- 3
- 1

**Unknown rating**
- 6
- 7

**Extended setting**
- Contest-based filtering
- Using the label matrix
- And three types of features
How does the data look like?
Another view needed to describe the third feature type

### Training data

<table>
<thead>
<tr>
<th>user</th>
<th>movie</th>
<th>date</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21</td>
<td>5/7/02</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>213</td>
<td>8/2/04</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>345</td>
<td>3/6/01</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>123</td>
<td>5/1/05</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>768</td>
<td>7/15/02</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>76</td>
<td>1/22/01</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>45</td>
<td>8/3/00</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>568</td>
<td>9/10/05</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>342</td>
<td>3/5/03</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>234</td>
<td>12/28/00</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>76</td>
<td>8/11/02</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>56</td>
<td>6/15/03</td>
<td>4</td>
</tr>
</tbody>
</table>

### Test data

<table>
<thead>
<tr>
<th>user</th>
<th>movie</th>
<th>date</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>62</td>
<td>1/6/05</td>
<td>?</td>
</tr>
<tr>
<td>1</td>
<td>96</td>
<td>9/13/04</td>
<td>?</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>8/18/05</td>
<td>?</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>11/22/05</td>
<td>?</td>
</tr>
<tr>
<td>3</td>
<td>47</td>
<td>6/13/02</td>
<td>?</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>8/12/01</td>
<td>?</td>
</tr>
<tr>
<td>4</td>
<td>41</td>
<td>9/1/00</td>
<td>?</td>
</tr>
<tr>
<td>4</td>
<td>28</td>
<td>8/27/05</td>
<td>?</td>
</tr>
<tr>
<td>5</td>
<td>93</td>
<td>4/4/05</td>
<td>?</td>
</tr>
<tr>
<td>5</td>
<td>74</td>
<td>7/16/03</td>
<td>?</td>
</tr>
<tr>
<td>6</td>
<td>69</td>
<td>2/14/04</td>
<td>?</td>
</tr>
<tr>
<td>6</td>
<td>83</td>
<td>10/3/03</td>
<td>?</td>
</tr>
</tbody>
</table>

**THIRD TYPE = FEATURES OVER USER-ITEM PAIRS = e.g. date in Netflix**
Another example of learning a dyadic relation: prediction of protein-ligand interactions
We did a number of experiments with the Karaman dataset (Nature 2008)

Based on joint work with: Thomas Fober, Eyke Huellermeier, Serghei Glinca, Denis Schmidt and Peter Kolb
Results on the Karaman dataset for different types of evaluations

<table>
<thead>
<tr>
<th>Setting</th>
<th>Queries</th>
<th>Rank loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>New query</td>
<td>Proteins</td>
<td>0.3279 (0.0883)</td>
</tr>
<tr>
<td>New query</td>
<td>Ligands</td>
<td>0.3240 (0.1293)</td>
</tr>
<tr>
<td>New query + new objects to be ranked</td>
<td>Proteins</td>
<td>0.3302 (0.2074)</td>
</tr>
<tr>
<td>New query + new objects to be ranked</td>
<td>Ligands</td>
<td>0.3811 (0.1775)</td>
</tr>
</tbody>
</table>
Flaws in evaluation schemes for pair-input computational predictions

To the Editor: Computational prediction methods that operate on pairs of objects by considering features of each (hereafter referred to as pair-input methods) have been crucial in many areas of biology and chemistry over the past decade. Among the most prominent examples are protein-protein interaction (PPI)\(^1\), protein-drug interaction\(^2,3\), protein-RNA interaction\(^4\) and drug indication\(^5\) prediction methods. A sampling of more than 50 published studies involving pair-input methods is provided in Supplementary Table 1. Here we demonstrate that the paired nature of inputs has significant, though not yet widely perceived, implications for the validation of pair-input methods.

The effects that the paired nature of inputs has on the cross-validation of pair-input methods can be seen in the following example. Proteochemometrics modeling\(^2\), a computational
A more formal description of the problem

- Bipartite graph $G = (U, P, E)$ with edges $e = (u, p)$ and $u \in U, p \in P$.

- Training data $T = \{(e, y_e) \mid e \in E\}$ with labels $y_e$ and features $e = (\phi(u), \phi(p))$.

- Primal and dual kernel model on edges:

$$h(\bar{e}) = h(\bar{u}, \bar{p}) = \langle w, \Phi(\bar{e}) \rangle = \sum_{e \in E} a_e K^\Phi(e, \bar{e}) = \sum_{e \in E} a_e K^\Phi(u, p, \bar{u}, \bar{p})$$
We use the Kronecker product to define a feature mapping or kernel on the edges.

Joint feature representation $\Phi$ for an edge:

$$\Phi(e) = \Phi(u, p) = \phi(u) \otimes \phi(p)$$

where $\otimes$ denotes the Kronecker-product:

$$A \otimes B = \begin{pmatrix} A_{1,1}B & \cdots & A_{1,n}B \\ \vdots & \ddots & \vdots \\ A_{m,1}B & \cdots & A_{m,n}B \end{pmatrix}$$

This yields the following kernel $K^{\Phi}$ for edges in the dual:

$$K^{\Phi}(e, \bar{e}) = K^{\Phi}(u, p, \bar{u}, \bar{p}) = K^{\Phi}(u, \bar{u})K^{\Phi}(p, \bar{p})$$

Results in a class of universal approximators for relations!
Different types of relations lead to different learning settings, for which standard algorithms can be adopted.

- Minimizing the regularized empirical error:
  \[
  \hat{h} = \arg\min_{h \in \mathcal{H}} \frac{1}{|T|} \sum_{y_e \in T} L(h(e), y_e) + \lambda \|h\|^2_{\mathcal{H}}
  \]

- Classification settings: e.g. logistic loss in kernel logistic regression or hinge loss in support vector machines

- Regression settings: e.g. the squared loss in kernel ridge regression or the $L^2$-insensitive loss in support vector regression

\begin{align*}
\begin{array}{c|c|c|c|c|c}
1 & 2 & 3 & 4 & 5 \\
\hline
1 & & & .8 & & \\
2 & & .2 & & & \\
3 & & & .5 & .6 & \\
4 & .3 & .6 & & 1 & \\
5 & & .7 & & & \\
\end{array}
\end{align*}

Regression

\begin{align*}
\begin{array}{c|c|c|c|c|c}
1 & 2 & 3 & 4 & 5 \\
\hline
1 & & & 1 & & \\
2 & & 0 & & & \\
3 & & & 0 & 1 & \\
4 & 0 & 1 & & 1 & \\
5 & 1 & & & & \\
\end{array}
\end{align*}

Classification
Outline

• Introduction
• Learning dyadic relations
• Conditional ranking
• Matrix factorization
• Learning monadic relations
• Conclusions and further perspectives
Reformulation as an information retrieval task

- **Query: Protein 1**
  - Ranking of ligands: 3 10 16 1 12 ...

- **Query: Protein 2**
  - Ranking of ligands: 5 13 2 3 11 ...

Pool of ligands

1 2 3 4 5 ...

Pool of proteins

1 2 3 4 5 ...

Google Belgïe
From learning relations to conditional ranking: optimize conditional rank loss (or Kendall’s Tau) instead of squared loss

Compute loss over all rankings simultaneously:

\[
L(h, T) = \sum_{u \in U} \sum_{(u, p), (u, \bar{p}) \in E_u : y_{u,p} < y_{u,\bar{p}}} \text{sgn}(h(u, p) - h(u, \bar{p}))
\]

Approximate by minimizing a squared ranking loss function:

\[
L(h, T) = \sum_{u \in U} \sum_{(u, p), (u, \bar{p}) \in E_u} (\text{sgn}(y_{u,p} - y_{u,\bar{p}}) - h(u, p) + h(u, \bar{p}))^2
\]

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>.8</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>.5</td>
<td>.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>.3</td>
<td>.1</td>
<td>.6</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Condition on node 4

Use 4 as a query

<table>
<thead>
<tr>
<th>Correct ranking</th>
<th>Predicted ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Rank loss</td>
<td>2 / 6</td>
</tr>
</tbody>
</table>

18
Optimization details: different methods are applicable

- **Standard algorithms** can be used for small to moderate sample sizes
- **Conjugate gradient with early stopping** descent methods are applied for large sparse label matrices
- An **efficient analytical solution** exists for fully-observed matrices (e.g. >25 000 000 entries)

T. Pahikkala, A. Airola, T. Salakoski, M. Stock, B. De Baets, W. Waegeman, Efficient least-squares algorithms for conditional ranking on relational data, Machine Learning, ARXIV
A third case study: predicting interactions between methanotrophic and heterotrophic bacteria

Testing interactions in lab experiments

Predicting interactions without experiments

Methanotrophic bacteria

Heterotrophic bacteria

Features

Machine Learning

Features
Interactions between methanotrophic and heterotrophic bacteria: results of lab experiments

**Extract parameters from the growth curves to obtain labels.**
Predicting interactions between methanotrophic and heterotrophic bacteria: results of our algorithms

<table>
<thead>
<tr>
<th>Setting</th>
<th>Best approach</th>
<th>Rank loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction for known pairs</td>
<td>Jukes-Cantor kernel Carbohydrate Tanimoto</td>
<td>0.1697</td>
</tr>
<tr>
<td>New methanotrophs</td>
<td>Jukes-Cantor kernel Carbohydrate linear</td>
<td>0.4030</td>
</tr>
<tr>
<td>New heterotrophs</td>
<td>Spectrum kernel Carbohydrate Tanimoto</td>
<td>0.1582</td>
</tr>
</tbody>
</table>
Outline

• Introduction
• Learning dyadic relations
• Conditional ranking
• **Matrix factorization**
• Learning monadic relations
• Conclusions and further perspectives
Instances are not i.i.d.

Find an approximation of the label matrix as a product of two smaller matrices:

\[ Y = MU^T \]

Yields latent features for every row and column

Works only for instances with observations in the training dataset

Use these latent variables together with the explicit variables using the same model as before
The latent variable view of matrix factorization methods
Outline

• Introduction
• Learning dyadic relations
• Conditional ranking
• Matrix factorization
• Learning monadic relations
• Conclusions and further perspectives
Monadic relations: the two objects that constitute a relation come from the same domain

Has been investigated in many areas of machine learning, such as link prediction, preference learning, statistical relational learning, metric learning, etc.
Learning monadic relations: functional ranking of enzymes

Convert EC numbers to similarity scores, from which rankings can be constructed

Queries = Objects to be Ranked = Proteins
Learning monadic relations: functional ranking of enzymes

Convert EC numbers to similarity scores, from which rankings can be constructed.

Users = Items = Queries = Objects to be Ranked = Proteins

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>X</td>
<td></td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>X</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>4</td>
<td>X</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td></td>
<td>1</td>
<td>X</td>
<td>2</td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td>3</td>
<td>2</td>
<td>X</td>
</tr>
</tbody>
</table>
Conditional ranking algorithms outperform traditional retrieval methods in this application domain.
Incorporating domain knowledge: the case of symmetric relations

A relation \( h : V^2 \rightarrow [0, 1] \) is called symmetric if

\[
h(v, v') = h(v', v) \quad \forall (v, v') \in V^2.
\]

Specific joint feature representation for an edge:

\[
\Phi(e) = \Phi(v, v') = \Psi(v, v') + \Psi(v', v),
\]

This yields the following kernel defined on the edges:

\[
K^\Phi(e, \overline{e}) = K^\Phi(v, v', \overline{v}, \overline{v}')
\]
\[
= K^\Psi(v, v', \overline{v}, \overline{v}') + K^\Psi(v', v, \overline{v}', \overline{v})
\]
\[
+ K^\Psi(v', v, \overline{v}, \overline{v}') + K^\Psi(v, v', \overline{v}', \overline{v}).
\]

Obtain a class of universal approximators by choosing for \( K^\Psi \) the Kronecker product kernel.
Incorporating domain knowledge: the case of reciprocal relations

A relation $h : V^2 \rightarrow [0, 1]$ is called reciprocal if

$$h(v, v') + h(v', v) = 1 \quad \forall (v, v') \in V^2.$$ 

Specific joint feature representation for an edge:

$$\Phi(e) = \Phi(v, v') = \Psi(v, v') - \Psi(v', v),$$

This yields the following kernel defined on the edges:

$$K^\Phi(e, \bar{e}) = K^\Phi(v, v', \bar{v}, \bar{v}') \\
    = K^\Psi(v, v', \bar{v}, \bar{v}') + K^\Psi(v', v, \bar{v}', \bar{v}) \\
    - K^\Psi(v', v, \bar{v}, \bar{v}') - K^\Psi(v, v', \bar{v}', \bar{v}).$$

Obtain a class of universal approximators by choosing for $K^\Psi$ the Kronecker product kernel
Other types of domain knowledge: metrics, total orders, etc.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>X</td>
<td>.4</td>
<td>.2</td>
</tr>
<tr>
<td>B</td>
<td>.6</td>
<td>X</td>
<td>.3</td>
</tr>
<tr>
<td>C</td>
<td>.8</td>
<td>.7</td>
<td>X</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>X</td>
<td>.1</td>
<td>.3</td>
</tr>
<tr>
<td>B</td>
<td>.1</td>
<td>X</td>
<td>.2</td>
</tr>
<tr>
<td>C</td>
<td>.3</td>
<td>.2</td>
<td>X</td>
</tr>
</tbody>
</table>

Kernel for utility models on reciprocal relations (Herbrich et al. 2000):

\[
K^\Phi_{fR}(e, \bar{e}) = K^\Phi(v, \bar{v}) + K^\Phi(v', \bar{v}') - K^\Phi(v, \bar{v}') - K^\Phi(v', \bar{v})
\]

Metric learning pairwise kernel (MLPK) for metrics (Vert et al. 2005):

\[
K^\Phi_{MLPK}(e, \bar{e}) = (K^\Phi_{fR}(e, \bar{e}))^2
\]

\[
= (K^\Phi(v, \bar{v}) + K^\Phi(v', \bar{v}') - K^\Phi(v, \bar{v}') - K^\Phi(v', \bar{v}))^2.
\]

Application to document retrieval: experimental results for the 20 Newsgroups dataset

alt.atheism
soc.religion.christian
talk.religion.misc
talk.politics.guns
talk.politics.mideast
talk.politics.misc
comp.sys.ibm.pc.hardware
comp.graphics
comp.os.ms-windows.misc
comp.sys.mac.hardware
comp.windows.x
rec.autos
rec.motorcycles
rec.sport.baseball
rec.sport.hockey
sci.crypt
sci.electronics
sci.space
sci.med
misc.forsale

Relation between documents:
3 = same topic  2 = similar topic  1 = totally different topic
Conditional ranking: experimental results for the 20 Newsgroups dataset

Incorporating symmetry helps

Optimizing a rank-based loss is beneficial for ranking purposes
Conclusions and further reading

- **A general kernel framework** for learning relations from paired-comparison data
- **Domain knowledge**: incorporated by defining specific similarity matrices / kernels
- **Observations are not independent**: use latent variables
- **Optimize ranking-based losses** in retrieval settings
- **Many applications** in biology, gaming, social networks, information retrieval, recommender systems


T. Pahikkala, A. Airola, T. Salakoski, M. Stock, B. De Baets, W. Waegeman, Efficient least-squares algorithms for conditional ranking on relational data, Machine Learning, ARXIV

[http://staff.cs.utu.fi/~aatapa/software/RPS](http://staff.cs.utu.fi/~aatapa/software/RPS)
Related problem settings: multivariate regression and label ranking

Users

<table>
<thead>
<tr>
<th></th>
<th>27</th>
<th>18</th>
<th>48</th>
<th>39</th>
<th>17</th>
<th>22</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>F</td>
<td>7</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>9</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Products / Items

Training dataset

Test dataset
Related problem settings: multilabel classification and (bipartite) label ranking

<table>
<thead>
<tr>
<th>Products / Items</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Users

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>M</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>F</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>48</td>
<td>F</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>39</td>
<td>M</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>M</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>M</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>F</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Training dataset

Test dataset
Related problem settings: structured output prediction and graph matching

![Diagram](image)

- **Users**:
  - 27 M
  - 18 F
  - 48 F
  - 39 M
  - 17 M
  - 22 M
  - 25 F

- **Products / Items**:
  - 22 13 18 29 31 12 48
  - 11 7 85 18 57 16 49

- **Training dataset**
- **Test dataset**
PLenty of available data, but do you really know your customers? Make big data work for your business.
Acknowledgments

• Dr. Tapio Pahikkala (University of Turku)
• Dr. Antti Airola (University of Turku)
• Prof. Bernard De Baets (Ghent University)
• Ir. Michiel Stock (Ghent University)
• Ayla De Paepe (Ghent University)
• Ir. Jan Verwaereren (Ghent University)
• Prof. Eyke Huellermeier (University of Marburg)
• Dr. Krzysztof Dembczynski (University of Poznan)
• Dr. Weiwei Cheng (University of Marburg)

And all people that delivered data!