Learning from Order Examples (LOE) is the learning task that handles order specifications.

- Formalize the LOE
- Propose several naive solution methods
- Experiments on Artificial Data:
  Analyze the characteristics of these methods
- Experiments on Real Data:
  Analyze the availability on Real Data

The 2002 IEEE International Conference on Data Mining
**What is the "order" ?**

**Order:** sorted lists of items according to some criterion

ex: an sorted list of sushi types *(items)* according to my preference *(a criterion)*

- fatty tunas > eggs > squids

This order specify that

"I prefer fatty tunas to eggs"

This order does NOT specify that

"How much I prefer fatty tunas to eggs"

※ "sushi" is a kind of Japanese food
Merit for Using Orders

An Application to a sensory test

SD (Semantic Differential) method

ex: specify one's preference by the following scale

[like] 5 4 3 2 1 [dislike]

assumption: all respondents share an understanding of its range, divisions and extremes.

→ UNREALISTIC ASSUMPTION

Specifications by using orders does not demand such an assumption
Formalization of LOE

Items are described by attribute value vectors

The acquired rule can handle the items that are not appeared in the example set
The LOE task can be viewed as a regression task targeting the orders.

Observed Order (example) = Absolute Order (model) + noise

Absolute Order: the order of all possible items
Noise: random swapping of items

The rule for sorting that is acquired by LOE tasks = The description of the absolute order by the attributes of items.
Evaluation method

Error measure:

**Spearman's Rank Correlation** ($\rho$ coefficient)

The correlation between ranks of items in the two orders of the same item set

0: no correlation, 1: complete match, -1: reverse

$t$ follows the Student $t$-distribution

with degree of freedom ($\#I - 2$)

$$t = \rho \sqrt{\frac{\#I - 2}{1 - \rho^2}}$$

$\#I$ : the length of orders
**Rank Correlation:** Spearman's $\rho$ or Kendall's $\tau$
measure for comparing two orders

**Paired Comparison:**
   Thurstone's method or Bradley's method

**input:** pairwise precedence information
   = which precedes the other between two items

**output:** real valued scale compatible with inputs as possible

The aim is different from LOE task:
   not estimating the orders
Related Works (2)

[Cohen et al. 99]
input: pairwise precedence information
   = which precedes the other between two items
output: estimate the order that preserves input information as possible

Difference from LOE task

target: LOE: totally well sorted orders
   Cohen: preserving pairwise information

error evaluation:
LOE: final orders are directly evaluated by Spearman's $\rho$
Cohen: evaluate the final orders indirectly by the accuracy of intermediate function
Method: Classification-Based (1)

Learning Stage:

rule for sorting = \[\text{PREF}(I^x, I^y)\]

The conditional probability of the event \(I^x > I^y\) given attribute values of these two items
Sorting Stage:
estimated order for the unordered item set $O_U$

= the order maximizing the criterion function

• Find the order by greedy search techniques

Two types of functions were examined

**type SC:** \[ \sum_{x,y: I^x > I^y} \text{PREF}(I^x, I^y) \]
compatible with the Cohen's criterion

**type PC:** \[ \prod_{x,y: I^x > I^y} \text{PREF}(I^x, I^y) \]
theoretical advantage over the Cohen's criterion
Method: Regression-Based

Learning Stage:
rule for sorting = \text{RANK}(I^x)

function of item's attribute values that outputs the expected item's rank

examples \((\{I\}_x, O^*_x)\)

combined order
\[\begin{align*}
I^x &< I^y \\
I^y &< I^z \\
I^x &< I^y < I^z
\end{align*}\]

item's attribute values \(A(I^x)\)

linar regression technique

Sorting Stage:
Find the order sorted according to RANK values

rule for sorting \(\text{RANK}(I^x, I^y)\)
Aim of Experiments on Artificial Data:
To analyze the characteristics of these methods

- **Item set types**
  - the number of attributes = \{3, 4, 5\}
  - the number of values per one attribute = \{3, 5, 7\}

- **Method to generate example orders**
  - 10 orders are generated for each item set types
  - orders are defined according to the score that is a linear function of weight or attribute values

- **Example sets**
  - the length of example order (= the size of item sets) = \{3, 5, 10\}
  - the number of examples = \{10, 30, 50\}

- Apply leave-one-out test (strict cross-validation)
- Error measure is the mean of $\rho$ between the estimated order and the original order
Artificial Data: means of $\rho$

<table>
<thead>
<tr>
<th></th>
<th>ALL</th>
<th>$#I = 3$</th>
<th>$#I = 5$</th>
<th>$#I = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC</td>
<td>0.808</td>
<td>0.667</td>
<td>0.825</td>
<td>0.932</td>
</tr>
<tr>
<td>PC</td>
<td>0.808</td>
<td>0.667</td>
<td>0.825</td>
<td>0.932</td>
</tr>
<tr>
<td>R</td>
<td>0.802</td>
<td>0.617</td>
<td>0.837</td>
<td>0.950</td>
</tr>
</tbody>
</table>

- $#I$: the length of examples = the size of items
- Rank Correlation $\rho$: 1=complete match, -1=reverse order
- SC: Classification-Based with Cohen compatible criterion
- PC: Classification-Based with the PREF product criterion
- R: Regression-Based method

### Overall Results:
- $#I$ becomes large $\rightarrow$ performance improves
- The number of examples increase $\rightarrow$ performance improves
Classfication vs Regression

Compare the classification-based method and the regression-based method.

**Paired $t$-test:** the difference between $\rho$ is statistically significant or not.

<table>
<thead>
<tr>
<th></th>
<th>ALL</th>
<th>$#I = 3$</th>
<th>$#I = 5$</th>
<th>$#I = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC-R</td>
<td>1.4430</td>
<td>4.4143</td>
<td>-2.2272</td>
<td>-8.5784</td>
</tr>
<tr>
<td>PC-R</td>
<td>1.4626</td>
<td>4.4254</td>
<td>-2.3547</td>
<td>-8.5023</td>
</tr>
</tbody>
</table>

Blue: Classification-based method is better
Red: Regression-based method is better

The length of example orders are:

**short**

better method: classification based

**long**

better method: regression based
Transitivity Consistency (1)

Accuracy of intermediates

A: The accuracy of PREF function of Classification Based Method

B: Correlation between ordinal and combined orders of Regression Based Method

<table>
<thead>
<tr>
<th></th>
<th>#I : the size of item sets</th>
<th>#EX : the number of examples</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>ALL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0.864</td>
<td>0.800</td>
</tr>
<tr>
<td>B</td>
<td>0.792</td>
<td>0.689</td>
</tr>
</tbody>
</table>

Summary

#I  increase ➔ ➔ A: increase ➔ ➔ B: drastically increase ↗

#EX increase ➔ ➔ A: increase ➔ ➔ B: not change ➔
The performance of the regression-based is worse if the length of example orders is short. But, for longer length, it surpasses the classification-based. **WHY?**

This is results from the performance of intermediates. The better combined orders can be derived from the longer example orders. **WHY?**

The longer example orders **HIGHLY** preserve the transitivity consistency:

\[(I^x < I^y) \land (I^y < I^z) \Rightarrow I^x < I^y < I^z\]

**The Regression-Based method:** This makes easy to combine of example orders

**The Classification-Based method:** This is not contribute the performance independency assumption of pairwise precedence info is violated
Additional Experiment on SC

**SC method (= compatible with Cohen's method)**

Find the sub-optimal order by greedy search

Order derived by optimal search

- pairwise precedence info. will more preserved

- \( t \)-values of difference

<table>
<thead>
<tr>
<th>Optimal Search</th>
<th>Greedy Search</th>
<th>( t )-values of difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>rank correlation</td>
<td>0.805</td>
<td>0.808</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-2.7915</td>
</tr>
</tbody>
</table>

the order estimated by optimal search is significantly worse than that by greedy search

**Cohen's GOAL:**
order preserving pairwise precedence info

**LOE's GOAL:**
totally well sorted orders

DIFFER!
Real Data: Overall

To investigate the LOE solution methods work well on Real Data

An Experiment:

**Ask 52 people to sort 10 types of sushi according to his/her preference**

By applying any of three methods, we could acquire the order of which mean correlation to given preference orders is *moderately high*

⇒ Our LOE methods works well on real data
The order derived by Regression-based method

One can know a summary of respondents in terms of preferences in sushi

ex. most popular type of sushi is "fatty tuna"

Rank correlation between the above order and the second author's preference order is HIGH (0.842)

He has ordinal tendency of preference in sushi
### Computational Complexity

<table>
<thead>
<tr>
<th></th>
<th>Learning Stage</th>
<th>Sorting Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC SC</td>
<td>$\sum_{i} (#I_i)^2$</td>
<td>$(#I_U)^3$</td>
</tr>
<tr>
<td>R</td>
<td>$(#I_C)^3$</td>
<td>$#I_U \log(#I_U)$</td>
</tr>
</tbody>
</table>

#EX: the number of examples  
#I_i: the size of $i$-th example item set  
#I_C: the size of combined item set  
#I_U: the size of the unordered item set

**Learning Stage:** classification-based method is better  
**Sorting Stage:** regression-based method is better
Conclusions

• We proposed a learning task that handles orders
• We showed several naive methods and analyzed these methods by applying them on artificial data
• We showed that these methods worked well also on real data

Errata

• In the last paragraph of the Section 6, (the first author's) ➔ (the second author's)