SHORTING NUMBER OF QUESTIONS IN LONG PSYCHOLOGICAL QUESTIONNAIRES

© Capp & Co Ltd, 2017
Psychological tests are useful for companies.

- Discrete type of response
- Paper-based / online-based

* https://www.jobmi.com/
** https://www.16personalities.com/
THE PROBLEM

I

Are the questions capturing what we want to capture?

II

Are there redundancy among questions such that we can reduce the size of the test?
THE PROBLEM

I
Are the questions capturing what we want to capture?

II
Are there redundancy among questions such that we can reduce the size of the test?
THE PROBLEM

- Dataset: collection of user responses (~30000)
- In our case the test has 90 questions with 4 possible answers:
  1  2  3  4
- Drawbacks?
**Drawback:** users perceive the scale in different ways

**Drawback:** users tend to choose high values
THE PROBLEM

- Dataset can be mapped into \( \{1, 2, 3, 4\}^{90} \)
- How does it look like?
Drawback: sparse data.

Drawback: data concentrated in few cells
HOW TO SOLVE IT

- Divide the problem:
HOW TO SOLVE IT

- Divide the problem:
  - 1. Find a set of predictors and a set of questions to be predicted

\[ Q = P \cup S \]
HOW TO SOLVE IT

- Divide the problem:
  - 1. Find a set of predictors and a set of questions to be predicted
    \[ Q = P \cup S \]
  - 2. Predict \( P \) using \( S \)
1. FEATURE SELECTION

- First part is a *feature selection* problem
- Ideally, find $P$ and $S$ automatically
- In reality, divide the problem
1. FEATURE SELECTION

- First part is a *feature selection* problem
- Ideally, find P and S automatically
- In reality, divide the problem
- **Fix one question** and find the best subset of predictors
1. FEATURE SELECTION

- First part is a *feature selection* problem
- Ideally, find P and S automatically
- In reality, divide the problem
- **Fix one question** and find the **best subset** of predictors
I. FEATURE SELECTION

Filter methods

Description
Intrinsic properties of data
Advantages
Computationally simple and fast
Disadvantages
Ignore interaction with the classifier
Examples used in our problem
Correlation-based
Mutual Information

[1] Feature selection review
I. FEATURE SELECTION

Filter methods

Intrinsic properties of data

Computationally simple and fast

Ignore interaction with the classifier

Correlation-based → Select a threshold

Mutual Information

1. FEATURE SELECTION

[1] Feature selection review
1. FEATURE SELECTION

Filter methods

Intrinsic properties of data

Computationally simple and fast

Ignore interaction with the classifier

Correlation-based

Mutual Information

Description

Advantages

Disadvantages

Examples used in our problem

1) Initialization: Set $F \leftarrow \text{“initial set of n features”}; S \leftarrow \text{“empty set.”}$
2) Computation of the MI with the output class: For each $f_i \in F$, compute $I(C; f_i)$.
3) Selection of the first feature: Find the feature $f_i$ that maximizes $I(C; f_i)$; set $F \leftarrow F\{f_i\}$; set $S \leftarrow \{f_i\}$.
4) Greedy selection: Repeat until $|S| = k$.
   a) Computation of the MI between variables: For all pairs $(f_i, f_s)$ with $f_i \in F$ and $f_s \in S$, compute $I(f_i; f_s)$, if it is not yet available.
   b) Selection of the next feature: Choose the feature $f_i \in F$ that maximizes

   $I(C; f_i) - \beta \sum_{f_s \in S} I(f_s; f_i)$

   Set $F \leftarrow F\{f_i\}$; set $S \leftarrow \{f_i\}$.
5) Output the set $S$ containing the selected features.

[1] Feature selection review
1. FEATURE SELECTION

**Embedded methods**

- The search of methods is built into the classifier
- Include interaction with the classifier
- Classifier dependent selection
- Random forest
- GLM using regularisation

**Description**

**Advantages**

**Disadvantages**

**Examples used in our problem**
1. FEATURE SELECTION

Embedded methods
- The search of methods is built into the classifier
- Include interaction with the classifier
- Classifier dependent selection

Feature Importance
- Random forest
  - GLM using regularisation

Description
Advantages
Disadvantages
Examples used in our problem
# 1. Feature Selection

**Description**

The search of methods is built into the classifier

Include interaction with the classifier

Classifier dependent selection

Random forest

**Embedded methods**

GLM using regularisation

\[
\min_{w \in \mathbb{R}^p} \frac{1}{n} \| \hat{X}w - \hat{Y} \|^2 + \lambda (\alpha \| w \|_1 + (1 - \alpha) \| w \|_2^2), \alpha \in [0, 1]
\]

---

[1] Feature selection review
1. BUILDING A GRAPH

- Feature selection methods are applied

- For each question we have a set of predictors
1. BUILDING A GRAPH

- Feature selection methods are applied

- For each question we have a set of predictors

- Together they form a graph
1. BUILDING A GRAPH

- Goal: find a ‘minimal’ subgraph with minimum number of predictors
1. BUILDING A GRAPH

- Goal: find a ‘minimal’ subgraph with minimum number of predictors
1. BUILDING A GRAPH

- Goal: find a ‘minimal’ subgraph with minimum number of predictors
1. BUILDING A GRAPH

- Goal: find a ‘minimal’ subgraph with minimum number of predictors
1. BUILDING A GRAPH

- Goal: find a ‘minimal’ subgraph with minimum number of predictors
I. BUILDING A GRAPH

- Goal: find a ‘minimal’ subgraph with minimum number of predictors
I. BUILDING A GRAPH

- Goal: find a ‘minimal’ subgraph with minimum number of predictors
1. BUILDING A GRAPH

- Goal: find a ‘minimal’ subgraph with minimum number of predictors
- It can be formalised as a matrix problem
1. BUILDING A GRAPH

- For each type of feature selection method we can build one graph
## 1. FEATURE SELECTION - REVIEW

### Graph construction

<table>
<thead>
<tr>
<th><strong>Method 1 (non-heuristic)</strong></th>
<th><strong>Method 2</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Choose model</td>
<td>Heuristic graph construction</td>
</tr>
<tr>
<td>1. Choose a model</td>
<td>Random graph construction</td>
</tr>
<tr>
<td>2. Study the algorithm. Can it be adapted to the problem?</td>
<td>e.g. dropping “Performance” questions</td>
</tr>
<tr>
<td>3. Set the parameters</td>
<td>e.g. choosing 10 random questions</td>
</tr>
<tr>
<td>Separate data 70%-30%</td>
<td>Feature selection</td>
</tr>
<tr>
<td>For each model:</td>
<td>For each sub-problem:</td>
</tr>
<tr>
<td>4. Select training data (70%)</td>
<td>7. Perform cross-validation to avoid over-fitting</td>
</tr>
<tr>
<td>5. Transform the data (1-7 or 0-1)</td>
<td>8. Select features that minimise the error</td>
</tr>
<tr>
<td>6. Divide the problem in sub-problems</td>
<td></td>
</tr>
<tr>
<td>Finally:</td>
<td>Finally:</td>
</tr>
<tr>
<td>9. Construct a graph with all the questions</td>
<td>10. Transform the graph into a tree</td>
</tr>
<tr>
<td>11. Calculate the final number of questions to be included in test</td>
<td></td>
</tr>
</tbody>
</table>
2. MULTI-CLASSIFICATION SUBPROBLEM

- Different types of models:
  - Random model
  - Generalised Linear models
  - Random forest
  - Support Vector Machines (linear kernel)
  - Some basic Neural Networks
2. REVIEW

Model construction

Choose tree
- Non-heuristic tree
- Heuristic tree
- Random tree

Choose model
- Fair model
- Gaussian linear model
- Binomial model
- Random forest
- Support v. machine

Perform Monte Carlo Cross-Validation

For each loop:
1. Simulate the results of the test
   - Ask N questions to the user
   - For each of the remaining 90-N questions:
     - train model (using 70%)
     - predict answer (using 30%)
     - calculate error

Finally:
2. Calculate the average error:
   - Answers absolute error
2. RESULTS

- Error vs. number of predictors
- Why are they performing in a similar way?
- Data is too noisy?
2. RESULTS

- Error vs. number

- Why are they performing in a similar way?

- Data is too noisy?
OTHER APPROACHES

- Information theory approach (no ML but useful)
- We want to find **redundancy** in data
- What is the **most redundant** set of questions

- Entropy!

\[ H(X) = - \sum_{i=1}^{n} p(x_i) \log p(x_i). \]

- \( H(X) = 0 \). High redundancy
- \( H(X) \) max. No redundancy
NEURAL NETWORKS

- Again, space $Q = P \cup S$
- Fix $s$ in $S$
- Classification problem with 4 classes
- We want a 0/1 output
- I. 4 NN, one output node for each class
- II. 1 NN, four output nodes

[3] Multi-class pattern classification using NN
Again, space $Q = P \cup S$

Fix $s$ in $S$

Classification problem with 4 classes

We want a 0/1 output

1. 4 NN, one output node for each class

2. 1 NN, four output nodes

Problem with unbalanced classes:

zoom-in regions with lower numbers of points (discrete space)
The problem is still open

Although, there are other steps omitted here that have been useful

If it cannot be solved, how to prove it?
REVIEW OF THE PROBLEM

- Dataset: collection of user responses (~30000)
- In our case the test has 90 questions with 4 possible answers:
  - **Drawback**: sparse data.
  - **Drawback**: data concentrated in few cells
  - **Drawback**: users perceive the scale in different way
  - **Drawback**: users tend to choose high values

\[ Q = P \cup S \]
CONCLUSIONS

Psychological test are relents in industry

Special type of data

How to extend traditional DM techniques to deal with these type of data?
SOME REFERENCES

- Many interesting ones…


[3] (Multiclass. NN)

[4] (Unbalanced data)

[5] (high-dimensional NN)
Thank you! Questions?