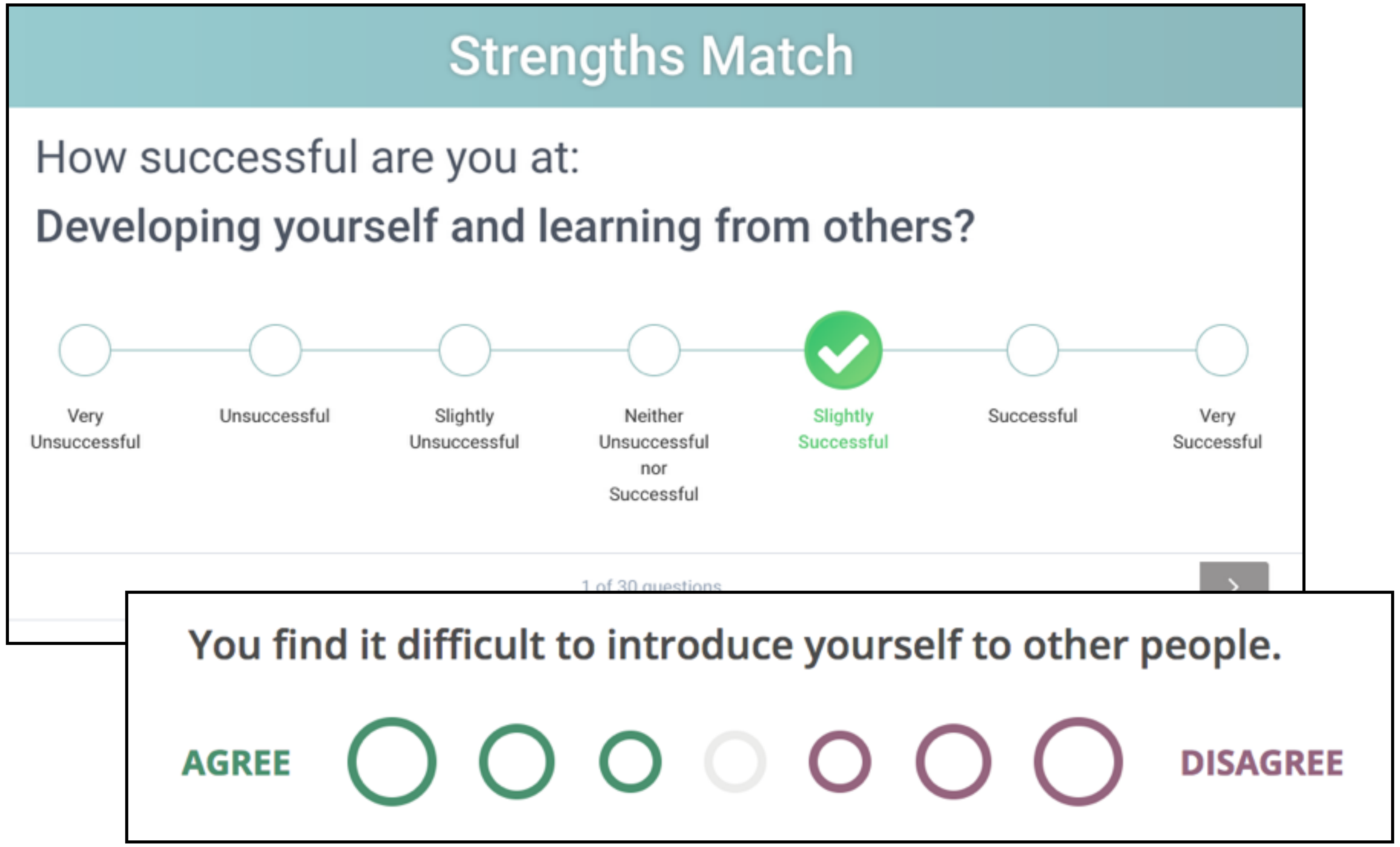
SHORTING NUMBER OF QUESTIONS IN LONG PSYCHOLOGICAL QUESTIONNAIRES

THE
STRENGTHS
EXPERTS

THE CONTEXT



- Psychological test 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:

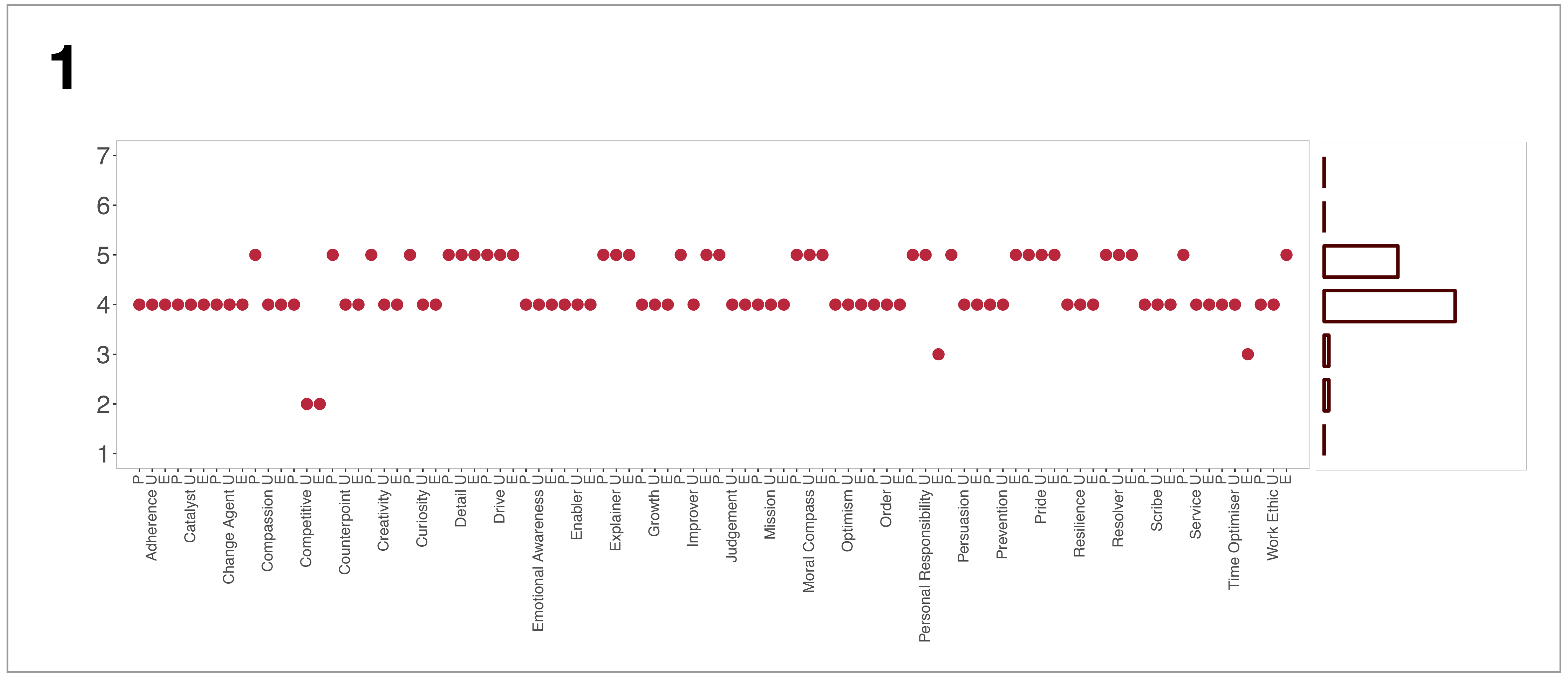
  1 2 3 4

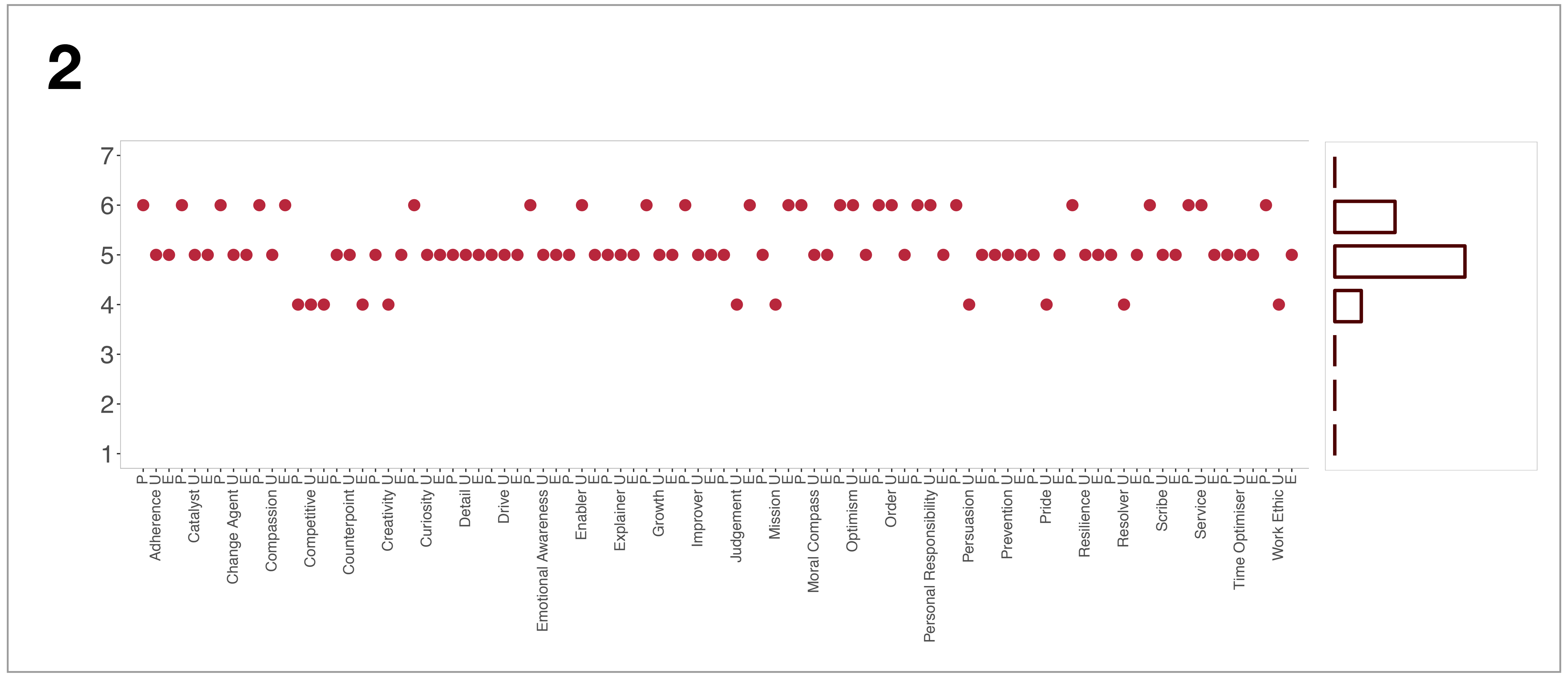
- Drawbacks?

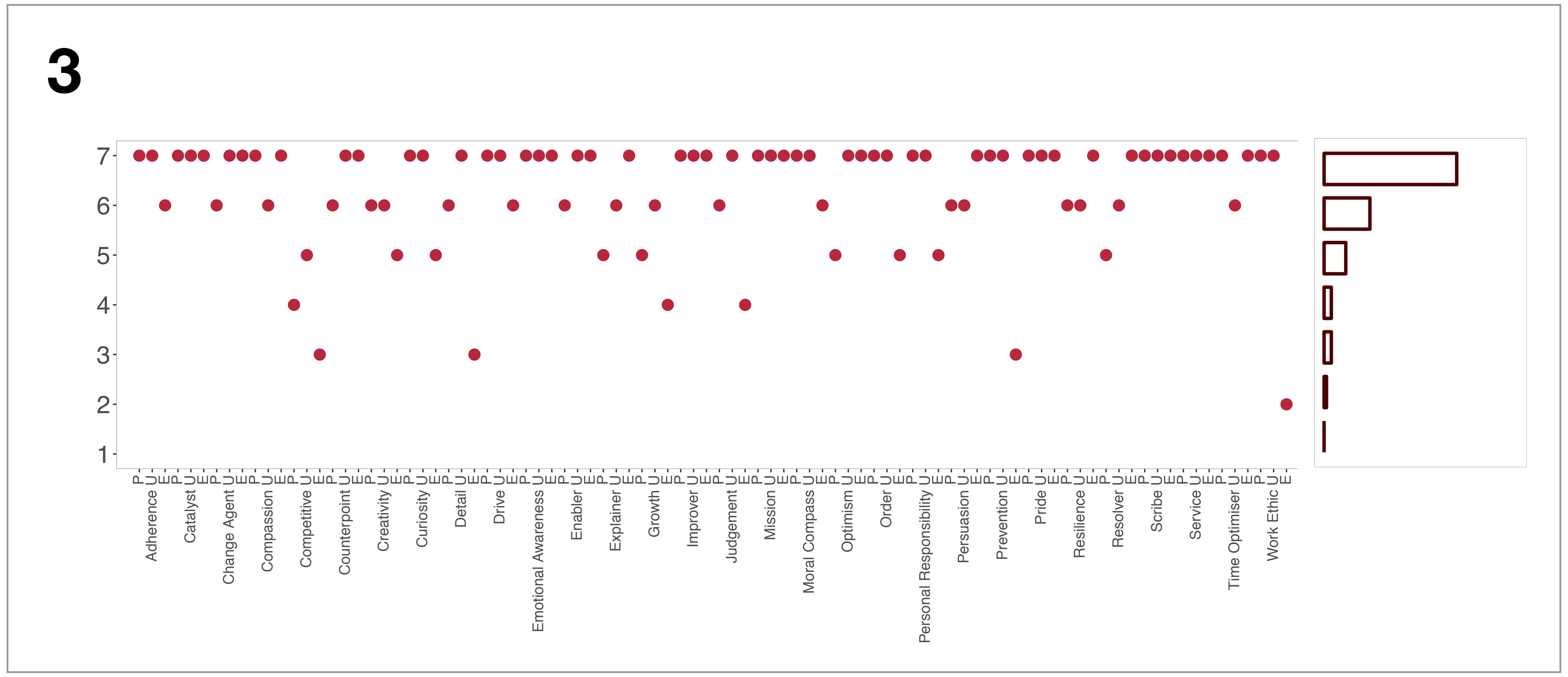
'ar	Var 4	Var 5	Var 6	Var 7	Var 8	Var 9	Var 10	Var 11	Var 12	Var 13	Var 14	Var 15	Var 16	Var 17	Var 18	Var 19	Va 20
1	4	4	3	3	1	3	1	3	3	3	4	1	3	1	3	1	[illegible]
1	1	4	2	2	3	3	2	3	1	1	3	4	3	3	3	2	[illegible]
1	1	1	4	3	4	3	3	3	3	2	3	1	3	3	1	1	[illegible]
2	3	2	1	1	3	1	1	1	3	1	1	1	2	1	3	1	[illegible]
1	1	2	3	1	3	3	2	1	4	1	1	2	1	3	1	3	[illegible]
1	2	1	1	3	2	2	2	2	2	3	2	1	1	3	1	1	[illegible]
2	3	4	1	1	1	3	3	4	3	3	3	4	2	2	3	2	[illegible]
1	3	2	1	1	3	3	3	2	3	3	3	1	3	1	1	1	[illegible]
2	2	1	1	1	1	3	1	3	3	1	2	1	1	3	3	3	[illegible]
4	3	4	3	2	3	3	2	1	3	3	1	1	3	2	3	3	[illegible]
2	2	3	3	3	1	3	3	3	3	4	1	1	3	1	1	2	[illegible]
3	1	1	3	1	3	1	2	3	3	2	3	1	4	4	1	4	[illegible]
3	3	4	3	1	4	3	3	3	3	3	3	2	4	2	2	1	[illegible]
2	1	3	3	3	3	3	3	2	3	3	1	1	3	2	3	1	[illegible]
1	4	1	2	2	3	3	1	1	3	4	1	1	1	3	1	4	[illegible]

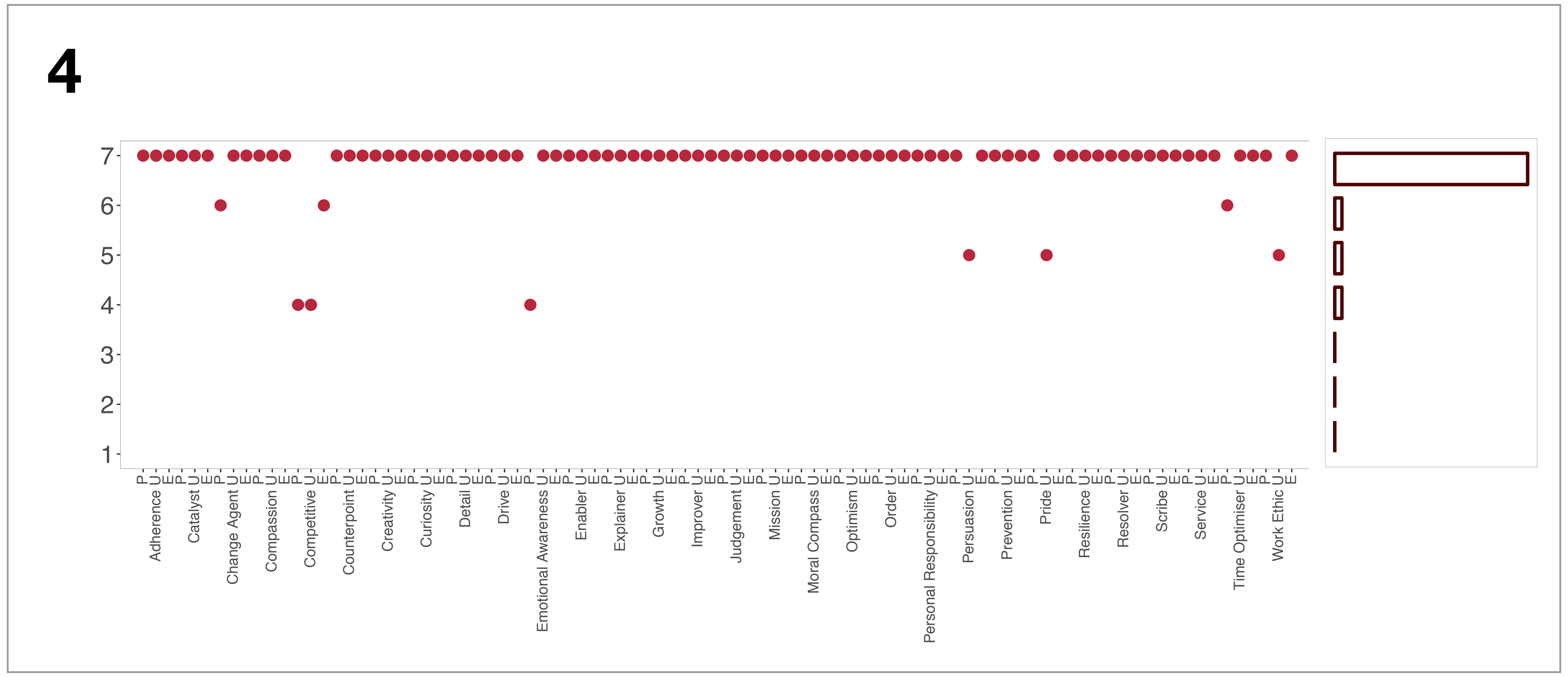
Drawback: users perceive the scale in different way

Drawback: users tend to choose high values







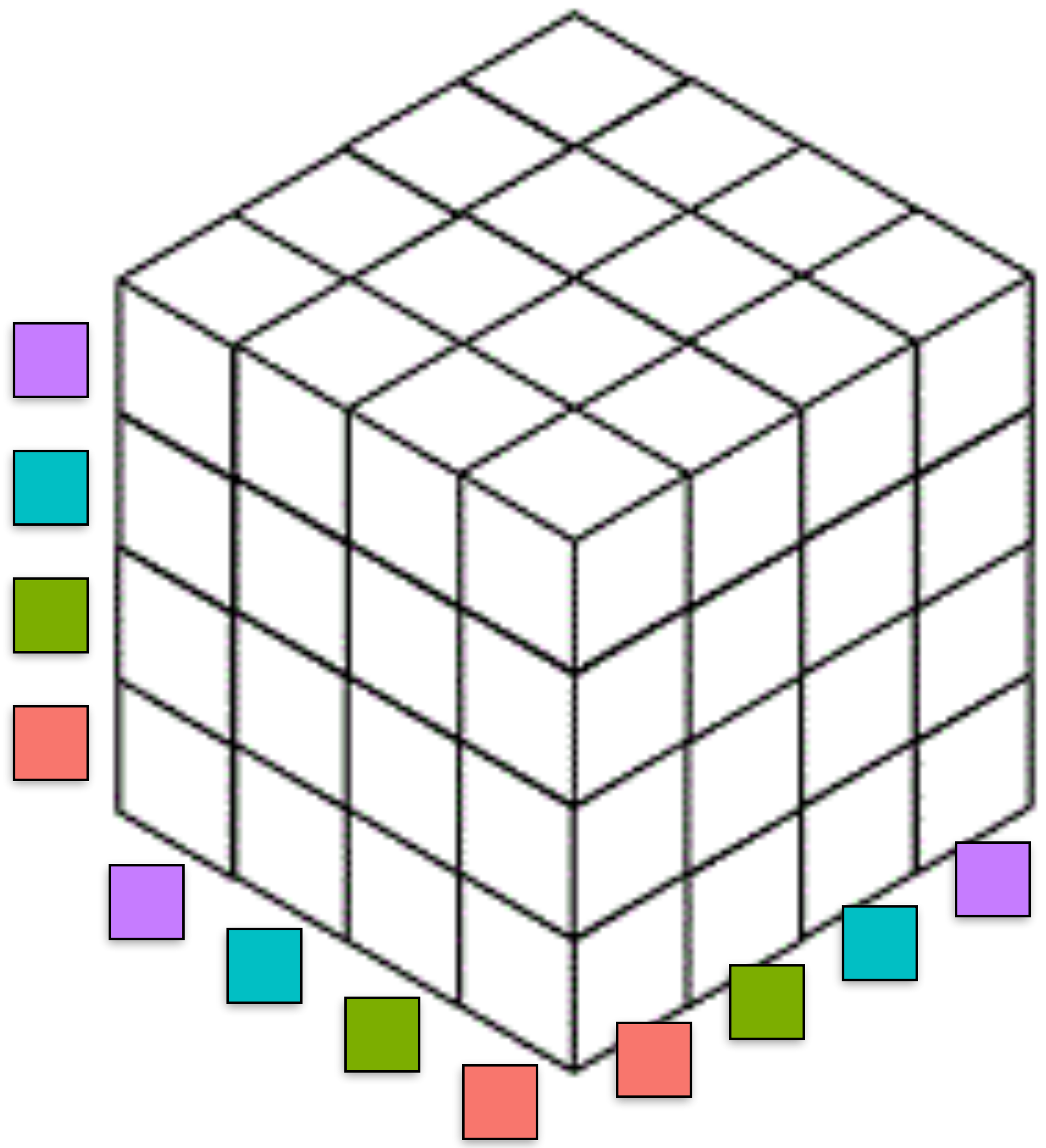


THE PROBLEM

- Dataset can be mapped into

$$\{1, 2, 3, 4\}^{90}$$

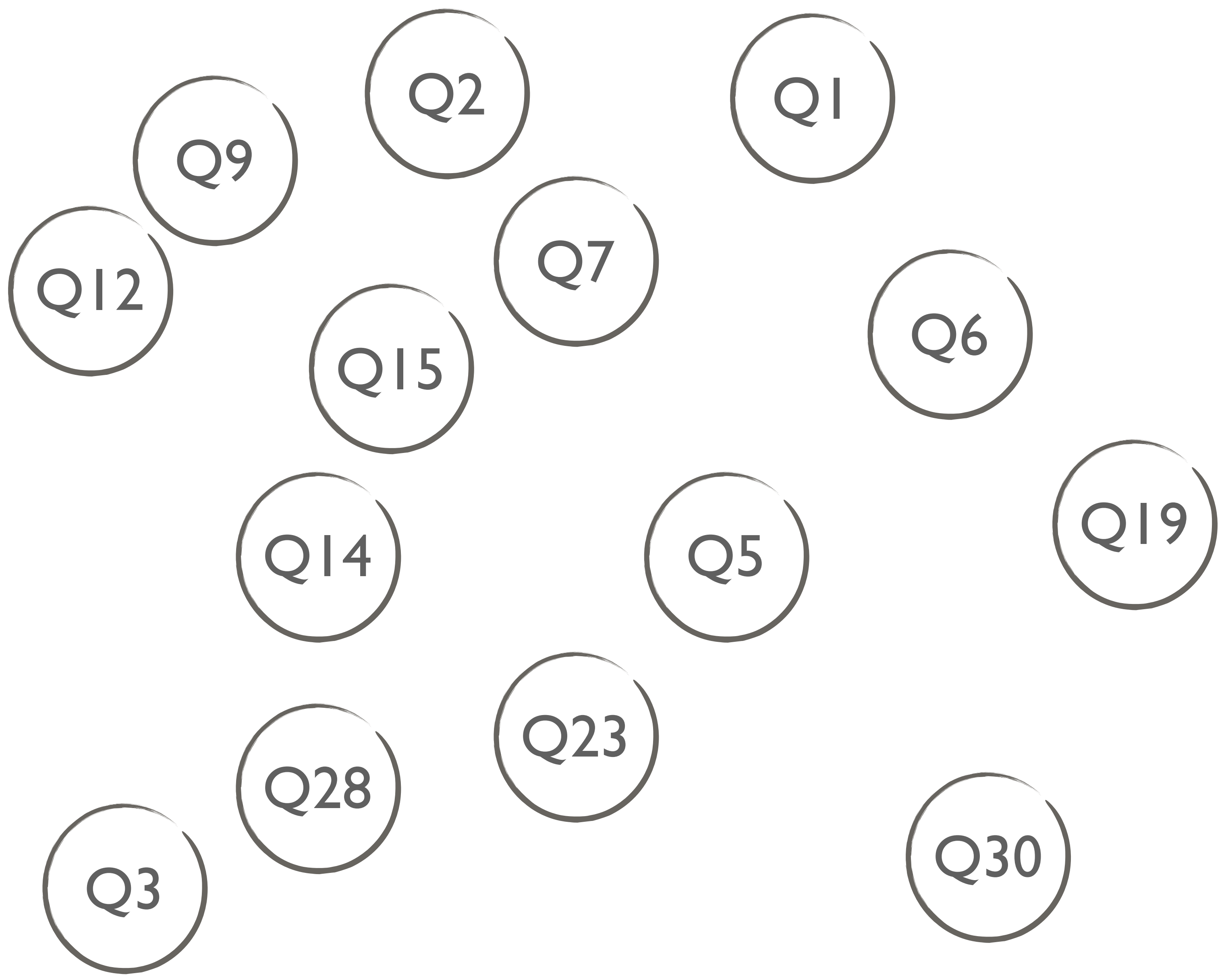
- How does it look like?



Drawback: sparse data.

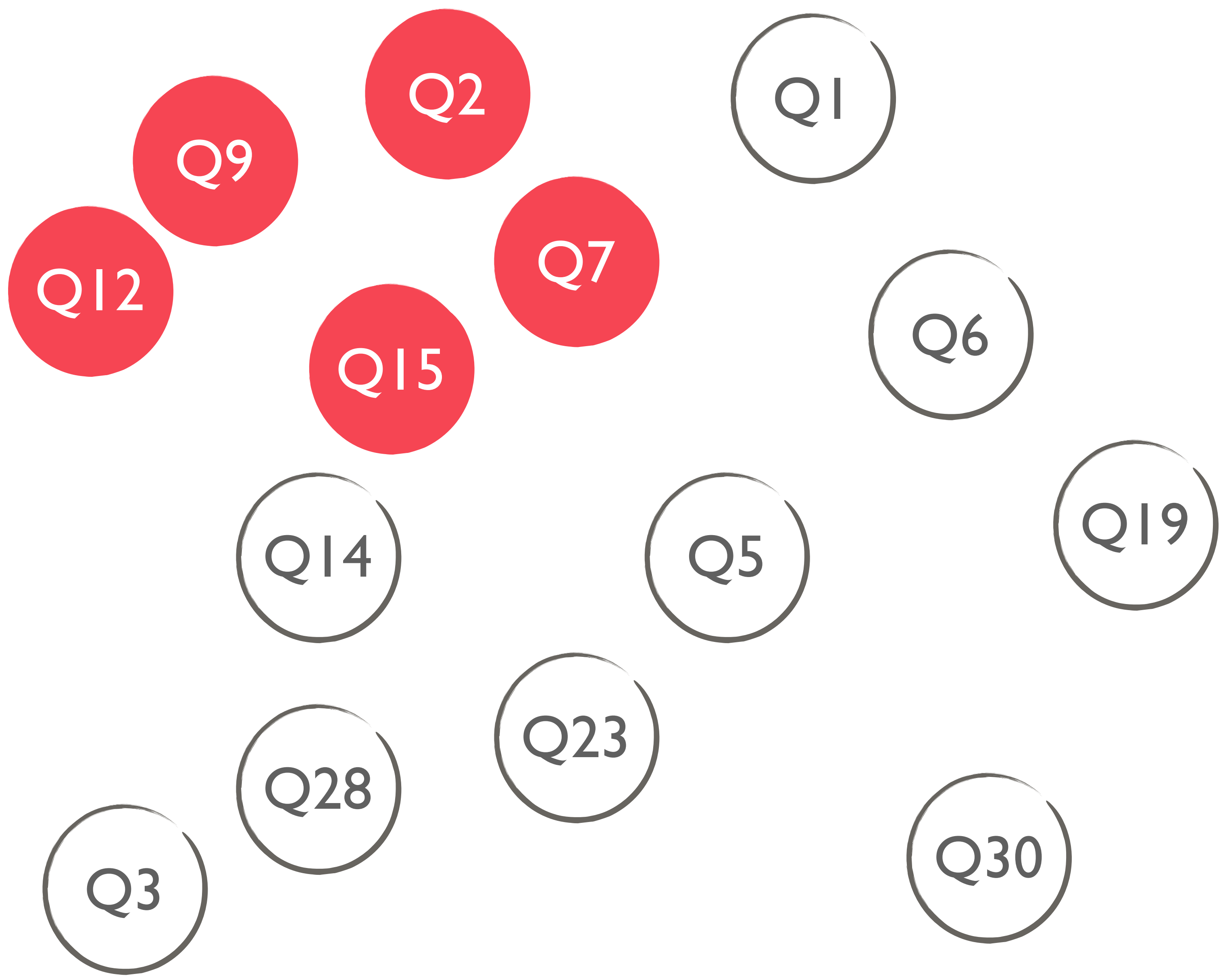
Drawback: data concentrated in few cells

HOW TO SOLVE IT



- Divide the problem:

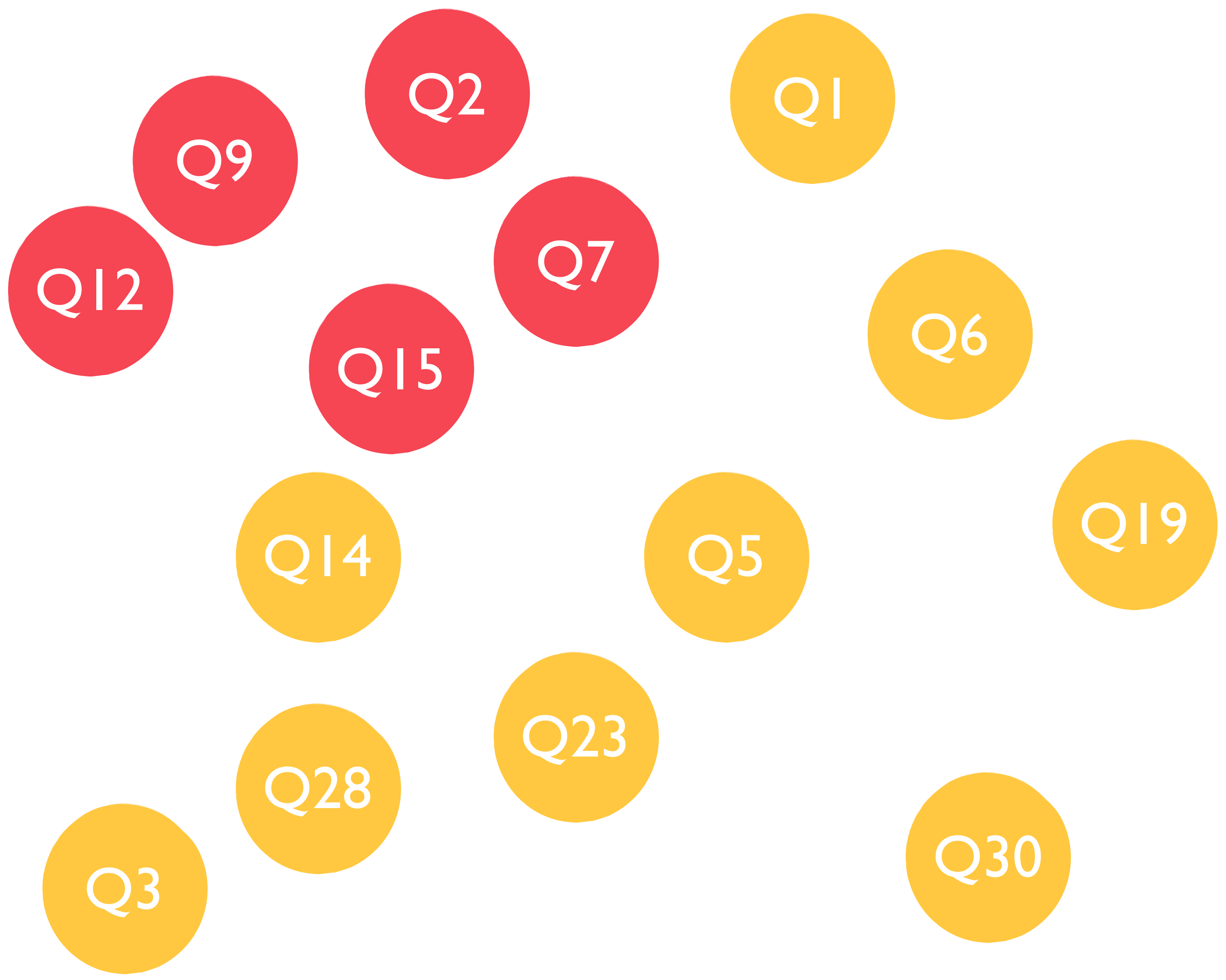
HOW TO SOLVE IT



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

$$Q = P\dot{\cup}S$$

HOW TO SOLVE IT



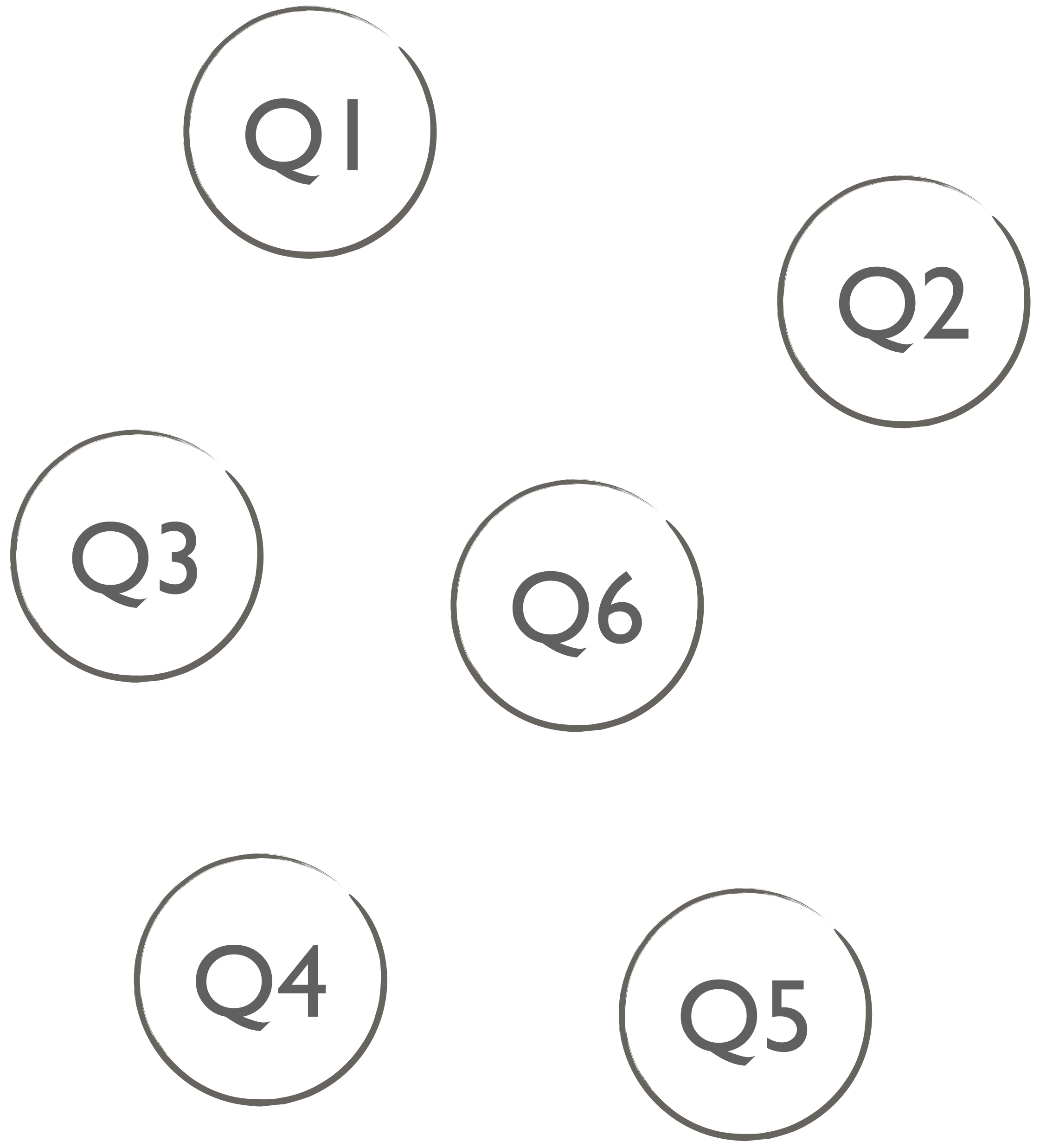
- Divide the problem:
- 1. Find a set of predictors and a set of questions to be predicted

$$Q = P \dot{\cup} S$$

- 2. Predict P using S

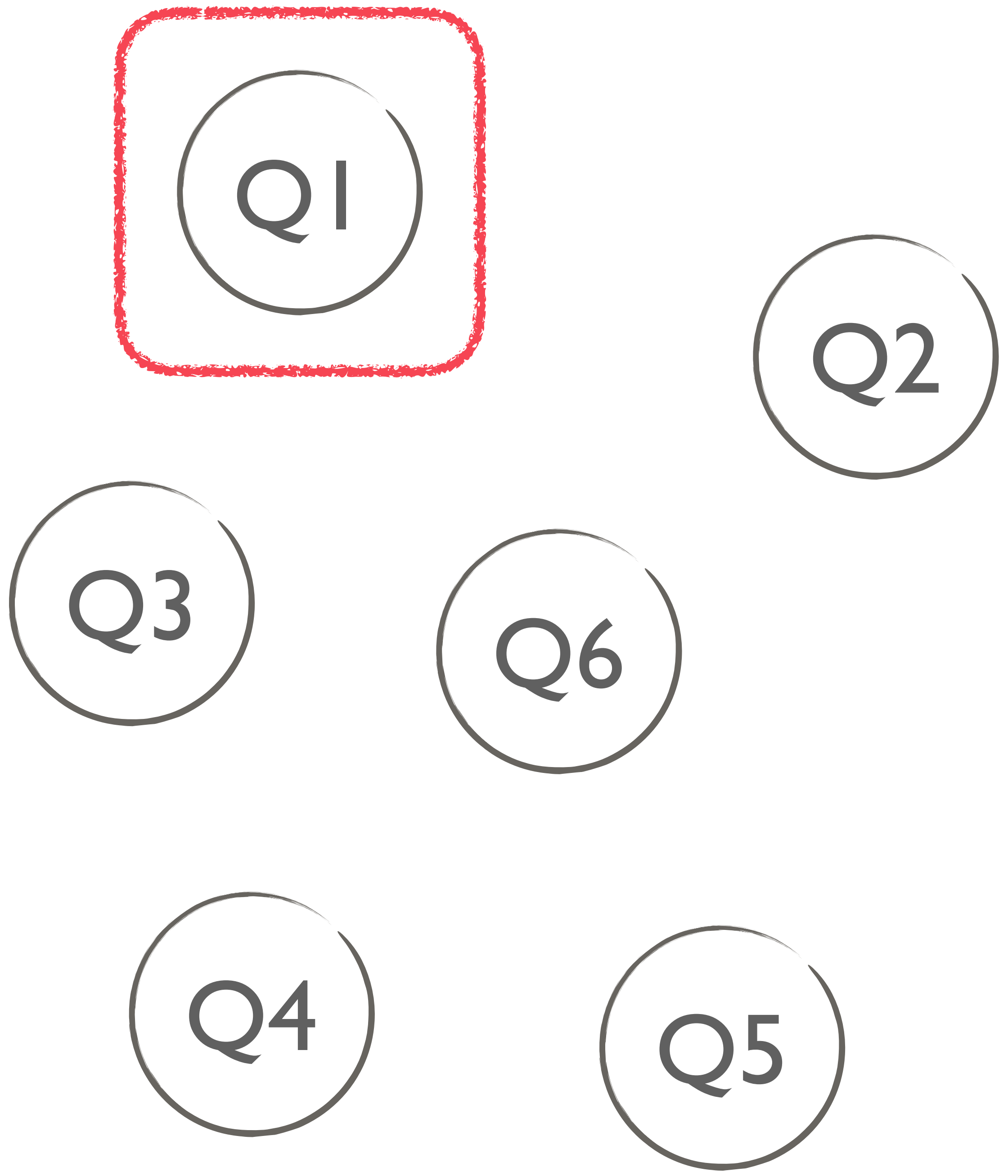
1. FEATURE SELECTION

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



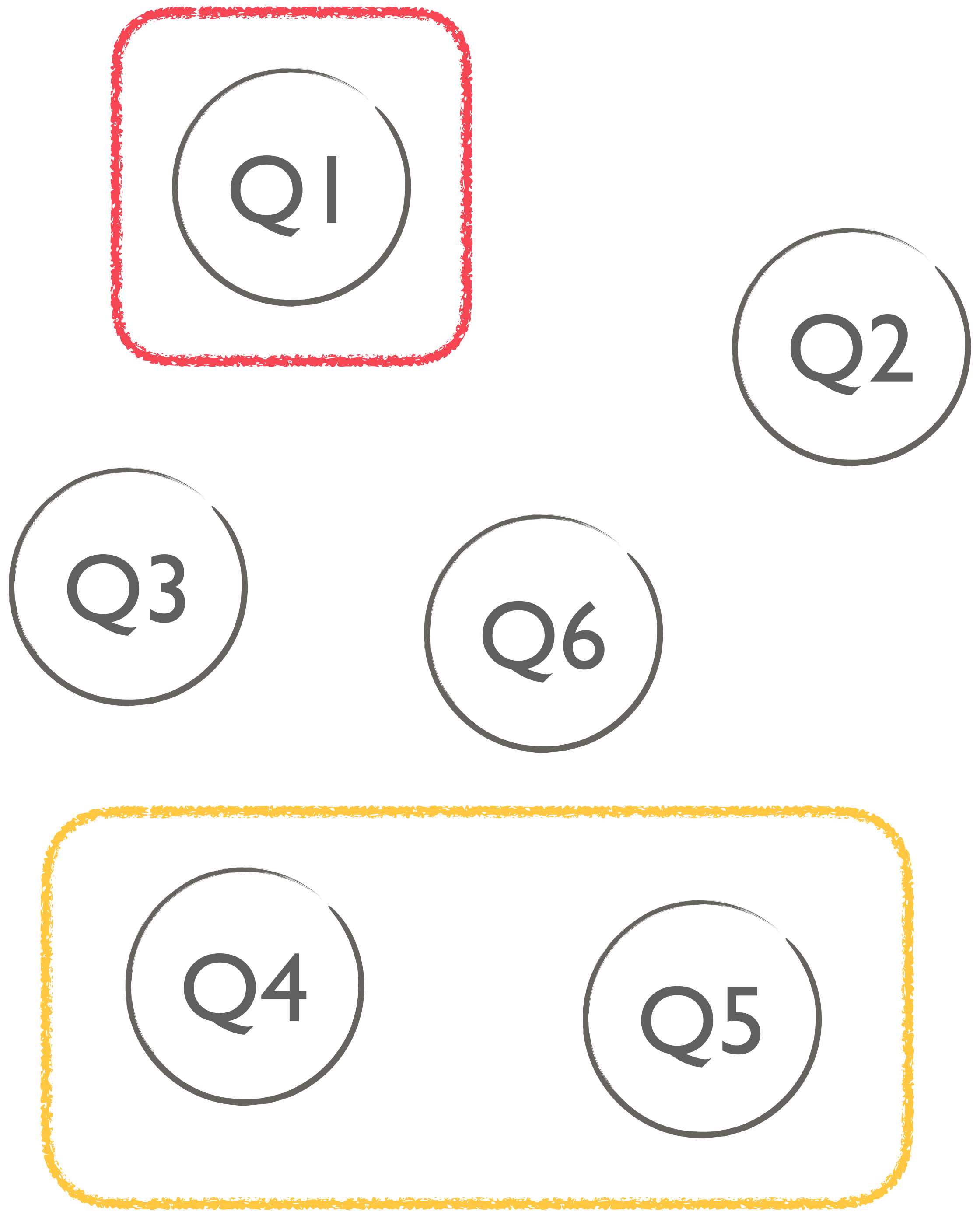
1. FEATURE SELECTION

- 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



1. FEATURE SELECTION

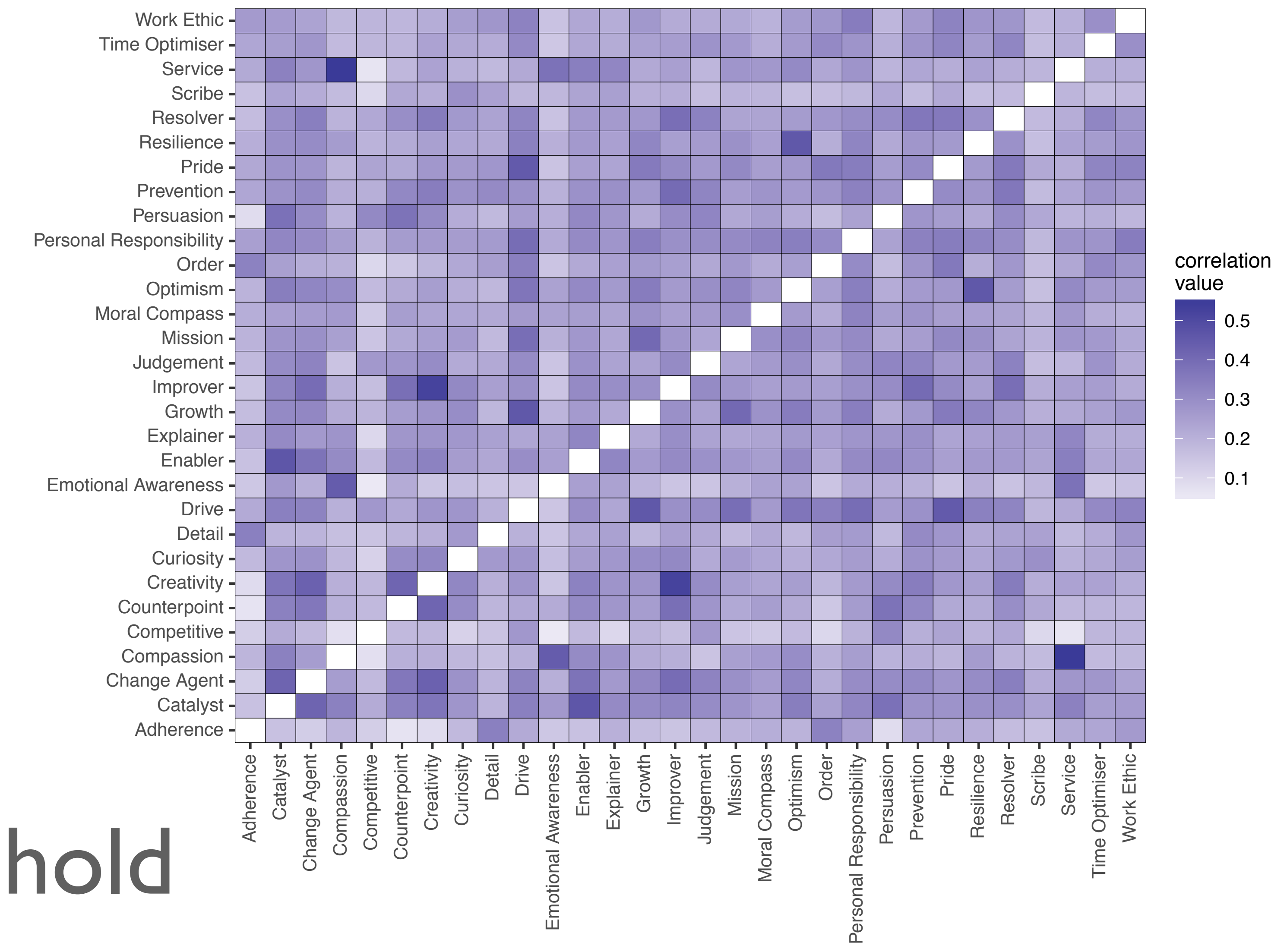
	Filter methods
Description	Intrinsic properties of data
Advantages	Computationally simple and fast
Disadvantages	Ignore interaction with the classifier
Examples used in our problem	Correlation-based Mutual Information

[1] Feature selection review

1. 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** → Select a threshold Mutual Information



1. 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) Initialization: Set $F \leftarrow$ "initial set of n features"; S $\leftarrow$ "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 $F \leftarrow F \backslash \{f_i\}$; 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 $F \leftarrow F \backslash \{f_i\}$; set $S \leftarrow \{f_i\}$.
5) Output the set S containing the selected features.

[1] Feature selection review

1. FEATURE SELECTION

Description

Advantages

Disadvantages

Examples used in our problem

Embedded methods

The search of methods is built into the classifier

Include interaction with the classifier

Classifier dependent selection

Random forest

GLM using regularisation

[1] Feature selection review

1. FEATURE SELECTION

Description

Advantages

Disadvantages

Examples used in our problem

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

I. FEATURE SELECTION

Description

Advantages

Disadvantages

Examples used in our problem

Embedded methods

The search of methods is built into the classifier

Include interaction with the classifier

Classifier dependent selection

Random forest

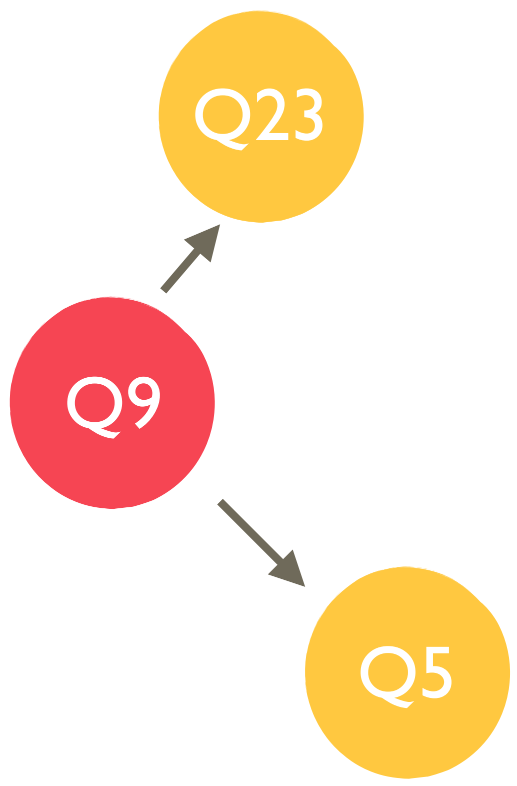
GLM using regularisation

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

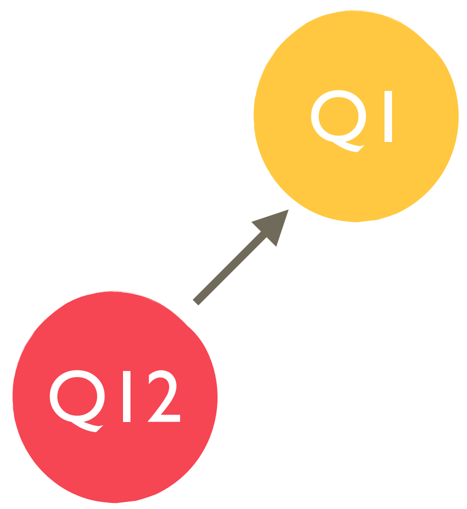
[1] Feature selection review

1. BUILDING A GRAPH

- Feature selection methods are applied

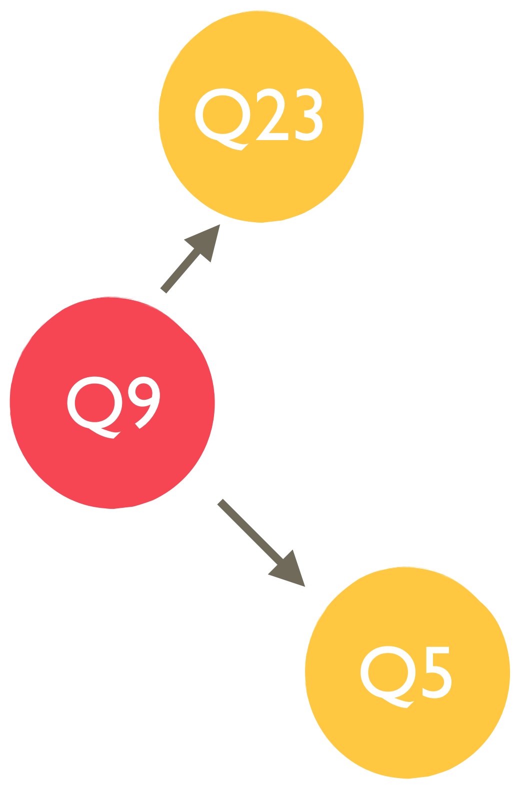


- For each question we have a set of predictors

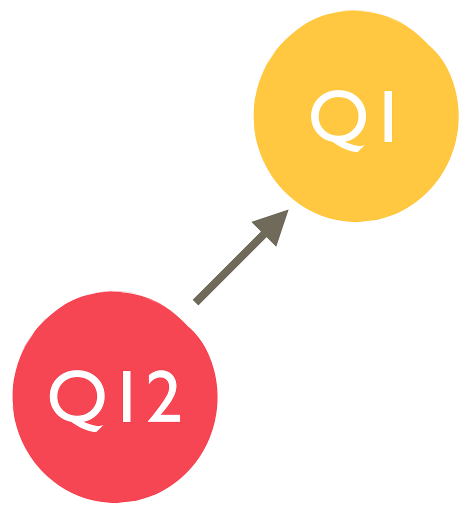


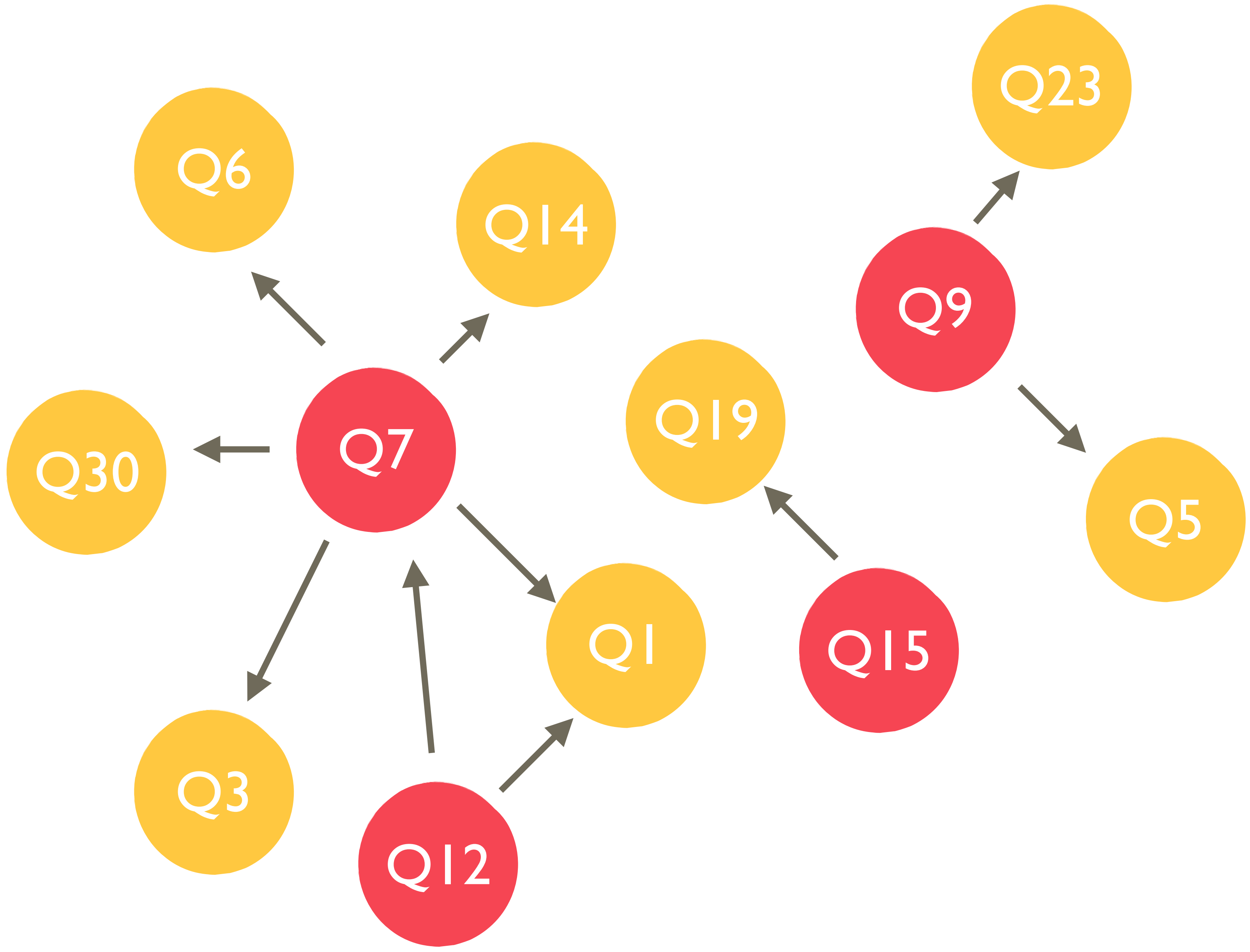
1. BUILDING A GRAPH

- Feature selection methods are applied



- For each question we have a set of predictors

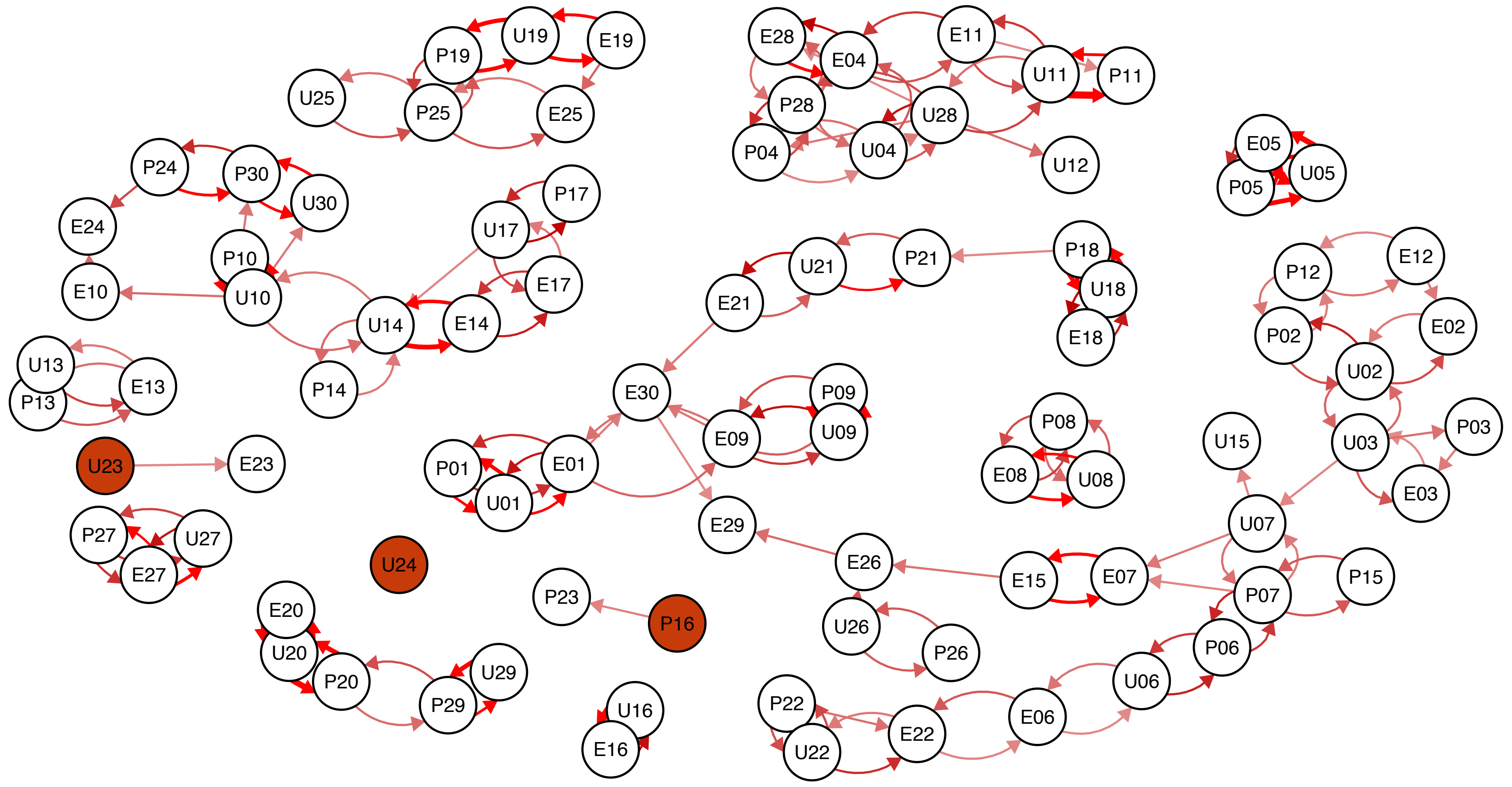




- Together they form a graph

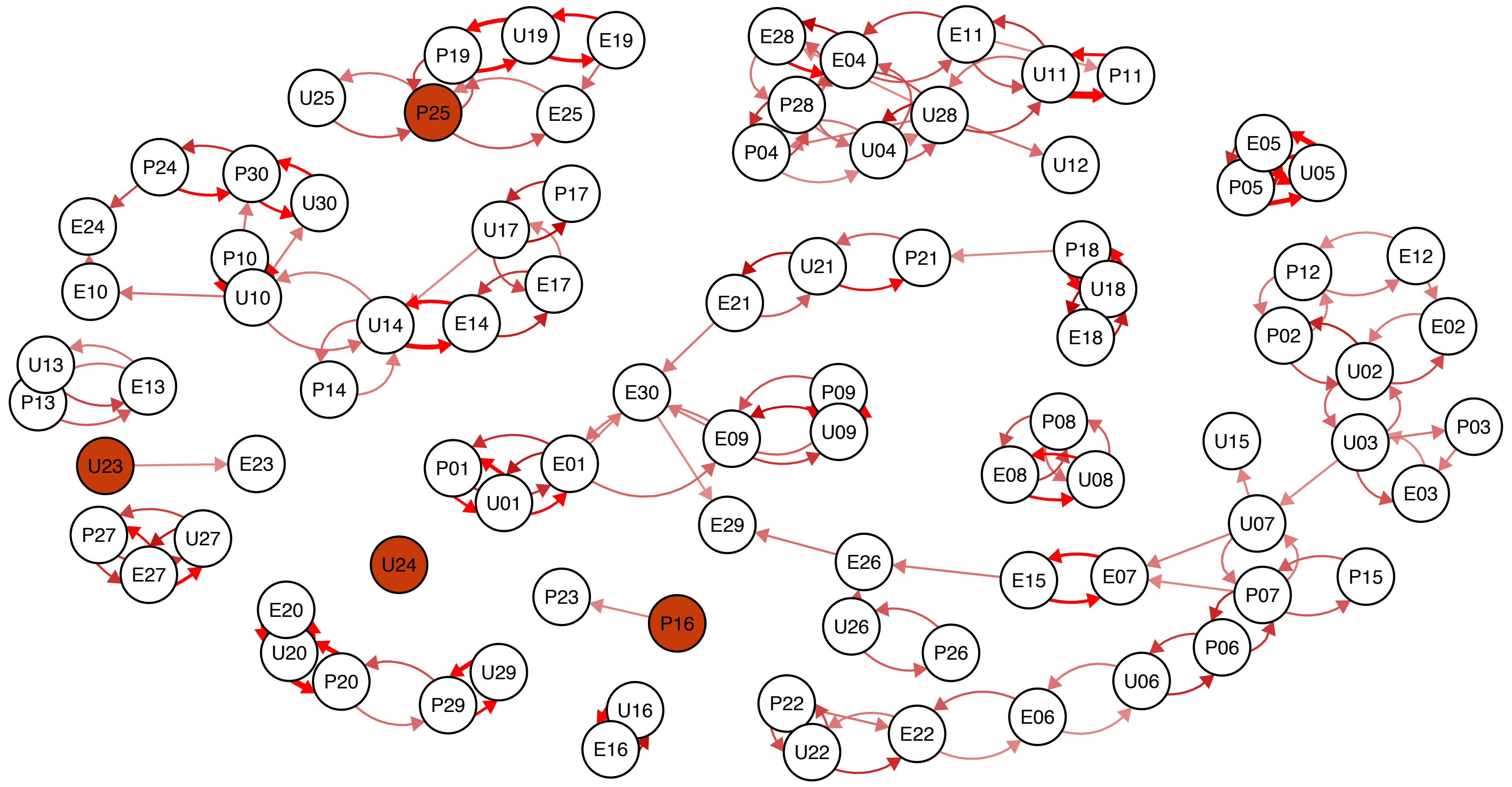
1. BUILDING A GRAPH

- Goal: find a 'minimal' subgraph with minimum number of predictors



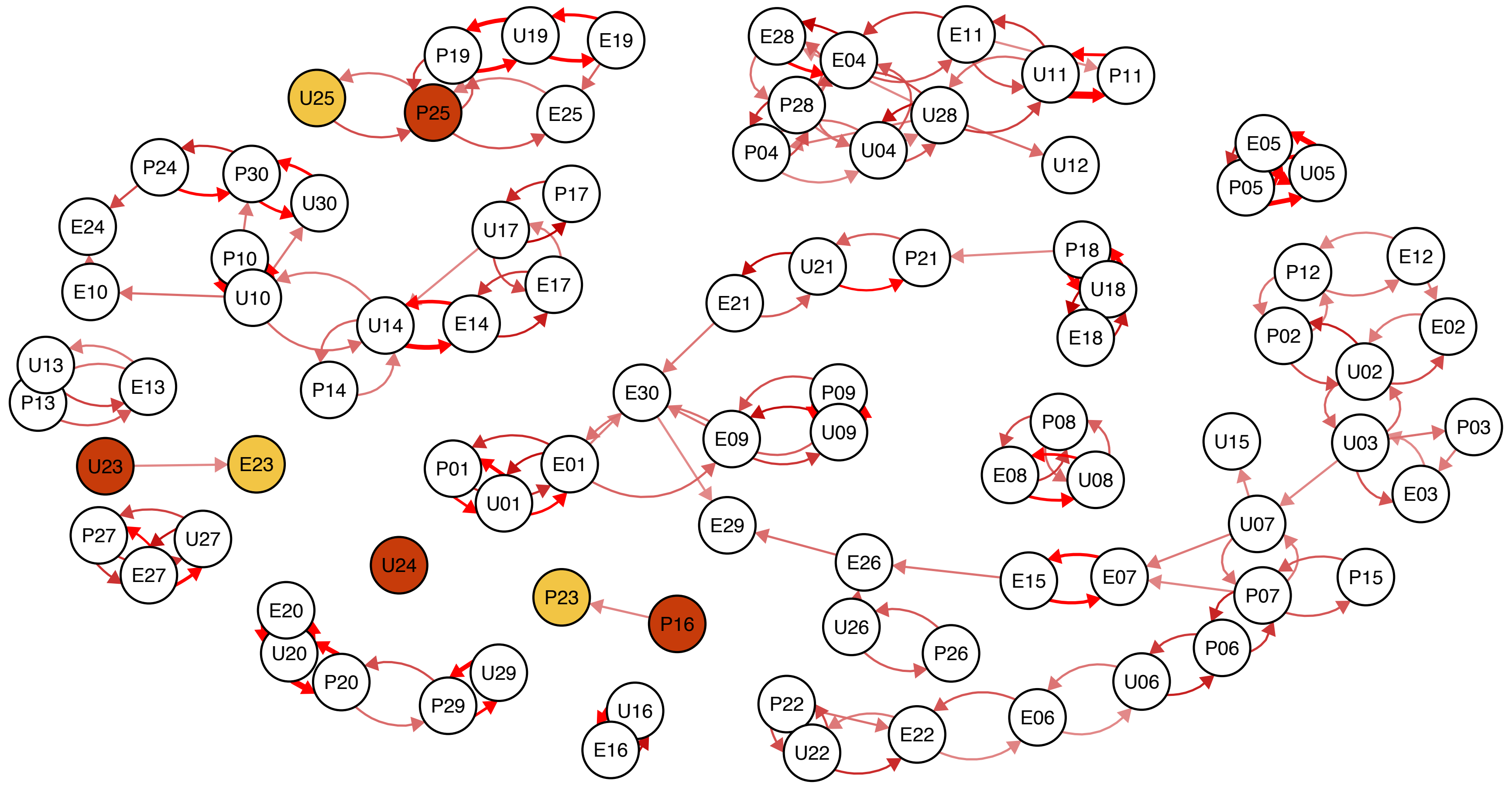
I. BUILDING A GRAPH

- Goal: find a ‘minimal’ subgraph with minimum number of predictors



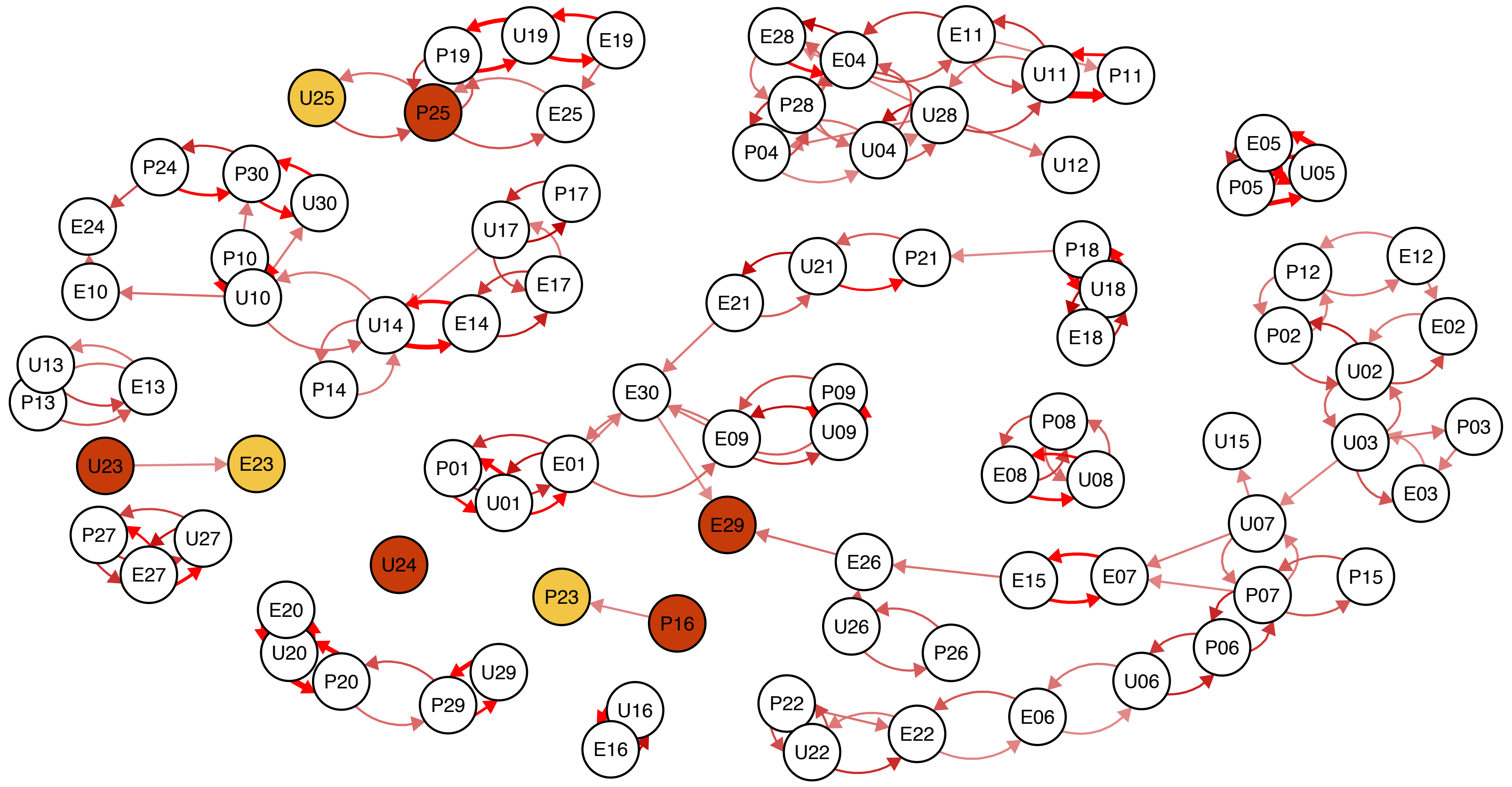
1. BUILDING A GRAPH

- Goal: find a ‘minimal’ subgraph with minimum number of predictors



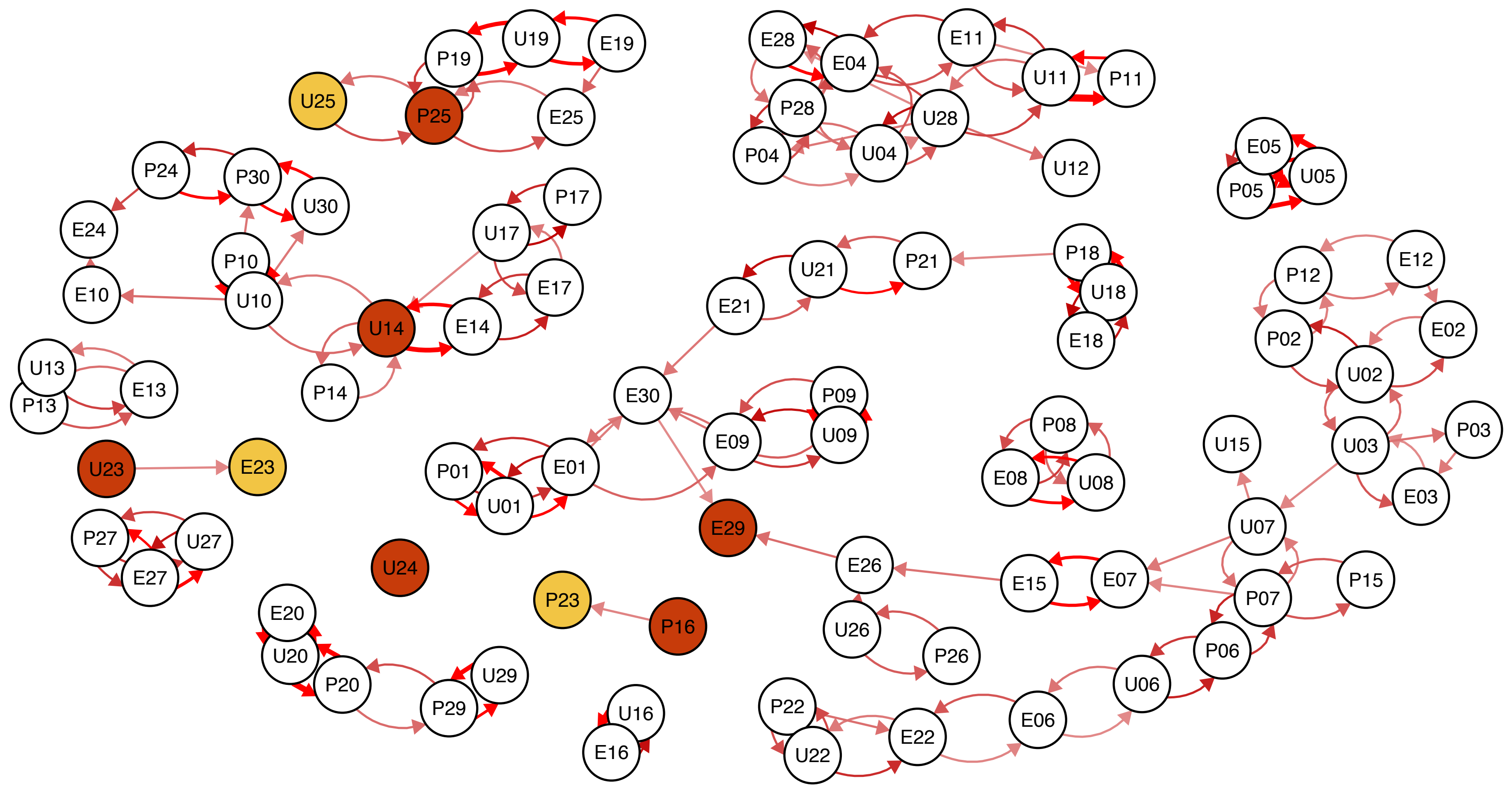
1. BUILDING A GRAPH

- Goal: find a 'minimal' subgraph with minimum number of predictors



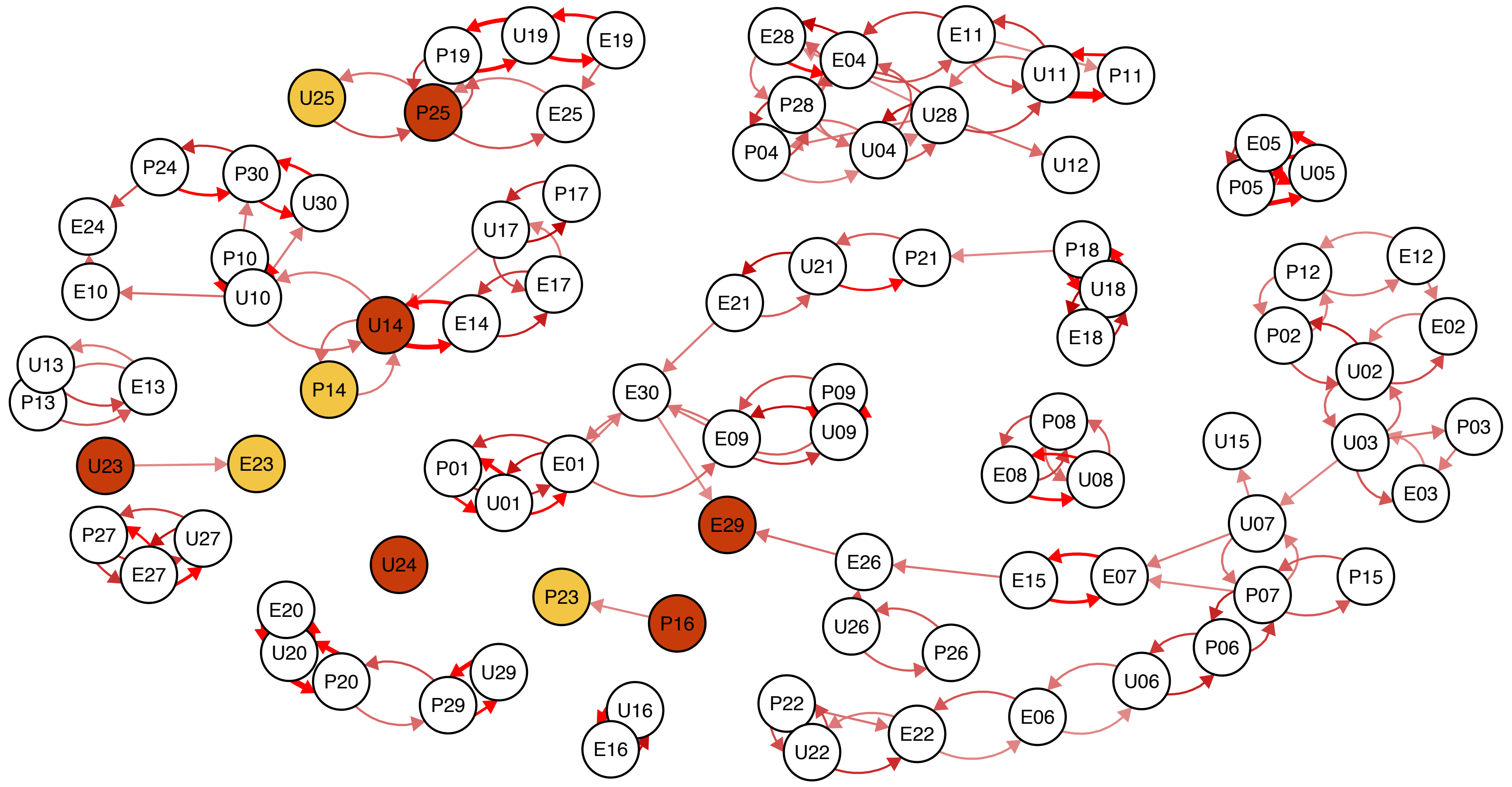
1. BUILDING A GRAPH

- Goal: find a ‘minimal’ subgraph with minimum number of predictors



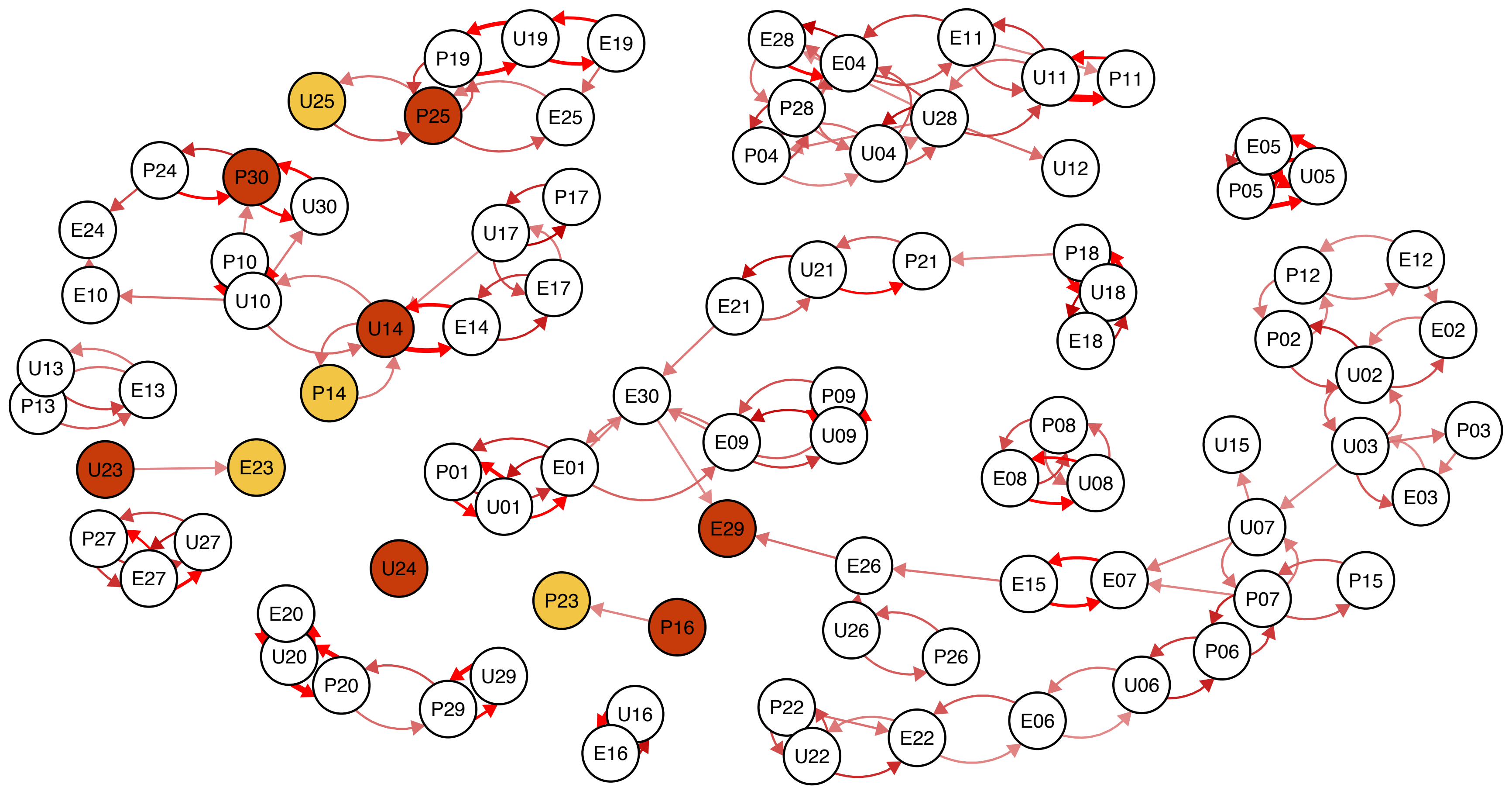
1. BUILDING A GRAPH

- Goal: find a ‘minimal’ subgraph with minimum number of predictors



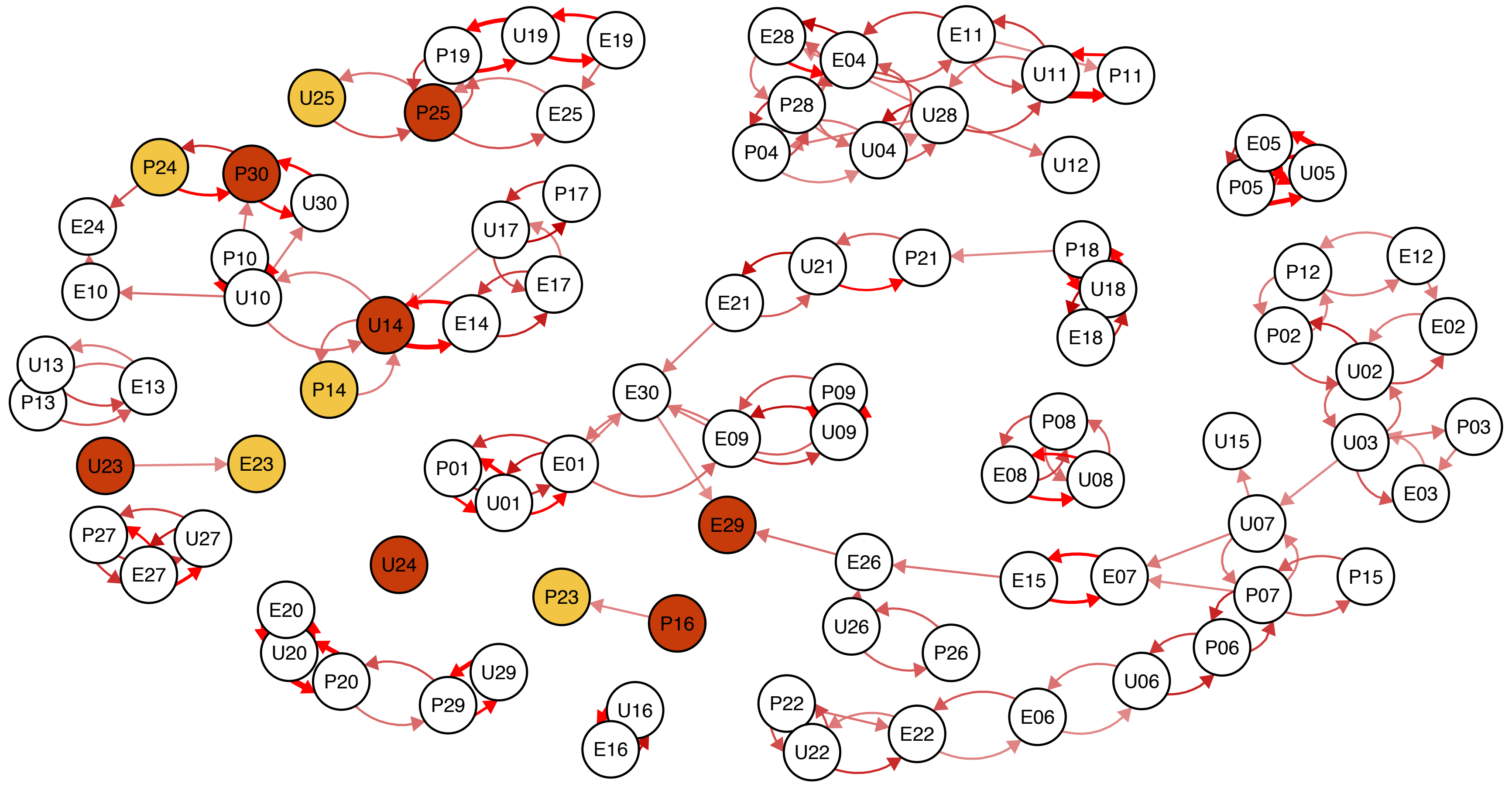
1. BUILDING A GRAPH

- Goal: find a ‘minimal’ subgraph with minimum number of predictors



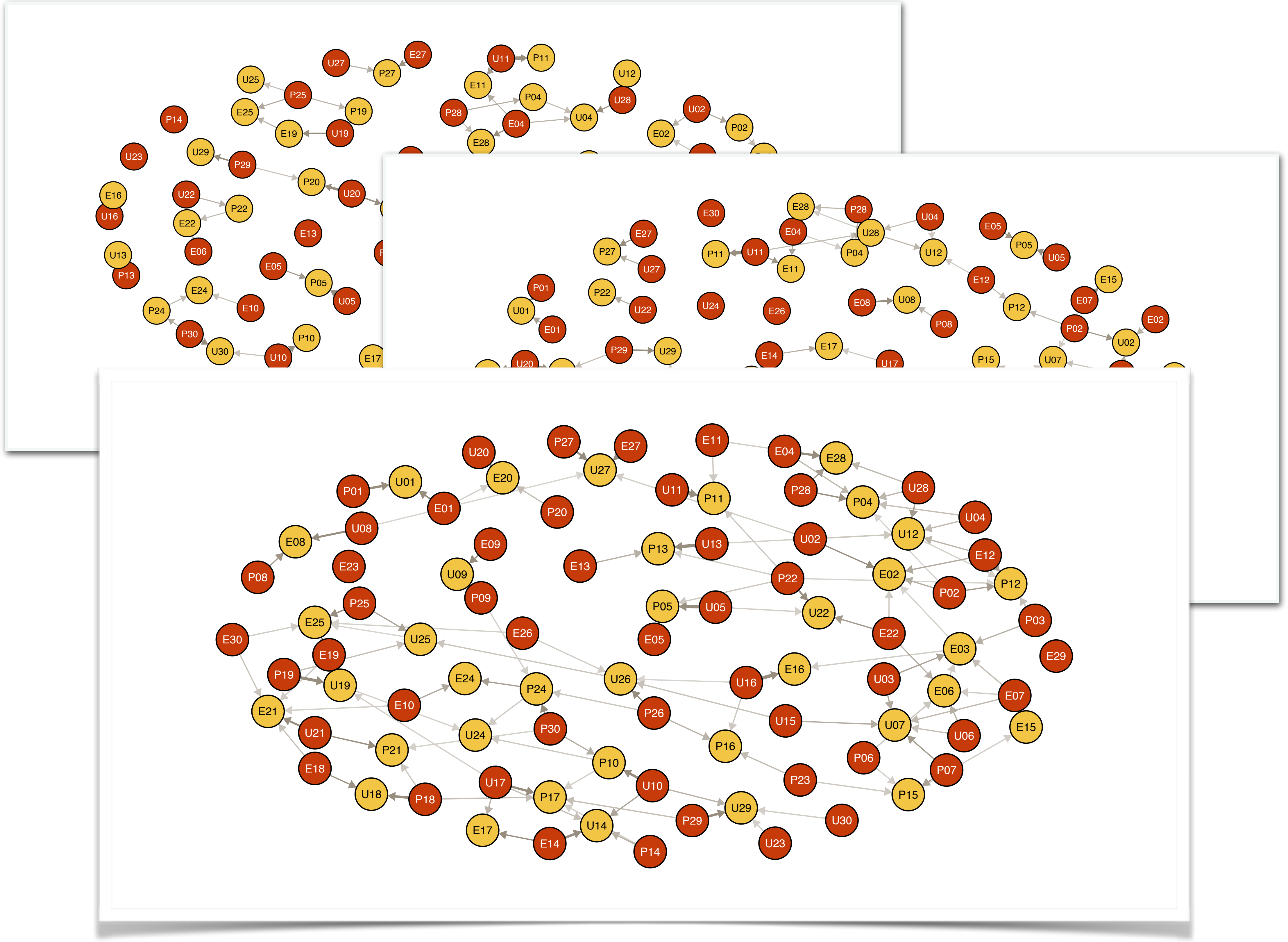
1. BUILDING A GRAPH

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



1. BUILDING A GRAPH

- For each type of feature selection method we can build one graph



I. FEATURE SELECTION - REVIEW

Graph construction

Method 1 (non-heuristic)

Choose model

1. Choose a model
2. Study the algorithm. Can it be adapted to the problem?
3. Set the parameters

Separate data 70%-30%

For each model:

4. Select training data (70%)
5. Transform the data (1-7 or 0-1)
6. Divide the problem in sub-problems

Feature selection

For each sub-problem:

7. Perform cross-validation to avoid over-fitting
8. Select features that minimise the error

Tree pruning

Finally:

9. Construct a graph with all the questions
10. Transform the graph into a tree
11. Calculate the final number of questions to be included in test

Method 2

Heuristic graph construction

e.g. dropping “Performance” questions

Random graph construction

e.g. choosing 10 random questions

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

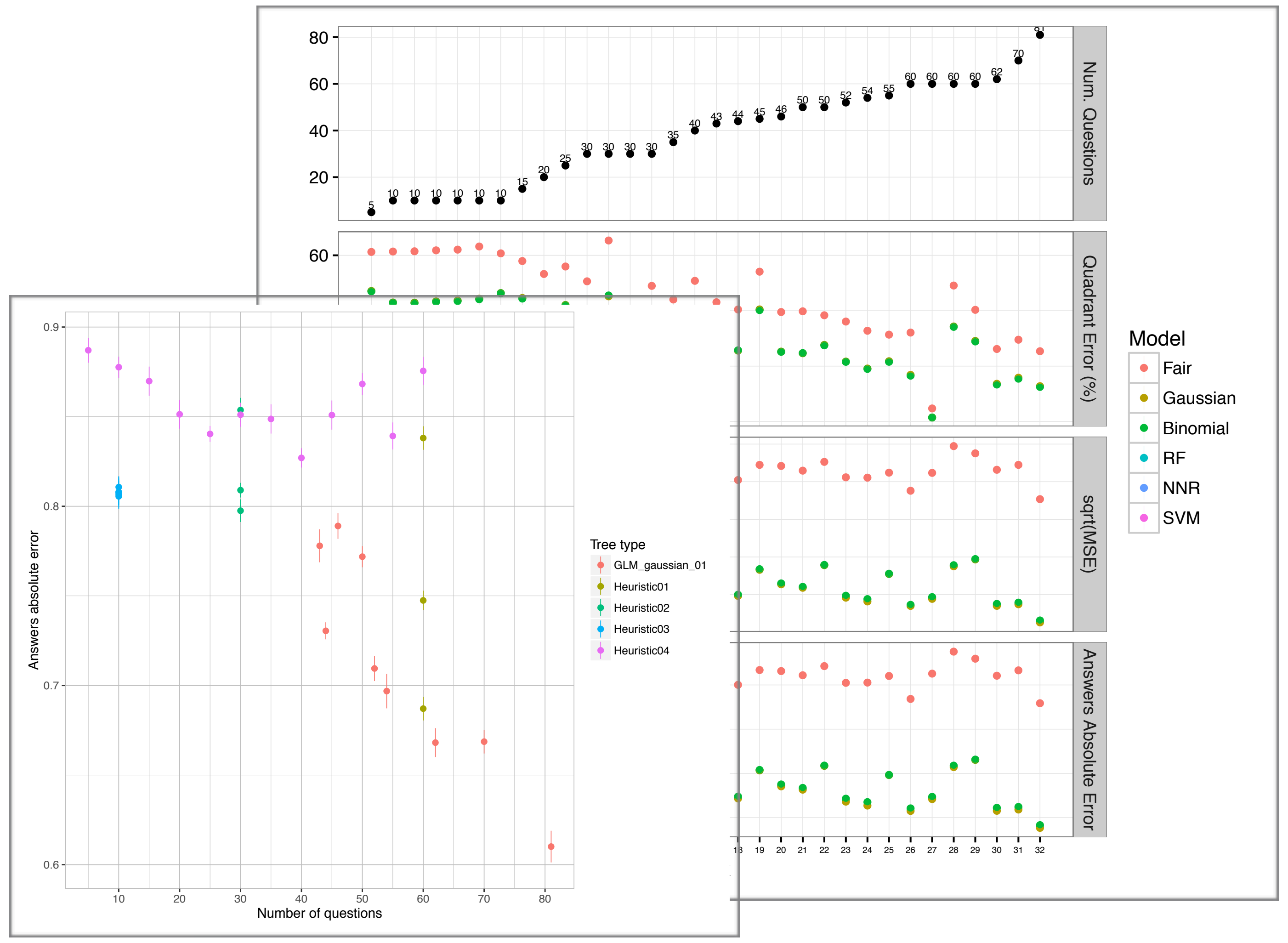
For each loop:

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*

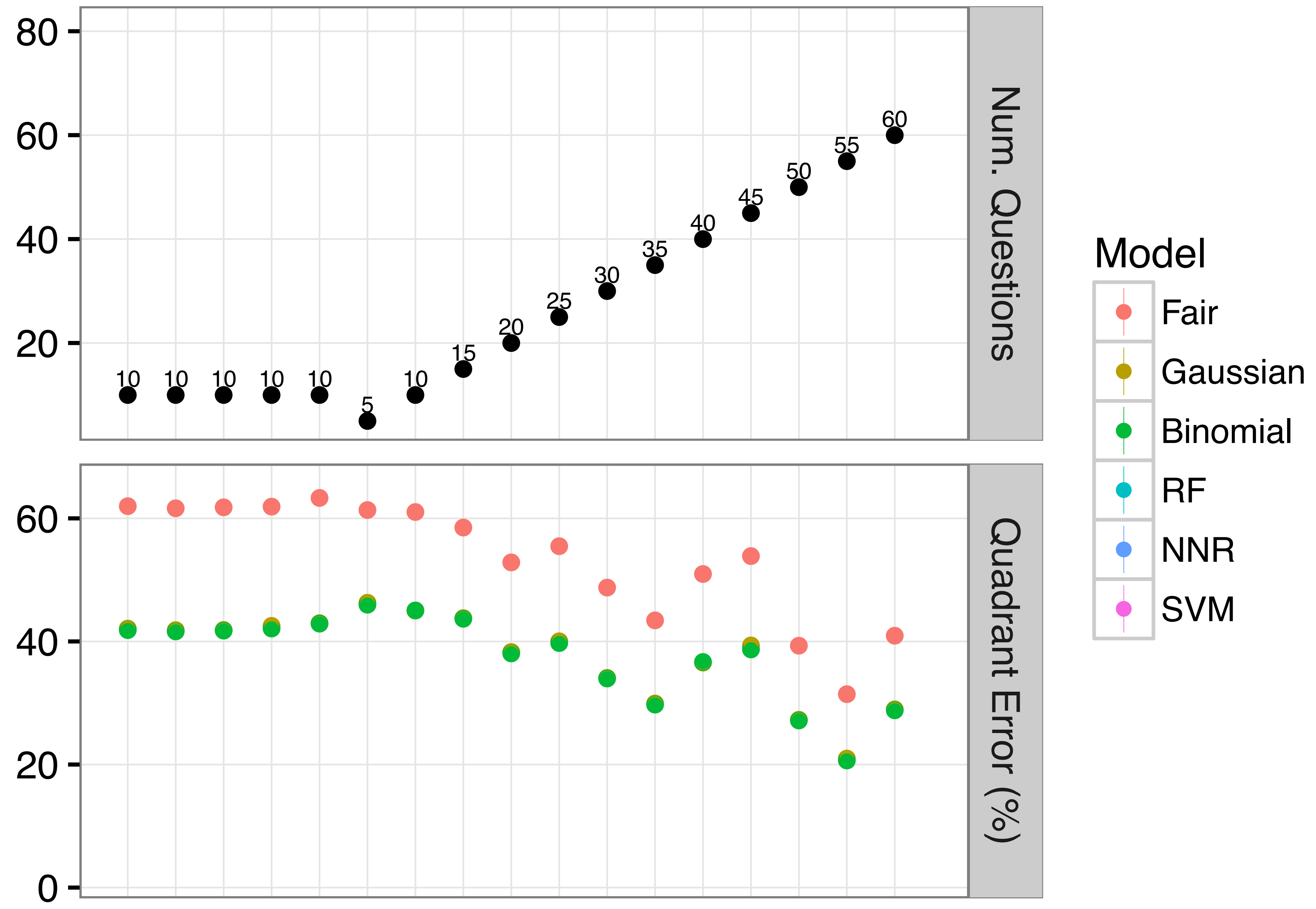
Results

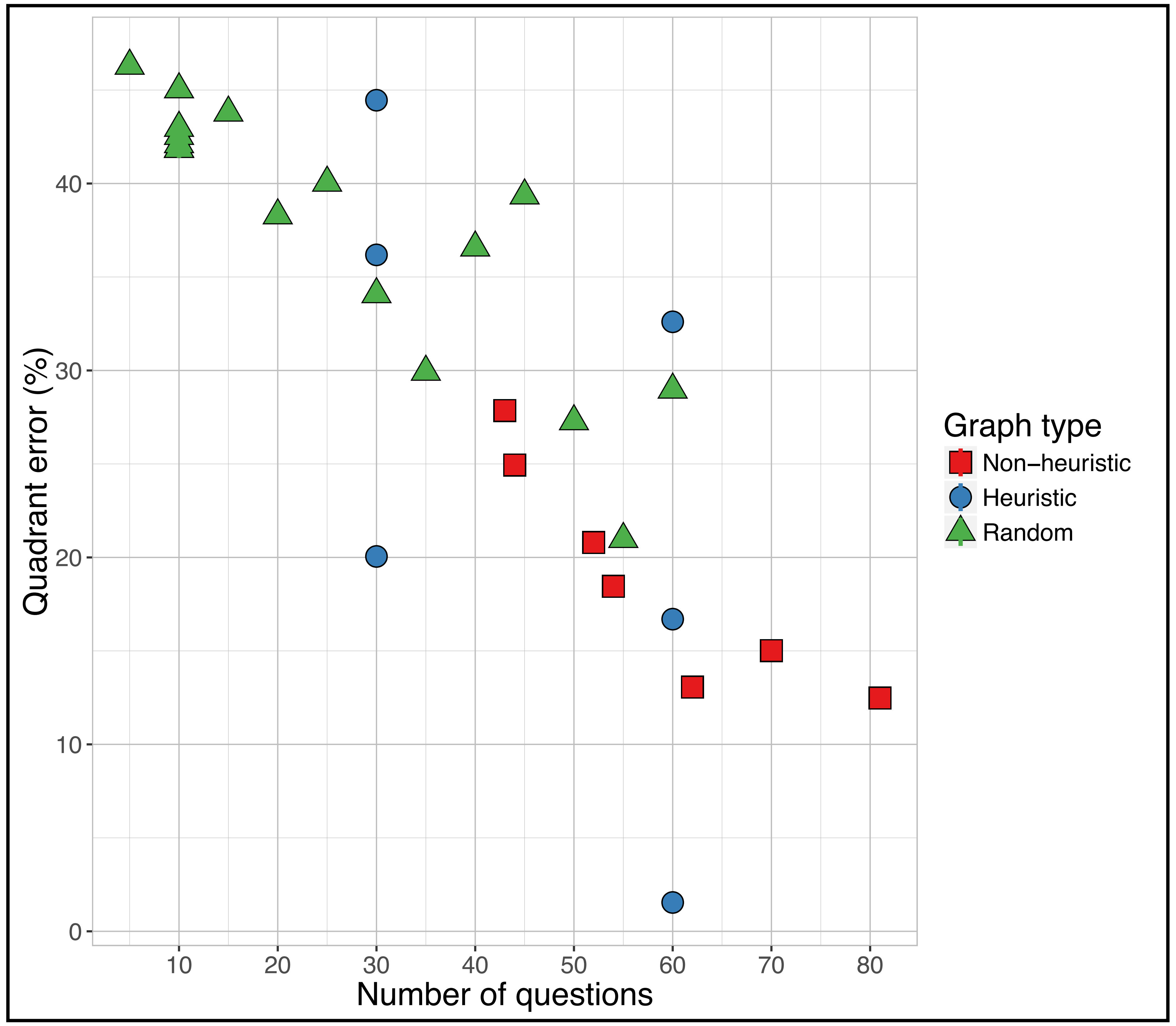


2. RESULTS

- Error vs. number of predictors

- Why are they performing in a similar way?
- Data is too noisy?



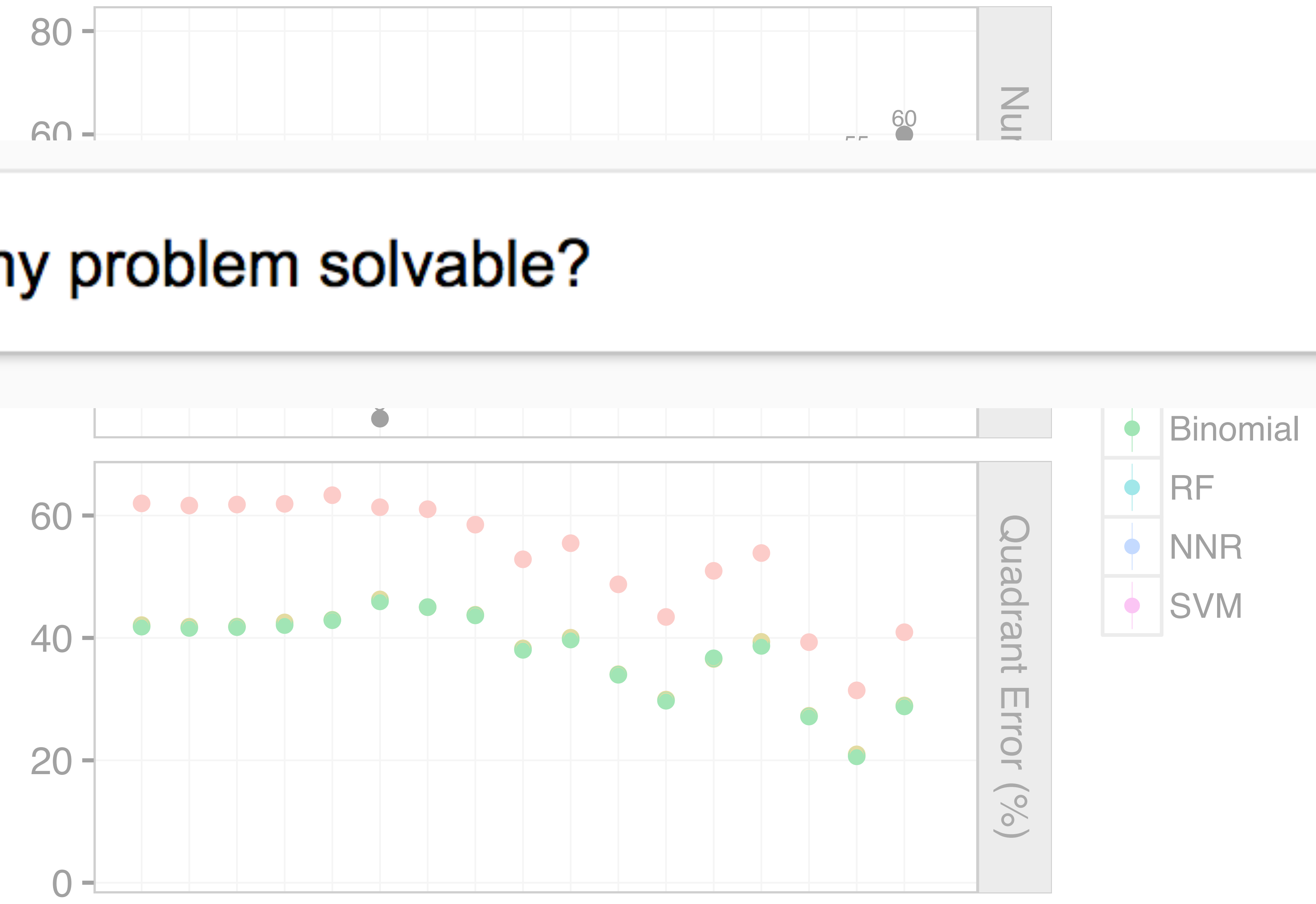


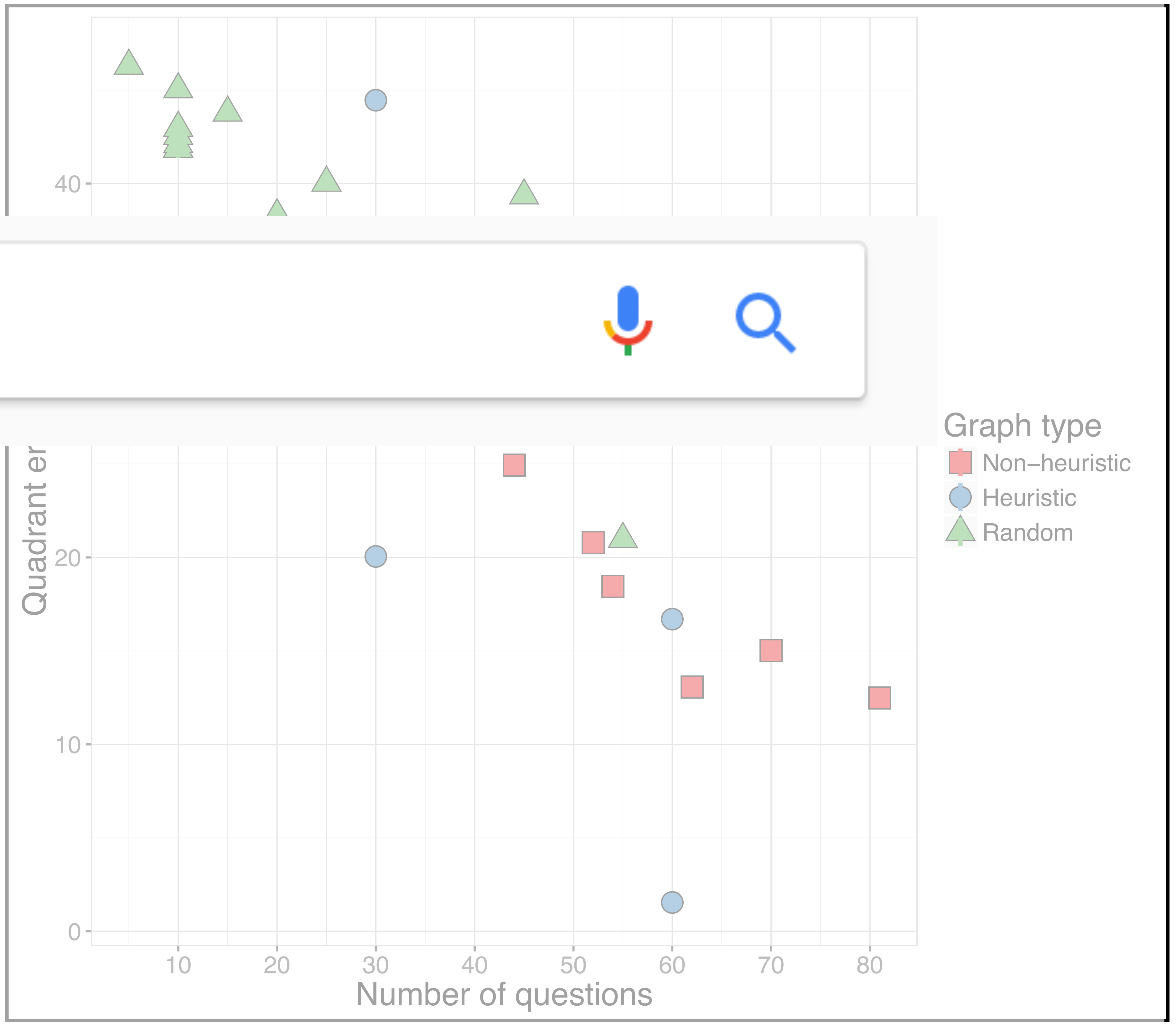
2. RESULTS

- Error vs. number

Google is my problem solvable?

- 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

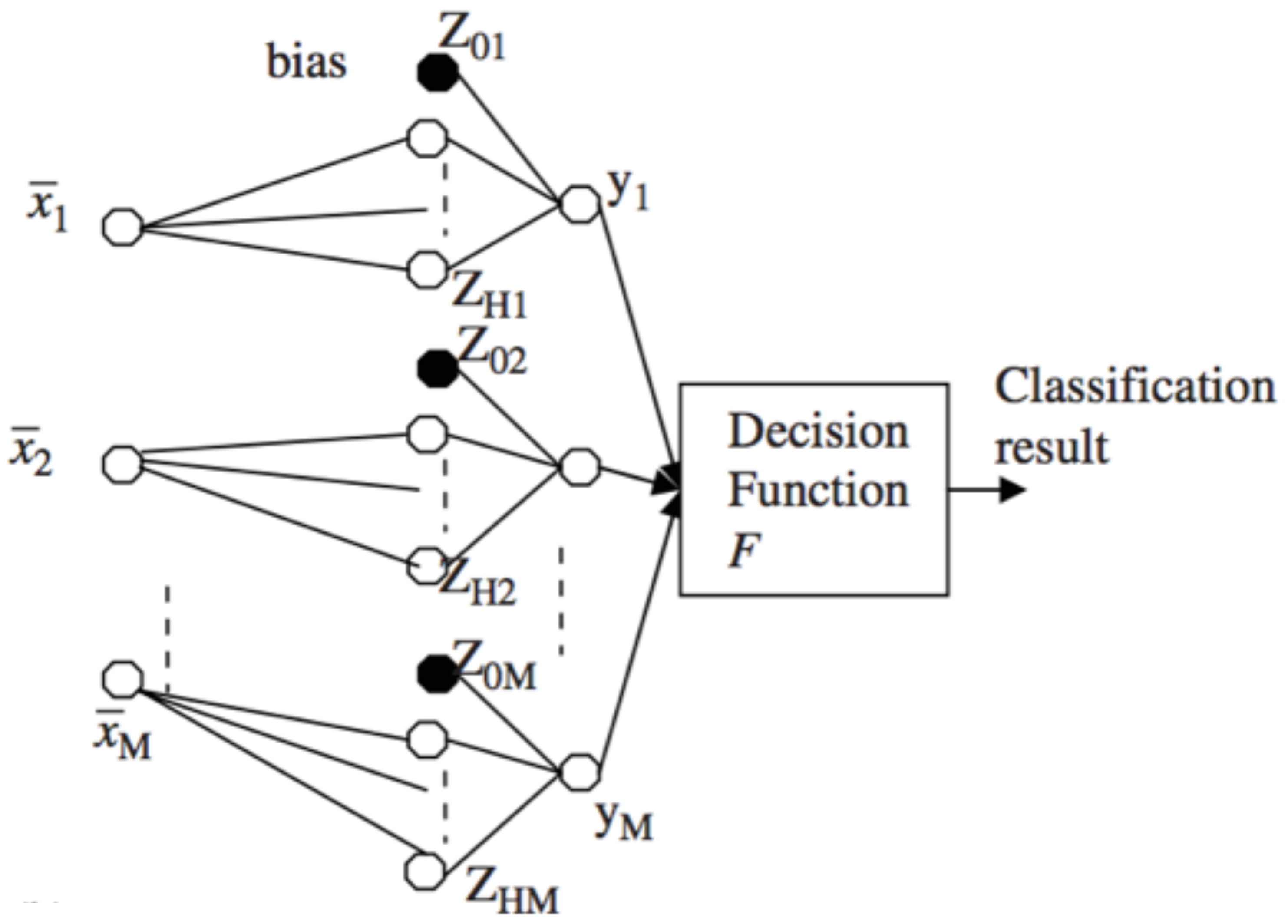
- Entropy!

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

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

NEURAL NETWORKS

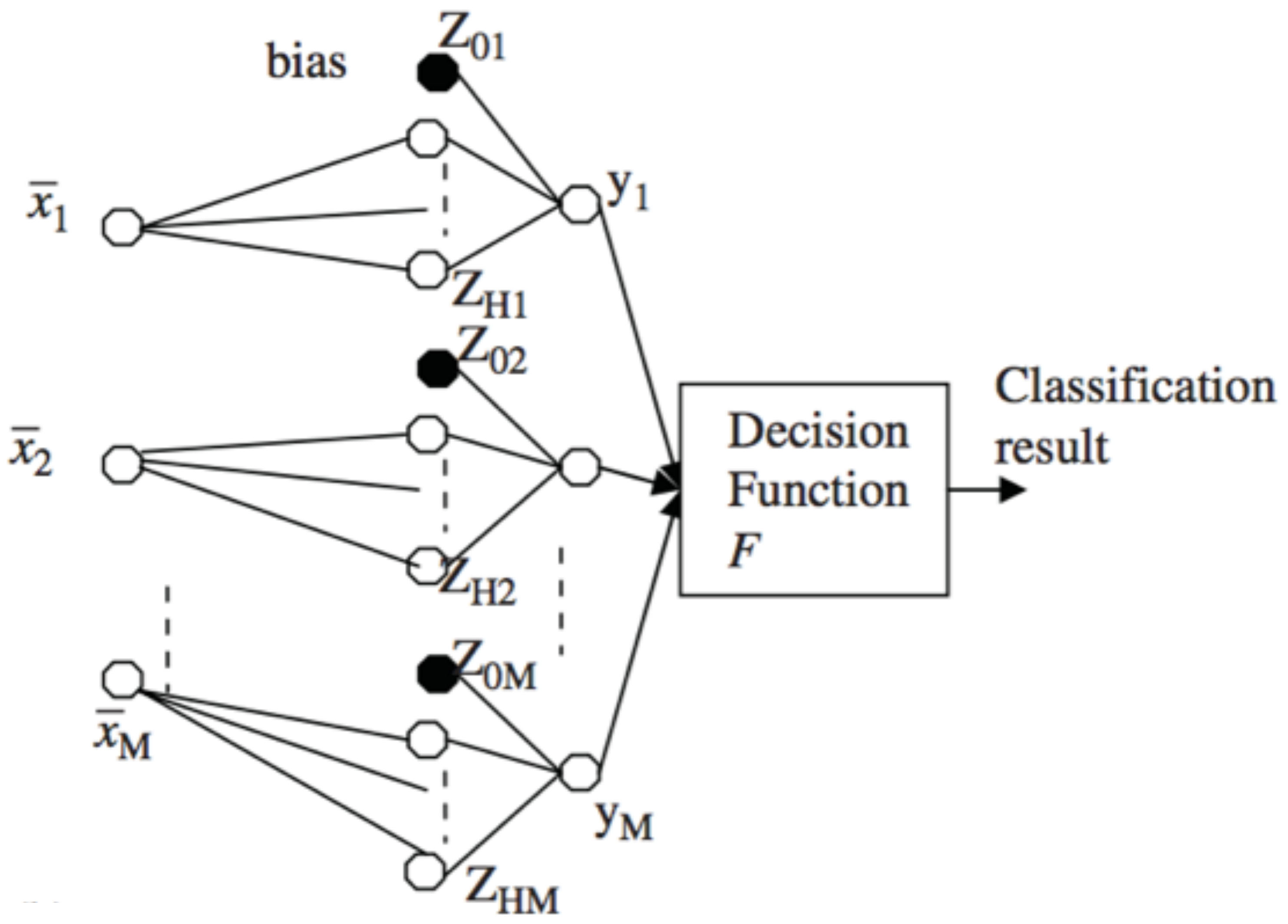
- Again, space $Q = P\dot{\cup}S$
- Fix s in S
- Classification problem with 4 classes
- We want a 0/1 output
- I. 4 NN, one output node for each class
- II. 1 NN, four output nodes



[3] Multi-class pattern classification using NN

NEURAL NETWORKS

- Again, space $Q = P\dot{\cup}S$
- 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



Problem with unbalanced classes:

zoom-in regions with lower numbers of points (discrete space)

[3] Multi-class pattern classification using NN

COMMENTS

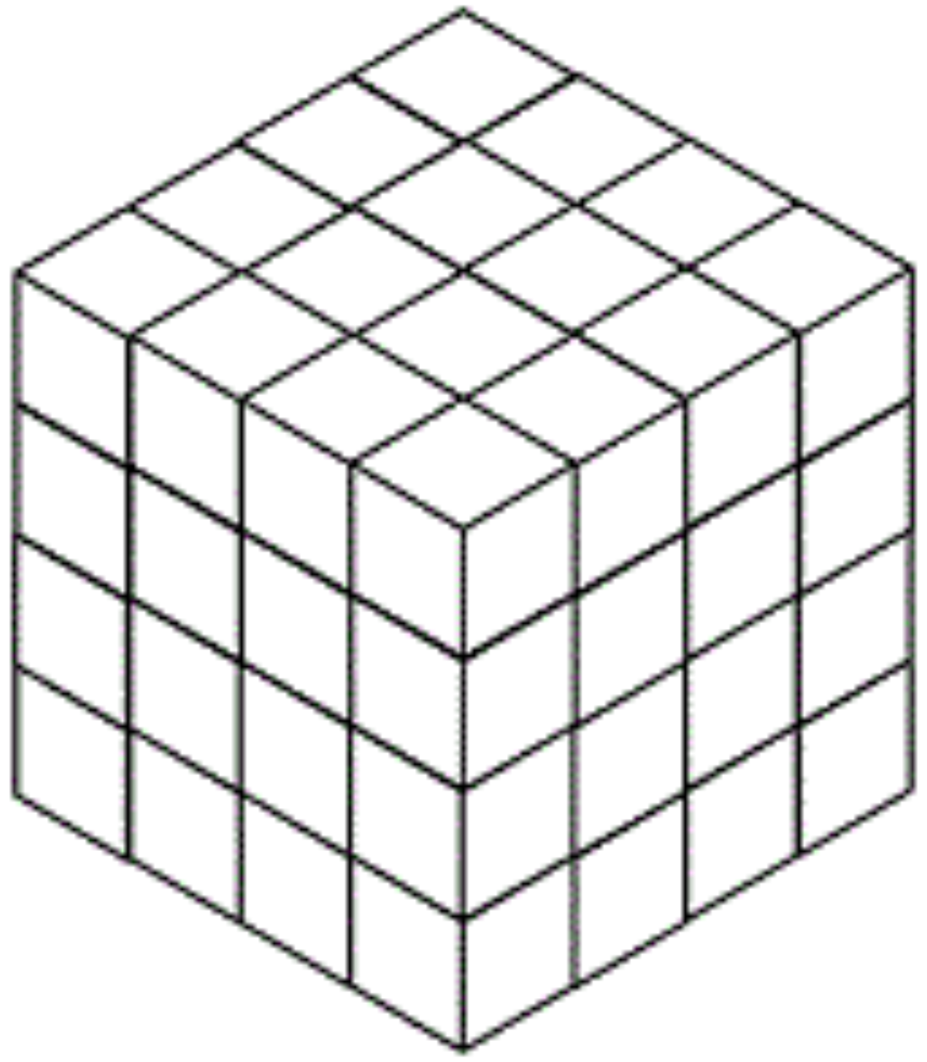
The problem is still open

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:



$$Q = P\dot{\cup}S$$

Drawback: sparse data.

Drawback: data concentrated in few cells

Drawback: users perceive the scale in different way

Drawback: users tend to choose high values

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…

[1] (Review FS) Saeys, Y. et al, A review of feature selection techniques in bioinformatics. *Bioinformatics*, 2007.

[2] (Normalised MI) Estévez, P. et al, Normalized Mutual Information Feature Selection. *IEEE Transactions On Neural Networks*, 2009.

[3] (Multiclass. NN)

[4] (Unbalanced data)

[5] (high-dimensional NN)

Thank you! Questions?