Webinars

Welcome!

The program will begin soon.
You will not hear audio until we begin.

SPE International ®

Webinars

Part 1

Incentive Engineering through Subgraph Matching

– with application to task allocation

Mark Ebden

Wednesday 15 December, 2021

SPE International

Speaker background

Webinars

Work & study at the University of Oxford:

- SPANN (Signal Processing And Neural Networks) res. group, 2002-2006
- PARG (Pattern Analysis Research Group), 2006-2012
- MLRG (Machine Learning Research Group), 2012-2015

Teaching at the University of Toronto, 2016-2019, 2021:

- Department of Statistical Sciences
- Faculty of Applied Science & Engineering

Work with companies:

- Datatrie / OBS Medical / Mind Foundry – tech startups
- Xilinx / BP / Celestica – larger engineering firms

SPE International ®

The ORCHID Project

- The science of human-agent collectives (HACs)
- January 2011 to December 2015
- Five subtopics, described at www.orchid.ac.uk

Jennings, N.R., Moreau, L., Nicholson, D., Ramchurn, S.D., Roberts, S.J., Rodden, T., and Rogers, A. (2014). **On human-agent collectives**. *Communications of the ACM*

Overview of Part 1

1. The problem: Agents $\not\rightarrow$ Humans

2. W3C PROV

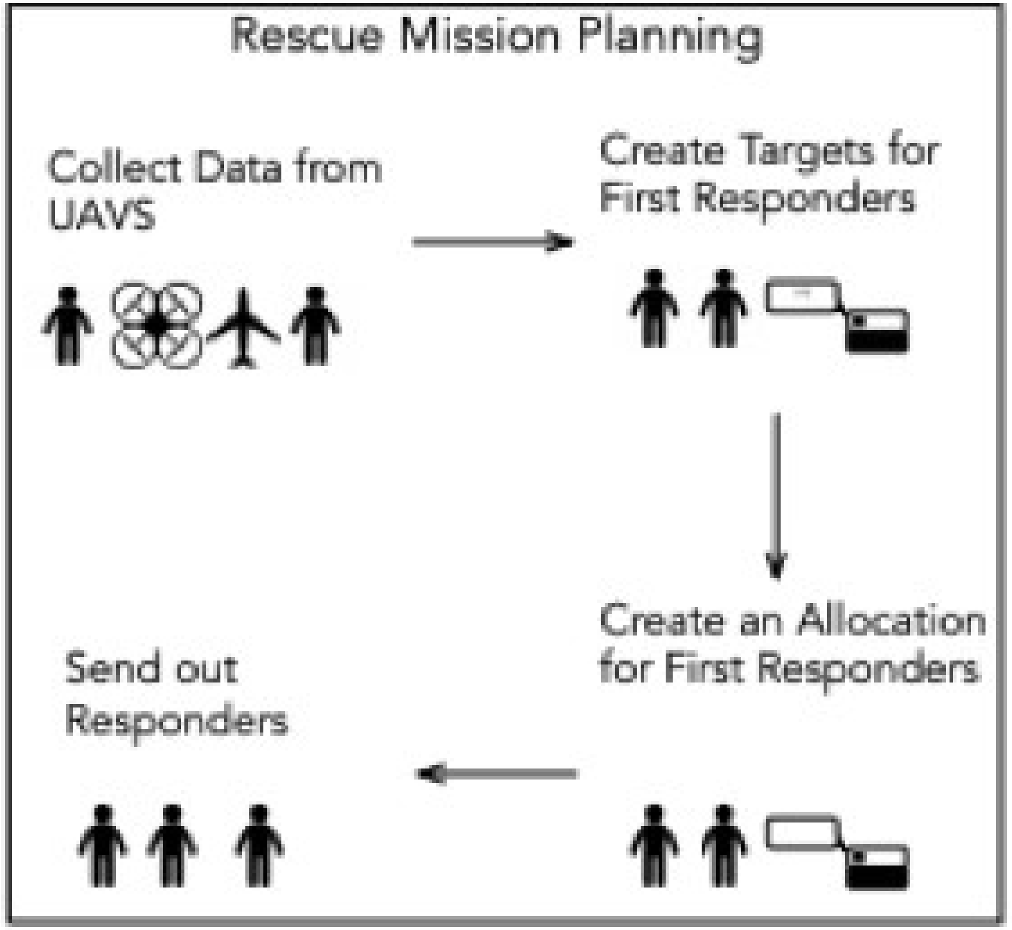
3. Subgraph matching

HAC-ER simulations at Nottingham University

Task-allocation game: **HAC-ER**

- Form pairs of complementarily skilled **humans** to travel together, pick up targets, and drop off the targets at designated locations
- A centralized planning **agent** provides each unoccupied player with a proposal: a target in need of attention and a teammate with the necessary complementary skills.
- Each player is allowed to reject tasks they judge to be personally unfavourable.

What the centralized planner does

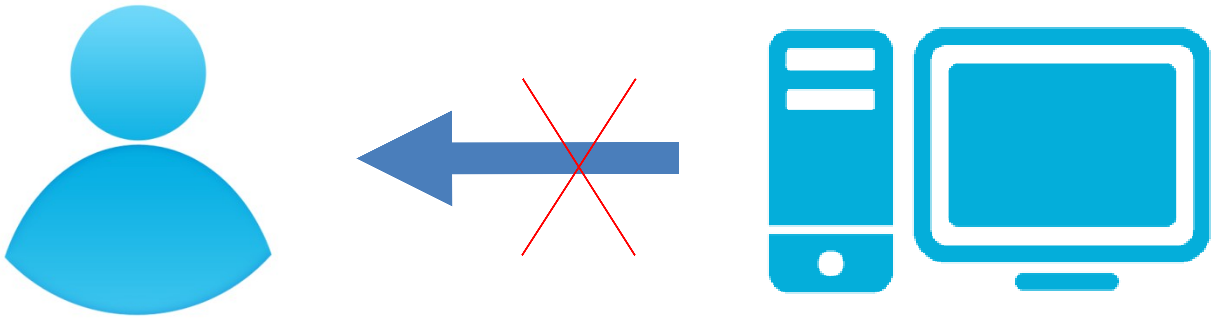


Incentive engineering for better HAI

Challenge: players decide to accept or reject tasks without necessarily knowing the global significance or the true personal enjoyability of that task.

Incentives would be desirable in certain cases, to increase the acceptance rate of the most useful tasks.

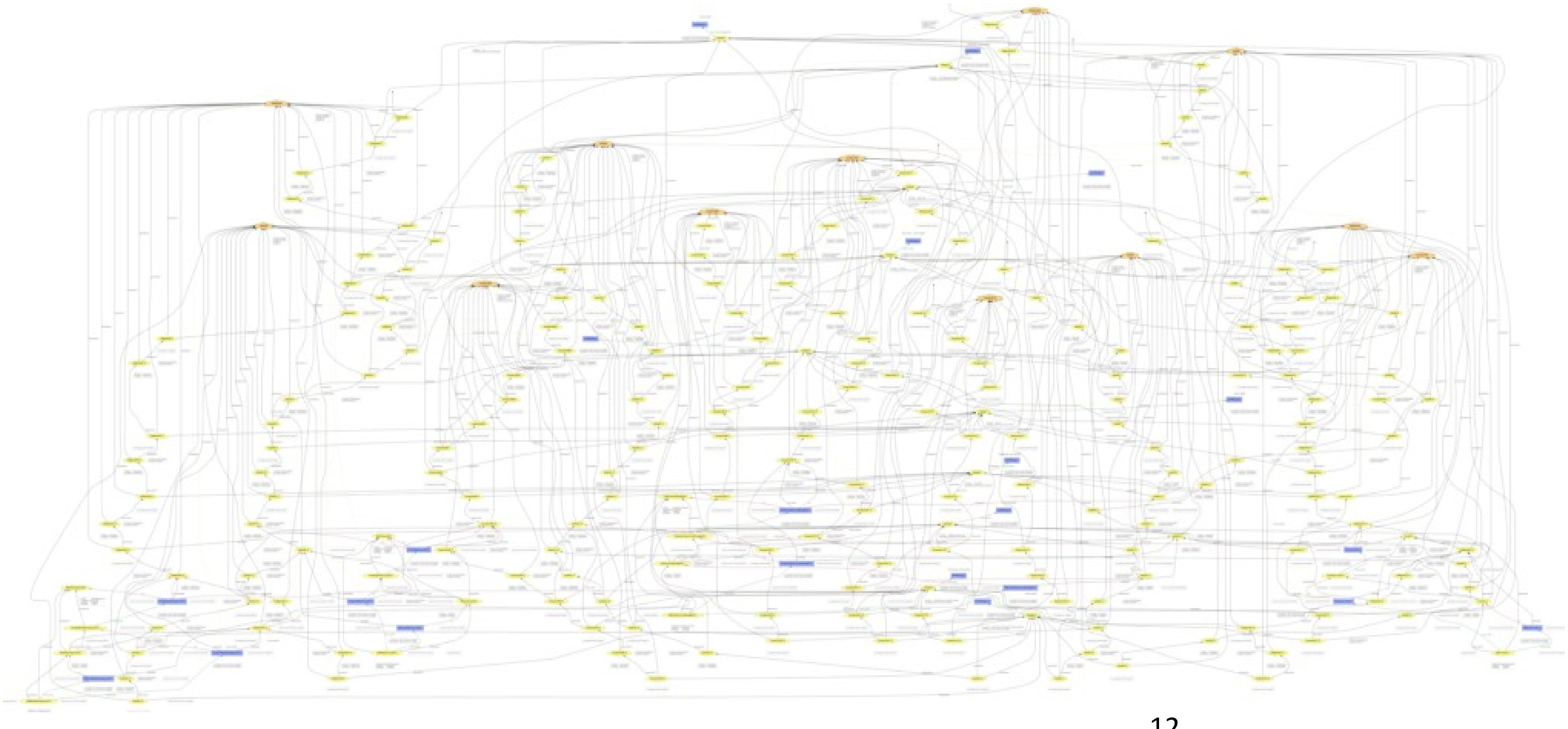
…task allocations suggested to people by a centralized planning agent "may need particular support" in order to succeed; there is a "need to ensure the agent's recommendations come with proper explanations in order to be trusted and further augment their capability"



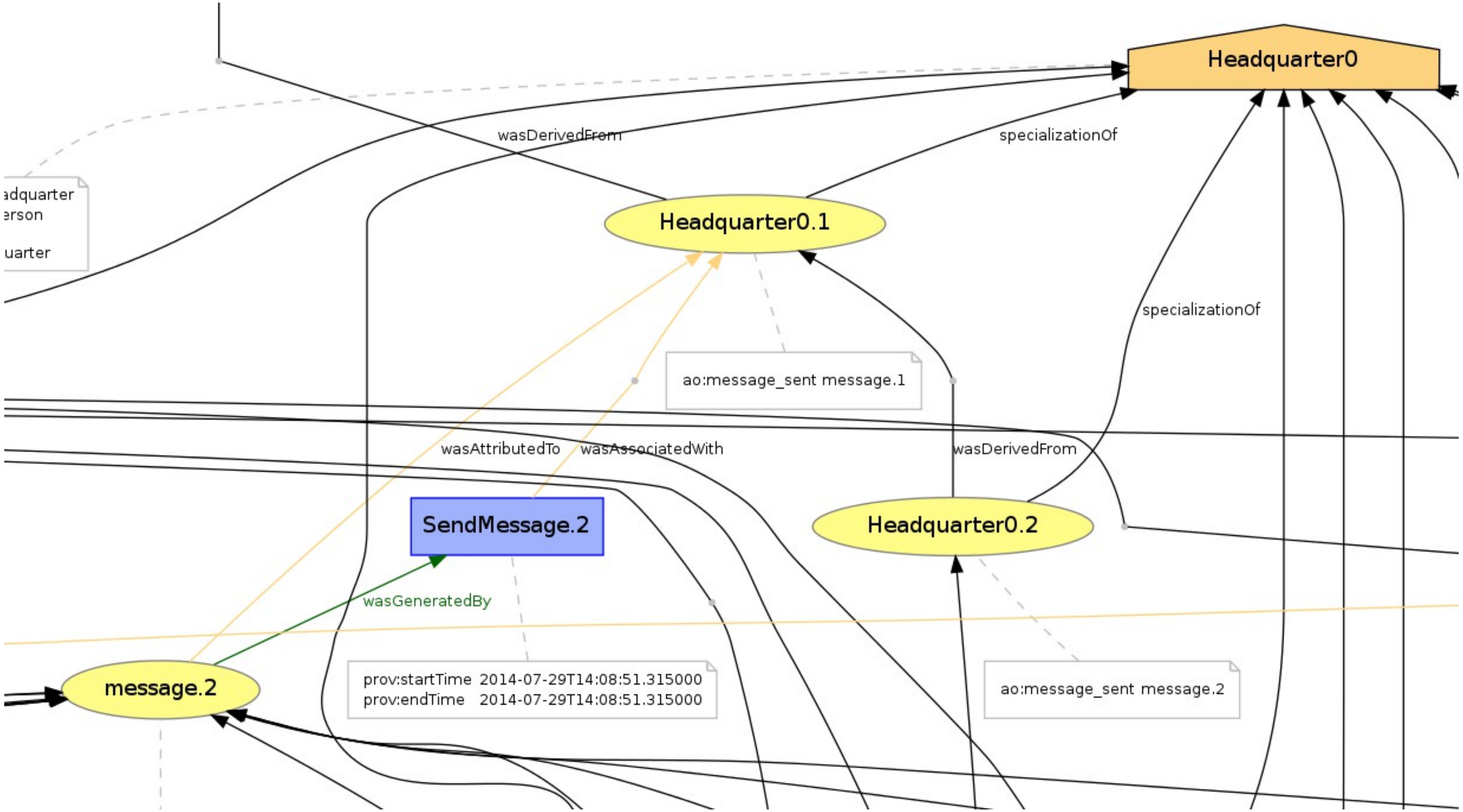
Examples of incentives

- **Reporting previous joint success:**
  - "You have worked successfully with this teammate before"
- **Reporting previously completed similar task:**
  - "You have completed this type of task before"
  - "Your teammate has completed this type of task before"
- **Reporting previously experienced geography:**
  - "The task is somewhere you've visited previously"
  - "The task is somewhere your teammate visited previously"
- **Reporting common experience with third parties:**
  - "You've each worked with player X before, successfully"

A provenance network



A HAC-ER provenance network



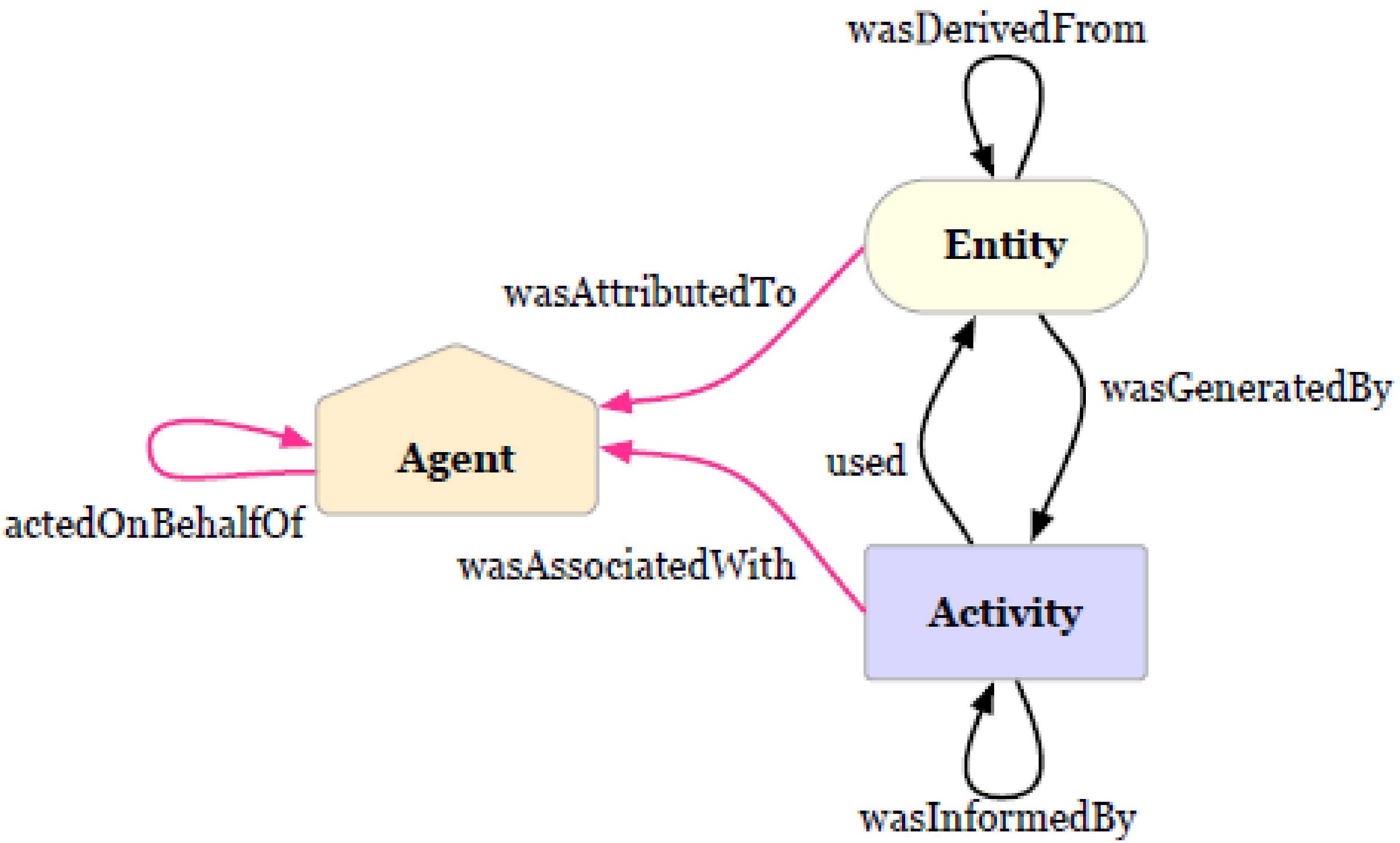
Introduction to Provenance

A tutorial (2019): http://openprovenance.org/tutorial/odsc2019.html and
https://www.youtube.com/watch?v=v8noJFOYFEk

A report on the use of PROV around the world (2020):
https://www.impact.science/blog/wp-content/uploads/2021/05/Evaluation-of-Impact-of-PROV.pdf

Additional presentation (2020): https://vimeo.com/478043709

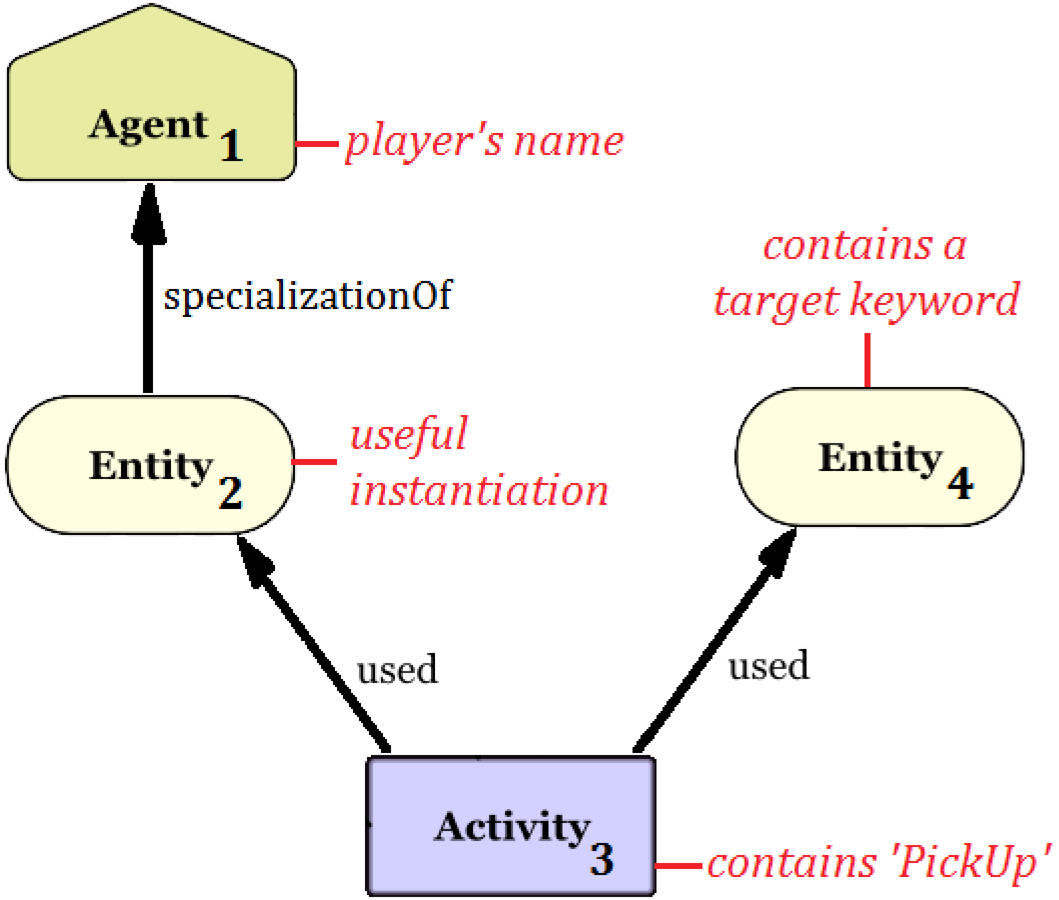
Nodes and edges of PROV



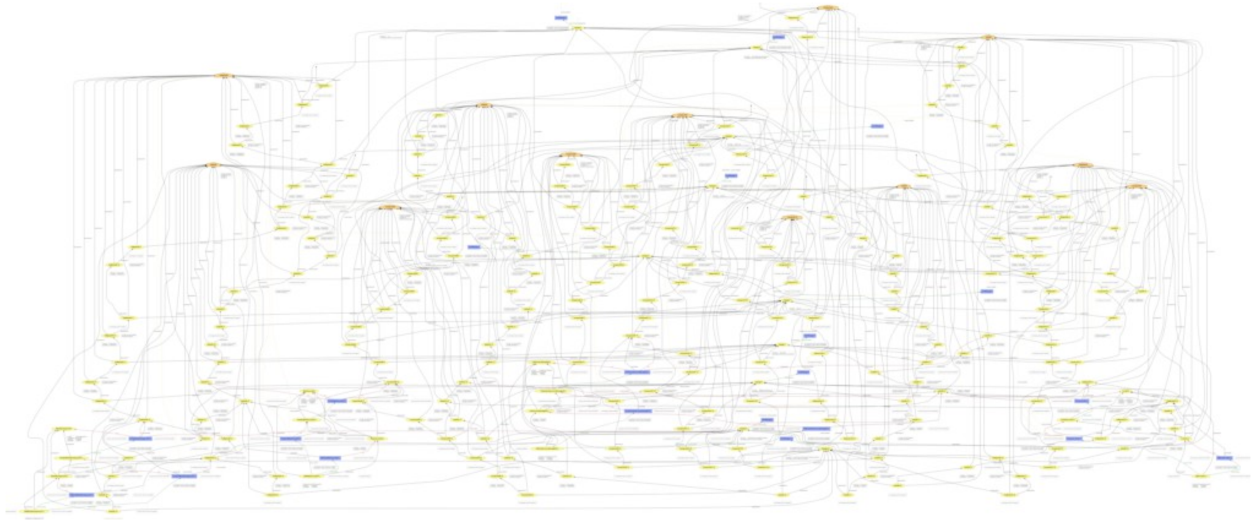
Graph properties

Graph name	Number of nodes	Number of edges	Density	Number of agent nodes
A	253	592	1.86%	12
B	875	2069	0.54%	7
C	1359	3279	0.36%	9
D	1485	3562	0.32%	9
E	1682	4173	0.30%	11
F	1784	4270	0.27%	7
G	2098	5142	0.23%	10

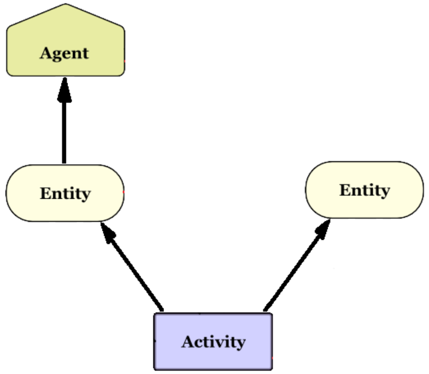
Example motif, for previously experienced geography



Motif-subgraph matching

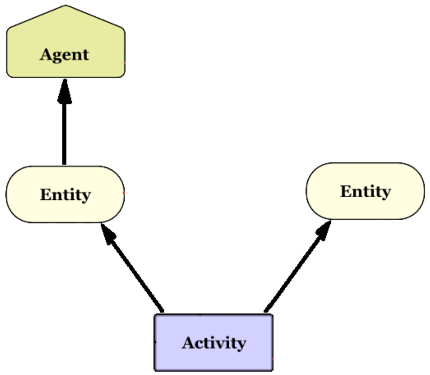


?



github.com/OxfordML/ENPG

The motif data



Each motif of interest was represented by a tuple with four constituents:

– An *m* x *m* adjacency matrix **H** in which the 1’s were replaced with codified edge attributes, e.g. ‘8’ for a PROV specializationOf

– An *m*-vector **t** listing codified node types, e.g. ‘1’ for a PROV entity

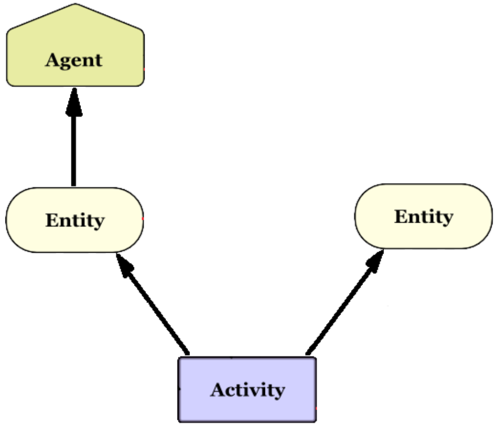
– A set of *m* strings *N* listing the names of each node, e.g. an agent might be called ‘medic95’

– A set of string pairs, *W*, with each pair giving the type of node attribute (e.g. ‘longitude’) and its value

Subgraph matching

Algorithm 1 Identify in a provenance graph $\mathcal{G}$ all instances of a particular subgraph $\mathcal{H}$

function FINDSUBGRAPHS $(\mathrm{p}, \mathcal{G}, \mathcal{H})$
    if $\mathrm{p}(i) > 0\ \forall i$ **then return** (p, 1)
    else
        $\mathrm{Y} \leftarrow \emptyset, c \leftarrow 0$
        $h \leftarrow$ index of first zero entity in p
        for $\{i | \mathrm{p}(i) > 0\}$ **do**
            for each neighbour g of the $\mathrm{p}(i)^{\mathrm{th}}$ node of G **do**
                if g's node type = $\mathrm{t}(h)$ AND g's name = $\mathcal{N}(h)$ AND
                edge attributes between nodes g and $\mathrm{p}(h) \supseteq \mathrm{H}(h, i)$ AND
                attributes of node $g \supseteq \mathcal{W}(h)$ **then**
                    $\mathrm{p}(h) \leftarrow g$
                    $(\mathrm{X}, r) \leftarrow$ FindSubgraphs $(\mathrm{p}, \mathcal{G}, \mathcal{H})$
                    if $r = 1$ **then**
                        $\mathrm{Y} \leftarrow \mathrm{Y} \cup \mathrm{X}$
                        $c \leftarrow 1$
                    end if
                end if
            end for
        end for
        return (Y, c)
    end if
end function

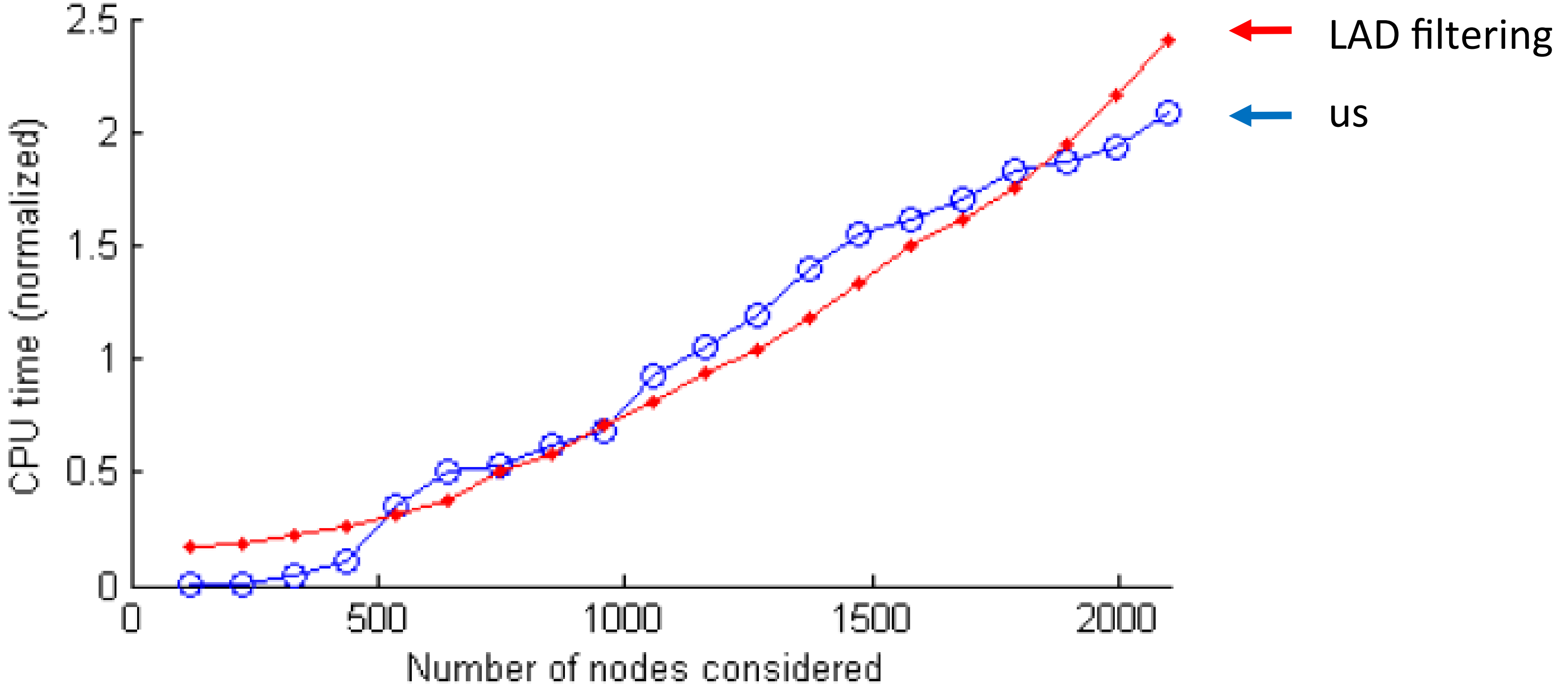


Results

Seven graphs with 7-12 agents each:

Number of motifs detected			
First type	**Second type**	**Third type**	**Fourth type**
11	22	26	13
8	16	20	0
8	16	16	10
9	18	22	13
4	8	8	0
16	32	32	24
12	24	24	26

Scalability



Messages to participants

The final output of the algorithm:

- text messages

If they were generated during a disaster trial, they would be transmissible with the specific intention of encouraging participants not to reject certain tasks.

A message template for each of the four incentive types was created, with any blanks filled in by text extracted automatically from the successfully located subgraphs

Provenance-based Explanations for Data Subjects

Questions | Automation | Inclusion | Exclusion | Sources | Relevance | Accuracy | Fairness

Q: Are the data used for assessing my loan application correct?

Data correctness may not be guaranteed: the applicant may have made a typo in their application or the data provided by a third-party may be inaccurate.

You can check the data supplied in your original application for any inaccuracy.

In addition, we obtained the following data from third-party providers:

- Accounts: 5.0
  The number of open credit lines in the borrower's credit file
- Mortgages: 0.0
  Number of mortgage accounts
- Revolving Balance: 1118.0
  Total credit revolving balance
- Credit Utilization Rate: 5.2
  Revolving line utilization rate, or the amount of credit the borrower is using
- Earliest Credit Line: Mar-2006
  The month the borrower's earliest reported credit line was opened
- Public Records: 0.0
  Number of derogatory public records
- Bankruptcies: 0.0
  Number of public record bankruptcies

We also calculated the following data from the above and your supplied application data:

- Interest Rate: 13.59
  Interest rate on the loan
- Installment: 169.9
  The monthly payment owed by the borrower when the loan starts
- Debt to Income: 0.38
  The ratio of debt over income
- Grade: C
  Loan grade
- Subgrade: C2
  Loan subgrade

From Luc Moreau and Dong Huynh, a 2020 application of messages:

https://explain.openprovenance.org

Summary

We use provenance graphs for incentive engineering.

We provide a viable feedback mechanism to allow incentivization through interaction with agents and information sources with the objective of improving task-acceptance rates.

Our algorithm was successfully able to find all the expected subgraphs within HAC-ER provenance graphs and to generate a variety of meaningful messages on an any-time basis, i.e. from whatever provenance data had been collected up to the point of execution.

Acknowledgments

Colleagues in the ORCHID project, particularly:

Stephen J. Roberts[1]
Trung Dong Huynh[2]
Luc Moreau[2]

1 Dept. of Engineering Science, University of Oxford

2 Electronics and Computer Science, University of Southampton

Webinars

Part 2

Organizing data-analysis tools for practitioners

Mark Ebden

Wednesday 15 December, 2021

SPE International ®

The problem:

You seek advice
with limited time.

Which classifier should I use?

Which hypothesis test
should I use?

Do I need a deep
neural network?

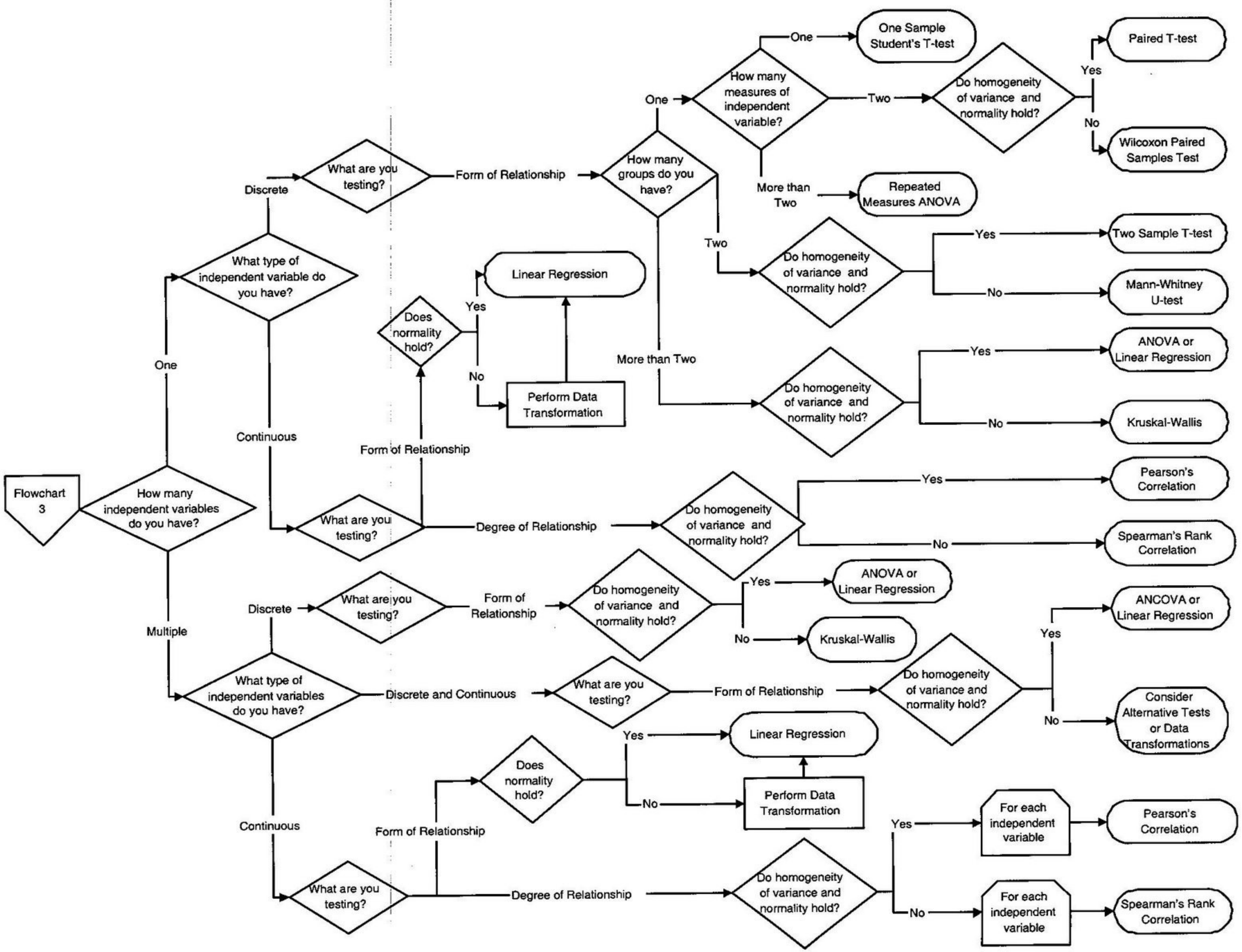
What was that procedure
I learned in undergrad for this?

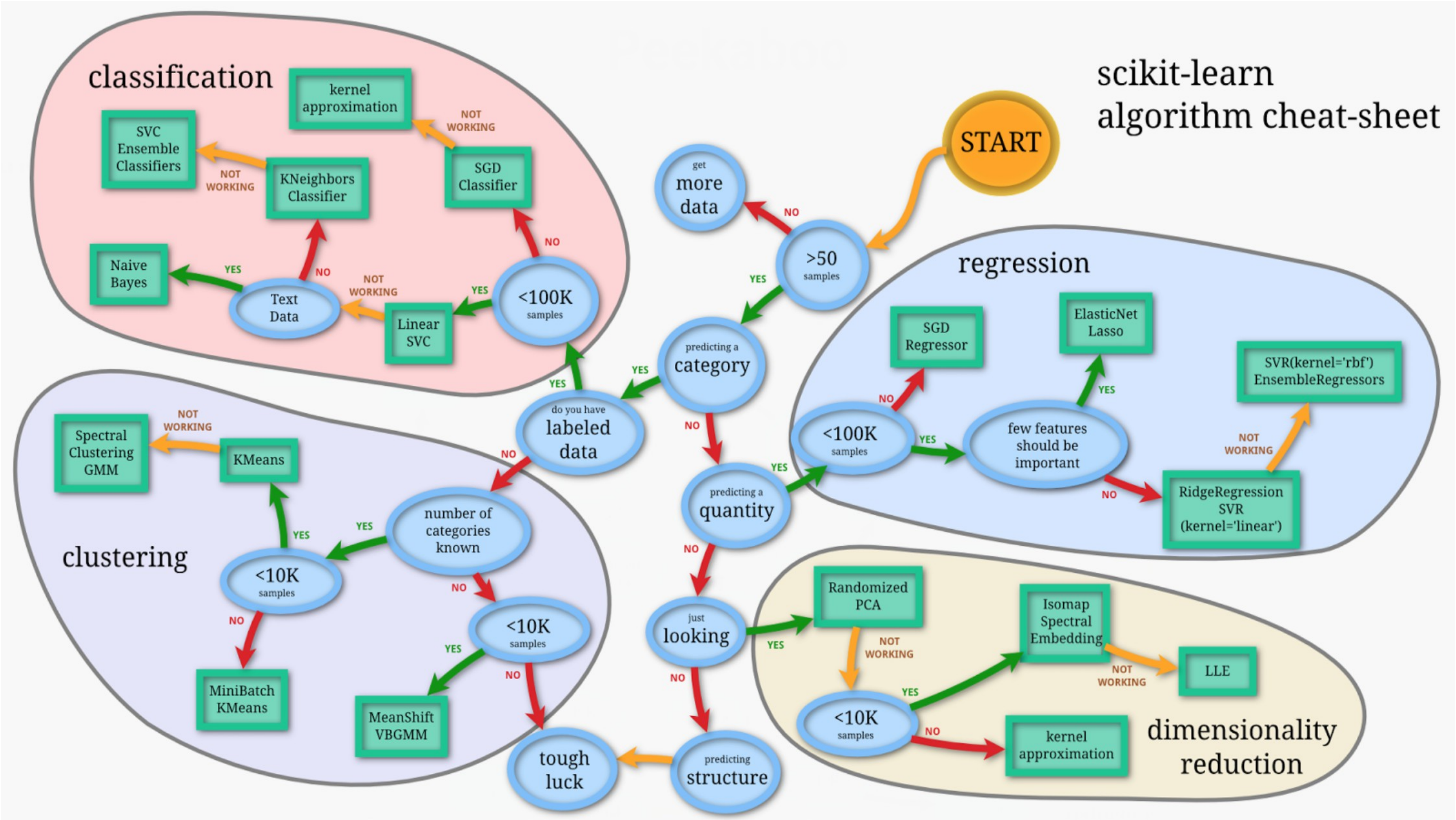
Current solutions

- Reviewing / purchasing textbooks
- Online courses

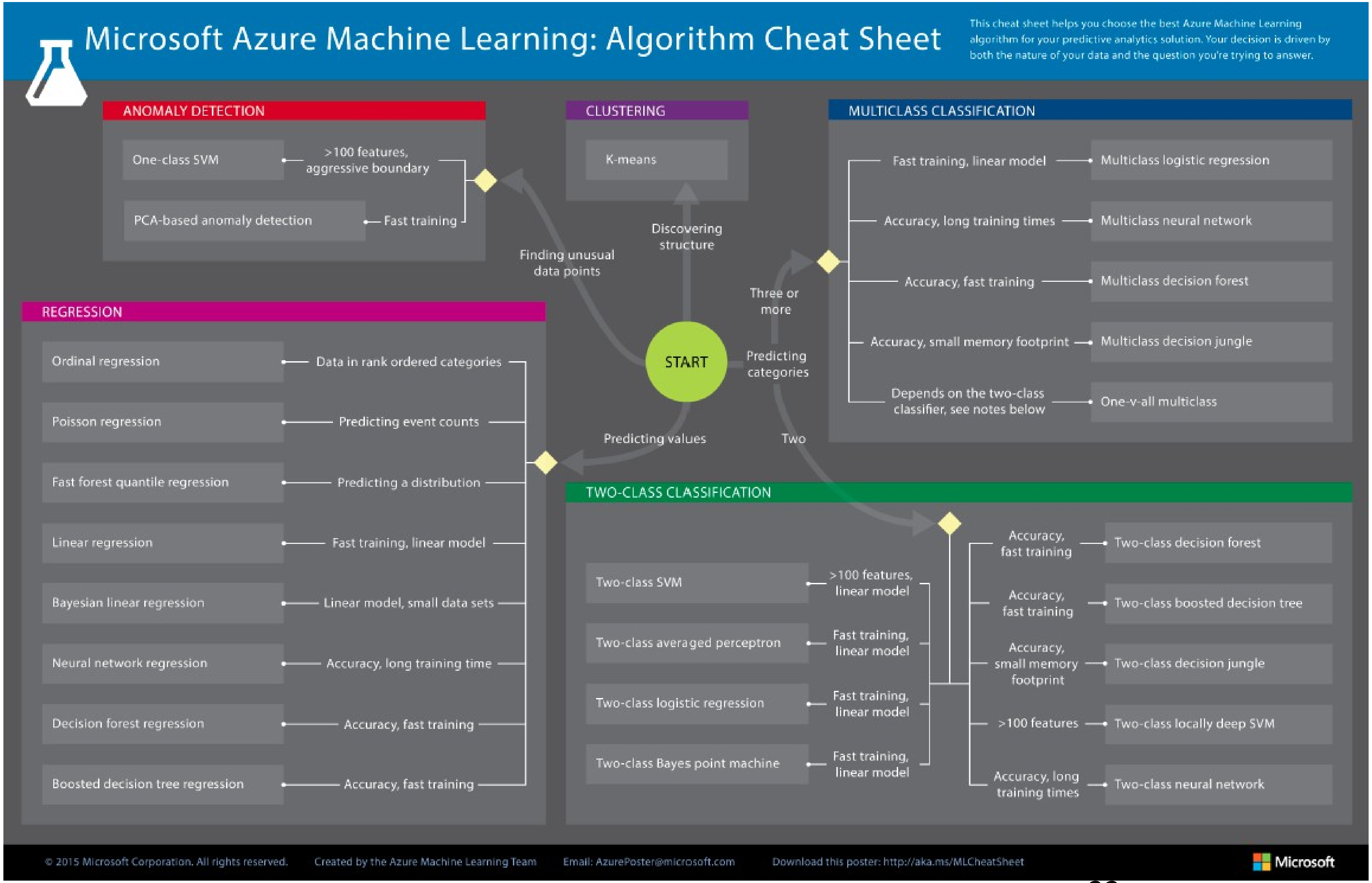
- Pestering classmates / colleagues
- Hiring / contracting
- stackoverflow.com, ResearchGate

- Flowcharts
- AutoML


Flowchart 3
How many independent variables do you have?
One
Multiple
What type of independent variable do you have?
Discrete
Continuous
What are you testing?
Form of Relationship
How many groups do you have?
One
How many measures of independent variable?
One
One Sample Student's T-test
Two
Do homogeneity of variance and normality hold?
Yes
Paired T-test
No
Wilcoxon Paired Samples Test
More than Two
Repeated Measures ANOVA
Two
Do homogeneity of variance and normality hold?
Yes
Two Sample T-test
No
Mann-Whitney U-test
More than Two
Do homogeneity of variance and normality hold?
Yes
ANOVA or Linear Regression
No
Kruskal-Wallis
What are you testing?
Form of Relationship
Does normality hold?
Yes
Linear Regression
No
Perform Data Transformation
Degree of Relationship
Do homogeneity of variance and normality hold?
Yes
Pearson's Correlation
No
Spearman's Rank Correlation
What type of independent variables do you have?
Discrete
What are you testing?
Form of Relationship
Do homogeneity of variance and normality hold?
Yes
ANOVA or Linear Regression
No
Kruskal-Wallis
Discrete and Continuous
What are you testing?
Form of Relationship
Do homogeneity of variance and normality hold?
Yes
ANCOVA or Linear Regression
No
Consider Alternative Tests or Data Transformations
Continuous
What are you testing?
Form of Relationship
Does normality hold?
Yes
Linear Regression
No
Perform Data Transformation
Degree of Relationship
Do homogeneity of variance and normality hold?
Yes
For each independent variable
Pearson's Correlation
No
For each independent variable
Spearman's Rank Correlation



scikit-learn
algorithm cheat-sheet
START
classification
regression
clustering
dimensionality reduction
get more data
>50 samples
predicting a category
do you have labeled data
predicting a quantity
just looking
predicting structure
tough luck
<100K samples
kernel approximation
SGD Classifier
Linear SVC
Text Data
KNeighbors Classifier
SVC Ensemble Classifiers
Naive Bayes
number of categories known
<10K samples
KMeans
Spectral Clustering GMM
MiniBatch KMeans
MeanShift VBGMM
SGD Regressor
few features should be important
ElasticNet Lasso
RidgeRegression SVR (kernel='linear')
SVR(kernel='rbf') EnsembleRegressors
Randomized PCA
Isomap Spectral Embedding
LLE
YES
NO
NOT WORKING



Microsoft Azure Machine Learning: Algorithm Cheat Sheet
This cheat sheet helps you choose the best Azure Machine Learning algorithm for your predictive analytics solution. Your decision is driven by both the nature of your data and the question you're trying to answer.
ANOMALY DETECTION
One-class SVM
>100 features, aggressive boundary
PCA-based anomaly detection
Fast training
CLUSTERING
K-means
Discovering structure
Finding unusual data points
MULTICLASS CLASSIFICATION
Fast training, linear model
Multiclass logistic regression
Accuracy, long training times
Multiclass neural network
Accuracy, fast training
Multiclass decision forest
Accuracy, small memory footprint
Multiclass decision jungle
Depends on the two-class classifier, see notes below
One-v-all multiclass
Three or more
START
Predicting categories
Predicting values
Two
REGRESSION
Ordinal regression
Data in rank ordered categories
Poisson regression
Predicting event counts
Fast forest quantile regression
Predicting a distribution
Linear regression
Fast training, linear model
Bayesian linear regression
Linear model, small data sets
Neural network regression
Accuracy, long training time
Decision forest regression
Accuracy, fast training
Boosted decision tree regression
Accuracy, fast training
TWO-CLASS CLASSIFICATION
Two-class SVM
>100 features, linear model
Two-class averaged perceptron
Fast training, linear model
Two-class logistic regression
Fast training, linear model
Two-class Bayes point machine
Fast training, linear model
Accuracy, fast training
Two-class decision forest
Accuracy, fast training
Two-class boosted decision tree
Accuracy, small memory footprint
Two-class decision jungle
>100 features
Two-class locally deep SVM
Accuracy, long training times
Two-class neural network
© 2015 Microsoft Corporation. All rights reserved.
Created by the Azure Machine Learning Team
Email: AzurePoster@microsoft.com
Download this poster: http://aka.ms/MLCheatSheet
Microsoft

CORNERED

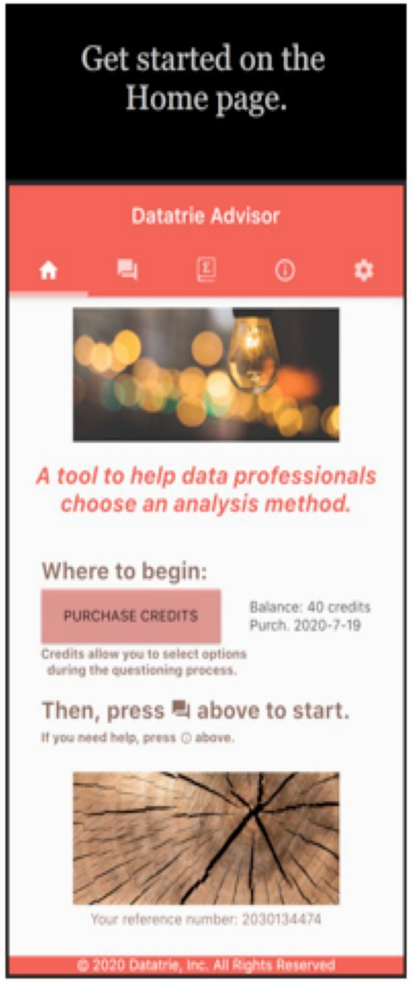
"This can't be right. Let me take another look at that flowchart."

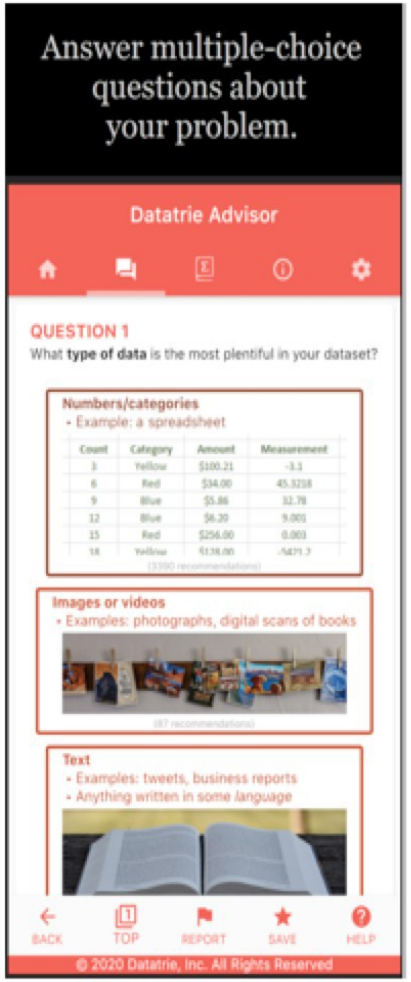
Solution: a trie

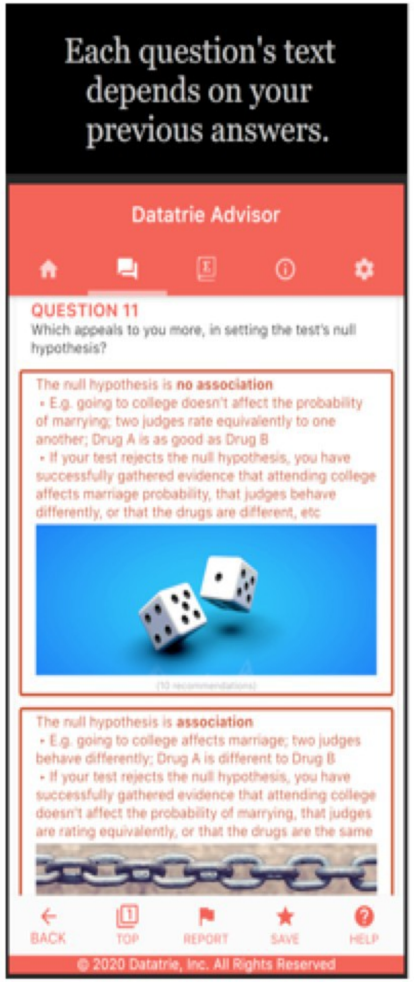
- About 1,800 statistical methods indexed
- Up to 30 questions asked per use case
- Flutter (.dart) targeting Android and iOS
- Launched summer 2020 at: www.datatrie.com/advisor

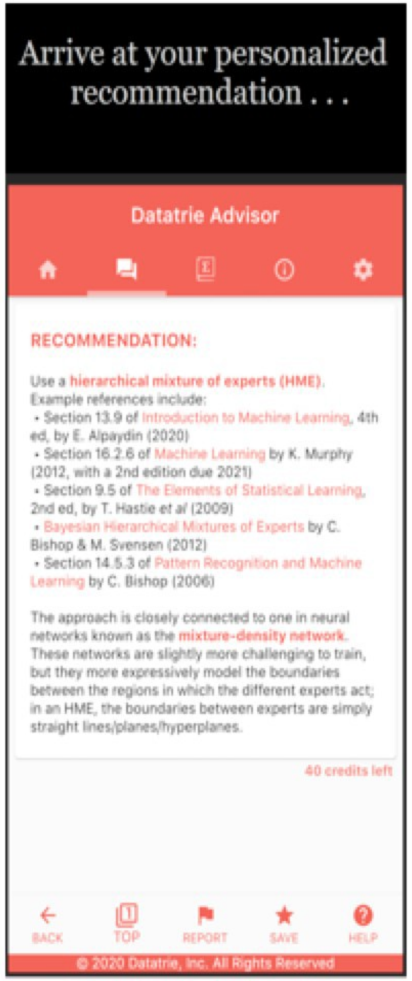
www.datatrie.com/advisor

Webinars









SPE Webinars are Sponsored by:

SPE Foundation