Statistics Automation in a Query-Answering System

Statistics Automation in a Query-Answering System

Thomas Fletcher - *T.Fletcher-6@sms.ed.ac.uk*[1], Alan Bundy - *A.Bundy@ed.ac.uk*[2], and Kwabena Nuamah - *K.Nuamah@ed.ac.uk*[3]

[1,2,3]The University of Edinburgh

Abstract

FRANK is a multi-domain Query Answering (QA) system which performs inferential and simple statistical reasoning on data which it automatically identifies and retrieves (e.g. to provide predictions). SMART is a statistics-advisor system designed to allow FRANK to answer queries of types which have not previously been within the purview of QA systems, including statistical significance, functional shape description and analysis of variance. The combined SMART FRANK system will allow users to quickly obtain answers to data-oriented questions involving choices of analysis, modelling and visualisation which are tedious for an expert and beyond the knowledge of the novice.

1 Introduction

Query Answering (QA) systems are capable of parsing requests of varying degrees of structure (e.g. query languages or natural language), fetch information from knowledge bases, and process it to produce the required answer. Functional Reasoning for Acquiring Novel Knowledge (FRANK) [12, 15] is a multi-domain QA system with a focus on producing new knowledge from the information it has access to. In achieving this, it utilises both symbolic and statistical reasoning, thus placing it in the DARPA "Third Wave of AI"[1]. Statistical Methodology Advisor at Reasoning Time (SMART) is a statistics system designed to enable FRANK to automatically select and carry out statistical methodologies based on the query and available data. This functionality will allow both expert and non-expert users to answer data-related questions quickly, without having to spend time implementing (or indeed having to know) the appropriate methods and visualisations; their reasonable uses for this would be early stages of research or, indeed, simply satisfying curiosities.

2 Background

2.1 Intelligent Data/Discovery Assistants

SMART FRANK lies mostly within the field of Intelligent Data/Discovery Assistants (IDAs), though it is narrower in scope due to its QA context, while the data processing it performs belongs to the field of AutoML [22]. Many varieties of IDA exist [17, 18], and the subset within which SMART FRANK overlaps the most is that of expert systems, which follow rules primarily defined by an expert designer. Systems of all IDA kinds have been researched and built in many ways [1], are applied mostly to specific but sometimes generic domains, and offer varying degrees of aid or automation to the user. READ [10] and the older AIDE [20, 19, 2, 21] are particularly interactive IDAs, guiding the user through a predetermined set of steps and providing suggestions and alternatives throughout. IDEA [3] is another IDA of interest since it lets the user specify weights for preferred model characteristics (e.g. explainability, accuracy or speed), a feature shared by SMART.

2.2 Current FRANK

An illustrative example of FRANK's current capabilities is answering the prediction query "What will the most populous country in Africa be in 2040?"; in order to do so multiple stages take place: query parsing, identification of relevant data sources, acquisition of the specific relevant data (if it exists; further query decomposition otherwise), aggregation into coherent datasets, application of simple regressions (which may be done together or independently for each country and data source), and finally comparison and selection of answers. FRANK's parsing is also able to decompose and handle

[1]https://machinelearning.technicacuriosa.com/2017/03/19/a-darpa-perspective-on-artificial-intelligence/

nested queries e.g. "What will be the GDP in 2030 of the most populous ...". A core feature of FRANK is its ability to generate explanations of its inference process [14], but two further points are worth mentioning regarding FRANK's capabilities: that every FRANK query returns a measure of uncertainty as part of its answer [13] (derived from the various processing steps involved), and that user context is taken into consideration in its processing [16]. What FRANK lacks is statistical understanding of the data it processes and variety in queries and outputs, as, despite its multi-domain nature, it only performs data retrieval (with simple linear regression if not possible) and simple boolean comparisons.

2.3 New Queries & Outputs

SMART gives FRANK the tools to select and carry out a collection of statistical methodologies, enabling it to answer a much wider variety of queries with multiple output types:

Multivariate Modelling How does the birth rate of European cities vary over population density, country GDP and time? [*model and formula description, scatter plots with fit projected to each covariate*]

Generic Analysis of Variance Taking GDP into account, does life expectancy vary between Italy, Japan and the UK? [*hypothesis statement with p-value, boxplots*]

Specific Relation How is rainfall related to population growth in Asia? [*correlation, univariate model, scatter plot with trend*]

Statistical Significance Are rainfall, government type, or GDP good predictors for population growth? [*boolean answers for each and underlying p-values*]

Specific Statistic What are the outlier thresholds for population growth in Africa? [*numerical values, explanation, boxplots*]

Description / Functional Shape How does rainfall in the UK behave over time? / Is Y multimodal/periodic/linear/exponential? [*text description / boolean answers, scatter plot with trend*]

Prediction (Not a new type, but using more sophisticated modelling methods) What will the population of Italy be in 2030? [*numerical values, model description, scatter and trace plots*]

In addition to the current numeric value retrievals and predictions, new output types include:

Specific Statistics Correlation, autocorrelation, percentiles, variance, bounds, distribution parameters, ...

Hypotheses P-Values e.g. "...X is different from Y with confidence C ..."

Various Plots Scatter plots with fits, boxplots, pairplots, high-dimensional visualisations, ...

Text Descriptions e.g. "...rainfall has a linear trend and a periodic component of period ..."

3 Design

SMART is called by FRANK at multiple stages of its execution; to explain how this occurs, a streamlined explanation of FRANK's processing is required. FRANK models queries as *alists*, i.e. association lists of attribute-value pairs with a compulsory predicate attribute (interpretable as typed n-ary relations). The values of these attributes can be marked as requiring instantiation, triggering value decomposition and generating an inference-tree with *alists* as nodes. When variables can be instantiated from a data source (i.e. once leaves are reached), parent nodes are recursively filled by aggregating their children's results through various operations (e.g. comparisons or regressions). SMART interacts with this processing pipeline as follows. Some query pre-processing is carried out immediately (by templates currently, by NLP in the near future) in order to generate simple description tags (e.g. 'Prediction', 'Significance', 'Shape', ...) to initialise the construct which handles the reasoning (**Reasoner**). Then the main SMART calls occur at the inference-tree nodes in which data-points are aggregated into datasets; there, statistical tags are generated for the data, and **Reasoner** progressively determines which steps to take, starting from the overall statistical methodology to apply (e.g. varieties of regression, analysis of variance, exploratory data analysis, ...). The system's "expertise" is contained in a small set of ontologies (a core one and one per methodology) which are navigated by a simple computational construct, called Graph State Machine (GSM). Finally, the various statistical methodologies are standardised components, each of which can produce all 3 types of outputs (the specifics of which depend on the query): *values*, *descriptions* and *visualisations*. Figure 1 depicts the interactions of the described components, which are implemented as Python libraries (though one has an extensive R back-end).

3.1 Graph State Machine

GSM [7] was designed with the goal of constructing a reasoning system which is not a blackbox, is easily interpretable and, crucially, easily programmable [2]. The result is a computational construct similar to a Turing Machine over a graph, where

[2] hardcoded logic, ontology frameworks and probabilistic languages like ProbLog did not meet these requirements

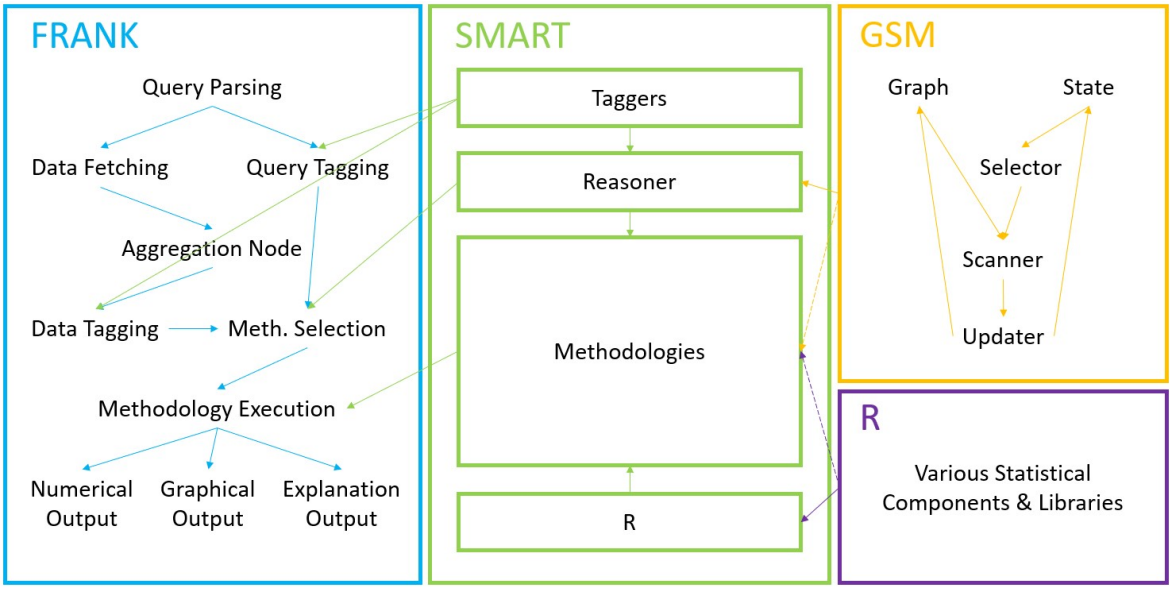

Figure 1: *Interactions of the various project components*

states are node combinations (though more information may be stored) and where the arbitrary transition function can update both state and graph. Appendix C compares the following definition to standard constructs.

Definition 3.1. Given a `Graph` with typed nodes and a `State` object of arbitrary structure, a **GSM** is defined by the functions it applies to perform a step:

`Selector` A function to extract a list of nodes from the arbitrarily-structured `State`
`Scanner` A generalised neighbourhood function which scans the graph "around" the state nodes, optionally applying some filter (e.g. node types), and returns a scored list of nodes
`Updater` A function to process the scan result and thus update the state and/or the graph itself

Figure 2 provides type signatures for the above and visualises their interactions.

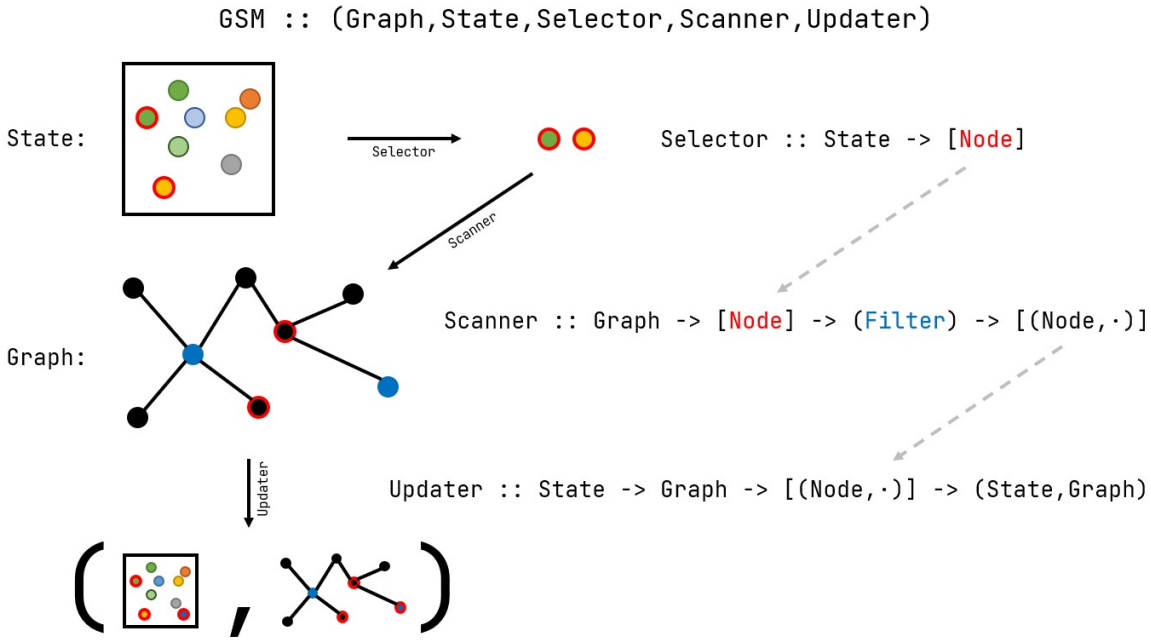

Figure 2: *Schematic representation of a GSM step (with data flow in dashed arrows): some nodes (in red) are extracted from an arbitrarily-structured* `State`*, their neighbourhoods are scanned with a filter (candidates in blue), and from the results a new node (in red) is selected*

3.2 Reasoner

The `Scanner` function of the GSM instance used by **Reasoner** (see Appendix C for details) is a Jaccard Similarity computed between the neighbours of the neighbours of the state and the state itself. In more practical terms, candidate nodes are ranked by how similar their neighbourhood is to the current state, and a candidate which has the same common nodes with the state as another but possesses a larger neighbourhood than it, is penalised. This neighbourhood-size bias is the practical translation of a preference for less generic candidates, i.e. Occam's razor, which is a key factor in the simplicity of **Reasoner**'s operation. **Reasoner**'s ontology is non-hierarchical, and its graph is interpretable on its own by knowing that an edge simply means 'is related to', which takes a more specific meaning based on the connected nodes' types. Its edges are undirected, but introducing directional 'necessity' and 'sufficiency' edge attributes is under consideration. SMART calls on **Reasoner** to make a sequence of decisions (e.g. probable response distribution, methodology to use, parameters, ...), which is extended (along with the ontology itself) when a statistical methodology is selected. Figure 3 in Appendix C depicts a step example on a portion of the ontology.

4 Methodologies

The statistical methodologies currently in SMART were selected to cover the aforementioned new query and output types; applicability overlaps are decided by the also aforementioned bias against genericness. Every statistical methodology is able to provide all three types of output to FRANK, which may highlight one and show the rest on request (when inspecting the inference-tree). Some sub-varieties of *value* outputs are predictions, bounds, parameters and statistics; some *visualisation* ones are trace plots, box plots, pair plots, combinations of them and more; and some *description* ones are shape descriptions, model definitions and process explanations. The implemented methodologies are: Generalised Linear Models (GLMs), Time Series analysis by joint Error-Trend-Seasonality Exponential Smoothing (ETS) and AutoRegressive Moving Average Process (ARMA) modelling, Analysis of Variance (ANOVA) (flexibly built on GLMs), Automatic Bayesian Covariance Discovery (ABCD) [5, 11], and a generic one for statistic computation and generation of summaries. Further methodologies under consideration are additional specific statistics and summaries, and extending GLMs to Generalised Linear Mixed Models (GLMMs) and/or Generalised Additive Models (GAMs). The back-end for all these methodologies is in R except for some statistics and ABCD, with the latter being built on the GPy-ABCD library [8, 6], also created for this project. Due to the QA nature of the system, unless a particular context is detected or provided [16], the intent is to return answers quickly, even if approximate, which is achieved in different ways across methodologies. Every methodology returns a form of uncertainty as part of its output: confidence/credibility interval, p-value or, for the current FRANK, relative standard deviation. In some cases, by assuming an underlying distribution, one type of uncertainty may be turned into another. An uncertainty example worth mentioning is that of ordered value-list output types, e.g. those arising from queries like "What is the largest X?", for which multiple values are retrieved/computed and then ordered; in these cases the returned uncertainty is essentially a p-value for the largest item actually being the largest (using the ANOVA methodology).

5 Conclusion, Limitations & Broader Impact

5.1 Statistical Automation

SMART FRANK is topical in the context of the recent "Automating Data Science" ACM article [4], meeting its predictions regarding the emphasis on interacting with and complementing the work of human users, but also managing to automate some tasks in all 4 data science quadrants: data engineering, data exploration, model building and exploitation. All 3 types of automation in the article are involved where expected: mechanization (e.g. fitting of various model types), composition (e.g. selecting and ordering analysis steps) and assistance (e.g. determining intent, providing visual and text descriptions of results and allowing interactivity throughout the process). There are, however, two general dangers related to automating statistics to any level:

- Execution of complex analyses by users who are unaware of the involved steps allows execution without understanding their limitations, making misuse of or over-reliance on results easy [9]
- Catering to expert users risks producing a steep learning curve for novices, and catering to novices risks frustrating or even actively obstructing experts [21]

With regards to the first point, FRANK's QA nature is a double-edged sword since not immediately showing used method details (i.e. showing them only on request) provides the user with the answer "as-is", and thus liable to be misused. On the other hand, FRANK's inference tree structure, and in particular its "explanation blanket" feature [14] (generating explanations for selections of inference nodes), is ideal to strike a balance in the second risk since the system *will* make decisions by itself unless otherwise instructed or in case of obvious ambivalence (also, full interactivity at decision nodes is available to provide expert users with fine control over the whole process).

5.2 Value & Future Work

SMART FRANK is in late stages of development, and a full evaluation of it is ongoing at multiple levels (individual methodologies, refinement/expansion of the ontology, full-stack SMART and entire SMART FRANK), but when complete it will be a useful and unique tool for both experts and curious novices.[3]. For the expert user its main value lies in the time and effort saved in preliminary investigation of research questions, either for pure exploration or to extract and build upon data and generated models, or even just to inspect how the answer was arrived at (sources & processing). A non-expert user may be principally interested in the direct answers to their queries, but something is to be said about the pedagogical value in the inspection and direct experimental tweaking of both SMART and FRANK features (i.e. the ontology-guided choices and the inference graph). Besides SMART, which, due to its modularity of methodologies and easy 'programming through ontology', has ample scope for expansion, FRANK is being improved on other fronts as well, such as natural language processing, user interface, and learning from past queries. A final feature which is within reach but not being worked on at the moment hinges on the interaction of the best features of both SMART and FRANK: an additional new type of query which could be implemented is that of asking data-wise open-ended questions such as "What are good predictors for X?", which FRANK would need to source intelligently by subject area and which SMART would then tackle in an appropriate variable selection process (the GLM methodology already does this on request, but variables need to be specified).

References

[1] Sara Alspaugh et al. "Building blocks for exploratory data analysis tools". In: 2013, pp. 9–17. DOI: 10.1145/2501511.2501515.

[2] Robert S. T. Amant and Paul R. Cohen. "Intelligent Support for Exploratory Data Analysis". In: *journal of computational and graphical statistics* 7.4 (1998), pp. 545–558. DOI: 10.1080/10618600.1998.10474794.

[3] A. Bernstein, F. Provost, and S. Hill. "Toward intelligent assistance for a data mining process: an ontology-based approach for cost-sensitive classification". In: *IEEE transactions on knowledge and data engineering* 17.4 (2005), pp. 503–518. DOI: 10.1109/TKDE.2005.67.

[4] Tijl de Bie et al. "Automating data science". In: *Communications of the ACM* 65.3 (2022), pp. 76–87. ISSN: 0001-0782. DOI: 10.1145/3495256.

[5] Duvenaud, David et al. "Structure Discovery in Nonparametric Regression through Compositional Kernel Search". In: 2013, pp. 1166–1174. URL: https://arxiv.org/abs/1302.4922.

[6] Thomas Fletcher. *GPy-ABCD*. 2020. URL: https://github.com/T-Flet/GPy-ABCD.

[7] Thomas Fletcher. *Graph-State-Machine*. 2020. URL: https://github.com/T-Flet/Graph-State-Machine.

[8] Thomas Fletcher, Alan Bundy, and Kwabena Nuamah. "GPy-ABCD: A configurable automatic Bayesian covariance discovery implementation". In: *8th ICML Workshop on Automated Machine Learning (AutoML)*. 2021. URL: https://openreview.net/forum?id=tyykiaedmw.

[9] Hand, David J. *Intelligent Data Analysis and Deep Understanding*. 1999.

[10] Udayan Khurana, Srinivasan Parthasarathy, and Deepak S. Turaga. "READ: Rapid data Exploration, Analysis and Discovery". In: 2014, pp. 612–615.

[11] Lloyd, James Robert et al. "Automatic construction and natural-language description of nonparametric regression models". In: 2014, pp. 1242–1250.

[3] GSM and GPy-ABCD are on PyPI and GitHub; SMART and the new FRANK will be after evaluation is complete

[12] Kwabena Nuamah. “Functional inferences over heterogeneous data”. PhD. University of Edinburgh, 2018. URL: https://era.ed.ac.uk/bitstream/handle/1842/31171/Nuamah2018.pdf.

[13] Kwabena Nuamah and Alan Bundy. “Calculating Error Bars on Inferences from Web Data”. In: 2018, pp. 618–640. DOI: 10.1007/978-3-030-01057-7{\textunderscore}48.

[14] Kwabena Nuamah and Alan Bundy. “Explainable Inference in the FRANK Question Answering System”. In: *ECAI* (2019).

[15] Kwabena Nuamah, Alan Bundy, and Christopher Lucas. “Functional Inferences over Heterogeneous Data”. In: 2016, pp. 159–166. DOI: 10.1007/978-3-319-45276-0{\textunderscore}12.

[16] Kwabena Nuamah, Alan Bundy, and Jia Yantao. “A Context Mechanism for an Inference-based Question Answering System”. In: *AAAI* (2020).

[17] Floarea Serban et al. “A survey of intelligent assistants for data analysis”. In: *acm computing surveys* 45.3 (2013).

[18] Serban, F, Kietz, J U, and Bernstein, A. *An overview of intelligent data assistants for data analysis*. 20/08/2010. DOI: 10.5167/uzh-44847.

[19] Robert St. Amant. “Navigation for Data Analysis Systems”. In: 1997, pp. 101–109.

[20] Robert St. Amant and Paul R. Cohen. “Evaluation of a semi-autonomous assistant for exploratory data analysis”. In: 1997, pp. 355–362.

[21] Robert St. Amant and Paul R. Cohen. “Interaction with a mixed-initiative system for exploratory data analysis”. In: *knowledge based systems* 10.5 (1998), pp. 265–273.

[22] Yao, Quanming et al. *Taking Human out of Learning Applications: A Survey on Automated Machine Learning*. 1/1/2018.

A Acronyms

ABCD Automatic Bayesian Covariance Discovery. 4

ANOVA Analysis of Variance. 2, 4

ARMA AutoRegressive Moving Average Process. 4, 8

ETS Error-Trend-Seasonality Exponential Smoothing. 4

FRANK Functional Reasoning for Acquiring Novel Knowledge. 1, 2, 4, 5

GAM Generalised Additive Model. 4, 8

GLM Generalised Linear Model. 4, 5

GLMM Generalised Linear Mixed Model. 4

GSM Graph State Machine. 2–5, 8

IDA Intelligent Data/Discovery Assistant. 1

NLP Natural Language Processing. 2

QA Query Answering. 1, 4, 5

SMART Statistical Methodology Advisor at Reasoning Time. 1, 2, 4, 5, 8, 9

B Basic Definitions

Analysis of Variance (ANOVA) Collection of methods to analyse differences in means between groups within a sample dataset and test a wide variety of statistical hypotheses involving them; basic ANOVA is model-less, but the most general class of model-based method is that of Mixed Models (where Covariates are labelled as Fixed vs Random Effects), which are particularly versatile. 2, 4, 6

Autocorrelation Though technically the term for correlation between any two elements X_s, X_t of the same series (or multi-dimensional indexed-set), the common use is restricted to specific-lag τ (or distance in the multi-dimensional case) correlation with the assumption of independence from specific indices (i.e. $\rho_\tau = Corr(X_s, X_{s+\tau}) = Corr(X_t, X_{t+\tau}) \, \forall \, s, t$). 8

AutoRegressive Moving Average Process A Stationary Time Series modelling method which combines the simpler Autoregressive and Moving Average processes models: ARMA models represent a given series observation X_t as a linear combination of both previous observations X_{t-i} and completely random components Z_{t-i}, as expressed in Equation B.1 (an ARMA(p,q) process)

$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{i=1}^{q} \theta_i Z_{t-i} + Z_t, \quad \text{or equivalently} \quad \Phi(B) X_t = \Theta(B) Z_t \tag{B.1}$$

They are therefore a direct combination of Autoregressive (X_t is a linear combination of previous observations plus a single random component) and Moving-Average models (X_t is just a linear combination of random components), and can thusly also be expressed in terms of their respective characteristic polynomials $\Phi(B)$ and $\Theta(B)$, where B is the backshift operator (i.e. $B(X_t) = X_{t-1}$ and $B(Z_t) = Z_{t-1}$). 4, 6

Covariate An independent variable which possibly influences an outcome variable of interest; it may or may not be of interest itself. 7

Exponential Family Distributions of random variable y with probability density function of the form $f(y; \theta) = s(y)t(\theta)e^{a(y)b(\theta)}$, where θ are some parameters and s, t, a, b are arbitrary functions. This large family includes common distributions like Normal, Exponential, Poisson, Geometric and fixed-trial-number Multinomial. 7, 8

Exponential Smoothing A Time Series smoothing technique placing exponential weights on past observations, giving the following 1-step-ahead forecast at t for smoothing parameter α:

$$\hat{X}_t(1) = \alpha \sum_{i=1}^{t} (1-\alpha)^i X_{t-i} = \alpha X_t + (1-\alpha)\hat{X}_{t-1}(1), \quad \alpha \in [0, 1] \tag{B.2}$$

. 4, 6

Fixed vs Random Effect Characterisation of Covariates with respect to their levels present in the sample vs the population:

Fixed Effects Levels of interest are finite and all present in the sample; model conclusions apply only to these levels

Random Effects The sample data contains a random selection of levels from a large population of levels; model conclusions apply to the whole level population

. 7

Generalised Additive Model An extremely versatile model type which encompasses Generalised Linear Models and (smooth) Additive Models, i.e. of the form:

$$g(\mu_i) = \eta_i = \beta_0 + \sum_{t=1}^{m} f_t(x_{ti}), \quad \text{where} \quad \boldsymbol{\mu} = E(\boldsymbol{y} | \boldsymbol{x_1}, \ldots, \boldsymbol{x_m}) \tag{B.3}$$

where the expected value $\boldsymbol{\mu}$ of a response variable $\boldsymbol{y}$ (given its predictors) from a distribution in the Exponential Family is related to a linear combination $\boldsymbol{\eta}$ of smooth functions f_t (often Smoothings functions) of its predictor variables $\boldsymbol{x_t}$ by a link function g. 4, 6

Generalised Linear Model An immediate generalisation of linear models in that the response variable is allowed to have any distribution in the Exponential Family of distributions (not just Normal); this is achieved by having the linear expression not model the response variable directly but through a link function:

$$g(\mu_i) = \eta_i = \boldsymbol{x_i}^T \boldsymbol{\beta}, \quad \text{where} \quad \boldsymbol{\mu} = E(\boldsymbol{y} | \boldsymbol{x_1}, \ldots, \boldsymbol{x_m}) \tag{B.4}$$

where the expected value $\boldsymbol{\mu}$ of a response variable $\boldsymbol{y}$ (given its predictors) from a distribution in the Exponential Family is related to a linear combination $\boldsymbol{\eta}$ of its predictor variables $\boldsymbol{x_t}$ by a link function g (the $\boldsymbol{x_i}$ above are observations of each $\boldsymbol{x_t}$, i.e. $\boldsymbol{x_i} = [x_{ti}]_{t \in [1,m]}$). 4, 6, 7

Jaccard Similarity Set similarity metric consisting of the ratio of the cardinalities of intersection over union: $J(A, B) = \frac{|A \cap B|}{|A \cup B|} \in [0, 1]$, and it is 0 and 1 respectively for disjoint and identical sets. 4

Smoothing Process and methods of modelling variable relationships without specifying regression function forms, thus often called non-parametric regression; used for both description and estimation purposes; often in GAMs. 7

Stationary Intuitively, the property of a series (or multi-dimensional indexed set) of its distribution properties (mean, variance, Autocorrelation) not varying throughout it; e.g. any Time Series with trend or seasonality is not stationary. Formal definitions and their strictness vary across literature and subject area; a standard Weak Stationarity one for Time Series is of constant & finite mean and variance with autocovariance (and Autocorrelation) depending solely on time-lag. 7

Time Series A series of values ordered by time, or, more formally, a collection of random variables $\{X_t \,|\, t \in T\}$ indexed by ordered one-dimensional set T, which may be discrete or continuous. The typical goals of Time Series analysis are description (interpretation), monitoring (for anomalies) or forecasting (prediction), and commonly trend, seasonal (periodic) effects and unexplained variation are isolated and analysed separately. The latter component is what the bulk of Time-Series-specific theory focusses on, and involves analysing Autocorrelation and applying models of the likes of ARMA. 4, 7, 8

Turing Machine A (the) universal computational construct, capable of computing any computable sequence. It is composed (in its simplest version) of a 'head' operating on an infinite tape of discrete cells on which symbols from a finite alphabet can be written. The 'head' can be in a finite set of states (including an initial and at least one final state), and its operation is determined by a transition function taking as inputs the 'head' state and the current cell symbol, and returning (applying) a new state, a new symbol and possibly a 1-cell movement to the left or right.. 2, 8

C Comparing GSM to Other Constructs

This computational construct is different from a finite state machine on a graph and from a graph cellular automaton, but it shares some similarities with both in that it generalises some of their features for the benefit of human ease of design and readability. For example, a GSM's graph generalises a finite state machine's state graph by allowing combinations of nodes to represent the state, and the scanner function is just a generalisation of a graph cellular automaton's neighbourhood function in both domain and codomain. As previously mentioned, it is closer to a Turing Machine on a graph than either of the above, one whose programming is split between the internal state rules and the graph topology, thus allowing programs to be simpler and with a more easily readable state. It is worth noting that the `Updater`'s ability to modify the graph along with the state is what makes the construct Turing complete since it allows implementing a Turing Machine with it (achieved by restricting the graph to a linear array and the state to a tuple of the current node name and "head" state). See GSM's repository [7] for a concrete (Python) version of Definition 3.1. A simple example of GSM whose `State` is a simple list of nodes but whose `Selector` is not the identity function is a Markovian `GSM` which only takes the last "visited" node into account. Going one step further, an intuitive example of `State` which is not a simple node-list is a dictionary of node-lists only some subsets of which are considered for graph exploration (while the others are, say, purely for state updating or logging, e.g. keeping track of which nodes were initial state and which ones were added by steps). This dictionary-of-node-lists variety is the type of GSM which SMART uses (with keys keeping

track of node sources, e.g. query, data, user preference, inferred, ...). In particular, SMART runs it for a finite sequence of node types, which is predefined up to selection of methodology (when domain-specific node-types are introduced), e.g. Methodology $\rightarrow \cdots \rightarrow$ Parameters $\rightarrow \cdots \rightarrow$ Output Type. Figure 3 depicts a small portion of the ontology with highlighted state nodes just before a preliminary distribution step. Ties are rare in the full ontology, and in most invoca-

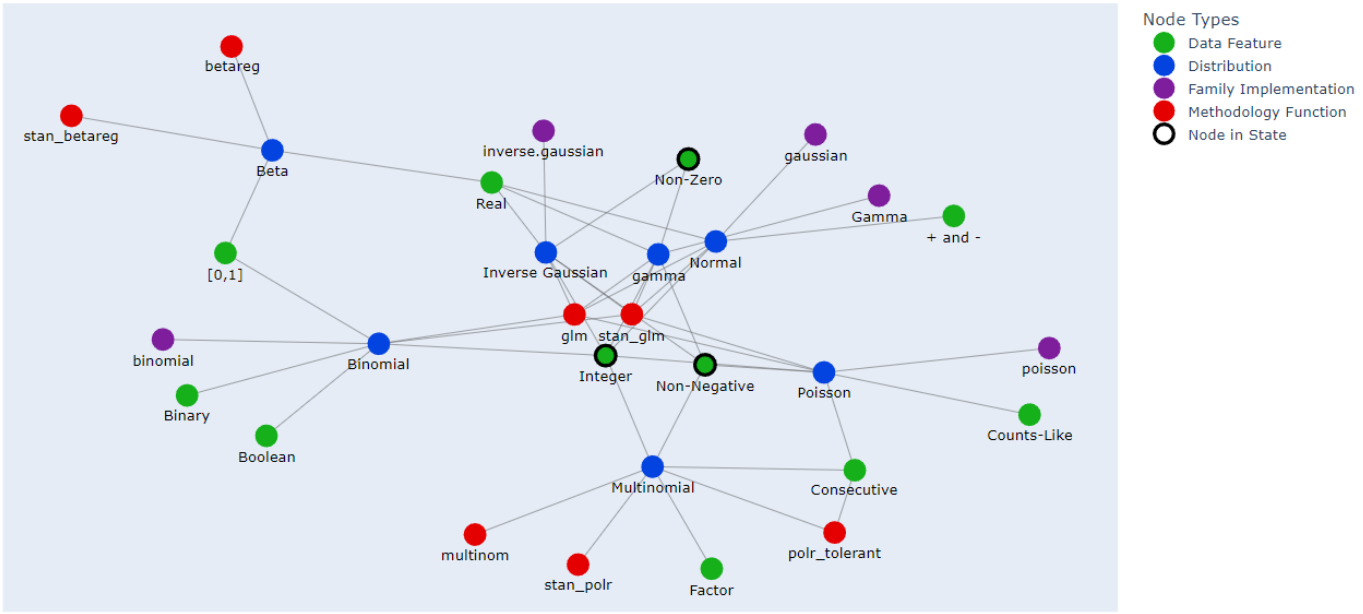

Figure 3: *Small ontology subset with highlighted state; note that 'gamma' and 'Inverse Gaussian' are the 'Distribution's whose neighbourhoods contains the most state nodes*

tions of **Reasoner** any top-scoring option is acceptable, but for the particular case of preliminary distributions guess, applicable methodologies may make use of more than one result by inspecting **Reasoner**'s log.