7/20/2016

# Fast Greedy Search (FGES) Algorithm for Discrete Variables

This document provides a brief overview of the FGES algorithm, focusing on a version of FGES that works with discrete variables called FGES-discrete (FGESd).

## Purpose

FGESd is an algorithm that takes as input a dataset of discrete variables, greedily searches over selected causal Bayesian network (CBN) structures[1] (models), and outputs the highest scoring model it finds. The model that FGESd returns serves as a data-supported hypothesis about causal relationships that exist among the variables in the dataset. Such models are intended to help scientists form hypotheses and guide the design of experiments to investigate these hypotheses.

## Methodological Approach

FGES is an optimized and parallelized version [Ramsey, 2015] of an algorithm developed by Meek [Meek, 1997] called the Greedy Equivalence Search (GES). The algorithm was further developed and studied by Chickering [Chickering, 2002]. GES is a Bayesian algorithm that heuristically searches the space of CBNs and returns the model with highest score it finds. In particular, GES starts its search with the empty graph. It then performs a forward stepping search in which edges are added between nodes in order to increase the Bayesian score. This process continues until no single edge addition increases the score. Finally, it performs a backward stepping search that removes edges until no single edge removal can increase the score.

FGESd uses the BDeu[2] scoring measure, which is described in detail in [Heckerman, 1995].

## Input Data and Parameters

FGESd has the following requirements for data input:

- the (training) data are in a table in which columns represent variables, rows represent samples, and the value of each variable in a sample is discrete.
- the first row of the table lists the variable names, in order and unique; the data and variable names are separated by a delimiter (default: tab).
- there are no missing values in the table.

FGESs takes the following parameters, which modify the behavior of the algorithm:

- depth - specifies the maximum number of edges to orient into a node during a single orientation step (FGESd uses a default search depth of -1, which indicates unlimited depth). Small search depth values will tend to reduce the search time.

[1] A CBN structure is a directed acyclic graph in which nodes represent variables and arcs represent direct causation among the nodes, where the meaning of *direct* is relative to the nodes in the CBN. For further information about CBNs, see [Spirtes, 2010; Lagani, 2016; Pearl 2016].

[2] BDeu stands for **B**ayesian **D**irichlet likelihood **e**quivalence and **u**niform. It is based on assuming a Dirichlet parameter prior probability and a multinomial likelihood. It uses Dirichlet parameter priors that guarantee that CBNs that represent the same dependence and independence relationships among the variables (by way of d-separation) are assigned the same score.

- structure-prior – for each node in a CBN, it provides the following prior probability that the node has a given set of parents:

  $$\left(\frac{e}{v-1}\right)^{p}\cdot\left(1-\frac{e}{v-1}\right)^{v-p-1}$$

  where *v* is the number of variables in the CBN, *p* is the number of parents of the child node, and *e* is a parameter that is approximately equal to the expected number of parents of the nodes in the CBN; by default we use e = 1. The structure prior of a network is equal to the product over the structure priors of each node in the network.
- sample-prior – a real valued parameter that specifies the parameter (probability) priors in the CBNs searched by FGESd. FGESd uses the BDeu scoring measure, for which the expectations of the prior probabilities are uniform. The sample-prior indicates how confident we are that these expectations are indeed uniform; the larger the sample-prior, the more confident we are. By default, sample-prior = 1, which reflects weak confidence that the probabilities in the data-generating CBN are uniform.
- disable-heuristic-speedup - FGESd by default applies a heuristic speedup; this condition is discussed briefly below. Simulation results indicate that using the heuristic has little or no negative effect on precision-recall performance and leads to a marked decrease in runtime. The user has the option to disable it by using this flag.
- skip-category-limit – FGESd has a default limit of 10 categories per variable. The user has the option to disable this validation step by using this flag.
- knowledge - the user may specify knowledge by providing a file that describes precedence and required and/or forbidden edges in the CBN structure. By default, the algorithm assumes no prior knowledge about the CBN structure
- exclude-variables – the user may specify which variables to exclude from the dataset by using this switch to point to a file that contains the name of a variable on each row.
- thread – by default the algorithm will run in a parallel fashion using as many threads as are needed and available on the system. The user has the option to specify a smaller number of threads.
- graphml - the user has the option to output the graph in standard GraphML format. By default the program will output a text file describing the causal graph and the search path.

### Output

FGESd outputs the most probable CBN structure it finds, according to the BDeu scoring measure. More precisely, it outputs a “pattern”[3] [Chickering, 2002] containing arcs[4] (→), which represent direct causation, and undirected edges (—), where such an edge indicates there is a causal arc, but its direction cannot be determined.[5]

### Algorithmic Assumptions

[3] Patterns are also known as PDAGs, essential graphs, and maximally oriented graphs.

[4] Arcs are direct edges.

[5] Methods exist for converting a pattern into a directed, acyclic graph (DAG) [Meek, 1995], which defines a CBN structure. In general, there are many possible DAGs consistent with a given pattern. Thus, a pattern defines an equivalence class of DAGs (i.e., CBN structures).

This section describes a sufficient set of assumptions for the application of FGESd to achieve the guarantees described in the next section. While the pattern output by FGESd may still include correct edges (and perhaps many correct edges) even if one or more of these assumptions are violated, there are no theoretical guarantees it will do so.

A sufficient set of conditions for recovering the causal structure of the data-generating process in the large sample limit (i.e., as the sample size grows without bound) is as follows. Assume that the causal process generating the data *D* given to FGESd is accurately modeled by a CBN containing only discrete, measured variables, which we call ***G***. Assume that each variable (node) in ***G*** is a function of its parents that is modeled by a multinomial probability distribution. Finally, assume that each case in *D* was obtained by randomly sampling all the variables in ***G*** according to the joint distribution defined by ***G***.

While the above procedure is simple, it includes several assumptions that may not be immediately obvious. Key among them are the following:

- cases (samples) in the data *D* are independent and identically distributed.
- the causal Markov condition holds [Spirtes, 2010]. This condition states that a variable is independent of its non-effects, given its direct causes (parents). It expresses a form of local causality.
- the causal faithfulness condition holds with probability 1 [Spirtes, 2010]. This condition states that all the independence relationships among the measured variables are implied by the causal Markov condition.
- there are no missing data. The user must fill in missing data before running FGESd. Many statistical packages provide methods for handling missing values, including imputing them.
- there is no measurement noise, so that the value of a node in the data generating process is equal to its measured value.
- there are no hidden confounders of the measured variables. That is, none of the measured variables have a common hidden variable as one of their direct causes (relative to the variable set). This is perhaps the strongest assumption of FGESd, because hidden confounders are typically replete in scientific data. Nonetheless, the output of FGESd may still provide helpful clues about the causal relationships among the measured variables. The Tetrad system [Tetrad, 2016] currently contains several algorithms that model latent confounders, including FCI and RFCI. The Center for Causal Discovery plans to release a relatively efficient algorithm called GFCI that models hidden confounders and uses FGES as an initial step.
- there is no selection bias. This means that the chance a case (sample) was selected from the population for inclusion in dataset *D* did not depend on the values of any of the measured variables in the data.
- there are no feedback cycles among the measured variables. Extensions to CBNs, such as causal Dynamic Bayesian Networks (DBNs) [Neapolitan, 2003], do allow feedback cycles, but they are not currently implemented in FGESd.

## Structure Learning Performance Guarantees

If the assumptions in the previous section hold, then in the large sample limit, the CBN structure output by FGESd will contain (1) an arc *X* → *Y* if and only if *X* causes *Y*; (2) an edge (—) if and only if either *X* causes *Y* or *Y* causes *X*; and (3) no edge between *X* and *Y* if and only if  *X* and *Y* have no direct causal relationship between them.

**Practice Dataset**

We used the CBN shown in Figure 1 and Table 1 to generate the simulated data shown in Table 2. The user may wish to apply FGESd (with its default settings) to the dataset in Table 2 and verify that the CBN structure obtained is the one shown in Figure 1.

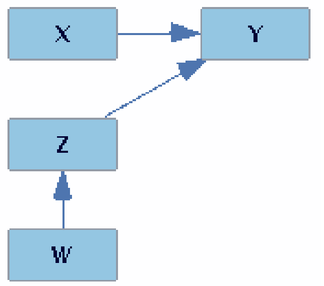

**Figure 1.** The CBN structure used to generated the practice dataset.

**Table 1**. Conditional and prior probabilities for the CBN used to generate the practice dataset. All variables are Boolean, represented by a value of 0 or 1.

| | |
|---|---|
| $P(X=0)$ | 0.5336 |
| $P(Y=0 \mid X=0, Z=0)$ | 0.5439 |
| $P(Y=0 \mid X=0, Z=1)$ | 0.4986 |
| $P(Y=0 \mid X=1, Z=0)$ | 0.8671 |
| $P(Y=0 \mid X=1, Z=1)$ | 0.9464 |
| $P(Z=0 \mid W=0)$ | 0.0603 |
| $P(Z=0 \mid W=1)$ | 0.5174 |
| $P(W=0)$ | 0.4958 |

**Table 2**. Simple practice dataset generated from the network structure and parameterization in Figure 1 and Table 1. The *MULT* column indicates how many instances of a given row we provided to the FGESd algorithm.

| *MULT* | *X* | *Y* | *Z* | *W* |
|---:|---|---|---|---|
| 8 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 |
| 55 | 0 | 0 | 0 | 1 |
| 135 | 0 | 0 | 1 | 0 |
| 71 | 0 | 0 | 1 | 1 |
| 12 | 0 | 1 | 0 | 0 |
| 51 | 0 | 1 | 0 | 1 |
| 123 | 0 | 1 | 1 | 0 |
| 64 | 0 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 |
| 96 | 1 | 0 | 0 | 1 |
| 225 | 1 | 0 | 1 | 0 |
| 111 | 1 | 0 | 1 | 1 |
| 23 | 1 | 1 | 0 | 1 |

## Performance on Simulated Data

We evaluated the performance of FGESd on simulated data. We first created a random CBN with a given number of nodes and edges, which we call $\mathrm{CBN}_{gen}$. We then randomly sampled the distribution defined by $\mathrm{CBN}_{gen}$ to generate a set of training data *D consisting of 1000 samples*. We provided that data to FGESd to obtain the pattern that it output, which we call $P_{out}$. We then derived the pattern of $\mathrm{CBN}_{gen}$, which we call $P_{gen}$. Thus, both $P_{gen}$ and $P_{out}$ are patterns. We compared $P_{gen}$ with $P_{out}$ to derive edge precision and arc recall. Two nodes are considered to have an edge between them if they have any edge type between them (i.e., $X \rightarrow Y$, $X \leftarrow Y$, or $X – Y$); in this case, we call *X* and *Y* adjacent. Edge recall is the fraction of pairs of variables adjacent in $P_{gen}$ that are also adjacent in $P_{out}$. Edge precision is the fraction of pairs of variables adjacent in $P_{out}$ that are also adjacent in $P_{gen}$. We also compared $P_{gen}$ with $P_{out}$ to derive arc (→) precision and arc recall. Arc recall is the fraction of arcs in $P_{gen}$ that appear in $P_{out}$. Arc precision is the fraction of arcs in $P_{out}$ that appear in $P_{gen}$. We also recorded the CPU time in minutes that FGESd required to derive $P_{out}$ when using a given number of processors on a machine at the Pittsburgh Supercomputing Center. The precision, recall, and timing results were averaged over 10 repeats of the process just described.

If the heuristic speedup option is selected, a few extra edges may be included in the graph in principle and some orientations may be missed that can in principle be made; nonetheless, the performance is still quite good. The performance shown in Table 2 is based on using the heuristic speedup.

**Table 2.** Performance results for FGESd on simulated data.

| # Nodes | # Edges | # Repeats | Average Edge Precision | Average Edge Recall | Average Arc Precision | Average Arc Recall | # Processors | Average Learning Time (minutes) |
|---|---|---|---|---|---|---|---|---|
| 1,000 | 1,000 | 10 | 99.9 | 79.1 | 90.7 | 43.2 | 3 | 0.03 |
| 1,000 | 2,000 | 10 | 99.9 | 83.5 | 93.4 | 66.2 | 6 | 0.03 |
| 10,000 | 10,000 | 10 | 99.8 | 72.9 | 91.3 | 39.4 | 120 | 0.75 |
| 10,000 | 20,000 | 10 | 99.5 | 46.7 | 84.6 | 25.7 | 120 | 0.76 |
| 100,000 | 100,000 | 10 | 100.0 | 71.1 | 93.1 | 90.1 | 120 | 98.1 |
| 100,000 | 200,000 | 10 | 100.0 | 44.8 | 87.4 | 23.2 | 120 | 97.1 |

The results in Table 2 provide benchmarks that may be helpful in estimating the performance of FGESd when it is applied to real datasets. We emphasize, however, that the recall and precision results obtained with such simulated data may be higher than those obtained with real datasets.

## References

Chickering DM. Optimal structure identification with greedy search. *Journal of Machine Learning Research* 3 (2002) 507-554.
http://www.jmlr.org/papers/volume3/chickering02b/chickering02b.pdf

Heckerman D, Geiger D, Chickering M. Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning (1995) 197-243.
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.222.5090&rep=rep1&type=pdf

Lagani V, Triantafillou S, Ball G, Tegner J, Tsamardinos I. Probabilistic computational causal discovery for systems biology. *Uncertainty in Biology* 17 (2016) 33-73.
http://www.mensxmachina.org/files/publications/Probabilistic%20Causal%20Discovery%20for%20Systems%20Biology_prePrint.pdf

Meek C. Causal inference and causal explanation with background knowledge. In: Proceedings of the Conference on Uncertainty in Artificial Intelligence (1995) 403-410.
https://arxiv.org/ftp/arxiv/papers/1302/1302.4972.pdf

Meek C. *Graphical Models: Selecting Causal and Statistical Models*. Ph.D. thesis, Carnegie Mellon University (1997).

Neapolitan RE. *Learning Bayesian Networks* (Pearson, 2003).

Pearl J, Glymour M, Jewell NP. *Causal Inference in Statistics – A Primer* (John Wiley & Sons, 2016).
https://books.google.com/books?hl=en&lr=&id=IqCECwAAQBAJ&oi=fnd&pg=PT1&dq=Causal+Inference+in+Statistics&ots=NPpnh1N4lC&sig=-CyGyDAsTQP1vFstnAZh3dt-Ih8#v=onepage&q=Causal%20Inference%20in%20Statistics&f=false

Ramsey J. Scaling up Greedy Equivalence Search for continuous variables (2015).
http://arxiv.org/ftp/arxiv/papers/1507/1507.07749.pdf

Spirtes P. Introduction to causal inference. *Journal of Machine Learning Research* 11 (2010) 1643-1662. http://jmlr.org/papers/volume11/spirtes10a/spirtes10a.pdf

Tetrad system (2016). http://www.phil.cmu.edu/tetrad/current.html