

Stable Specification Searches in Structural Equation Modeling Using a Multi-objective Evolutionary Algorithm

Ridho Rahmadi^{1,2}, Perry Groot², Tom Heskes²

¹ Department of Informatics, Faculty of Industrial Technology, Universitas Islam Indonesia

² Machine Learning Group, Institute for Computing and Information Science
Faculty of Science, Radboud University Nijmegen, the Netherlands
{r.rahmadi, perry, t.heskes}@cs.ru.nl

Abstract—Structural equation modelling (SEM) is a statistical technique for testing and estimating causal relations using a combination of statistical data and qualitative causal assumptions [1]–[3]. SEM allows for both confirmatory and exploratory modeling. In exploratory modeling one starts with the specification of a hypothesis, which is tested against measurements by measuring how well the model fits the data. In exploratory modeling one searches the model space without stating a prior hypothesis. Exploratory modeling has the benefit that no prior background knowledge is needed, but has the drawback that the model search space grows super-exponentially since for n variables the number of SEM models is n^{4n} . In the present paper we use an evolutionary algorithm approach to deal with the large search space in order to obtain good solutions within a reasonable amount of computation time. In addition, instead of dealing with one objective, we deal with multiple objectives to obtain more robust specifications. For this we employ the multi-objective evolutionary algorithm (MOEA) approach by using the Non-Dominated Sorting Genetic Algorithm-II (NSGA-II). At the end, to confirm the stability of a specification, we employ a stability selection approach. We validate our approach on a data set which is generated from an artificial model. Experimental results show that our procedure allows for stable inference of a causal model.

Keywords—Stable specification search, Structural Equation Modeling, Multi-objective optimization.

I. INTRODUCTION

Covariance structure modeling, which is an alternative name of structural equation modeling [4], typically has the following steps [5]: 1) set a hypothesis as the prior model, 2) fit the model to the data, 3) hypothesis testing or evaluation of the model, and 4) modification of the model to improve the parsimony and fit statistics. The last step is called *specification search* [6], [7].

Step 1 requires background knowledge of the problem domain. In practice, however, we may not always have the necessary knowledge about the underlying causal mechanism to formulate a good hypothesis. Furthermore, there may be more than one hypothesis that is similarly plausible.

Typical approaches in the literature for addressing the exploratory search problem are based on *restricted* and *unrestricted* search [5], or employing heuristic approaches such as tabu search [8], genetic algorithm [9], genetic algorithm with factor reordering [10], ant colony [11], and others [12]–[15].

Existing specification search methods are mostly driven by a single fit statistic, e.g., *chi-square*, *non-centrality parameter*, *root mean square error of approximation*. It is, however, questionable to see a single fit statistic as a magic number that tells everything about a model [16]. Instead, a fit statistic only reflects on a particular part of model fit. Therefore, it is worth trying to deal with more than one fit statistic, in order to get a more robust model specification. In addition, most of the previous work starts the specification search after an initial specified model, because presumes to have background knowledge of the data beforehand.

In this study, we propose a procedure that accounts for more than one fit statistic, using *multi-objective optimization evolutionary algorithm*. We do not start the search from an initial model and assume that all hypotheses are equally plausible a priori. We validate our approach on a data set which is generated from an artificial model. We show that our approach allows for stable inference of a causal model.

This paper is organized as follows. In Section II we describe the employed methods. In Section III we explain in detail the proposed procedure. In Section IV we describe the experimental set up and its corresponding results. In Section V and Section VI we discuss the results and conclude the study, respectively.

II. BACKGROUND

A. Structural Equation Modeling (SEM)

SEM with observed variables has two types, *recursive* and *nonrecursive*. In this study we employ the recursive one, since we presume that there are no reciprocal or feedback relations among variables. SEM requires a number of basic steps [16], [17].

The first step is *model specification*. This can be done by drawing a path diagram or by giving equations of the following form [18]:

$$\mathbf{y} = \mathbf{B}\mathbf{y} + \mathbf{\Gamma}\mathbf{x} + \boldsymbol{\zeta} \quad (1)$$

where \mathbf{y} and \mathbf{x} are vectors of endogenous and exogenous variables, respectively, \mathbf{B} and $\mathbf{\Gamma}$ are the parameter matrices, and $\boldsymbol{\zeta}$ is a vector of errors which is assumed to be uncorrelated with \mathbf{x} .

The second step is *Identification*. This step ensures that there is a unique estimate for every model parameter. There are several rules that can be used in this step, namely *t-Rule* and *Recursive Rule*. The first one is a necessary condition, while the second one is a sufficient condition. The *t-Rule* ensures that the number of parameters to estimate is less than or equal to the number of nonredundant element in the covariance matrix. The *Recursive Rule* is to ensure that the model contains no reciprocal relations by requiring \mathbf{B} to be triangular and the covariance matrix of the errors Ψ to be diagonal.

The third step is *estimation*. Let $\hat{\theta}$ contain all model parameters, including e.g., \mathbf{B} , Γ , and Ψ . Each value of $\hat{\theta}$ corresponds to a model-implied covariance matrix $\Sigma(\hat{\theta})$. In the *estimation* step we want to estimate $\hat{\theta}$ such as to get the model-implied covariance matrix $\Sigma(\hat{\theta})$ as close as possible to the sample covariance \mathbf{S} . This can be done by minimizing the residual matrix $(\mathbf{S} - \Sigma(\hat{\theta}))$, e.g., using *maximum likelihood*.

The fourth step is *hypothesis testing*. In this step, we want to validate how well our model fits the data by means of fit statistics. There are two categories of fit statistic [16], these are *model test statistic*, e.g., *chi-square* χ_M^2 , and *approximate fit indices*, e.g., *goodness-of-fit* (GFI), *standardized root mean square residual* (SRMSR).

B. Concept of Domination

The *dominance* relation between two models can be interpreted as follows. A model \mathbf{x}_1 is said to dominate model \mathbf{x}_2 , if the following conditions are satisfied [19]:

$$\mathbf{x}_1 \preceq \mathbf{x}_2 \text{ iff } \begin{cases} f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2) & \forall i \in 1, \dots, M \\ \exists j \in 1, \dots, M & f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2) \end{cases} \quad (2)$$

The first condition states that the model \mathbf{x}_1 is no worse than \mathbf{x}_2 in all objectives. The second condition states that the model \mathbf{x}_1 is strictly better than \mathbf{x}_2 in at least one objective. By using this concept, given the population of models P , we want to get a *Pareto-optimal set* or a *non-dominated set* $P^* \subseteq P$ such that models in P^* are not dominated by any member of P . Further, we have the *non-domination front* or *Pareto Front* $P\mathcal{F}^*$ which is a visualization of the non-dominated set P^* [20].

C. Non-dominated Sorting Genetic Algorithm-II (NSGA-II)

NSGA-II [21] is a non-dominated sorting-based multi-objective evolutionary algorithm (MOEA). This method has been developed to improve upon NSGA, which has problems such as time complexity of computation when running the sorting algorithm, lack of elitism (best solution of a generation), and a need of sharing parameters.

NSGA-II also introduces a number of novel procedures. First, NSGA-II uses *fast non-dominated sorting*, which sorts solutions based on the concept of domination. The time complexity has order $\mathcal{O}(MN^2)$, which is better than a naïve approach with $\mathcal{O}(MN^3)$ where M is the number of objectives and N is the population size. The second and third procedures are *crowded-distance assignment* and *crowded-comparison operator*. These procedures are built to preserve the diversity among the solutions in the Pareto fronts.

NSGA-II uses the following initialization procedure. Firstly, it creates a random population P_0 . This population

is sorted, based on the concept of domination. Thus, every solution will be assigned a rank such that 1 is the best one. Secondly, the *tournament* selection, and some manipulation operators such as *crossover* and *mutation* are implemented. These steps will generate an offspring population Q_0 of size N .

After initialization, the iterative procedure given in Algorithm 1 will be applied in the length of predetermined number of generations t . NSGA-II unifies parent population P_t and

Algorithm 1 Iterative Procedure of NSGA-II

```

1:  $R_t \leftarrow P_t \cup Q_t$ 
2:  $\mathcal{F} \leftarrow \text{fastNonDominatedSort}(R_t)$ 
3:  $P_{t+1} \leftarrow \emptyset$  and  $i \leftarrow 1$ 
4: until  $|P_{t+1}| + |\mathcal{F}_i| \leq N$ 
5:    $\text{crowdingDistanceAssignment}(\mathcal{F}_i)$ 
6:    $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_i$ 
7:    $i \leftarrow i + 1$ 
8: end
9:  $\text{Sort}(\mathcal{F}_i, \prec_n)$ 
10:  $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_i[1 : (N - |P_{t+1}|)]$ 
11:  $Q_{t+1} \leftarrow \text{makeNewPop}(P_{t+1})$ 
12:  $t \leftarrow t + 1$ 

```

offspring population Q_t into R_t having size $2N$. The next step is to apply non-dominated sorting on R_t . This step assigns all solutions in R_t with similar rank into groups, called fronts, such that \mathcal{F}_i contains all solutions with rank i .

After that, each front will be assigned with *crowding-distance* and included in P_t . If \mathcal{F}_i is bigger than N we use the crowded-comparison operator to choose the best solutions from \mathcal{F}_i in order to obtain a population of size N .

The last step is to produce a new generation Q_t by computing P_t using selection, crossover, and mutation as aforementioned. The only slight difference is that, the selection is now based on *crowded-comparison* operator \prec_n . All steps are repeated until the termination criterion is met.

D. Stability Selection

Stability selection [22] is a general method for estimating discrete structures. The method is based on subsampling together with selection algorithms. Further, it is required to set the tuning parameter $\lambda \in \Lambda \subseteq \mathbb{R}^+$ which will determine the amount of regularisation. For every $\lambda \in \Lambda$, we will get an estimate of structure $\hat{S}^\lambda \subseteq \{1, \dots, p\}$ where p is the data dimension. One also needs to set the threshold π_{thr} but according to [22], sensible values in the range (0.6, 0.9) tend to give similar results.

III. PROPOSED PROCEDURE

This section explains about the proposed procedure which is a combination of methods that are described in the previous section. Generally, the procedure works as shown in Figure 1. It has two main stages. The first stage, *search*, is the main part of specification search. It consists of an *inner loop* and an *outer loop*. The second stage, *inference*, is to infer model(s) that pass the given threshold in stability selection. The following pseudocode describes the general steps of the procedure.

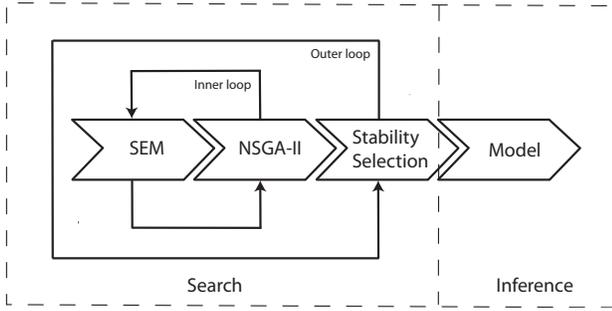


Fig. 1: Proposed procedure.

Algorithm 2 Proposed procedure

- 1: **for all** iteration j in outer loop **do**
 - 2: $M_j \leftarrow$ sample with size $\lfloor D/2 \rfloor$ without replacement
 - 3: $P_0 \leftarrow$ random specifications of size N
 - 4: $P_0 \leftarrow$ apply SEM on P_0
 - 5: $Q_0 \leftarrow \text{fastNonDominatedSort}(P_0)$
 - 6: **for all** iteration i in inner loop **do**
 - 7: $\mathcal{F}_i \leftarrow$ all NSGA-II steps on Q_0
 - 8: **end for**
 - 9: $R_j \leftarrow$ the first front of \mathcal{F}_i
 - 10: **end for**
 - 11: $G \leftarrow$ CPDAG of R
 - 12: $edge \leftarrow$ undirected edge stability of G
 - 13: $causal \leftarrow$ causal path stability of G
 - 14: $model \leftarrow$ inferring from $edge$ and $causal$
-

In the search stage, we implement two main loops namely the outer and inner loop. The outer loop consists of the iteration of the stability selection and some initial steps of NSGA-II. For each iteration in this loop, we do step 2 to 5. On step 2 we set M_j , a subset of random sampling of the data D with size $\lfloor D/2 \rfloor$ without replacement. So, the SEM computation in step 4 and in the inner loop will be based on the particular subset M_j of the current outer loop iteration. Step 3 is to generate initial random model specifications P_0 which consists of binary string of length N . All model in P_0 contains no cycle or acyclic. After that, in step 4, SEM is applied to all model specifications in P_0 . This step returns fit statistics for all models. In step 5 all model specifications are sorted based on their fit statistics using a function $\text{fastNonDominatedSort}()$ from NSGA-II. From step 5, we can see that the initial step of NSGA-II is a bit modified such that we do not implement the manipulation operators.

Still in the search stage, but we are now in the inner loop which is the iterative procedure of NSGA-II as explained in algorithm 1. Step 7 returns i first fronts where i is the number of generations for NSGA-II. This step is also a modification from the original NSGA-II which returns only one first front after all iterations. In step 9, again, function $\text{fastNonDominatedSort}()$ is employed to sort all the first fronts in \mathcal{F}_i to get a single first front R_j .

Previously, we mentioned that all generated model specifications contain no cycles. To ensure this, we developed an operator $\text{cycleRepair}()$ to handle the cycles. Generally, this operator works by detecting and repairing. Once it detects a

cycle in a generated model specification, it adjusts the model to remove cycles. Furthermore, we generate model specifications in *binary representation*. This representation is suitable for manipulation in NSGA-II, and it is suitable to represent the status of parameters in SEM such that, 1 is free or available to estimate and 0 otherwise. If the model specification contains no cycle, it can be visually represented as a *directed acyclic graph* (DAG), which is a graph with directed edges. An *acyclic partially directed graph* (PDAG) is a graph that contains both directed and undirected edges. A *completed PDAG* or CPDAG is a PDAG that consists of a directed edge for every compelled edge and an undirected edge for every reversible edge [23]. A CPDAG can be used to represent the equivalence class of DAGs [24]. DAGs that represent the same probability distribution (model) are represented by a unique CPDAG.

In the inference stage, we infer model specification(s). In step 11, we convert all DAGs to their corresponding CPDAGs. The stability of undirected edges and causal paths is measured by counting the occurrence of reversible edges and compelled edges, respectively, for a given CPDAG in each *Degree of Freedom* (DF). Namely for the causal path, the occurrence of a directed path between two variables with any length, will be considered as a causal path. Finally, in order to infer, we employ two thresholds π_{thr} and τ_{thr} . Threshold π_{thr} is a threshold of occurrence of both undirected edges and causal paths. Threshold τ_{thr} is a threshold of the *Bayesian Information Criterion* (BIC) and is set equal to the DF value at which the lowest BIC value occurs. We use the models with a DF value below τ_{thr} and above π_{thr} to infer a stable and parsimonious model specification.

IV. SIMULATION STUDY AND RESULTS

This section describes the implementation of the proposed method and the results of a simulation study. The implementation was done in R [25]. We used the *SEM* package to compute SEMs [26], the *pcalg* package to compute CPDAGs [27], some functions adopted from the *nsga2R* package [28], and a number of packages for handling matrices and graphs [29]–[31].

A. Data Set

For simulation purpose, we generated a data set of 400 samples from an artificial model represented by the following multiple regression equations:

$$\mathbf{x}_6 = 2 + 0.75\mathbf{x}_1 + 0.5\mathbf{x}_3 + 0.65\mathbf{x}_4 - 0.3\mathbf{x}_5 + \epsilon_1 \quad (3)$$

$$\mathbf{x}_1 = 1 + 1.25\mathbf{x}_2 + \epsilon_2 \quad (4)$$

The data set consists of two dependent variables $\mathbf{x}_6, \mathbf{x}_1$ and four independent variables $\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5$. They all, including error ϵ_1 and ϵ_2 , are normally distributed with mean μ and standard deviation σ as listed in Table I.

In our simulation we do not enforce any constraints, i.e., allow for interactions in both directions between all pairs of variables. In SEM, as generally explained in Section III-A about *t-Rule*, with six variables, we will have t free parameters to estimate in θ with $t \leq \frac{V(V+1)}{2}$ where V is the number of the observed variables. So that, by having six variables, there will be at most 21 free parameters to estimate. From those parameters, 15 parameters represent the relation among variables, while 6 parameters are variances of the variables.

TABLE I: Means and standard deviations used to generate the simulation data

Variables	μ	σ
x_2	0	0.5
x_3	0	1
x_4	0	0.8
x_5	0	0.7
ϵ_1	0	0.5
ϵ_2	0	0.5

B. Parameter Setting of NSGA-II

The NSGA-II algorithm was set up with the following parameters. The number of generations (inner loop) is 10, the size of initial population P_0 is 50, *one-point* crossover with rate 1.0, *1-bit flip* mutation with rate 0.2. For the initial steps of NSGA-II, we did not employ the *tournament* selection, instead, we used *truncation* selection with threshold 80% and complete the 20% of the population by newly-random generated solutions (model specifications). In the NSGA-II multi-objective, we use the *chi-square* χ_M^2 and *degree of freedom* (DF). Minimizing χ_M^2 results in minimizing the discrepancy between \mathbf{S} and $\Sigma(\hat{\theta})$ and maximizing DF results in a more parsimonious model.

C. Parameter Setting of Stability Selection

We generated 400 subsets, randomly sampling without replacement with size $\lfloor D/2 \rfloor$, where $D = 400$. Thus, the outer loop has 400 iterations and hence the number of first fronts. We also set DF as tuning parameter λ , threshold $\pi_{thr} = 0.6$ and threshold $\tau_{thr} = \min(\text{average}(BIC))$.

D. Results

The results can be seen in Figures 2, 3, 4 and 5. The simulation is based on 400 subsets. Each subset consists of 10 generations and each generation consists of 50 model specifications. Thus, approximately, 200,000 models are evaluated. Figure 2 depicts the mean and the standard deviation of χ_M^2 for every DF. Since the objectives are to minimize χ_M^2 and to maximize DF, in Figure 2, the preferred model is any model that is close to the right bottom corner. In addition, we can see that the deviation of χ_M^2 is higher in higher degrees of freedom, except for DF = 15. Note that, the higher DF means the more parsimonious model specification.

From Figures 3 and 4, we can observe the stability of the undirected edges and causal paths between variables. The y axis is normalized to the range of [0,1]. The x axis is set from the most parsimonious model specification, which is 15, to the most complex one, which is 0. So, DF = 15 means there are 15 free parameters while DF = 0 means there is no free parameter. Threshold π_{thr} (stability) and τ_{thr} (BIC) are plotted horizontally and vertically, respectively. As mentioned before, $\pi_{thr} = 0.6$ and τ_{thr} is set to the minimum average of BIC. $\tau_{thr} = -27.90786$ is found as the minimum average of BIC at DF = 7.

By plotting the thresholds, we get four divisions of the region on the plot. The region in which all the undirected edges and causal paths pass both thresholds is the top left region. We

can infer the stable model specification by selecting the line which has at least a connecting dot inside the inferring region.

In this simulation, we find six stable undirected edges (Figure 3) and five stable causal paths (Figure 4). The stable undirected edges are between variables $x_1 - x_2$ (line a), $x_1 - x_6$ (line b), $x_2 - x_6$ (line f), $x_3 - x_6$ (line d), $x_4 - x_6$ (line c) and $x_5 - x_6$ (line e). The stable causal paths are from variable x_1 to x_6 (line 3), x_2 to x_6 (line 4), x_3 to x_6 (line 2), x_4 to x_6 (line 1) and x_5 to x_6 (line 5). Figure 5 shows the corresponding model.

V. DISCUSSION

The inferred model specification can be interpreted as follows. A stable edge between two variables indicates that there is a relation between two variables which is not mediated by any other (observed) variable. A causal path can be both direct and indirect. Comparing with the model that generated the data (Equations 3 and 4), we see that all edges are correctly inferred, but there is one superfluous edge between x_2 and x_6 . Apparently, it is difficult to detect that the strong correlation between x_2 and x_6 is mediated by x_1 . All causal paths are correctly inferred, even the one from x_2 to x_6 . The causal path from x_2 to x_1 is impossible to detect from purely observed data: a causal link from x_1 to x_2 gives exactly the same conditional dependencies and independencies as a causal link from x_2 to x_1 .

From the part of specification search, the inner loop (NSGA-II) is able to search through the search space of the models while still maintaining two important features: the trade-off between conflicting objectives and the diversity preservation of the front. The diversity of the front can be seen from Figure 2, that is, the deviation of χ_M^2 in different DF among the first fronts is quite fluctuating, especially with a higher DF. Further, with Intel Core i5 machine, 4GB RAM and under Windows 7, it took time approximately 160 minutes to run the whole analysis.

The idea of subsampling with size $\lfloor D/2 \rfloor$ in stability selection ensures that all edges and causal paths in the inferring region are reasonably stable. This is because they appear through different various subsets. So that by inferring a model from this region, we can expect to get a stable and parsimonious model.

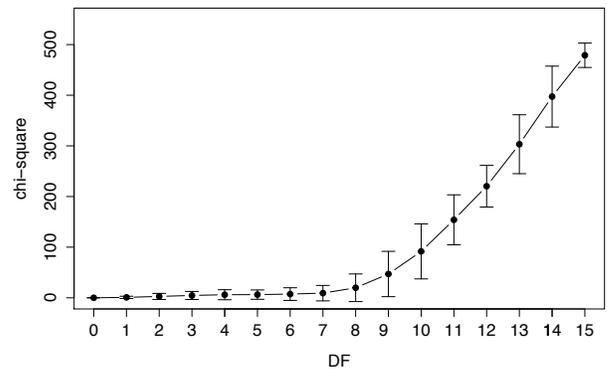


Fig. 2: The mean and standard deviation of 400 fronts

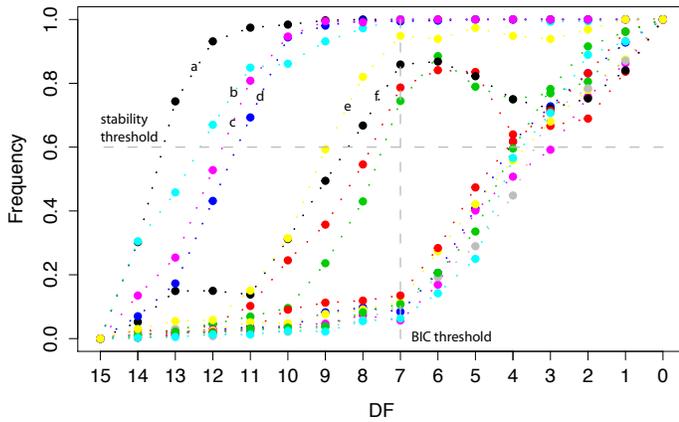


Fig. 3: Stability of edges.

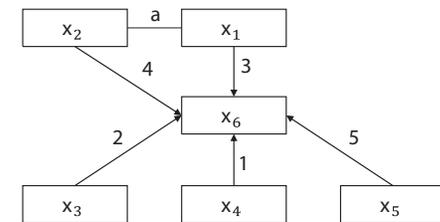
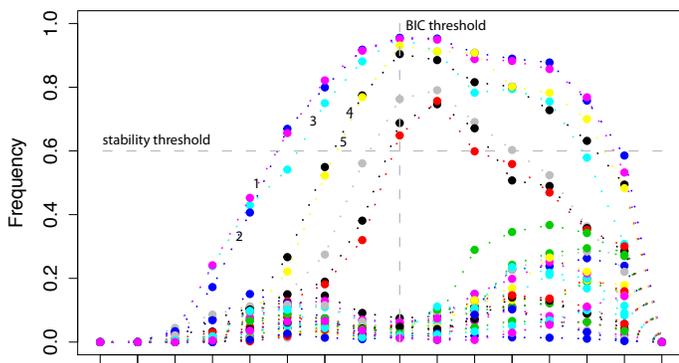


Fig. 5: The inferred model specification.

VI. CONCLUSION

The structure in structural equation models is often based on prior knowledge. When insufficient prior knowledge is available to fully determine the model structure, specification search provides an alternative approach. Specification search boils down to a high-dimensional discrete optimization problem: it aims to find the structure that minimizes some error criterion measuring how well the SEM fits the available data. Such high-dimensional discrete optimization problems are known to be both computationally expensive (since the search space over structures can be huge) and highly unstable

(small changes in the data can lead to quite different optimal solutions). In this paper, we propose a procedure which tackles both issues, using a combination of stability selection and multi-objective evolutionary optimization.

Stability selection is a recently introduced resampling procedure for robust estimation of model structure. It needs to compute the optimal model structures for a whole range of model complexities (in the original paper in terms of a regularization path, here by varying the number of degrees of freedom) and for different draws of the original data (here and in the original paper, half of the samples, randomly drawn). The key insight in this paper is that we can efficiently generate the optimal solutions for all degrees of freedom in one go by making use of a multi-objective evolutionary algorithm that takes some fit criterion (here chi-squared) and the degrees of freedom itself as the two conflicting objectives to be jointly optimized. These optimal solutions can then be treated as in standard stability selection to obtain those parts of the model structure that are stable across statistical fluctuations of the data.

Experimental results on the artificial data set show that our procedure allows for stable inference of a causal model. However, our current implementation still considers only observed variables at one time point. In future work, we also aim to account for latent variables and to extend our procedure to longitudinal data covering multiple time points.

ACKNOWLEDGMENT

The authors would like to thank to DGHE of Indonesia and OPTIMISTIC to support this study.

REFERENCES

- [1] S. Wright, "Correlation and causation," *Journal of agricultural research*, vol. 20, no. 7, pp. 557–585, 1921.
- [2] H. A. Simon, "Causal ordering and identifiability," in *Models of Discovery*. Springer, 1977, pp. 53–80.
- [3] J. Pearl, *Causality: models, reasoning and inference*. Cambridge Univ Press, 2000, vol. 29.
- [4] J. Hox and T. Bechger, "An introduction to structural equation modelling," *Family Science Review*, vol. 11, no. 354–373, 1998.
- [5] R. MacCallum, "Specification searches in covariance structure modeling," *Psychological Bulletin*, vol. 100, no. 1, p. 107, 1986.
- [6] E. E. Leamer and E. E. Leamer, *Specification searches: Ad hoc inference with nonexperimental data*. Wiley New York, 1978.
- [7] J. S. Long, *Covariance structure models: An introduction to LISREL*. Sage, 1983, no. 34.
- [8] G. A. Marcoulides, Z. Drezner, and R. E. Schumacker, "Model specification searches in structural equation modeling using tabu search," *Structural Equation Modeling: A Multidisciplinary Journal*, vol. 5, no. 4, pp. 365–376, 1998.
- [9] G. A. Marcoulides and Z. Drezner, "Specification searches in structural equation modeling with a genetic algorithm," *New developments and techniques in structural equation modeling*, pp. 247–268, 2001.
- [10] H. Murohashi and H. Toyoda, "Model specification search using a genetic algorithm with factor reordering for a simple structure factor analysis model," *Japanese Psychological Research*, vol. 49, no. 3, pp. 179–191, 2007.
- [11] G. A. Marcoulides and Z. Drezner, "Model specification searches using ant colony optimization algorithms," *Structural Equation Modeling: A Multidisciplinary Journal*, vol. 10, no. 1, pp. 154–164, 2003.
- [12] J. Herting and H. Costner, "Respecification in multiple indicator models," *Causal Models in the Social Sciences*, pp. 321–393, 1985.

- [13]P. Spirtes, R. Scheines, and C. Glymour, "Simulation studies of the reliability of computer-aided model specification using the TETRAD II, EQS, and LISREL programs," *Sociological Methods & Research*, vol. 19, no. 1, pp. 3–66, 1990.
- [14]W. E. Saris, A. Satorra, and D. Sörbom, "The detection and correction of specification errors in structural equation models," *Sociological methodology*, vol. 17, pp. 105–129, 1987.
- [15]S. Harwood and R. Scheines, "Genetic algorithm search over causal models," 2002.
- [16]R. Kline, *Principles and Practice of Structural Equation Modeling*, ser. Methodology in the social sciences. Guilford Press, 2011.
- [17]K. Bollen, *Structural Equations With Latent Variables*. WILEY, 1989.
- [18]J. Pearl, "The causal foundations of structural equation modeling," DTIC Document, Tech. Rep., 2012.
- [19]K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons Chichester, 2001, vol. 2012.
- [20]K. Deb, "Introduction to evolutionary multiobjective optimization," in *Multiobjective Optimization*. Springer, 2008, pp. 59–96.
- [21]K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [22]N. Meinshausen and P. Bühlmann, "Stability selection," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 4, pp. 417–473, 2010.
- [23]D. M. Chickering, "Learning equivalence classes of Bayesian-network structures," *The Journal of Machine Learning Research*, vol. 2, pp. 445–498, 2002.
- [24]M. Kalisch and P. Bühlmann, "Robustification of the PC-algorithm for directed acyclic graphs," *Journal of Computational and Graphical Statistics*, vol. 17, no. 4, pp. 773–789, 2008.
- [25]R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2013. [Online]. Available: <http://www.R-project.org/>
- [26]J. Fox, Z. Nie, and J. Byrnes, *sem: Structural Equation Models*, 2013, R package version 3.1-3. [Online]. Available: <http://CRAN.R-project.org/package=sem>
- [27]M. Kalisch, M. Machler, D. Colombo, M. H. Maathuis, and P. Bühlmann, "Causal inference using graphical models with the R package pcalg," *Journal of Statistical Software*, vol. 47, no. 11, pp. 1–26, 2012. [Online]. Available: <http://www.jstatsoft.org/v47/i11/>
- [28]C.-S. Tsou, *nsga2R: Elitist Non-dominated Sorting Genetic Algorithm based on R*, 2013, R package version 1.0. [Online]. Available: <http://CRAN.R-project.org/package=nsga2R>
- [29]F. Novomestky, *matrixcalc: Collection of functions for matrix calculations*, 2012, R package version 1.0-3. [Online]. Available: <http://CRAN.R-project.org/package=matrixcalc>
- [30]R. Gentleman, E. Whalen, W. Huber, and S. Falcon, *graph: A package to handle graph data structures*, R package version 1.40.1.
- [31]J. Fox and S. Weisberg, *An R Companion to Applied Regression*, 2nd ed. Thousand Oaks CA: Sage, 2011. [Online]. Available: <http://socserv.socsci.mcmaster.ca/jfox/Books/Companion>