WIRTSCHAFTS
UNIVERSITÄT
WIEN VIENNA
UNIVERSITY OF
ECONOMICS
AND BUSINESS

Chain Graph Models in R: Implementing the Cox-Wermuth Procedure

Outline

1. Brief Introduction to Graphical Models
2. The `coxwer` function: Fitting Chain Graph Models via the Cox-Wermuth Heuristic
3. Illustration: Contraceptive Method Choice
4. Conclusion and Outlook

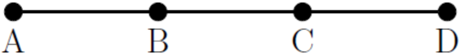
This is joint work with Marcus Wurzer and Reinhold Hatzinger.

Graphical Models: General

- Graphical models (GM) allow multivariate analysis of complex dependency structures
- They are probability distributions over a multidimensional space encoded by graphs (as a set of vertices/variables, V, and a set of edges/relationships between variables, E)
- Different types: undirected GM (e.g., Markov random fields), directed GM (e.g., Bayesian Networks, DAG), Chain GM
- GM represent multivariate dependencies by conditional dependence and independence statements
- Thus they can help in reducing overall complexity and allow model formulation, identification and selection

Graphical Models: Example

A simple graphical model (a Markov random field):



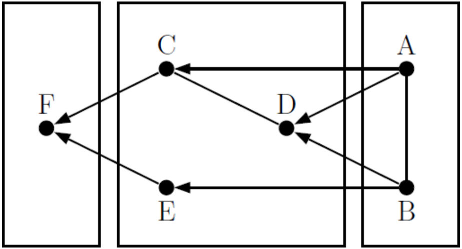
- In GM the Markov property of graphs allows to factorize the distribution F_V into a set of conditional distributions, e.g., for $V = \{A, B, C, D\}$ by way of densities: $f_V = f_{A|B} \times f_{B|C} \times f_{C|D} \times f_D$
- Thus the problem of fitting graphical models effectively reduces to estimating a series of conditional distributions

Chain Graph Models: General

- Chain graph models (CGM) are a mixture of directed and undirected graphical models
- They are particularly interesting for social and behavioral sciences (observational studies, complex multivariate dependencies, existing substantive knowledge)
- In CGM, all variables are assigned to boxes (disjoint variable subsets $V_t, V = \bigcup_t V_t$) by theory or substantive knowledge
- Between boxes exist directed edges, within boxes the edges are undirected
- Two types of CGM:
  - Univariate recursive regression graph model (URRG; one variable per block)
  - Joint response chain graph model (JRCG; more than one variable per block)

Chain Graph Models: Factorization

A joint response chain graph model:



- In CGM factorization happens at least recursively between blocks: $f_V = f_{V_T | V_{T-1}, \ldots, V_1} \times f_{V_{T-1} | V_{T-2}, \ldots, V_1} \times \cdots \times f_{V_1}$.
- Possibly additional conditional independence by missing edges, e.g., for the above graph $f_V = f_{F|C,E,D,A,B} \times f_{C,E,D|A,B} \times f_{A,B} = f_{F|C,E} \times f_{C,D|A,B} \times f_{E|B} \times f_{A,B}$

Chain Graph Models: Estimation

- For CGM there are no theoretical restrictions on the form of the conditional distributions (though usually conditional Gaussian distributions; Lauritzen & Wermuth, 1989)
- In particular variable types can be of mixed type within and between boxes (discrete and continuous components)
- General algorithms for computing estimates in every CGM under every possible variable type specification are not yet available
- Fitting the conditional distributions of the factorization with a series of multiple univariate conditional regressions is feasible (Wermuth & Cox, 2001)
- Cox & Wermuth (1996; see also Caputo et al., 1997) lay out ideas for a data-driven, heuristic selection strategy to approximate the CGM by univariate conditional regressions

The coxwer Functionality in R

We implemented an algorithm based on the ideas of the Cox-Wermuth heuristic in R for approximate fitting of JRCG and URRG models.

Currently, there are the following functions intended for the user:

`cw-class`	S3 class for objects from a Cox-Wermuth fit
`coxwer`	Fit a JRCG or a URRG via Cox-Wermuth selection strategy
`prep_coxwer`	Setup of variable frame, block membership and variable type (interactive)
`summary, print` `plot, predict`	S3 methods for class cw
`adjmatrix`	Extracts the adjacency matrix
`write_cw`	Writes and saves the graph in igraph format

Using the coxwer Function I

- `coxwer` arguments are a variable frame and an observations × variables data frame.
- The variable frame defines the block and type of a variable. It must have the same row names as the data frame has column names.

```
              type block
age           cont     5
wifeEdu        ord     4
husbEdu        ord     4
nrChild      count     1
wifeRel        bin     4
wifeWork       bin     4
husbOcc      categ     4
solIndex       ord     3
mediaExp       bin     2
contraceptive categ    1
```

- The `prep_coxwer` function allows to define the variable frame interactively.

Using the `coxwer` Function II

- Further arguments to `coxwer` are:
  - `adjfile`: Save the adjacency matrix to this file.
  - `autodetect`: Automatically assign the data type to the variables in the data frame according to variable type in the variable frame.
  - `pen`, `signif`: Parameters for screening and model selection. `pen` is the penalty for the information criterion used in `stepAIC` and `signif` the significance level when screening for higher-order effects and non-linearities.
  - `contrasts`: The contrasts to be used for categorical predictors. Defaults to dummy coding for ordered and unordered factors.
  - `silent`: Flag for whether model fitting progress should be printed.

The coxwer Selection Algorithm

- Our algorithm is roughly the following (cf. Caputo et. al., 1997):
  1. Start in the block with the lowest number
  2. Take one variable from that block. Fit main effects model with all the variables in the same block or higher block.
  3. Screen for quadratic effects (metric variables) and two-way interactions by adding of single terms. Retain the ones with an associated p-value < `signif`.
  4. Fit the model with main and retained effects.
  5. Use backward selection to reduce the model.
  6. Re-enter interactions for the terms that remain in the model.
  7. Use backward selection.
  8. Re-enter quadratic terms for remaining effects.
  9. Use backward selection.
  10. If other variables in the same block: Repeat for them. Else: jump to next block and repeat.

Univariate Models used by coxwer

- For binary targets: binomial logistic models
  `stats::glm(...,family=binomial,link=logit)`
- For unrestricted continuous targets: OLS/Gaussian linear models
  `stats::glm(...,family=gaussian,link=identity)`
- For positive continuous targets: gamma or inverse Gaussian GLM
  `stats::glm(...,family=Gamma,link=inverse)`
  `stats::glm(...,family=inverse.gaussian,link=1/mu^2)`
- For count targets: Poisson/negative binomial loglinear models
  `MASS::glm.nb(...,link=log)`
- For categorical targets: multinomial logistic models
  `nnet::multinom(...,link=logit)`
- For ordinal targets: proportional odds logistic models
  `MASS::polr(...,link=logit)`

CMC: Data

- For illustration we fit a JRCGM for contraceptive methods choice (CMC) in a subset of the 1987 National Indonesia Contraceptive Prevalence Survey (Lim et. al., 1999)
- Overall we have 1473 observations of married women on 10 variables.
  - Age (age; continuous)
  - Education (wifeEdu; ordinal 1=low, 2, 3, 4=high)
  - Husband's education (husbEdu; ordinal 1=low, 2, 3, 4=high)
  - Number of children ever born (nrChild; count)
  - Religion (wifeRel; binary; 0=Non-Islam 1=Islam)
  - Wife's now working? (wifeWork; binary 0=Yes, 1=No)
  - Husband's occupation (husbOcc; categorical 1, 2, 3, 4)
  - Standard-of-living index (soliNdex; ordinal 1=low, 2, 3, 4=high)
  - Media exposure (mediaExp; binary 0=Good, 1=Not good)
  - Contraceptive method used (contraceptive; categorical 1=No-use 2=Long-term 3=Short-term)

CMC: Blocks

- Blocks
  - Block 1 - Dependent variables: contraceptive, nrChild
  - Block 2 - Intermediate variable: mediaExp
  - Block 3 - Intermediate variable: solIndex
  - Block 4 - Intermediate variables: wifeEdu, husbEdu, wifeRel, wifeWork, husbOcc
  - Block 5 - Purely explanatory variable: age

CMC: coxwer Results

```
> cmc_prep <- prep_coxwer(cmc)
> res.cmc <- coxwer(cmc_prep, cmc)
```

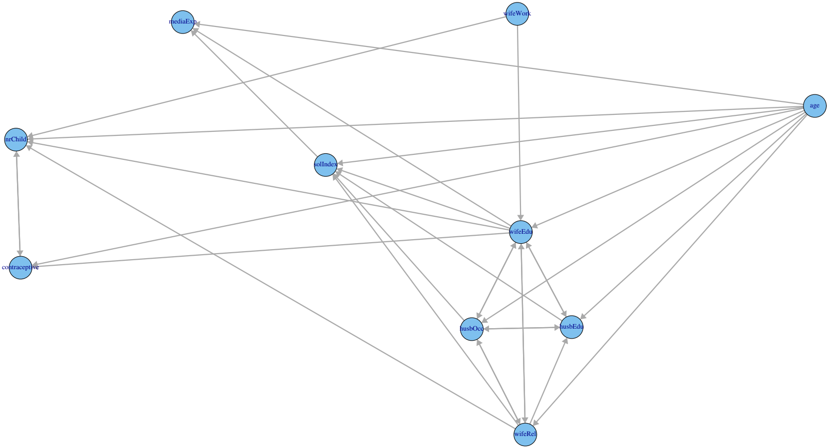
```
TARGET: nrChild (poisson loglinear model)
TARGET: contraceptive (multinomial logit model)
TARGET: mediaExp (binomial logit model)
TARGET: solIndex (proportional odds logit model)
TARGET: wifeEdu (proportional odds logit model)
TARGET: husbEdu (proportional odds logit model)
TARGET: wifeRel (binomial logit model)
TARGET: wifeWork (binomial logit model)
TARGET: husbOcc (multinomial logit model)
```

```
> print(res.cmc)
```

```
Adjacency Matrix:

                  1  2  3  4  5  6  7  8  9 10
 1 age            0  1  1  1  1  0  1  1  1  1
 2 wifeEdu        0  0  1  1  1  0  1  1  1  1
 3 husbEdu        0  1  0  0  0  0  1  1  0  0
 4 nrChild        0  0  0  0  0  0  0  0  0  1
 5 wifeRel        0  1  1  1  0  0  1  1  0  0
 6 wifeWork       0  1  0  1  0  0  0  0  0  0
 7 husbOcc        0  1  1  0  1  0  0  1  0  0
 8 solIndex       0  0  0  0  0  0  0  0  1  0
 9 mediaExp       0  0  0  0  0  0  0  0  0  0
10 contraceptive  0  0  0  1  0  0  0  0  0  0
```

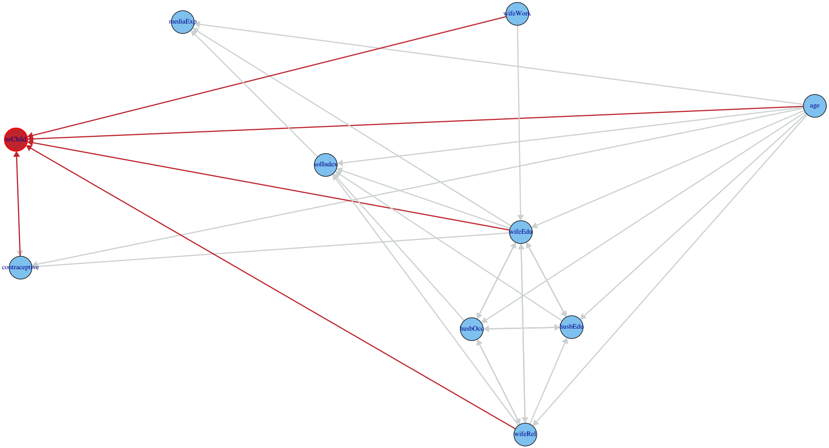
CMC: Joint Response Chain Graph

```
> plot(res.cmc)
```



CMC: Target “nrChild”

```
> plot(res.cmc)
```

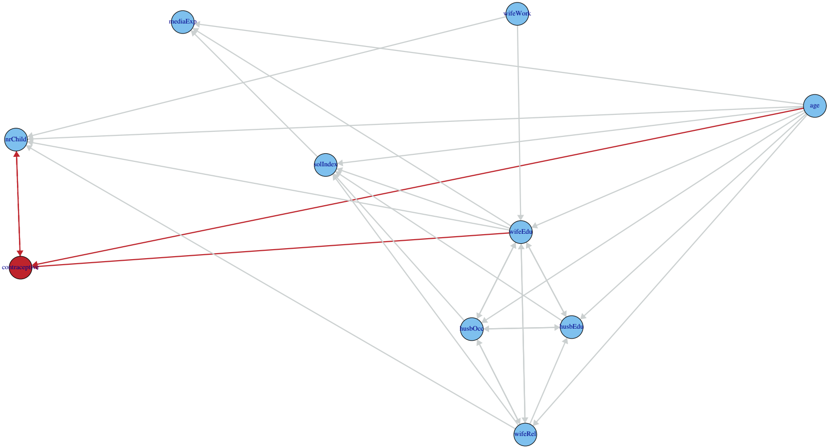


CMC: Model for “nrChild”

```
> summary(res.cmc,target=c("nrChild","contraceptive"))
---------- Summary for dependent variable: nrChild ----------

Call:
stats::glm(formula = y ~ age + wifeEdu + wifeRel + wifeWork +
    contraceptive + I(poly(age, 2)[, 2]), family = curr.family,
    data = dat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.3620  -0.6483  -0.1031   0.5343   3.5907

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)          -1.228343   0.110211 -11.145  < 2e-16 ***
age                   0.058168   0.002117  27.480  < 2e-16 ***
wifeEdu2              0.012220   0.050068   0.244    0.807
wifeEdu3             -0.075736   0.049643  -1.526    0.127
wifeEdu4             -0.351352   0.049615  -7.082 1.42e-12 ***
wifeRel1              0.263919   0.044373   5.948 2.72e-09 ***
wifeWork1             0.171091   0.035053   4.881 1.06e-06 ***
contraceptive2        0.334047   0.039516   8.454  < 2e-16 ***
contraceptive3        0.348241   0.035753   9.740  < 2e-16 ***
I(poly(age, 2)[, 2]) -5.163229   0.622035  -8.301  < 2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

(Dispersion parameter for poisson family taken to be 1)
```

CMC: Target “contraceptive”

```
> plot(res.cmc)
```



CMC: Model for “contraceptive”

```
> summary(res.cmc,target=c("nrChild","contraceptive"))
---------- Summary for dependent variable: contraceptive ----------
Call:
nnet::multinom(formula = y ~ age + wifeEdu + nrChild + I(poly(nrChild,
    2)[, 2]), data = dat, Hess = TRUE, trace = FALSE, MaxNWts = 5000)

Coefficients:
  (Intercept)         age  wifeEdu2  wifeEdu3 wifeEdu4   nrChild
2   -2.292873 -0.04835992 0.8820847 1.8373202 3.096257 0.3578242
3    1.745353 -0.11908511 0.2365778 0.6442521 1.337352 0.3558117
  I(poly(nrChild, 2)[, 2])
2                -25.60374
3                -26.44224

Std. Errors:
  (Intercept)        age  wifeEdu2  wifeEdu3  wifeEdu4    nrChild
2   0.5138863 0.01211590 0.4047368 0.3869659 0.3816910 0.04444398
3   0.3756312 0.01136707 0.2482052 0.2452609 0.2461524 0.04057962
  I(poly(nrChild, 2)[, 2])
2                 3.570454
3                 3.223996


Residual Deviance: 2708.166
AIC: 2736.166
```

Conclusion

- Applicability
  - The procedure allows to explore multivariate dependencies and approximate the real CGM
  - Neglects some information in the multivariate structure (loss of efficiency)
  - Validity of equivalence of Markovian properties for the whole graph is not ensured
- Program
  - Intended to further broaden the availability and applicability of algorithms for graphical models in R.
  - Provides a unified, user-friendly way of approximately fitting CGM with mixed variable types.
  - Implementation can be used as a building block in even more complicated computational tasks, e.g., Wurzer & Hatzinger (2013).
  - The coxwer procedure is not very fast and computing time increases massively for a large number of variables.

Outlook

Current future plans

- Release it (look for gRchain or chaingraphs on R-Forge)
- Extend support to other variable types
- Formula interface, normalizing of inputs and standardized effects
- New screening option that does not rely on *p* values
- New model selection option by L1-regularization
- New way of treating within-block association
- Unified model summary
- Add support for model diagnostics and interpretation
- Leverage/use/embed functionality offered in packages such as ggraph, gRBase, igraph,...
- Incorporate measurement models/latent variables

References I

- Caputo, A., Heinicke, A. & Pigeot, I. (1997). A graphical chain model derived from a model selection strategy for the sociologists graduates study. *Collaborative Research Center 386, Discussion Paper 73.*
- Cox, D. & Wermuth, N. (1996). *Multivariate Dependencies: Models, Analysis, Interpretation*. Florida:Chapman&Hall/CRC.
- Cox, D. & Wermuth, N. (2001). Joint response graphs and separation induced by triangular systems. *Research Report, Australian National University*.
- Lauritzen, S. & Wermuth, N. (1989). Graphical models for associations between variables, some of which are qualitative and some quantitative. *The Annals of Statistics, 31–57.*
- Lim, T.-S., Loh, W.-Y. & Shih, Y.-S. (1999). A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-three Old and New Classification Algorithms.*Machine Learning, 40, 203–238.*

References II

- Wermuth, N. (1998). Graphical Markov models. *Encyclopedia of Statistical Science. Update, 2*.
- Wermuth, N. (2003). Analysing social science data with graphical Markov models. *Oxford Statistical Science Series, 33–38*.
- Wurzer, M. & Hatzinger, R. (2013). Using Graphical Models in Microsimulation. In C. O'Donoghue (ed.), *New Pathways In Microsimulation*. Ashgate Publishing, forthcoming.

Thank you for your Attention

Thomas Rusch
Center for Empirical Research Methods
email: `thomas.rusch@wu.ac.at`
URL: `http://wu.ac.at/methods/en/hum/trusch`

WU Vienna University of Economics and Business
Augasse 2–6, A-1090 Vienna