Modeling the Effects of Nutrition with Mixed-Effect Bayesian Network

Jari Turkia, jari.turkia@cgi.com
CGI
University of Eastern Finland

Abstract

This work proposes Mixed-Effect Bayesian Network (MEBN) as a method for modeling the effects of nutrition. It allows identifying both typical and personal correlations between nutrients and their bodily responses. Predicting a personal network of nutritional reactions would allow interesting applications at personal diets and in understanding this complex system. Brief theory of MEBN is first given, followed by the implementation in R and Stan. A real life dataset from a nutritional study (Sysdimet) is then analyzed with this method and the results are visualized with a responsive JavaScript-visualization.

The Effects of Nutrition

Nutrition experts have known for a long by their experience that people can react very differently to the same nutrition. Typical reactions are quite well known from existing nutritional studies, but personal reactions may differ from them. Are some people more sensitive to some nutrients than the others? If this information could be systematically quantified, it would allow us to create personal models of reaction types. This in turn, would open up a lot of new applications in personal dietary recommendations and in personal health care.

I am proposing a Mixed-Effect Bayesian Network (MEBN) as a method for modeling the effects of nutrition in both population and personal level. By the effects of nutrition we mean the way how people react to different levels of nutrients at their diets. There have been studies of these effects on specific cases, like personal glucose metabolism (Zeevi et al. 2015), but MEBN would allow more general modeling.

The Bayesian Networks (BN) are directed acyclic graphical models that can contain both observed and latent random variables as vertices and edges as indicators of connection. In the setting of nutritional modeling the observed variables are nutrients at person's diet and their corresponding bodily responses, like blood characteristics. The connections, and especially the indirect connections, between these variables can be very complex and the BN seems to be an intuitively appealing method for modeling such a system. The latent variables at the graph correspond to both typical and personal variables indicating the significance of these connections.

For capturing these personal variances we need a set of data that contains several repeated measurements from number of persons. The measurements from each person are correlated with each other and a general method for modeling this kind of correlations is a hierarchical, or mixed-effect, model. Previously BNs have been considered mainly for uncorrelated observations (Scutari and Denis 2014, Nagarajan and Scutari (2013), Aussem et al. (2010)), but in this work we are using Mixed-Effect Bayesian Network parameterization (Bae et al. 2016) that allows combining hierarchical modeling to Bayesian networks.

This presentation covers first briefly the theory of graphical models and how it can be expanded to correlated observations. Then it is shown how this modeling can be implemented with Stan and what benefits that fully Bayesian estimation can offer in understanding the uncertainty at the model.

Mixed-Effects Bayesian Network

Let us denote the graph of interconnected nutrients and responses with G. We can then formulate the modeling problem as finding the graph G that is the most probable given the data D

$$P(G|D) \tag{1}$$

By using the Bayes' Rule we can be split this probability into proportions of the data likelihood of the given graph and any prior information we might have about suitable graphs

$$P(G|D) \propto P(D|G)P(G) \tag{2}$$

Now the problem is converted into a search of the maximum likelihood graph for the given data. If all the graphs are equally probable then $P(G)$ is a constant and does not affect the search, but it can be also beneficial to use it to guide the search towards meaningful graphs (Bishop 2006, Nagarajan and Scutari (2013)).

Decomposition of the likelihood. Bayesian network factorizes into local distributions according to *local Markov property* stating that a variable X_i is independent of its non-descendants given its parents at the graph. The *global Markov property* states that a variable is independent of all the remaining variables in the graph conditionally on its *Markov blanket* that consists its parents and child nodes at the graph, and additional parents of the child nodes (Bae et al. 2016, Koller and Friedman (2009)). With this decomposition, the joint probability of the graph can be calculated with sum and product rules of probability as a product of the independent local graphs G_i. The graph structure depends also on the parameters ϕ_i describing the dependencies, and they should also be taken into account at the structure estimation. We assume that ϕ_i is a set that pools all the parameters that describe the relationship.

Likelihood of the data is then

$$P(D|G) = \prod_{i=1}^{v} P(X_i|pa(X_i), \phi_i, G_i)P(\phi_i|G_i) \tag{3}$$

assuming that we have v independent local distributions at the graph. The notation $pa(X_i)$ denotes the parent variables of variable X_i according to graph structure G_i. Since the probability of data in the graph depends on the parameters of the local distributions, ϕ_i, they have to be integrated out from the equation to make the probability of graph independent of any specific choice of parameters

$$\prod_{i=1}^{v} \int P(X_i|pa(X_i), \phi_i, G_i)P(\phi_i|G_i)d\phi_i = \prod_{i=1}^{v} P(X_i|pa(X_i), G_i)P(G_i) \tag{4}$$

Besides the Markov properties, we also assume *global independence of the parameters*

$$\phi_i \neq \phi_j, i \neq j \tag{5}$$

and for the Bayesian estimation we assume *hyper-Markov law* (Dawid and Lauritzen 1993) for ensure that these decompositions are indeed independent.

Linear dependency between variables. As we are more interested in the system as a whole and less considered about the details of any specific nutritional response, we consider it adequate to model the dependency between the nutrients and the bodily responses with an approximate linear model. However, a simple linear model is not enough, as we need a parameterization that is able to reflect the correlations

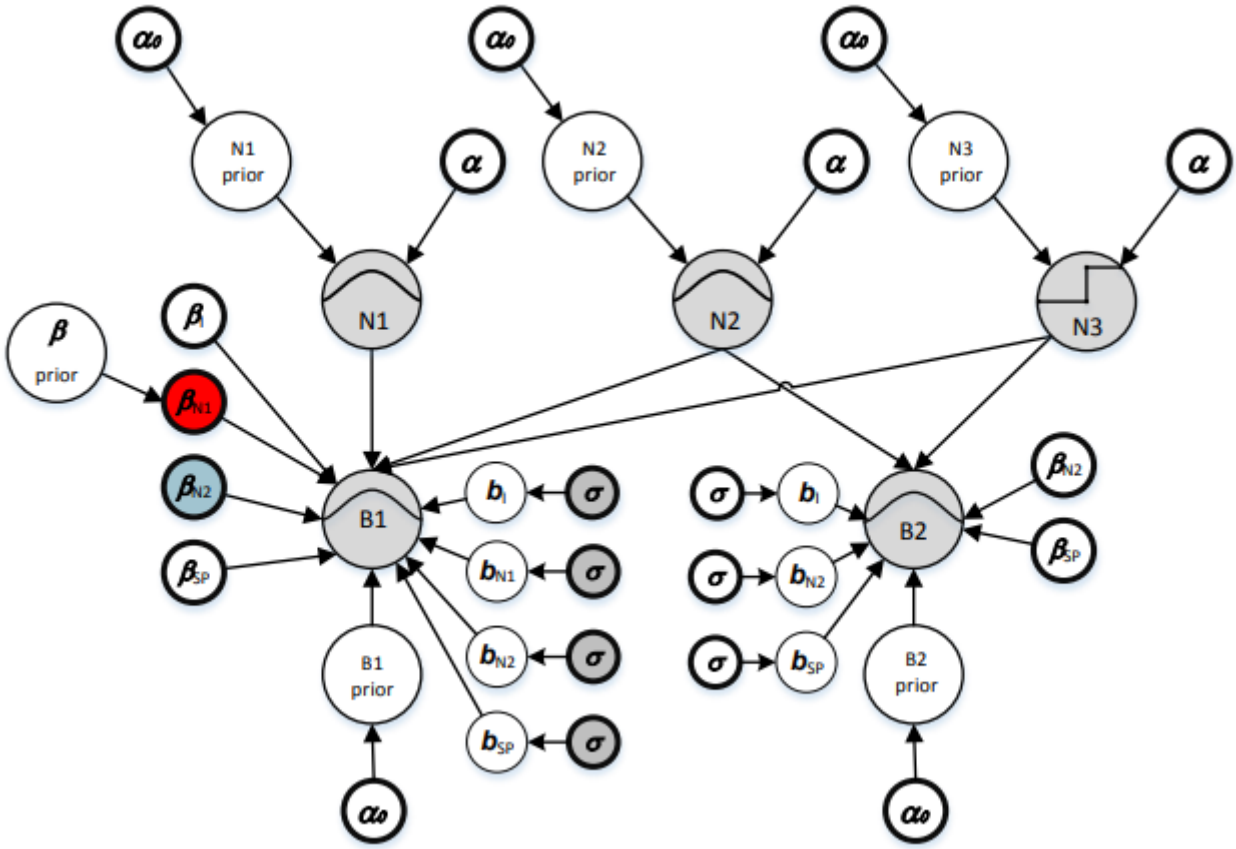


Figure 1: This is an example MEBN as a graphical model. The grey nodes are the observed variables of nutrients (N) and blood tests (B). The white beta and b nodes are the latent nodes to be estimated. Red, blue and dark grey correspond to the colors used in the visualization at figure 2.

between observations and to express the amount of variability between persons since the data consists of several repeated measurements from different persons.

Generally, the local probability distributions can be from exponential family of distributions, but in this case we consider only normally distributed response variables. The subset of parent variables, that we assume containing personal variance, is denoted with $pa_Z(X_i)$. For the mixed-effect modeling we need to estimate parameters $\phi_i = \{\beta_i, b_i\}$ for expressing the typical and personal reaction types. In a multivariate normal model the uncertainty is furthermore defined by variance-covariance matrix V_i

$$P(X_i|pa(X_i), \phi_i, G_i) = N(X_i|pa(X_i)\beta_i + pa_Z(X_i)b_i, V_i) \tag{6}$$

This theory motivates our search for the optimal graph with Stan. By decomposing the joint likelihood into local probability distributions according to Markov properties, it is possible to find the optimal graph by estimating one local distributions one by one.

Estimating the Hierarchical Local Distributions with Stan

In the mixed-effect modeling, the goal is to explain some of the model's variance in V_i with the latent personal effect variables b_i. These in turn offer us a way to detect and express the personal variations at the nutritional effects. Let us assume that matrix Z is a design matrix of the personal effects. Then variance-covariance matrix is defined by

$$V = ZDZ' + R \tag{7}$$

where R is a variance-covariance matrix of residuals and D is a variance-covariance matrix of personal, or random-effects,

$$D = \mathcal{T}C\mathcal{T}' \tag{8}$$

where $\mathcal{T}$ is a diagonal matrix of personal effect variances and C is correlation matrix that can be divided into Cholesky decompositions as

$$C = LL' \tag{9}$$

and with L we can define the personal effects as

$$b = Lu, u \sim N(0, I) \tag{10}$$

as we assume for now that the personal random-effects are drawn from Normal distribution.

This is implemented in Stan as follows

```
transformed parameters {
  // ...
  // Create diagonal matrix from sigma_b and premultiply it with L
  D = diag_pre_multiply(sigma_b, L);
```

```
  // Group-level effects are generated by multipying D with z that has standard normal distribution
  for(j in 1:J)
    b[j] = D * z[j];
}
```

The actual model with Normal distribution having the linear mixed-effect likelihood is defined below. Notice that instead of matrix V, in Stan we are using vectors `group` and scalar `sigma_e`.

```
model {
  // ...
  // Standard normal prior for random effects
  for (j in 1:J)
    z[j] ~ normal(0,1);

  // Likelihood
  // - link function (identity function for Normal dist.) for typical correlation
  mu = temp_Intercept + Xc * beta;

  // - add personal (group) effects
  for (i in 1:N)
  {
     mu[i] = mu[i] + Z[i] * b[group[i]];
  }

  // Y and mu are vectors, sigma_e is a scalar that is estimated for whole vector
  Y ~ normal(mu, sigma_e);
}
```

Constructing the Population Level Graph of Nutrional Effects

The dataset in this example comes from Sysdimet study (Lankinen et al. 2011) that studied altogether 106 men and women with impaired glucose metabolism. The original study is a randomized control trial, but each of the control groups are assumed to react to the nutrition basically at the same way. The only prior knowledge about the difference in reactions is related to the cholesterol medication. After taking this variable into account it is possible to use all the persons from the study in this modeling.

For each person we have four observations from their nutritional diary and from blood tests. The blood tests are taken a week after the diet observation. We have picked few interesting variables indicating person's diet, blood test results and personal information, like gender and medication. Altogether we have 22 variables of personal and dietary information, and 5 variables from blood tests.

There exist plenty of general algorithms for constructing BNs, but for this special case we can constrain the search to biologically plausible reaction graphs. We assume that all possible graphs are directed bipartite graphs with nutrients and personal information as root nodes and blood tests as targets.

```
# Read the data description
datadesc <- read.csv(file="Data description.csv", header = TRUE, sep = ";")

# Read the actual data matching the description
sysdimet <- read.csv(file="data\\SYSDIMET_diet.csv", sep=";", dec=",")

# Define how to iterate through the graph
assumedpredictors <- datadesc[datadesc$Order==100,]
assumedtargets <- datadesc[datadesc$Order==200,]
```

Pruning the edges. We start the graph construction from a fully connected graph, but for gaining the

nutritional knowledge of significant connections, it is necessary to prune out the insignificant connections from the graph. This also factorizes the joint likelihood of BN as formulated earlier.

For pruning we use shrinkage prior on beta coefficients to push the insignificant coefficients towards zero. Especially, we use regularized horseshoe prior (Piironen and Vehtari 2017a) that allows specifying the number of non-zero coefficients for each target. In the nutritional setting this provides a way for specifying prior knowledge about the relevant nutrients for each response. For now, we approximate that one third of the predictive nutrients are relevant, but finer approximation will be done based on the previous nutritional research. See (Piironen and Vehtari 2017a) for detailed information on the shrinkage parameters.

```
shrinkage_parameters <- within(list(),
{
    scale_icept  <- 1         # prior std for the intercept
    scale_global <- 0.01821   # scale for the half-t prior for tau:
                              # (p0=6) / (D=22-6)*sqrt(n=106*4)
    nu_global    <- 1         # degrees of freedom for the half-t priors for tau
    nu_local     <- 1         # degrees of freedom for the half-t priors for lambdas
    slab_scale   <- 1         # slab scale for the regularized horseshoe
    slab_df      <- 1         # slab degrees of freedom for the regularized horseshoe
})
```

If the shrinkage prior does not shrink the coefficients to exactly zero, we are pruning out the insignificant connections with following test. Notice that in the population level graph we are keeping the connections that have large variance between persons even though they are not typically relevant. Personal random-effect variance means that the connection is relevant for someone.

The effect of shrinkage can be studied by using an alternative with Stan model "BLMM.stan" that omits the shrinkage.

```
my.RanefTest <- function(localsummary, PredictorId)
{
  abs(localsummary$fixef[PredictorId]) > 0.001 || localsummary$ranef_sd[PredictorId] > 0.05
}
```

To assure that this pruning does not affect predictive accuracy of the model, a projection method (Piironen and Vehtari 2017b) could be also used here. In the projection approach the edges are removed if the removal doesn't affect the distance from the true model measured with the KL-divergence.

Construction of the graph. The data structure of the graph is based on iGraph package. The process of BN construction starts by adding a node for every observed variable at the dataset.

```
# Add data columns describing random variables as nodes to the graph
# - initial_graph is iGraph object with only nodes and no edges
initial_graph <- mebn.new_graph_with_randomvariables(datadesc)
```

For estimating the typical MEBN graph, we iterate the hierarchical Stan-model through all the assumed predictors and targets. This builds up the joint distribution of MEBN, one local distribution at a time. These local distributions correspond to the hierarchical regression models that are estimated with Stan and the resulting HMC samplings are cached to files.

The result is an iGraph object that contains a directed bipartite graph with nutrients as predictors and blood test results as targets. We normalize all the values to unit scale and center before the estimation. Instead of sampling, it is also possible to estimate the same model with Stan's implementation of variational Bayes by switching the local_estimation-paramter.

```
sysdimet_graph <- mebn.typical_graph(reaction_graph = initial_graph,
                                     inputdata = sysdimet,
                                     predictor_columns = assumedpredictors,
                                     assumed_targets = assumedtargets,
```

```
                                    group_column = "SUBJECT_ID",
                                    local_estimation = mebn.sampling,
                                    local_model_cache = "models",
                                    stan_model_file = "mebn\\BLMM_rhs.stan",
                                    edge_significance_test = my.RanefTest,
                                    normalize_values = TRUE,
                                    reg_params = shrinkage_parameters)
```

```
## [1] "Loading models\\fshdl_blmm.rds"
## [1] "Loading models\\fsldl_blmm.rds"
## [1] "Loading models\\fsins_blmm.rds"
## [1] "Loading models\\fskol_blmm.rds"
## [1] "Loading models\\fpgluk_blmm.rds"
```

Sampling may still cause some divergent transitions and errors on estimating correlation matrix that is quite restricted data type. Estimations of then parameters seem nevertheless realistic.

Now "sysdimet_graph" is a Mixed Effect Bayesian Network for whole sample population. We can store it in GEXF-format and load to visualization for general inspection. The visual nature of the Bayesian Networks, and the graphical models in general, provide a useful framework for creating visualizations. In comparison to the schematic figure (fig. 1), the coefficients and latent variables are removed and denoted by different colors. Blue edge denotes typically negative correlation, red edge denotes positive correlation and gray shade denotes the amount of personal variation at the correlation.

The shrinkage leaves still quite a few subtle connections at the graph and to remove the clutter we will only visualize the few most relevant of them, having either typically large effect or having large personal variance. In that case it might be relevant to some individuals, but not for all.

A few observations raise from this visualization: Women tend to have typically higher HDL-cholesterol levels than men, but lower blood insulin levels. On the other hand saturated fats (safa) lower HDL-cholesterol, but raise the blood insulin. One interesting correlation can be also found between protein and blood insulin. It has typically very little positive correlation, but there is a quite large variance between persons. This has also new clinical support.

It can be also seen from the visualization that many of the nutrients affect to several responses. This multiresponse effect is not yet implemented to the model, but should be taken into account. It allows, for example, to gain knowledge about responses that have missing measurements for some persons, but have known connections to other responses and predictors. Besides the visual inspection, we can also do a numerical inference on the graph.

Inference

The generated graph object allows us to query some interesting insights. We can, for example, investigate the most significant typical reactions by quering largest beta coefficients

```
# Let's query the graph for beta coefficients that denote typical effects, (see beta nodes in fig 1)
allnodes <- V(sysdimet_graph)
beta <- allnodes[allnodes$type=="beta"]

# Separate data frame is constructed for printing
typical_effects<-data.frame(matrix(NA, nrow=length(beta), ncol=0))

# - let's translate the names of beta-nodes with the data description metadata
typical_effects$effect <- unlist(lapply(strsplit(gsub("beta_","", beta$name), "_"), function(x) paste0(

# - sort most positive and negative values
```

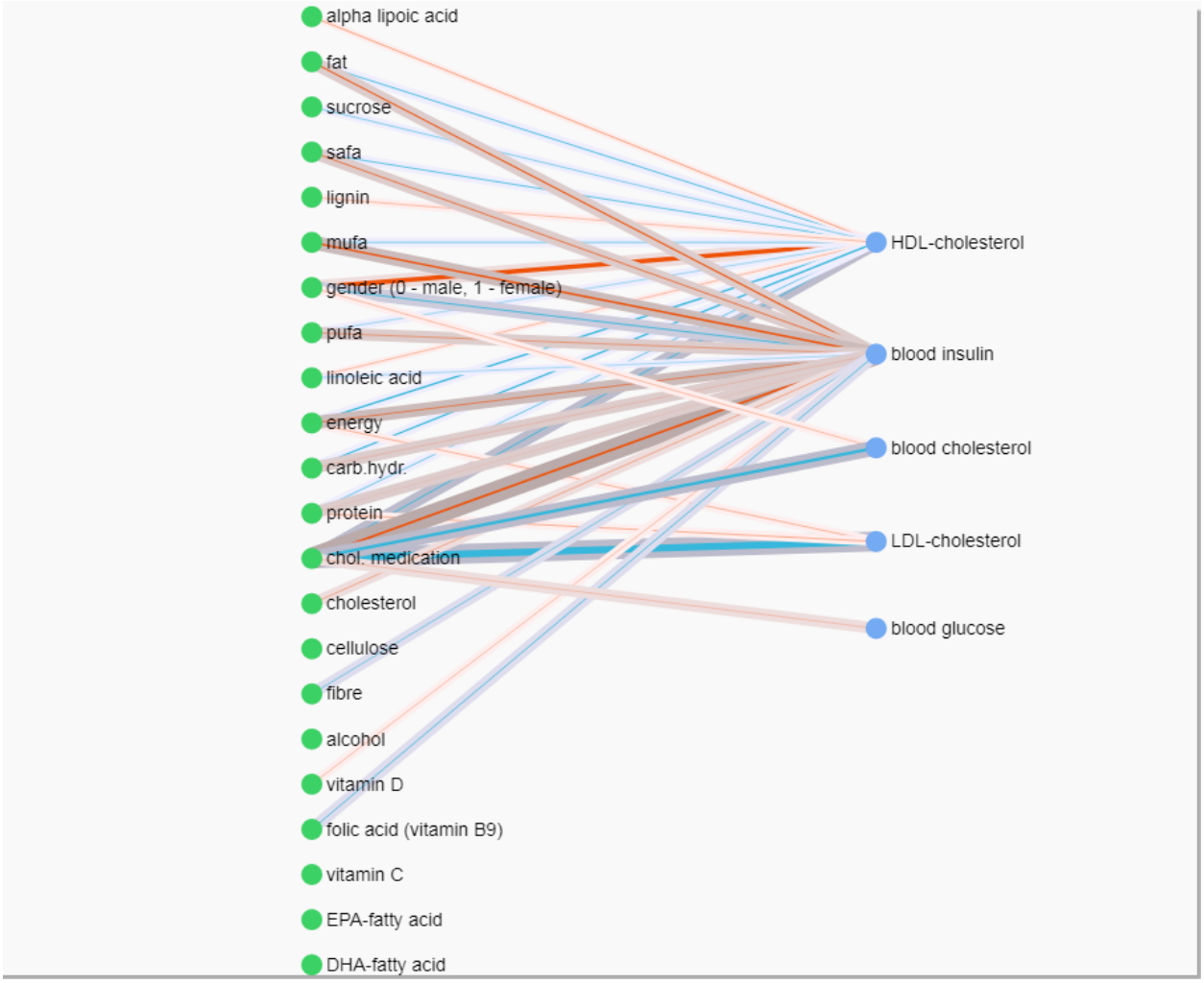


Figure 2: Visualization of the typical reaction graph with edge thickness and color indicating the level of effect (beta coefficient). Grey shade at the edge indicates the amount of variance between personal effects (sigma_b). Notice that less significant connections have been filtered out for clarity.

```
typical_effects$value <- round(beta$value,3)
largest_typical_negative <- typical_effects[order(typical_effects$value),]
largest_typical_positive <- typical_effects[order(-typical_effects$value),]
```

Largest typical negative effects

```
##                                    effect  value
## 24   chol. medication -> LDL-cholesterol -0.027
## 68 chol. medication -> blood cholesterol -0.012
## 3              energy -> HDL-cholesterol -0.006
## 12         carb.hydr. -> HDL-cholesterol -0.006
## 2    chol. medication -> HDL-cholesterol -0.004
## 5                 fat -> HDL-cholesterol -0.004
## 6                safa -> HDL-cholesterol -0.004
```

Largest typical positive effects

```
##                                            effect value
## 1  gender (0 - male, 1 - female) -> HDL-cholesterol 0.017
## 46                chol. medication -> blood insulin 0.008
## 51                           mufa -> blood insulin 0.007
## 49                            fat -> blood insulin 0.005
## 47                         energy -> blood insulin 0.003
## 50                           safa -> blood insulin 0.003
## 9                 linoleic acid -> HDL-cholesterol 0.002
```

As the visualization showed, females have typically higher levels of HDL cholesterol. Also the cholesterol medication, besides of lowering the cholesterol levels, also raises the blood insulin level.

What we are really interested in, though, are the variations between persons. For this, we can query the largest variances of random-effects..

```
# Query the graph for personal variances, denoted by b_sigma-nodes
b_sigma <- allnodes[allnodes$type=="b_sigma"]

# Again, a separate data frame is constructed for printing
personal_variances<-data.frame(matrix(NA, nrow=length(b_sigma), ncol=0))

personal_variances$effect <- unlist(lapply(strsplit(gsub("b_sigma_","", b_sigma$name), "_"), function(x)

personal_variances$variance <- round(b_sigma$value,3)
largest_personal_variance <- personal_variances[order(-personal_variances$variance),]
```

Largest personal variance

The variance value here is the variance of the random-effect denoted with *sigma_b* in the Stan code. It can be interpreted as the amount of variability in reactions between persons.

```
##                                effect variance
## 47           energy -> blood insulin    0.160
## 3          energy -> HDL-cholesterol    0.107
## 25         energy -> LDL-cholesterol    0.107
## 48          protein -> blood insulin    0.103
## 50             safa -> blood insulin    0.100
## 52             pufa -> blood insulin    0.100
## 69       energy -> blood cholesterol    0.100
## 46 chol. medication -> blood insulin    0.099
## 49              fat -> blood insulin    0.097
```

```
## 51              mufa -> blood insulin    0.097
```

So, there seems to be a large variance in how the energy intake affects to insulin and cholesterol levels. Also the effect of protein to insulin levels is interesting. As the visualization hinted, the protein typically increases the insulin level a bit and there is a quite large personal variance in the effect.

Confidence of the personal variances

Relevance of these estimations can be inspected from their posterior distributions. Let us consider the estimated variance (sigma_b) in the personal effect from the protein level in diet to the blood insulin level (fsins).

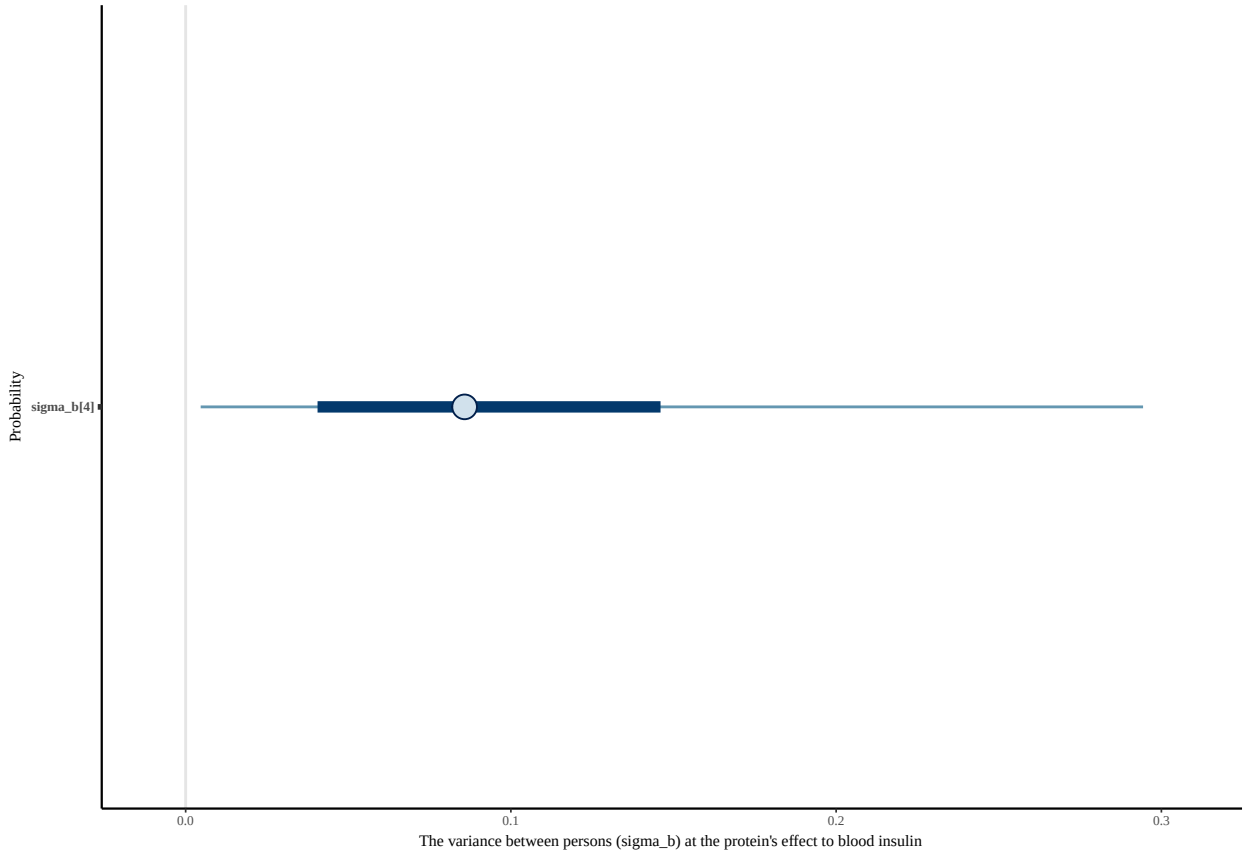
```
library("bayesplot")

# Local distribution for fsins
fsins_blmm <- mebn.get_localfit("fsins")
```

```
## [1] "Loading models\\fsins_blmm.rds"
```

```
# Index of predictor 'protein' -- this is sigma_b[4] at the posterior plot
id <- match("prot", datadesc$Name)

posterior <- as.array(fsins_blmm)
mcmc_intervals(posterior, pars = c(paste0("sigma_b[",id,"]")), prob_outer = 0.95) +
  xlab("The variance between persons (sigma_b) at the protein's effect to blood insulin") +
  ylab("Probability")
```



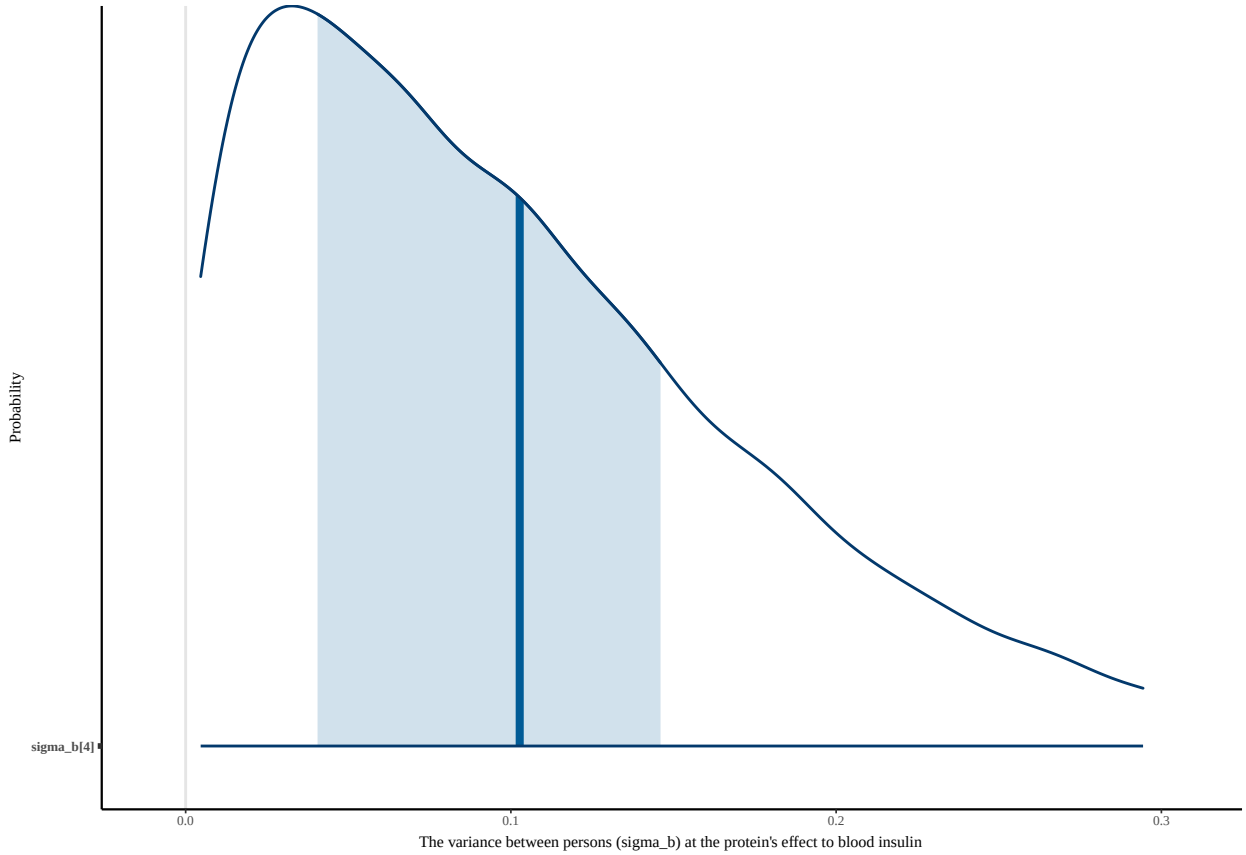


Figure 3: Posterior distribution of variance between persons at protein's effect to blood insulin. This hyperparameter corresponds to the thick grey shade at figure 2 between protein and blood insulin nodes. In 95% probability people have real differences in this reaction and this is one of the connections that may be used for personal nutrition. Variance between persons means a possibility for personalization.

This shows that the there exists personal variance in how of protein level at diet affects to blood insulin levels as the 95% credible interval is above zero. The wide probability distribution shows that exact estimate is quite uncertain and we would need more observations or stricter prior information for more precise estimation. One can also observe the difference between typical choices of Bayesian point estimation; upper interval chart points at the posterior median, while the lower area chart shows the posterior mean and the MAP estimate at the high point of the probability.

Conclusions

Mixed-effect Bayesian networks offer an appealing way to model the system of nutritional effects. By using the mixed-effect models as local probability distributions in the graph, we can estimate the effects in both population and personal level. Furthermore, the fully Bayesian estimation of the distributions allow direct means for adding the prior information to the model and also addressing the uncertainty of the estimates.

The model could be enhanced by adding a correlation structure between observations. For longer time series, for example, a moving average or ARMA structure could be beneficial. This might level out some of the noise that human observations contain. For studying the indirect connections of the nutrients and bodily responses, a multiresponse modeling needs to be added. This might be effectively estimated only after the factorization of the graph is done and significant parents of the nodes are found.

In my future work I will make a procedure that combines the personal predictions as a personal reaction graph, similar to the population level graph that we constructed in this notebook. With the previously estimated random-effect variances and covariances, this kind of personal reaction graph can be constructed with linear predictions from very limited data. These personal graphs allow inference on any of the variables resulting interesting new applications, for example, in personal diet recommendations and in personal health care.

References

Aussem, Alex, André Tchernof, Sérgio Rodrigues de Morais, and Sophie Rome. 2010. "Analysis of Lifestyle and Metabolic Predictors of Visceral Obesity with Bayesian Networks." *BMC Bioinformatics* 11 (1): 487. doi:10.1186/1471-2105-11-487.

Bae, Harold, Stefano Monti, Monty Montano, Martin H. Steinberg, Thomas T. Perls, and Paola Sebastiani. 2016. "Learning Bayesian Networks from Correlated Data" 6 (May). The Author(s) SN -: 25156 EP. http://dx.doi.org/10.1038/srep25156.

Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Dawid, A. P., and S. L. Lauritzen. 1993. "Hyper Markov Laws in the Statistical Analysis of Decomposable Graphical Models." *Ann. Statist.* 21 (3). The Institute of Mathematical Statistics: 1272–1317. doi:10.1214/aos/1176349260.

Koller, Daphne, and Nir Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press.

Lankinen, M., U. Schwab, M. Kolehmainen, J. Paananen, K. Poutanen, H. Mykkanen, T. Seppanen-Laakso, H. Gylling, M. Uusitupa, and M. Ore?i? 2011. "Whole grain products, fish and bilberries alter glucose and lipid metabolism in a randomized, controlled trial: the Sysdimet study." *PLoS ONE* 6 (8): e22646.

Nagarajan, Radhakrishnan, and Marco Scutari. 2013. *Bayesian Networks in R with Applications in Systems Biology*. New York: Springer. doi:10.1007/978-1-4614-6446-4.

Piironen, Juho, and Aki Vehtari. 2017a. "Sparsity Information and Regularization in the Horseshoe and Other Shrinkage Priors." *Electron. J. Statist.* 11 (2). The Institute of Mathematical Statistics; the Bernoulli Society: 5018–51. doi:10.1214/17-EJS1337SI.

———. 2017b. "Comparison of Bayesian Predictive Methods for Model Selection." *Statistics and Computing* 27 (3): 711–35. doi:10.1007/s11222-016-9649-y.

Scutari, Marco, and Jean-Baptiste Denis. 2014. *Bayesian Networks with Examples in R*. Boca Raton: Chapman; Hall.

Zeevi, David, Tal Korem, Niv Zmora, David Israeli, Daphna Rothschild, Adina Weinberger, Orly Ben-Yacov, et al. 2015. "Personalized Nutrition by Prediction of Glycemic Responses." *Cell* 163 (5). Elsevier: 1079–94. doi:10.1016/j.cell.2015.11.001.