

Bayesian Sparse Factor Models and DAGs Inference and Comparison

Authored by:

Ole Winther
Ricardo Henao

Abstract

In this paper we present a novel approach to learn directed acyclic graphs (DAG) and factor models within the same framework while also allowing for model comparison between them. For this purpose, we exploit the connection between factor models and DAGs to propose Bayesian hierarchies based on spike and slab priors to promote sparsity, heavy-tailed priors to ensure identifiability and predictive densities to perform the model comparison. We require identifiability to be able to produce variable orderings leading to valid DAGs and sparsity to learn the structures. The effectiveness of our approach is demonstrated through extensive experiments on artificial and biological data showing that our approach outperform a number of state of the art methods.

1 Paper Body

Sparse factor models have proven to be a very versatile tool for detailed modeling and interpretation of multivariate data, for example in the context of gene expression data analysis [1, 2]. A sparse factor model encodes the prior knowledge that the latent factors only affect a limited number of the observed variables. An alternative way of modeling the data is through linear regression between the measured quantities. This multiple regression model is a well-defined multivariate probabilistic model if the connectivity (non-zero weights) defines a directed acyclic graph (DAG). What usually is done in practice is to consider either factor or DAG models. Modeling the data with both types of models at the same time and then perform model comparison should provide additional insight as these models are complementary and often closely related. Unfortunately, existing off-the-shelf models are specified in such a way that makes direct comparison difficult. A more principled idea that can be phrased in Bayesian terms is for example to find an equivalence between both models, then represent them using a common/comparable hierarchy, and finally use a marginal likelihood or a predictive density to select one of them. Although a formal connection between

factor models and DAGs has been already established in [3], this paper makes important extensions such as explicitly modeling sparsity, stochastic search over the order of the variables and model comparison. It is well known that learning the structure of graphical models, in particular DAGs is a very difficult task because it turns out to be a combinatorial optimization problem known to be NP-hard [4]. A commonly used approach for structure learning is to split the problem into two stages using the fact that the space of variable orderings is far more smaller than the space of all possible structures, e.g. by first attempting to learn a suitable permutation of the variables and then the skeleton of the structure given the already found ordering or viceversa. Most of the work so far for continuous data assumes linearity and Gaussian variables hence they can only recover the DAG structure up to

Markov equivalence [5, 6, 7, 8], which means that some subset of links can be reversed without changing the likelihood [9]. To break the Markov equivalence usually experimental (interventional) data in addition to the observational (non-interventional) data is required [10]. In order to obtain identifiability from purely observational data, strong assumptions have to be made [11, 3, 12]. In this work we follow the line of [3] by starting from a linear factor model and ensure identifiability by using non-normal heavy-tailed latent variables. As a byproduct we find a set of candidate orderings compatible with a linear DAG, i.e. a mixing matrix which is "close to" triangular. Finally, we may perform model comparison between the factor and DAG models inferred with fixed orderings taken from the candidate set. The rest of the paper is organized as follows. Sections 2 to 5 we motivate and describe the different ingredients in our method, in Section 6 we discuss existing work, in Section 7 experiments on both artificial and real data are presented, and Section 8 concludes with a discussion and perspectives for future work.

2

From DAGs to factor models

We will assume that an ordered d -dimensional data vector \mathbf{x} can be represented as a directed acyclic graph with only observed nodes, where \mathbf{P} is the usually unknown true permutation matrix. We will focus entirely on linear models such that the value of each variable is a linear weight combination of parent nodes plus a driving signal $\mathbf{x} = \mathbf{P}^{-1} \mathbf{B} \mathbf{x} + \mathbf{z}$, (1) where \mathbf{B} is a strictly lower triangular square matrix. In this setting, each non-zero element of \mathbf{B} corresponds to a link in the DAG. Solving for \mathbf{x} we can rewrite the problem as $\mathbf{x} = \mathbf{P}^{-1} \mathbf{A} \mathbf{z} = \mathbf{P}^{-1} (\mathbf{I} - \mathbf{B})^{-1} \mathbf{P} \mathbf{z}$, (2) which corresponds to a noise-free linear factor model with the restriction that $\mathbf{P}^{-1} \mathbf{A} \mathbf{P}$ must have a sparsity pattern that can be permuted to a triangular form since $(\mathbf{I} - \mathbf{B})^{-1}$ is triangular. This requirement alone is not enough to ensure identifiability (up to scaling and permutation of columns \mathbf{P}^{-1}). We further have to use prior knowledge about the distribution of the factors \mathbf{z} . A necessary condition is that these must be a set of non-Gaussian independent variables [11]. For heavy-tailed data it is often sufficient in practice to use a model with heavier tails than Gaussian [13]. If the requirements for \mathbf{A} and for the distribution of \mathbf{z} are met, we can first estimate $\mathbf{P}^{-1} \mathbf{A} \mathbf{P}$ and subsequently find \mathbf{P} searching over the space of all

possible orderings. Recently, [3] applied the fastICA algorithm to solve for the inverse mixing matrix $P^{-1}A^{-1}P$. To find a candidate solution for B , P is set such that B found from the direct relation equation (1), $B = I - A^{-1}$ (according to magnitude-based criterion) is as close as possible to lower triangular. In the final step the Wald statistic is used for pruning B and the chi-square test is used for model selection. In our work we also exploit the relation between the factor models and linear DAGs. We apply a Bayesian approach to learning a sparse factor models and DAGs, and the stochastic search for P is performed as an integrated part of inference of the sparse factor model. The inference of factor model (including order) and DAG parameters are performed as two separate inferences such that the only input that comes from the first part is a set of candidate orders.

3

From factor models to DAGs

Our first goal is to perform model inference in the families of factor and linear DAG models. We specify the joint distribution or probability of everything, e.g. for the factor model, as $p(X, A, Z, P, ?) = p(X|A, Z, P, ?)p(A|Z, P, ?)p(Z|P, ?)p(P|?)$, where $X = [x_1, \dots, x_N]$, $Z = [z_1, \dots, z_N]$, N is the number of observations and $(?)$ indicates additional parameters in the hierarchical models. The prior over permutation $p(P|?)$ will always be chosen to be uniform over the $d!$ possible values. The actual sampling based inference for P is discussed in the next section and the standard Gibbs sampling components are provided in the supplementary material. Model comparison should ideally be performed using the marginal likelihood. This is more difficult to calculate with sampling than obtaining samples from the posterior so we use the predictive densities on a test set as a yardstick. 1

These ambiguities are not affecting our ability to find correct permutation P of the rows.

2

Factor model Instead of using the noise-free factor model of equation (2) we allow for additive noise $x = P^{-1}rAPc z + ?$, where $?$ is an additional Gaussian noise term with diagonal covariance matrix $?$, i.e. uncorrelated noise, to account for independent measurement noise, $Pr = P$ is the permutation matrix for the rows of A and $Pc = Pf Pr$ another permutation for the columns with Pf accounting for the permutation freedom of the factors. We will not restrict the mixing matrix A to be triangular. Instead we infer Pr and Pc using a stochastic search based upon closeness to triangular as measured by a masked likelihood, see below. Now we can specify a hierarchy for the Bayesian model as follows $X|Pr, A, Pc, Z, ? \sim N(X|P^{-1}rAPc Z, ?)$, $?|ss, sr$

?

$\text{Gamma}(?^{-1}ss, sr)$

$Z|?(Z-?)$,

(3)

$A|?(A-?)$,

,

where θ_i are elements of θ . For convenience, to exploit conjugate exponential families we are placing a gamma prior on the precision of θ with shape ss and rate sr . Given that the data is standardized, the selection of hyperparameters for θ_i is not very critical as long as both 'signal and noise' are supported. The prior should favor small values of θ_i as well as providing support for $\theta_i = 1$ such that certain variables can be explained solely by noise (we set $ss = 2$ and $sr = 0.05$ in the experiments). For the factors we use a heavy-tailed prior $\theta(Z = ?)$ in the form of a Laplace distribution parameterized for convenience as a scale mixture of Gaussians [14] $Z \sim N(z_{jn} | \theta, ?)$ $\text{Exponential}(\theta_{jn} | \theta^2) d\theta_{jn}$, (4) $z_{jn} | \theta, ? \sim \text{Laplace}(z_{jn} | \theta, ?) = 0$

$$\theta^2 \sim \theta^s, \theta^r \sim \text{Gamma}(\theta^2 | \theta^s, \theta^r), \quad (5)$$

where z_{jn} is an element of Z , θ is the rate and θ has an exponential distribution acting as mixing density. Furthermore, we place a gamma distribution on θ^2 to get conditionals for θ and θ^2 in standard conjugate families. We let the components of Z have on average unit variance. This is achieved by setting $\theta^s / \theta^r = 2$ (we set $\theta^s = 4$ and $\theta^r = 2$). Alternatively one may use a t distribution again as scale mixture of Gaussians which can to interpolate between very heavy-tailed (power law) and very light tails, i.e. becoming Gaussian when degrees of freedom approaches infinity. However such flexibility comes at the price of being more difficult to select its hyperparameters, because the model could become unidentified for some settings.

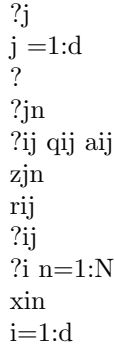


Figure 1: Graphical model for Bayesian hierarchy in equation (3).

The prior $\theta(A = ?)$ for the mixing matrix should be biased towards sparsity because we want to infer something close to a triangular matrix. Here we adopt a two-layer discrete spike and slab prior for the elements a_{ij} of A similar to the one in [2]. The first layer in the prior control the sparsity of each element a_{ij} individually, whereas the second layer impose a per-factor sparsity level to allow elements within the same factor to share information. The hierarchy can be written as $a_{ij} | r_{ij}, \theta_i, \theta_{ij} \sim (1 - r_{ij}) \delta(a_{ij}) + r_{ij} N(a_{ij} | 0, \theta_i \theta_{ij})$, $\theta_i \sim \theta^s, \theta^r \sim \text{Gamma}(\theta_i | \theta^s, \theta^r)$,

$$r_{ij} | \theta_{ij}, \theta_{ij} | q_{ij}, \theta_p, \theta_m q_{ij} | \theta_j, \theta_j | \theta_m, \theta_p \sim \text{Bernoulli}(r_{ij} | \theta_{ij}), \theta_j | (1 - q_{ij}) \delta(\theta_j) + q_{ij} \text{Beta}(\theta_j | \theta_p \theta_m, \theta_p (1 - \theta_m)), \theta_j \sim \text{Bernoulli}(q_{ij} | \theta_j), \theta_j \sim \text{Beta}(\theta_j | \theta_p \theta_m, \theta_p (1 - \theta_m)), \quad (6)$$

where $\delta(\cdot)$ is a Dirac δ -function. The prior above specifies a point mass mixture over a_{ij} with mask r_{ij} . The expected probability of a_{ij} to be non-zero is π_{ij} and is controlled through a beta hyperprior with mean π_m and precision π_p . Besides, each factor has a common sparsity rate π_j that lets the elements π_{ij} to be exactly zero with probability $1 - \pi_j$ through a beta distribution with mean π_m and 3

precision π_p , turning the distribution of π_{ij} bimodal over the unit interval. The magnitude of nonzero elements in A is specified through the slab distribution depending on π_{ij} . The parameters for π_{ij} should be specified in the same fashion as π_i but putting more probability mass around $a_{ij} = 1$, for instance $t_s = 4$ and $t_r = 10$. Note that we scale the variances with π_i since it makes the model easier to specify and tend to have better mixing properties [15]. The masking matrix r_{ij} with parameters π_{ij} should be somewhat diffuse while favoring relatively large masking probabilities, e.g. $\pi_p = 10$ and $\pi_m = 0.9$. Additionally, q_j and should favor very small values with low variance, this is for example $\pi_p = 1000$ and $\pi_m = 0.005$. The graphical model for the entire hierarchy in (3) omitting parameters is shown in Figure 1. DAG

We make the following Bayesian specification of linear DAG model of equation (1) as $X \sim \text{Pr}(B, X, \pi) \mid (X \sim P \mid r \mid B \sim \pi)$,

$$B \sim \pi(B \sim \pi),$$

(7)

where π and π are given by equations (4) and (6). The Bayesian specification for the DAG has a similar graphical model to the one in Figure 1 but without noise variances π . The factor model needs only shared variance parameter π for the Laplace distributed z_{jn} because a change of scale in A is equivalent to change of variance in z_{jn} . The DAG on the other hand, needs individual variance parameters because it has no scaling freedom. Given that we know that B is strictly lower triangular, it should be in general less sparse than A , thus we use a different setting for the sparsity prior, i.e. $\pi_p = 100$ and $\pi_m = 0.01$.

4

Sampling based inference

For given permutation P , Gibbs sampling can be used for inference of the remaining parameters. Details of Gibbs sampler is given in the supplementary material and we will focus on the non-standard inference corresponding to the sampling over permutations. There are basically two approaches to find P , one is perform the inference for parameters and P jointly with B restricted to be triangular. The other is to let the factor model be unrestricted and search for P according to a criterion that does not affect parameter inference. Here we prefer the latter for two reasons. First, joint combinatorial and parameter inference in this model will probably have poor mixing with slow convergence. Second, we are also interested in comparing the factor model against the DAG for cases when we cannot really assume that the data is well approximated by a DAG. In our approach the proposal $P?$ corresponds to picking two of the elements in the order vector by random and exchanging them. Other approaches such as restricting to pick two adjacent elements have been suggested as well [16,

7]. For the linear DAG model we are not performing joint inference of P and the model parameters. Rather we use a set of P s found for the factor model to be good candidates for the DAG. The stochastic search for $P = P_c$ goes as follows: we make inference for the unrestricted factor model, propose $P^?r$ and $P^?c$ independently according $q(P^?r | Pr)q(P^?c | Pc)$ which is the uniform two variable random exchange. With this proposal and the flat prior over P , we use a MetropolisHastings acceptance probability simply as the ratio of likelihoods with A masked to have zeros above its diagonal (through masking matrix M) $?? =$

$$\frac{N(X | (P^?r)^?1 (M^? P^?r A(P^?c)^?1) P^?c, ?)}{N(X | (P^?r)^?1 (M^? Pr A(P_c)^?1) P_c, ?)}$$

The procedure can be seen as a simple approach for generating hypotheses about good, close to triangular A , orderings in a model where the spike and slab prior provides bias towards sparsity. To learn DAGs we first perform inference on the factor model specified by the hierarchy in (3) to obtain a set of ordering candidates sorted according to their usage during sampling?after the burnin period. It is possible that the estimation of A might contain errors, e.g. a false zero entry on A allowing several orderings leading to several lower triangular versions of A , only one of those being actually correct. Thus, we propose not only to use the best candidate but a set of top candidates of size $m_{top} = 10$. Then we perform inference on the DAG model corresponding to the structure search (m) (1) hierarchy in (7), for each one of the permutation candidates being considered, Pr, \dots, Pr_{top} . Finally, we select the DAG model among candidates using the predictive distribution for the DAG when a test set is available or just the likelihood if not. 4

5

Predictive distributions and model comparison

Given that our model produces both DAG and a factor model estimates at the same time, it could be interesting to estimate also whether one option is better than the other given the observed data, for example in exploratory analysis when the DAG assumption is just one reasonable option. In order to perform the model comparison, we use predictive densities $p(X^? | X, M)$ with $M = \{MFA, MDAG\}$, instead of marginal likelihoods because the latter is difficult and expensive to compute by sampling, requiring for example thermodynamic integration. With Gibbs sampling, we draw samples from the posterior distributions $p(A, ?, ? | X, ?)$ and $p(B, ?_1, \dots, ?_m | X, ?)$. The average over the extensive variables associated with the test points $p(Z^? | ?)$ is a bit more complicated because naively drawing samples from $p(Z^? | ?)$ gives an estimator with high variance?for $?_i, \dots, ?_n$. In the following we describe how to do it for each model, omitting the permutation matrices for clarity. Factor model We can compute the predictive distribution by taking the likelihood in equation (3) and marginalizing Z . Since the integral has no closed form we can approximate it using the Gaussian distribution from the scale mixture representation as $Z \sim \int p(X | A, ?, ?) p(X | A, Z, ?) p(Z | ?) dZ \approx \int p(X | A, Z, ?) p(Z | ?) dZ$ (x?n $\rightarrow 0$, $A^? \sim Un(A + ?)$), $rep = n$ where $Un = \text{diag}(?_1n, \dots, ?_dn)$, the $?_jn$ are sampled from the prior and rep is the number of samples generated to

approximate the intractable integral (rep = 500 in the experiments). Then we can average over $p(A, ?, ?|X, ?)$ to obtain $p(X?|X, MFA)$. DAG In this case the predictive distribution is rather easy because the marginal over Z in equation (4) is just a Laplace distribution with mean $BX Z Y$ $p(X?|B, ?) = p(X?|B, X, Z)p(Z|?)dZ = \text{Laplace}(x_{ij} - [BX]_{in}, ?i) , i, n$

where $[BX]_{ij}$ is the element indexed by the i -th row and n -th column of BX . In practice we compute the predictive densities for a particular $X?$ during sampling and then select the model based on its ratio. Note that both predictive distributions depend directly on $?$ the rate of Laplace distribution, making the estimates highly dependent on its value. This is why it is important to have the hyperprior on $?$ of equation (5) instead of just fixing its value.

6

Existing work

Among the existing approaches to DAG learning, our work is most closely related to LiNGAM (Linear Non-Gaussian Acyclic Model for causal discovery) [3] with several important differences: Since LiNGAM relies on fastICA to learn the mixing is not inherently sparse, hence a pruning procedure based on Wald statistic and model fit second order information should be applied after obtaining an ordering for the variables. The order search in LiNGAM assumes that there is not estimation errors during fastICA model inference, then a single ordering candidate is produced. LiNGAM produces and select a final model among several candidates, but in contrast to our method such candidates are not different DAGs with different variable orderings but DAGs with different sparsity levels. The factor model inference in LiNGAM, namely fastICA is very efficient however their structure search involves repeated inversions of matrices of sizes $d_2 \times d_2$ which can make it prohibitive for large problems. More explicitly, the computational complexity of LiNGAM is roughly $O(N_{fit} d_6)$ where N_{fit} is the number of model fit evaluations. In contrast, the complexity in our case is $O(N_{ite} d_2 N)$ where N_{ite} is the total number of samples including burn-in periods for both, factor model and DAG inferences. Finally, our model is more principled in the sense that all the approach is within the same Bayesian framework, as a result it can be extended to for example binary data or time series by selecting some suitable prior distributions. Much work on Bayesian models for DAG learning already exist. For example, the approach presented in [16] is a Gaussian Bayesian network and therefore suffers from lack of identifiability. Besides, order search is performed directly for the DAG model making necessary the use of longer

5
sampler runs with a number of computational tricks when the problem is large ($d \geq 10$), i.e. when exhaustive order enumeration is not an option.

7

Experiments

We consider four sets of experiments in the following. The first two consist on extensive experiments using artificial data, the third addresses the model comparison scenario and the last one uses real data previously published in [17]. In every case we ran 2000 samples after a burn-in period of 4000 iterations and three independent chains for the factor model, and a single chain with

1000 samples and 2000 as burn-in for the DAG2 . Hyperparameter settings are discussed in Section 3. LiNGAM suite We evaluate the performance of our model against LiNGAM3 using the artificial model generator presented in [3]. The generator produces both dense and sparse networks with different degree of sparsity, Z is generated from a non-Gaussian heavy-tailed distribution, X is generated using equation (1) and then randomly permuted to hide the correct order, P . For the experiment we have generated 1000 different dataset/models using $d = \{5, 10\}$, $N = \{200, 500, 1000, 2000\}$ and the DAG was selected using the (training set) likelihood in equation (7). Results are summarized in Figure 2 using several performance measures. For the particular case of the area under the ROC curve (AUC), we use the conditional posterior of the masking matrix, i.e. $p(R|X, ?)$ where R is a matrix with elements r_{ij} . AUC is an important measure because it quantifies how the model accounts for the uncertainty of presence or absence of links in the DAG. Such uncertainty assessment is not possible in LiNGAM where the probability of having a link is simply zero or one, however the AUC can be still computed. 0.94

0.8	0.9	0.85
0.92		
0.7	0.8	
0.9		
0.8		
0.6		
0.84	0.82	0.8
0.75	0.6	
AUC		
True negative rate		
True positive rate		
0.86		
0.7		
0.65		
0.5	0.4	
0.6		
0.78		
Orderings error rate		
0.7		
0.88		
0.55		
d=5 Ours	d=5 LINGAM	d=10 Ours
d=10 LINGAM		
0.5		
0.4		
0.3		
0.3		
0.2		
0.2		
0.1		
0.76	200	

500
1000
2000
0.5 200
500
1000
2000
0.1 200
500
1000
2000
0 200
500
1000
N
N
N
N
(a)
(b)
(c)
(d)
2000

Figure 2: Performance measures for LiNGAM suite. Symbols are: square for 5 variables, star for 10 variables, solid line for sFA and dashed line for LiNGAM. (a) True positive rate. (b) True negative rate. (c) Frequency of AUC being greater than 0.9. (d) Number of estimated correct orderings. In terms of true negative rates, AUC and ordering error rate, our approach is significantly better than LiNGAM. The true positive rate results in Figure 2(a) show that LiNGAM outperform our approach only for $N = 2000$. However by comparing it to the true positive rate, it seems than LiNGAM prefer more dense models which could be an indication of overfitting. Looking to the ordering errors, our model is clearly superior. It is important to mention that being able to compute a probability for a link in the DAG to be zero, $p(\text{bij } 6= 0 \text{---} X, ?)$, turns out to be very useful in practice, for example to reject links with high uncertainty or to rank them. To give an idea of running times on a regular two-core 2.5GHz machine, for $d = 10$ and $N = 500$: LiNGAM took in average 10 seconds and our method 170 seconds. However, when doubling the number of variables the times were 730 and 550 seconds for LiNGAM and our method respectively, which is in agreement with our complexity estimates. 2 3

Source code available upon request (C with Matlab interface). Matlab package available at <http://www.cs.helsinki.fi/group/neuroinf/lingam/>.

6

Bayesian networks repository Next we want to compare some of the state of the art (Gaussian) approaches to DAG learning on 7 well known structures⁴, namely alarm, barley, carpo, hailfinder, insurance, mildew and water ($d = 37$,

48, 61, 56, 27, 35, 32 respectively). A single dataset of size 1000 per structure was generated using a similar procedure to the one used before. Apart from ours (sFA), we considered the following methods5 : standard DAG search (DS), order-search (OS), sparse candidate pruning then DAG-search (DSC) [6], L1MB then DAG-search (DSL) [8], sparsecandidate pruning then order-search (OSC) [7]. Results are shown in Figure 3, including the number of reversed links found due to ordering errors. water

water
water
water
mildew
mildew
mildew
mildew
insurance hailfinder DS OS OSC DSC DSL sFA
carpo barley alarm

0
0.1 0.2 False positive rate

(a)
insurance
insurance
insurance
hailfinder
hailfinder
hailfinder

carpo
carpo
carpo
barley
barley
barley
alarm
alarm

0
0.2 0.4 False negative rate

alarm
0
(b)
0.2
0.4 0.6 AUC

(c)
0.8
0
0.2 0.4 0.6 Reversed links
(d)

Figure 3: Performance measures for Bayesian networks repository experiments. In this case, our approach obtained slightly better results when looking at the false positive rate, Figure 3(a). The true negative rate is comparable to the other methods suggesting that our model in some cases is sparser than the others. AUC estimates are significantly better because we have continuous probabilities for links to be zero (in the other methods we had to use a binary value). From Figure 3(d), the number of reversed links in the other methods is quite high as expected due to lack of identifiability. Our model produced a small amount reversed links because it was not able to find any of the true orderings, but indeed something quite close. This results could be improved by running the sampler for a longer time or by considering more candidates. We also tried to run the other approaches with data generated from Gaussian distributions but the results were approximately equal to those shown in Figure 3. On the other hand, our approach performs similarly but the number of reversed links increases significantly since the model is no longer identified. The most important advantage of the (Gaussian) methods used in this experiment is their speed. In all cases they are considerably faster than sampling based methods. Their speed make them very suitable for large scale problems regardless of their identifiability issues. Model comparison For this experiment we have generated 1000 different datasets/models with $d = 5$ and $N = \{500, 1000\}$ in a similar way to the first experiment but this time we selected the true model to be a factor model or a DAG uniformly. In order to generate a factor model we basically just need to be sure that A cannot be permuted to a triangular form. We kept 20% of the data to compute the predictive densities to then select between all estimated DAG candidates and the factor model. We found that for $N = 500$ our approach was able to select true DAGs 91.5% of the times and true factor models 89.2%, corresponding to an overall error of 9.6%, For $N = 1000$ the true DAG and true factor model rates increased to 98.5% and 94.6% respectively. This results demonstrate that our approach is very effective at selecting the true underlying structure in the data between the two proposed hypotheses. Protein-signaling network The dataset introduced in [17] consists on flow cytometry measurements of 11 phosphorylated proteins and phospholipids (Raf, Erk, p38, Jnk, Akt, Mek, PKA, PKC, PIP2, PIP3, PLC?). Each observation is a vector of quantitative amounts measured from single cells, generated from a series of stimulatory cues and inhibitory interventions. The dataset contains both observational and experimental data. Here we are only using 1755 samples corresponding to 4

<http://compbio.cs.huji.ac.il/Repository/>. Parameters: 10000 iterations, 5 candidates (SC, DSC), max fan-in of 5 (OS, OSC) and Or strategy and MDL penalty (DSL). 5

7

Raf

PLC?

p38

PLC?

0.23

Mek PKC
 Erk
 PKA
 p38
 Akt Erk
 0.23
 PIP3
 0.93
 0.44
 PIP2
 0.46
 Jnk
 PKC
 0.15
 Jnk
 PIP2
 PIP2
 PLC? PKA
 PKA
 Akt
 PKC
 0.37
 Jnk
 0.02 0.24
 0.77
 Raf PIP3
 PIP3
 Erk
 Akt
 (a)
 Mek
 p38
 Mek 0.93
 (b)
 Raf
 (c)
 ?3
 8
 x 10
 1
 0.9
 7
 6
 4
 0.95
 0.8

3
2
1
0 ?4000 ?3800 ?3600 ?3400 ?3200 ?3000
Likelihood
(d)
0.9 1
2
3
4
5
6
7
Orderings
8
9
0.7 10
Accuracy
Ratio
5

Figure 4: Result for protein-signaling network. (a) Textbook signaling network as reported in [17]. (b) Estimated structure using Bayesian networks [17]. (c) Estimated structure using our model. (e) Test likelihoods for the best ordering DAG (dashed) and the factor model (solid). (d) Likelihood ratios (solid) and structure errors (dashed) for all candidates considered by our method and their usage. The Bayesian network is not able to identify the direction of the links with only observational data.

(e)

pure observational data and randomly selected 20% of the data to compute the predictive densities. Using the entire set will produce a richer model, however interventions are out of the scope of this paper. The textbook ground truth and results are presented in figure 4. From the 21 possible links in figure 4(a), the model from [17] was able to find 9, but also one falsely added link. In 4(b), a marginal likelihood equivalent prior is used and they therefore cannot make any inferences about directionality from observational data alone, see Figure 4(b). Our model in Figure 4(c) was able to find 10 true links, one falsely added link and only two reversed links (RL), one of them is PIP2 ? PIP3 which according to the ground truth is bidirectional and the other one, PLC? ? PIP3 which was also found reversed using experimental data in [17]. Note from figure 4(e) that the predictive density ratios correlate quite well with the structural accuracy. The predictive densities for the best candidate (sixth in Figure 4(e)) is shown in Figure 4(d) and suggests that the factor model is a better option which makes sense considering that estimated DAG in figure 4(c) is a substructure of the ground truth. We also examined the estimated factor model and we found out that three factors could correspond to unmeasured proteins (PI3K, MKK and IP3), see Figure 2 and table 3 in [17]. We also tried the above methods. Results

were very similar to our method in terms of true positives (? 9) and true negatives (? 32), however none of them were able to produce less than 6 reversed links that corresponds to approximately two-thirds of total true positives.

8

Discussion

We have proposed a novel approach to perform inference and model comparison of sparse factor models and DAGs within the same framework. The key ingredients for both Bayesian models are spike and slab priors to promote sparsity, heavy-tailed priors to ensure identifiability and predictive densities to perform the comparison. A set of candidate orderings is produced by the factor model. Subsequently, a linear DAG is learned for each of the candidates. To the authors' knowledge this is the first time that a method for comparing such a closely related linear models is proposed. This setting can be very beneficial in situations where the prior evidence suggests both DAG structure and/or unmeasured variables in the data. For example in the protein signaling network [17], the textbook ground truth suggests both DAG structure and a number of unmeasured proteins. The previous approach [17] only performed structure learning in DAGs but our results suggest that the data is better explained by the factor model. For further exploration of this data set, we obviously need to modify our approach to handle hybrid models, i.e. graphs with directed/undirected links and observed/latent nodes as well as being able to use experimental data. Our Bayesian hierarchical approach is very flexible. We are currently investigating extensions to other source distributions (non-parametric Dirichlet process, temporal Gaussian processes and discrete). 8

2 References

- [1] M. West. Bayesian factor regression models in the "large p, small n" paradigm. In J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, editors, *Bayesian Statistics 7*, pages 723-732. Oxford University Press, 2003.
- [2] J. Lucas, C. Carvalho, Q. Wang, A. Bild, J. R. Nevins, and M. West. *Bayesian Inference for Gene Expression and Proteomics*, chapter *Sparse Statistical Modeling in Gene Expression Genomics*, pages 155-176. Cambridge University Press, 2006.
- [3] S. Shimizu, P. O. Hoyer, A. Hyvärinen, and A. Kerminen. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7:2003-2030, October 2006.
- [4] D. M. Chickering. Learning Bayesian networks is NP-complete. In D. Fisher and H.-J. Lenz, editors, *Learning from Data: AI and Statistics*, pages 121-130. Springer-Verlag, 1996.
- [5] I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31-78, October 2006.
- [6] N. Friedman, I. Nachman, and D. Pe'er. Learning Bayesian network structure from massive datasets: The "sparse candidate" algorithm. In K. B. Laskey and H. Prade, editors, *UAI*, pages 206-215, 1999.
- [7] M. Teyssier and D. Koller. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. In *UAI*, pages 548-549, 2005.
- [8] M. W. Schmidt,

A. Niculescu-Mizil, and K. P. Murphy. Learning graphical model structure using L_1 regularization paths. In AAAI, pages 1278–1283, 2007. [9] D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, January 1995. [10] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, March 2000. [11] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, December 1994. [12] C. M. Carvalho, J. Chang, J. E. Lucas, J. R. Nevins, Q. Wang, and M. West. High-dimensional sparse factor modeling: Applications in gene expression genomics. *Journal of the American Statistical Association*, 103(484):1438–1456, December 2008. [13] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley-Interscience, May 2001. [14] D. F. Andrews and C. L. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society: Series B (Methodology)*, 36(1):99–102, 1974. [15] T. Park and G. Casella. The Bayesian lasso. 103(482):681–686, June 2008.

Journal of the American Statistical Association,

[16] N. Friedman and D. Koller. Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50(1–2):95–125, January 2003. [17] K. Sachs, O. Perez, D. Pe’er, D. A. Lauffenburger, and G. P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, April 2005.