Supplement: Active Learning of Continuous-time Bayesian Networks through Interventions

Dominik Linzner   Heinz Koeppl

1. Algorithms

Algorithm 1 Computation of VBHC for Parameter Learning

Input: Proposed intervention i, current initial state s_0, desired number of posterior-samples N_S, current belief over parameters $p(\Lambda \mid \mathcal{H}, G)$, initial variational parameters $\boldsymbol{\alpha}^{\kappa,0}$, $\boldsymbol{\beta}^{\kappa,0}$. As initial values for the optimization, we set the initial values for the optimization to the posterior counts $\boldsymbol{\alpha}^{\kappa,0} = \bar{\boldsymbol{\alpha}}$, $\boldsymbol{\beta}^{\kappa,0} = \bar{\boldsymbol{\beta}}$, see main-text.
for n_S = 1 : N_S **do**
  Draw $\hat{\Lambda}_{n_S} \sim p(\Lambda \mid \mathcal{H}, G)$.
  Perform intervention by setting $\hat{\Lambda}_{n_S,n} = 0$ and initial state x_n^0 for all $n \notin \aleph$.
  Calculate W from $\hat{\Lambda}_{n_S}$ and G by amalgamation.
  Solve the master-equation main-text (3) subject to W and s_0 and recover $\mathsf{E}[\hat{M}(s,s')]$ and $\mathsf{E}[\hat{T}(s)]$ using appendix (6) and (5) respectively.
  for n = 1 : N **do**
    Compute expected statistics $\mathsf{E}[\hat{M}_n \mid \hat{\Lambda}_{n_S}, G, i]$ and $\mathsf{E}[\hat{T}_n \mid \hat{\Lambda}_{n_S}, G, i]$ from $\mathsf{E}[\hat{M}(s,s')]$ and $\mathsf{E}[\hat{T}(s)]$, see appendix (3) and (4).
  end for
end for
Calculate $\mathrm{VBHC}(i,\kappa)$ via main-text (12) and gradients appendix (7) and (8) with weighted posterior samples replacing $\int \mathrm{d}p(\Lambda \mid \mathcal{H}, G)$ with $\sum_{n_S=1}^{N_S} p(\hat{\Lambda}_{n_S} \mid \mathcal{H}, G)$.
Minimize w.r.t κ.
Output: $\min_\kappa \mathrm{VBHC}(i,\kappa)$.

2. Derivations

All derivations are done for a fixed set of conditions i and respective initial states s_0. We will omit those in the following derivations for readability.

2.1. Kullback–Leibler divergence between two CTBNs

Evaluation of our design criteria (V)BHC, requires the calculation of the KL-divergence between two cCTBNs.

The likelihood of observing a CTBN path $D = S^{[0,T]}$ is (expressed in terms of its sufficient statistics)

$$p(S^{[0,T]} \mid \Lambda, G) = \prod_n \prod_{x,x',u} \Lambda_n(x,x',u)^{M_n(x,x',u)} \exp\left\{-\Lambda_n(x,x',u) T_n(x,u)\right\}. \tag{1}$$

The KL between two measures is defined via the integration over all paths

$$\mathrm{KL}\left(p(S^{[0,T]} \mid \Lambda, G) \,||\, p(S^{[0,T]} \mid \Lambda', G)\right) = \int \mathrm{d}p(S^{[0,T]} \mid \Lambda, G) \ln \frac{p(S^{[0,T]} \mid \Lambda, G)}{p(S^{[0,T]} \mid \Lambda', G)}.$$

Algorithm 2 Computation of VBHC for Structure Learning

```
Input: Proposed intervention $i$, current initial state $s_0$, desired number of posterior-samples $N_S$, current belief over parameters $p(\Lambda \mid \mathcal{H}, G)$ and structures $p(G \mid \mathcal{H})$, initial variational parameters $\kappa$.
for $n = 1 : N$ do
    for $n_S = 1 : N_S$ do
        for $\mathrm{par}(n)$ in power-set of $\mathcal{V}$ do
            Draw $\hat{G}_{n_S} \sim p(G \mid \mathcal{H}, \mathrm{par}(n))$.
            Draw $\hat{\Lambda}_{n_S} \sim p(\Lambda \mid \mathcal{H}, \hat{G}_{n_S})$.
            Perform intervention by setting $\hat{\Lambda}_{n_S,n} = 0$ and initial state $x_n^0$ for all $n \notin \aleph$.
            Calculate $W$ from $\hat{\Lambda}_s$ and $\hat{G}_{n_S}$ by amalgamation.
            Solve the master-equation main-text (3) subject to $W$ and recover $\mathsf{E}[\hat{M}(s,s')]$ and $\mathsf{E}[\hat{T}(s)]$.
            Compute expected statistics $\mathsf{E}[\hat{M}_n \mid \hat{\Lambda}_{n_S}, \hat{G}_{n_S}, i]$ and $\mathsf{E}[\hat{T}_n \mid \hat{\Lambda}_{n_S}, \hat{G}_{n_S}, i]$ from $\mathsf{E}[\hat{M}(s,s')]$ and $\mathsf{E}[\hat{T}(s)]$, see appendix (3) and (4).
        end for
    end for
end for
Calculate $\mathrm{VBHC}(i,\kappa)$ using appendix (13) and gradients appendix (14) with weighted posterior samples replacing $\sum_{G|\mathrm{par}(n)} \sum_{G'|\mathrm{par}'(n)} p(G \mid \mathcal{H}) q(G') \int \mathrm{d}p(\Lambda \mid G)$ with $\sum_{n_S, n'_S} p(\hat{G}_{n_S} \mid \mathcal{H}) q(\hat{G}_{n'_S}) p(\hat{\Lambda}_{n_S} \mid \hat{G}_{n_S})$.
Minimize w.r.t $\kappa$.
Output: $\min_\kappa \mathrm{VBHC}(i,\kappa)$.
```

Algorithm 3 Computation of the EIG for Parameter Learning

```
Input: Proposed intervention $i$, current initial state $s_0$, desired number of posterior-samples $N_S$, number of path samples $N_P$, current belief over parameters $p(\Lambda \mid \mathcal{H}, G)$.
Set EIG $= 0$.
for $n_S = 1 : N_S$ do
    Draw parameter $\hat{\Lambda}_{n_S} \sim p(\Lambda \mid \mathcal{H}, G)$
    Perform intervention by setting $\hat{\Lambda}_{n_S,n} = 0$ and initial state $x_n^0$ for all $n \notin \aleph$.
    for $n_p = 1 : N_P$ do
        Draw path $\hat{S}^{[0,T]} \sim p(S^{[0,T]} \mid \hat{\Lambda}_{n_S}, G, i, s_0)$.
        Set EIG $=$ EIG $+ \frac{1}{N_P N_S} \Big( \ln p(\hat{\Lambda}_{n_S} \mid \hat{S}^{[0,T]}, \mathcal{H}) - \ln p(\hat{\Lambda}_{n_S} \mid \mathcal{H}) \Big)$.
    end for
end for
Output: Estimate EIG.
```

Inserting (1) yields

$$\mathrm{KL}\left(p(S^{[0,T]} \mid \Lambda, G) \,||\, p(S^{[0,T]} \mid \Lambda', G) \right) = \sum_{n,x,x' \neq x, u} \tag{2}$$

$$\left\{ \Lambda'_n(x,x',u) - \Lambda_n(x,x',u) \right\} \mathsf{E}\left[\hat{T}_n(x,u) \mid \Lambda, G \right] - \ln \frac{\Lambda_n(x,x',u)}{\Lambda'_n(x,x',u)} \mathsf{E}\left[\hat{M}_n(x,x',u) \mid \Lambda, G \right],$$

with the expectations being taken with respect to the process $p(S^{[0,T]} \mid \Lambda, G)$. The expected moments can not be calculated from the parametric form of $p(S^{[0,T]} \mid \Lambda, G)$ directly. Instead, we will construct an ODE for the moments of the CTMC recovered after amalgamation, and recover its expectation as solutions. The moments of the CTBN can then be calculated as projections of the CTMC moments, the dwelling times per state $T(s)$ and the number of transitions $M(s,s')$

$$\hat{M}_n(x,x',u) = \sum_{s,s'} M(s,s') \mathbb{1}(s'_n = x') \mathbb{1}(s_n = x) \mathbb{1}(s_{\mathrm{par}(n)} = u), \tag{3}$$

$$\hat{T}_n(x,u) = \sum_{s} T(s) \mathbb{1}(s_n = x) \mathbb{1}(s_{\mathrm{par}(n)} = u). \tag{4}$$

Algorithm 4 Computation of the EIG for Structure Learning

Input: Proposed intervention i, current initial state s_0, number of path samples N_S, current belief over parameters $p(\Lambda \mid \mathcal{H}, G)$ and structures $p(G \mid \mathcal{H})$.
Set EIG $= 0$.
for $n = 1 : N$ **do**
  for $\mathrm{par}(n)$ in power-set of $\mathcal{V}$ **do**
    for $n_s = 1 : N_S$ **do**
      Draw $\hat{G}_{n_S} \sim p(G \mid \mathcal{H}, \mathrm{par}(n))$.
      Draw parameter $\hat{\Lambda}_{n_S} \sim p(\Lambda \mid \mathcal{H}, \hat{G}_{n_S})$.
      Perform intervention by setting $\hat{\Lambda}_{n_S,n} = 0$ and initial state x_n^0 for all $n \notin \aleph$.
      Draw path $\hat{S}^{[0,T]} \sim p(S^{[0,T]} \mid \hat{\Lambda}_{n_S}, \hat{G}_{n_S}, i, s_0)$.
      Set EIG $=$ EIG $+ \frac{1}{N_S}\Big(\ln p(\mathrm{par}(n) \mid \mathcal{H}, \hat{S}^{[0,T]}) - \ln p(\mathrm{par}(n) \mid \mathcal{H})\Big)$,
      see appendix (12).
    end for
  end for
end for
Output: Estimate EIG.

2.2. Moment ODEs of a CTMC

Expected Dwelling-times. The expected dwelling-times $\mathsf{E}\left[T(s)\right]$ in a state $s \in \mathcal{S}$ of a CTMC are calculated as solution of an ODE. For this, we need to consider the evolution of the stochastic process $T(s,t)$, the dwelling times in state $s \in \mathcal{S}$ up to time t. For this process we can denote transition probabilities, by considering the dynamics of the CTMC

$$\begin{aligned} p(T(s,t+h) = \tau + h \mid T(s,t) = \tau) &= p(S(t+h) = s, S(t) = s), \\ p(T(s,t+h) = \tau \mid T(s,t) = \tau) &= 1 - p(S(t+h) = s, S(t) = s). \end{aligned}$$

Thus $T(s,t)$ evolves according to

$$\begin{aligned} p(T(s,t+h) = \tau) = {} & p(S(t+h) = s, S(t) = s) p(T(s,t) = \tau - h) \\ & + \left[1 - p(S(t+h) = s, S(t) = s)\right] p(T(s,t) = \tau). \end{aligned}$$

For small h, we can expand $p(T(s,t) = \tau - h) = p(T(s,t) = \tau) - h\partial_\tau p(T(s,t) = \tau) + o(h)$, and we arrive at

$$\frac{p(T(s,t+h) = \tau) - p(T(s,t) = \tau)}{h} = -p(S(t) = s)\partial_\tau p(T(s,t) = \tau) + o(h).$$

We can now take the expectation $\mathsf{E}\left[T(s,t)\right] = \int_0^\infty \mathrm{d}\tau' \tau' p(T(s,t) = \tau')$, and the continuum limit $h \to 0$ in order to arrive at

$$\partial_t \mathsf{E}\left[T(s,t)\right] = -p(S(t) = s) \int_0^\infty \mathrm{d}\tau' \, \tau' \partial_{\tau'} p(T(s,t) = \tau'),$$

which, after integration by parts, reduces to simply

$$\partial_t \mathsf{E}\left[T(s,t)\right] = p(S(t) = s).$$

Thus, the expected dwelling-time is given by the solution

$$\mathsf{E}\left[T(s)\right] = \int_0^T \mathrm{d}t \; p(S(t) = s). \tag{5}$$

Expected Number of Transitions. Similarly to above, we can compute the expected number of transitions of a CTMC $\mathsf{E}\left[M(s,s')\right]$. The computation is analogous to above. We consider the stochastic process $M(s,s',t)$ of transitions from s to s' till time t. Transition probabilities are

$$\begin{aligned} p(M(s,s',t+h) = k \mid M(s,s',t) = k-1) &= p(S(t+h) = s', S(t) = s), \\ p(M(s,s',t+h) = k \mid M(s,s',t) = k) &= 1 - p(S(t+h) = s', S(t) = s). \end{aligned}$$

After inserting the identity $p(S(t+h)=s', S(t)=s) = \mathbb{1}(s=s') + hW(s,s') + o(h)$, we arrive at

$$\frac{p(M(s,s',t+h)=k) - p(M(s,s',t)=k)}{h}$$
$$= W(s,s')p(S(t)=s)\left[p(M(s,s',t)=k-1) - p(M(s,s',t)=k)\right] + o(h).$$

The expected number of transitions can be calculated via $\mathsf{E}\left[M(s,s',t)\right] = \sum_{k=0}^{\infty} p(M(s,s',t)=k)$. Noticing that $p(M(s,s',t)=k-1)=0$ for $k<1$, we can perform an index-shift $k \to k+1$and arrive at

$$\frac{\mathsf{E}\left[M(s,s',t+h)\right] - \mathsf{E}\left[M(s,s',t)\right]}{h}$$
$$= W(s,s')p(S(t)=s)\left[\mathsf{E}\left[M(s,s',t)\right] - \mathsf{E}\left[M(s,s',t)\right] + 1\right] + o(h),$$

and thus in the continuum limit $h \to 0$ we recover the ODE,

$$\partial_t \mathsf{E}\left[M(s,s',t)\right] = W(s,s')p(S(t)=s),$$

with the solution

$$\mathsf{E}\left[M(s,s')\right] = W(s,s')\mathsf{E}\left[T(s)\right]. \tag{6}$$

2.3. (V)BHC for Parameter Learning

Equipped with the moments derived in the last Section, we can now derive the (V)BHC. The VBHC takes the form of an expected KL-divergence

$$\mathrm{VBHC} = \int \mathrm{d}\Lambda \int \mathrm{d}\Lambda'\, p(\Lambda \mid \mathcal{H}, G) q_\kappa(\Lambda') \mathrm{KL}\left(p(S^{[0,T]} \mid \Lambda, G) \,||\, p(S^{[0,T]} \mid \Lambda', G)\right)$$
$$+ \mathrm{KL}\left(q_\kappa(\Lambda) \,||\, p(\Lambda \mid \mathcal{H}, G)\right),$$

with the KL given in appendix (2). As explained in the main-text, we have $p(\Lambda \mid \mathcal{H}, G) = \prod_{n,x,x',u} \mathrm{Gam}\left(\Lambda_n(x,x',u) \mid \bar{\alpha}_n(x,x',u), \bar{\beta}_n(x,u)\right)$ and choose $q_\kappa(\Lambda') = \prod_{n,x,x',u} \mathrm{Gam}\left(\Lambda_n(x,x',u) \mid \alpha_n^\kappa(x,x',u), \beta_n^\kappa(x,u)\right)$. As the expected moments in (2) only depend on Λ, we can calculate the integral over Λ' analytically. For this, we notice that the moments

$$\mathsf{E}\left[\Lambda_n(x,x',u)\right] = \frac{\alpha_n^\kappa(x,x',u)}{\beta_n^\kappa(x,u)},$$
$$\mathsf{E}\left[\ln \Lambda_n(x,x',u)\right] = \psi^{(0)}(\alpha_n^\kappa(x,x',u)) - \ln \beta_n^\kappa(x,u),$$

where the expectation is w.r.t $q_\kappa(\Lambda)$, have a closed form expression. By insertion into (2), we recover the expression from the main-text. Finally, we notice that

$$\mathrm{KL}\left(q_\kappa(\Lambda') \,||\, p(\Lambda \mid \mathcal{H}, G)\right) =$$
$$\sum_{n,x,x',u} \mathrm{KL}\left(\mathrm{Gam}(\alpha_n^\kappa(x,x',u), \beta_n^\kappa(x,u)) \,||\, \mathrm{Gam}(\bar{\alpha}_n(x,x',u), \bar{\beta}_n(x,u))\right),$$

with the KL-divergence between two gamma-distributions (5)

$$\mathrm{KL}\left(\mathrm{Gam}(\alpha_n^\kappa(x,x',u), \beta_n^\kappa(x,u)) \,||\, \mathrm{Gam}(\bar{\alpha}_n(x,x',u), \bar{\beta}_n(x,u))\right) =$$
$$\bar{\alpha}_n(x,x',u) \ln\left(\frac{\beta_n^\kappa(x,u)}{\bar{\beta}_n(x,u)}\right) - \ln\left(\frac{\Gamma(\alpha_n^\kappa(x,x',u))}{\Gamma(\bar{\alpha}_n(x,x',u))}\right)$$
$$+ (\alpha_n^\kappa(x,x',u) - \bar{\alpha}_n(x,x',u)))\psi(\alpha_n^\kappa(x,x',u))) - (\beta_n^\kappa(x,u) - \bar{\beta}_n(x,u))\frac{\alpha_n^\kappa(x,x',u)}{\beta_n^\kappa(x,u)}.$$

Gradients. The gradients of the VBHC can be calculated in (semi-)analytical form

$$\partial_{\alpha_n^\kappa(x,x',u)}\text{VBHC} = \int d\Lambda\, p(\Lambda \mid \mathcal{H}, G)\mathsf{E}\left[\hat{T}_n \mid \Lambda, G\right]\left\{\Lambda\psi^{(1)}(\alpha_n^\kappa(x,x',u)) - \frac{1}{\beta_n^\kappa(x,u)}\right\} \tag{7}$$
$$+ \alpha_n^\kappa(x,x',u)\psi^{(1)}(\alpha_n^\kappa(x,x',u)) - \frac{(\beta_n^\kappa(x,u) - \bar{\beta}_n(x,u))}{\beta_n^\kappa(x,u)}$$
$$\partial_{\beta_n^\kappa(x,u)}\text{VBHC} = \int d\Lambda\, p(\Lambda \mid \mathcal{H}, G)\mathsf{E}\left[\hat{T}_n \mid \Lambda, G\right]\left\{\frac{\alpha_n^\kappa(x,x',u)}{\beta_n^\kappa(x,u)^2} - \frac{\Lambda}{\beta_n^\kappa(x,u)}\right\} \tag{8}$$
$$+ \frac{\bar{\alpha}_n(x,x',u)}{\beta_n^\kappa(x,u)} - \frac{\alpha_n^\kappa(x,x',u)}{\beta_n^\kappa(x,u)} + (\beta_n^\kappa(x,u) - \bar{\beta}_n(x,u))\frac{\alpha_n^\kappa(x,x',u)}{\beta_n^\kappa(x,u)^2}.$$

If necessary, also higher-order derivatives can be computed in principle.

In all results above, the corresponding BHC expressions are recovered by setting $\beta_n^\kappa(x,u) = \bar{\beta}_n(x,u)$ and $\alpha_n^\kappa(x,x',u) = \bar{\alpha}_n(x,x',u)$.

2.4. (V)BHC for Structure Learning

KL-divergence between Marginal CTBNs. The marginal likelihood of a path $\hat{S}^{[0,T]} \sim p(\hat{S}^{[0,T]} \mid \Lambda, G)$, with statistics $\hat{T}_n(x,u)$ and $\hat{M}_n(x,x',u)$, given a structure and history $\mathcal{H}$ can be calculated via marginalization of (1)

$$p(\hat{S}^{[0,T]} \mid G, \mathcal{H}) = \int d\Lambda\, p(\Lambda \mid \mathcal{H}, G)\prod_n \prod_{x,x',u} \Lambda_n(x,x',u)^{\hat{M}_n(x,x',u)} \exp\left\{-\Lambda_n(x,x',u) + \hat{T}_n(x,u)\right\}$$
$$\propto \prod_n \prod_{x,x'\neq x,u} \Gamma(\hat{\alpha}_n(x,x',u))\hat{\beta}_n(x,u)^{-\hat{\alpha}_n(x,x',u)}, \tag{9}$$

where $\hat{\alpha}_n(x,x',u) = \hat{M}_n(x,x',u) + \bar{\alpha}_n(x,x',u)$ and $\hat{\beta}_n(x,u) = \hat{T}_n(x,u) + \bar{\beta}_n(x,u)$ and $\bar{\alpha}_n(x,x',u) = \alpha_n(x,x',u) + M_n(x,x',u,i=0)$ and $\bar{\beta}_n(x,x',u) = \beta_n(x,u) + T_n(x,u,i=0)$, see main-text. The KL between two measures, in this case CTBNs with different graphs, is defined via the integration over all paths

$$\text{KL}\left(p(S^{[0,T]} \mid G, \mathcal{H})\,||\,p(S^{[0,T]} \mid G', \mathcal{H})\right) = \int \mathrm{d}p(S^{[0,T]} \mid G, \mathcal{H}) \ln \frac{p(S^{[0,T]} \mid G, \mathcal{H})}{p(S^{[0,T]} \mid G', \mathcal{H})}.$$

In order to avoid solving the computationally taxing solution of the marginal master-equation (6; 2) (which is an integro-differential equation), we can express this in terms of the original path-measure

$$\text{KL}\left(p(S^{[0,T]} \mid G, \mathcal{H})\,||\,p(S^{[0,T]} \mid G', \mathcal{H})\right) =$$
$$\int \mathrm{d}p(\Lambda \mid G, \mathcal{H}) \int \mathrm{d}p(S^{[0,T]} \mid \Lambda, G) \ln \frac{p(S^{[0,T]} \mid G, \mathcal{H})}{p(S^{[0,T]} \mid G', \mathcal{H})}.$$

Inserting (9) yields

$$\text{KL}\left(p(S^{[0,T]} \mid G, \mathcal{H})\,||\,p(S^{[0,T]} \mid G', \mathcal{H})\right) = \int \mathrm{d}p(\Lambda \mid G, \mathcal{H}) \int \mathrm{d}p(S^{[0,T]} \mid \Lambda, G) \sum_{n\in\aleph} \tag{10}$$
$$\sum_{u\in\mathcal{U}_n^G} \sum_{u'\in\mathcal{U}_n^{G'}} \sum_{x,x'\neq x} \left[\ln \frac{\Gamma(\hat{\alpha}_n(x,x',u\,))}{\Gamma(\hat{\alpha}_n(x,x',u'))} + \hat{\alpha}_n(x,x',u') \ln \hat{\beta}_n(x,u') - \hat{\alpha}_n(x,x',u) \ln \hat{\beta}_n(x,u)\right].$$

As mentioned in the main-text, exact computation of the integral w.r.t $p(S^{[0,T]} \mid \Lambda, G)$ is not feasible, due to non-linearity. For this reason we expand this KL around the expected transitions and dwelling times and arrive at

$$\text{KL}\left(p(S^{[0,T]} \mid G, \mathcal{H})\,||\,p(S^{[0,T]} \mid G', \mathcal{H})\right) \approx \int \mathrm{d}p(\Lambda \mid G, \mathcal{H}) \sum_{n\in\aleph} \sum_{u\in\mathcal{U}_n^G} \sum_{u'\in\mathcal{U}_n^{G'}} \sum_{x,x'\neq x} \tag{11}$$
$$\left[\ln \frac{\Gamma(\mathsf{E}[\hat{\alpha}_n(x,x',u\,)])}{\Gamma(\mathsf{E}[\hat{\alpha}_n(x,x',u')])} + \mathsf{E}[\hat{\alpha}_n(x,x',u')] \ln \mathsf{E}[\hat{\beta}_n(x,u')] - \mathsf{E}[\hat{\alpha}_n(x,x',u\,) \ln \mathsf{E}[\hat{\beta}_n(x,u)]\right],$$
$$\equiv \mathcal{F}[\kappa, p(G \mid \mathcal{H})]$$

with $\mathsf{E}[\hat{\alpha}_n(x,x',u)] \equiv \bar{\alpha}_n(x,x',u) + \mathsf{E}[\hat{M}_n(x,x',u) \mid \Lambda, G]$ and $\mathsf{E}[\hat{\beta}_n(x,u)] \equiv \bar{\beta}_n(x,u) + \mathsf{E}[\hat{T}_n(x,u) \mid \Lambda, G]$.

Below, we derive higher-order moments of the transitions and dwelling-times. This allows to compute higher-order approximations of this KL-divergence, under higher computational costs. However, in this work a first order approximation was sufficient to demonstrate effectiveness of our method.

VBHC for Structure Learning. We can then approximate the VBHC by

$$\mathrm{VBHC} \approx \mathcal{F}[\kappa, p(G \mid \mathcal{H})] + \mathrm{KL}\left(q_\kappa(G) \,||\, p(G \mid \mathcal{H})\right),$$

with the KL-divergence, between two categoricals

$$\mathrm{KL}\left(p(G \mid \mathcal{H}) \,||\, q_\kappa(G)\right) = \sum_G q_\kappa(G)\left(\ln q_\kappa(G) - \ln p(G \mid \mathcal{H})\right).$$

While the form of $\mathcal{F}$ is compact, it is helpful for computational reasons to re-order this summation into a node-wise form. This is helpful, as it will allow is to compute sample approximations of the VBHC, where only a summation over local parent-sets instead of global graphs needs to be performed

$$\mathcal{F}[\kappa, p(G \mid \mathcal{H})] = \sum_{n \in \aleph} \sum_{G,G'} p(G \mid \mathcal{H}) q(G') \int \mathrm{d}p(\Lambda \mid G, \mathcal{H}) \sum_{u \in \mathcal{U}_n^G} \sum_{u' \in \mathcal{U}_n^{G'}} \sum_{x, x' \neq x}$$
$$\left[\ln \frac{\Gamma(\mathsf{E}[\hat{\alpha}_n(x,x',u)])}{\Gamma(\mathsf{E}[\hat{\alpha}_n(x,x',u')])} + \mathsf{E}[\hat{\alpha}_n(x,x',u')] \ln \mathsf{E}[\hat{\beta}_n(x,u')] - \mathsf{E}[\hat{\alpha}_n(x,x',u) \ln \mathsf{E}[\hat{\beta}_n(x,u)]\right].$$

The product form of (9) translates to a product posterior, if not broken by the prior, over parent-sets

$$\begin{aligned} p(G \mid \mathcal{H}) &= \prod_n p(\mathrm{par}^G(n) \mid \mathcal{H}) \\ &\propto \prod_n p(\mathrm{par}^G(n)) \prod_{x, x' \neq x} \prod_{u \in \mathcal{U}_n^G} \Gamma(\bar{\alpha}_n(x,x',u)) \bar{\beta}_n(x,u)^{-\bar{\alpha}_n(x,x',u)}. \end{aligned} \tag{12}$$

This allows us to rewrite

$$\begin{aligned} &\sum_{n \in \aleph} \sum_{G,G'} p(G) q_\kappa(G') = \\ &\sum_{n \in \aleph} \sum_{\mathrm{par}(n), \mathrm{par}'(n) \subset \mathcal{V}} p(\mathrm{par}(n) \mid \mathcal{H}) q_\kappa(\mathrm{par}'(n)) \sum_{G \mid \mathrm{par}(n), G' \mid \mathrm{par}'(n)} p(G \mid \mathcal{H}) q_\kappa(G'). \end{aligned}$$

We then get the form of the VBHC for structure learning, as used in algorithm 2

$$\mathcal{F}[\kappa, p(G \mid \mathcal{H})] = \sum_{n \in \aleph} \sum_{\mathrm{par}(n), \mathrm{par}'(n)} p(\mathrm{par}(n) \mid \mathcal{H}) q_\kappa(\mathrm{par}'(n)) \tag{13}$$
$$\sum_{u \in \mathcal{U}_n^{\mathrm{par}(n)}} \sum_{u' \in \mathcal{U}_n^{\mathrm{par}(n)'}} \sum_{G \mid \mathrm{par}(n)} \sum_{G' \mid \mathrm{par}'(n)} p(G \mid \mathcal{H}) q_\kappa(G') \int \mathrm{d}p(\Lambda \mid G, \mathcal{H}) \sum_{x, x' \neq x}$$
$$\left[\ln \frac{\Gamma(\mathsf{E}[\hat{\alpha}_n(x,x',u)])}{\Gamma(\mathsf{E}[\hat{\alpha}_n(x,x',u')])} + \mathsf{E}[\hat{\alpha}_n(x,x',u')] \ln \mathsf{E}[\hat{\beta}_n(x,u')] - \mathsf{E}[\hat{\alpha}_n(x,x',u) \ln \mathsf{E}[\hat{\beta}_n(x,u)]\right].$$

Similarly, we make the ansatz for $q_\kappa(G) = \prod_n q_\kappa(\mathrm{par}^G(n))$, then the KL-divergence decomposes

$$\mathrm{KL}\left(p(G \mid \mathcal{H}) \,||\, q_\kappa(G)\right) = \sum_n \sum_{\mathrm{par}(n)} q_\kappa(\mathrm{par}(n))\left(\ln q_\kappa(\mathrm{par}(n)) - \ln p(\mathrm{par}(n) \mid \mathcal{H})\right).$$

Gradients. The gradient for the parameter $q_\kappa(\mathrm{par}'(n))$ can be calculated to be

$$\partial_{q_\kappa(\mathrm{par}'(n))} \mathrm{VBHC} = 1 + \ln q_\kappa(\mathrm{par}'(n)) - \ln p(\mathrm{par}'(n) \mid \mathcal{H}) + \sum_{\mathrm{par}(n)} p(\mathrm{par}(n) \mid \mathcal{H}) \tag{14}$$
$$\sum_{u \in \mathcal{U}_n^{\mathrm{par}(n)}} \sum_{u' \in \mathcal{U}_n^{\mathrm{par}(n)'}} \sum_{G \mid \mathrm{par}(n)} \sum_{G' \mid \mathrm{par}'(n)} p(G \mid \mathcal{H}) q_\kappa(G') \int \mathrm{d}p(\Lambda \mid G, \mathcal{H}) \sum_{x, x' \neq x}$$
$$\left[\ln \frac{\Gamma(\mathsf{E}[\hat{\alpha}_n(x,x',u)])}{\Gamma(\mathsf{E}[\hat{\alpha}_n(x,x',u')])} + \mathsf{E}[\hat{\alpha}_n(x,x',u')] \ln \mathsf{E}[\hat{\beta}_n(x,u')] - \mathsf{E}[\hat{\alpha}_n(x,x',u) \ln \mathsf{E}[\hat{\beta}_n(x,u)]\right].$$

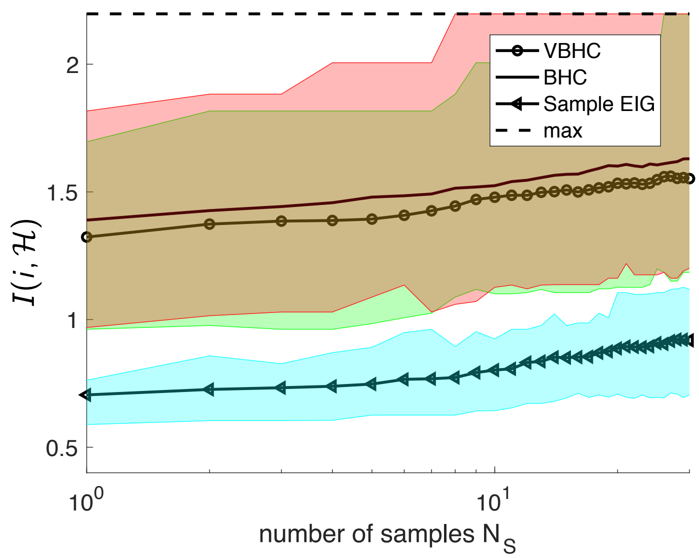

Figure 1. Mutual information between design sample estimates and recommended interventions for different number of samples N_S. Areas denote 25-75% confidence intervals.

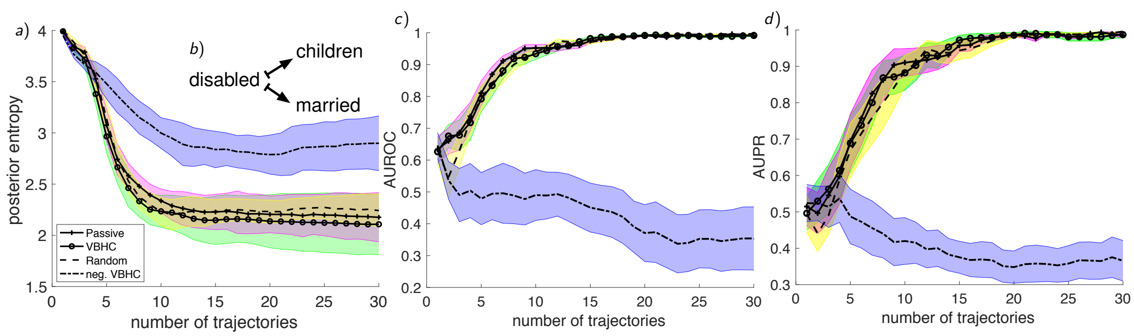

Figure 2. a) Mean and variance (area) of the evolution of the posterior entropy in BHPS data-set for 100 repetitions. b) Sketch of the underlying network. c) AUROC and d) AUPR converge equally fast to the inferred network b) for all criteria but negative VBHC.

3. Experiments

3.1. Additional Experiments

Sample Estimates of Design Criteria. We want to investigate the viability of using sample estimates of different criteria for active learning of CTBNs. One basic requirement on such an estimate is that its recommendations actually depend on the history of observations $\mathcal{H} \underset{\text{design}}{\longrightarrow} i$. We can make this formal by the following non-parametric dependency check: The recommended intervention i is dependent on experimental sequence $\mathcal{H}$ if they share high mutual information $I(i, \mathcal{H})$. We stress, that this does not reflect the quality of recommended interventions! We calculate the MI for random graphs of size $L = 3$ for different sample sizes N_S for random histories $\mathcal{H}$ consisting of 30 trajectories drawn from our synthetic network. The results are displayed in figure 1. For all sample sizes considered, (V)BHC shares a much higher MI with their recommended interventions, than the sample estimate of the EIG.

3.2. Processing of British-Household Data-set

As mentioned in the main-text, the British-Household Data-set is incomplete, as no complete paths of variables are provided, but only their measurement at singular time-points $t_i \in \{1, \ldots, 11\}$ (yearly for 11 years). In order to process, this kind of data, we employ a standard forward backward filter for continuous-time Markov jump processes, as in (4; 3; 1; 2). For this data $Y^{[0,T]} \equiv \{Y(t_i) \mid t_i \in \{1,\dots,11\}\}$ and $Y(t_i) \sim p(Y(t_i) \mid S(t_i))$ some observation model, with measurements at singular time-points, posterior inference of the marginals $p(S(t)=s \mid Y^{[0,T]})$ is implemented by solving a time-dependent master-equation

$$\frac{\mathrm{d}}{\mathrm{d}t}p(S(t)=s \mid Y^{[0,T]}) = \\ \sum_{s'\neq s}\left[\hat{W}(s',s,t)p(S(t)=s \mid Y^{[0,T]}) - \hat{W}(s,s',t)p(S(t)=s' \mid Y^{[0,T]})\right]$$

with $\hat{W}(s,s',t) = W(s,s')\frac{\rho(s',t)}{\rho(s,t)}$ and

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho(s',t) = -\sum_{s'\neq s}\left[W(s',s)\rho(s,t) - W(s,s')\rho(s',t)\right]$$

$$\text{subject to: } \lim_{t\to t_i^-}\rho(s,t) = \lim_{t\to t_i^+}\rho(s,t)\ln p(Y(t_i) \mid S(t_i)=s).$$

This allows to calculate the marginal likelihood

$$p(Y^{[0,T]} \mid W) = \prod_{s,s'\neq s} W(s,s')^{\mathsf{E}\left[M(s,s')\mid Y^{[0,T]}\right]}\exp\left\{W(s,s)\mathsf{E}\left[T(s) \mid Y^{[0,T]}\right]\right\},$$

with $\mathsf{E}\left[T(s) \mid Y^{[0,T]}\right] \equiv \int \mathrm{d}t\; p(S(t)=s \mid Y^{[0,T]})$ and $\mathsf{E}\left[M(s,s') \mid Y^{[0,T]}\right] \equiv W(s,s')\mathsf{E}\left[T(s) \mid Y^{[0,T]}\right]$. By calculation of the corresponding moments of the CTBNs by appendix (4) and (3), we can also write this likelihood in terms of rates Λ and structure G

$$p(Y^{[0,T]} \mid \Lambda, G) = \\ \prod_{n,x,x'\neq x,u}\Lambda_n(x,x',u)^{\mathsf{E}\left[M_n(x,x',u)\mid Y^{[0,T]}\right]}\exp\left\{\Lambda_n(x,x,u)\mathsf{E}\left[T_n(x,u) \mid Y^{[0,T]}\right]\right\}.$$

As can be seen in (2), this finally allows to form a posterior over parameters $p(\Lambda \mid Y^{[0,T]}) \propto p(Y^{[0,T]} \mid \Lambda)p(\Lambda)$, which is again a Gamma distribution, if $p(\Lambda)$ is gamma-distributed. Similarly, this holds for structures, by marginalization. Aside from this posterior calculation, everything about our method remains the same for incomplete data.

In Fig. 2 a), we track the evolution of the posterior entropy over structures for 100 independent runs. In Fig. 2 b) and c), we show that for all designs (except the "worst" design neg. VBHC) the inferred network converges against the one inferred using the full data-set (using AUROC and AUPR as metrics). We note that the effect of active learning can be expected to be small in this synthetic scenario, as we were only able to intervene on a single node.

References

[1] Ido Cohn, Tal El-Hay, Nir Friedman, and Raz Kupferman. Mean field variational approximation for continuous-time bayesian networks. *Journal Of Machine Learning Research*, 11:2745–2783, 2010.

[2] Dominik Linzner and Heinz Koeppl. Cluster variational approximations for structure learning of continuous-time bayesian networks from incomplete data. *Advances in Neural Information Processing Systems*, (NeurIPS):7880–7890, 2018.

[3] Manfred Opper and Guido Sanguinetti. Variational inference for markov jump processes. *Advances in Neural Information Processing Systems 20*, pages 1105–1112, 2008.

[4] Manfred Opper and Ole Winther. Tractable approximations for probabilistic models: The adaptive thouless-anderson-palmer mean field approach. *Physical Review Letters*, 86(17):3695–3699, 2001.

[5] Joram Soch and Carsten Allefeld. Kullback-leibler divergence for the normal-gamma distribution. pages 0–10, 2016.

[6] Lukas Studer, Christoph Zechner, Matthias Reumann, Loic Pauleve, Maria Rodriguez Martinez, and Heinz Koeppl. Marginalized continuous time bayesian networks for network reconstruction from incomplete observations. *Proceedings of the 30th Conference on Artificial Intelligence (AAAI 2016)*, pages 2051–2057, 2016.