Convex Relaxations for Learning Bounded-Treewidth Decomposable Graphs. Supplementary Material, ICML 2013.

K. S. Sesh Kumar Francis Bach

- Note that all the equations we refer to belong to this supplementary material

Derivation of Cost Function. The projection of the joint probability distribution of the random variables $X = (X_1, X_2, \ldots, X_n)$, associated with the vertices in V, on a decomposable graph G is given by:

$$p_G(x) = \frac{\prod_{C \in \mathcal{C}(G)} p_C(x_C)}{\prod_{(C,D) \in \mathcal{T}(G)} p_{C \cap D}(x_{C \cap D})} \ , \tag{1}$$

where x is an instance in the domain of X, which we denote by $\mathcal{X}$. $p_C(x_C)$ denotes the marginal distribution of random variables belonging to $C \in \mathcal{C}(G)$ and $p_{C \cap D}(x_{C \cap D})$ denotes the marginal distribution of random variables belonging to the *separator* set $C \cap D$, such that $(C, D) \in \mathcal{T}(G)$.

Let $\hat{p}(x)$ denote the empirical distribution and $\hat{p}_G(x)$ denotes the projected distribution on a decomposable graph G. Estimating the maximum likelihood decomposable graph which best approximates $\hat{p}$ is equivalent to finding the graph, G, which minimizes the KL-divergence between the target distribution and the projected distribution, $\hat{p}_G$, defined by $D(\hat{p}||\hat{p}_G)$.

$$\begin{aligned}
D(\hat{p}||\hat{p}_G) &= \sum_{x \in \mathcal{X}} \hat{p}(x) \log \frac{\hat{p}(x)}{\hat{p}_G(x)} \\
&\propto \sum_{x \in \mathcal{X}} -\hat{p}(x) \log \hat{p}_G(x) \text{ as } \hat{p}(x) \text{ is independent of } G, \\
&= \sum_{x \in \mathcal{X}} -\hat{p}(x) \log \frac{\prod_{C \in \mathcal{C}(G)} \hat{p}_C(x_C)}{\prod_{(C,D) \in \mathcal{T}(G)} \hat{p}_{C \cap D}(x_{C \cap D})} \text{ from Eq. (1)}, \\
&= \sum_{x \in \mathcal{X}} \Big(- \hat{p}(x) \log \prod_{C \in \mathcal{C}(G)} \hat{p}_C(x_C) \Big) - \sum_{x \in X} \Big(- \hat{p}(x) \log \prod_{(C,D) \in \mathcal{T}(G)} \hat{p}_{C \cap D}(x_{C \cap D}) \Big) \\
&= \sum_{C \in \mathcal{C}(G)} \sum_{x \in \mathcal{X}} -\hat{p}(x) \log \hat{p}_C(x_C) - \sum_{(C,D) \in \mathcal{T}(G)} \sum_{x \in \mathcal{X}} -\hat{p}(x) \log \hat{p}_{C \cap D}(x_{C \cap D}) \\
&= \sum_{C \in \mathcal{C}(G)} \sum_{x_C \in \mathcal{X}_C} -\hat{p}_C(x_C) \log \hat{p}_C(x_C) - \sum_{(C,D) \in \mathcal{T}(G)} \sum_{x_{C \cap D} \in \mathcal{X}_{C \cap D}} -\hat{p}_{C \cap D}(x_{C \cap D}) \log \hat{p}_{C \cap D}(x_{C \cap D}) \\
&= \sum_{C \in \mathcal{C}(G)} H(C) - \sum_{(C,D) \in \mathcal{T}(G)} H(C \cap D) \ , \qquad (2)
\end{aligned}$$

where $H(S)$ is the entropy of the random variables representing the set $S \subseteq V$, defined by $H(S) = \sum_{x_S \in \mathcal{X}_S} -\hat{p}_S(x_S) \log \hat{p}_S(x_S)$.

Primal optimization problem. $\mathcal{P}(\tau, \rho)$ denotes the primal cost function.

$$\mathcal{P}(\tau, \rho) = \sum_{C \in \mathcal{D}} H(C)\tau(C) - \sum_{(C,D) \in \mathcal{E}} H(C \cap D)\rho(C, D). \tag{3}$$

The constraints of the combinatorial optimization problem are given by:

$$\forall i \in V, \textstyle\sum_{C \in \mathcal{D}} 1_{i \in C} \tau(C) \geq 1, \tag{4}$$
$$\textstyle\sum_{(C,D) \in \mathcal{E}} \rho(C, D) = n - k - 1, \tag{5}$$
$$\textstyle\sum_{C \in \mathcal{D}} \tau(C) = n - k. \tag{6}$$
$$\forall i \in V, \textstyle\sum_{(C,D) \in \mathcal{E}} 1_{i \in (C \cap D)} \rho(C, D) - \sum_{C \in \mathcal{D}} 1_{i \in C} \tau(C) + 1 = 0, \tag{7}$$
$$\forall C \in \mathcal{D}, \forall (C, D) \in \mathcal{E}, \rho(C, D) \leq \tau(C), \tag{8}$$
$$\forall C \in \mathcal{D}, \tau(C) \leq \textstyle\sum_{(C,D) \in \mathcal{E}} \rho(C, D), \tag{9}$$
$$\rho \text{ is in the forest polytope of } (\mathcal{D}, \mathcal{E}), \tag{10}$$
$$\tau \text{ is in the hyperforest polytope of } (V, \mathcal{D}). \tag{11}$$

The combinatorial optimization problem is given by

$$\begin{array}{l} \min \mathcal{P}(\tau, \rho) \text{ subject to} \\ \tau \in \{0, 1\}^{\mathcal{D}},\ \rho \in \{0, 1\}^{\mathcal{E}}, \\ \text{Eq. (4), Eq. (5), Eq. (6), Eq. (7),} \\ \text{Eq. (8), Eq. (9), Eq. (10) and Eq. (11).} \end{array} \tag{12}$$

The τ-relaxed primal optimization problem is

$$\begin{array}{l} \min \mathcal{P}(\tau, \rho) \text{ subject to} \\ \tau \in [0, 1]^{\mathcal{D}},\ \rho \in \{0, 1\}^{\mathcal{E}}, \\ \text{Eq. (4), Eq. (5), Eq. (6), Eq. (7),} \\ \text{Eq. (8), Eq. (9), Eq. (10) and Eq. (11).} \end{array} \tag{13}$$

Proposition 1 *The non-convex primal and the τ-relaxed primal are equivalent.*

Proof Let us assume (τ^*, ρ^*) be a feasible solution for the relaxed primal with $0 < \tau^*(C) < 1$ for some $C \in \mathcal{D}$. The edge constraint in Eq. (8) ensures that there are no incident edges on C selected by ρ^* (as ρ^* is integral). This violates the clique constraint in Eq. (9). Therefore, the feasible solutions of relaxed primal are integral. Hence the optimal solutions of the primal and the relaxed primal are identical. ■

The convex relaxation of the primal optimization problem is

$$\begin{array}{l} \min \mathcal{P}(\tau, \rho) \text{ subject to} \\ \tau \in [0, 1]^{\mathcal{D}},\ \rho \in [0, 1]^{\mathcal{E}}, \\ \text{Eq. (4), Eq. (5), Eq. (6), Eq. (7),} \\ \text{Eq. (8), Eq. (9), Eq. (10) and Eq. (11).} \end{array} \tag{14}$$

Dual Derivation. The dual variables are defined by :

– Set cover constraints in Eq. (4): $\gamma \in \mathbb{R}_+^V$.

– Running intersection property in Eq. (7): $\mu \in \mathbb{R}^V$.

– Edge constraints in Eq. (8): $\lambda \in \mathbb{R}_+^{\mathcal{E}} \times \mathbb{R}_+^{\mathcal{E}}$.

– Clique constraints in Eq. (9): $\eta \in \mathbb{R}_+^{\mathcal{D}}$.

Therefore, the dual space is represented by $(\gamma, \mu, \lambda, \eta)$.

Let $\mathcal{L}(\tau, \rho, \gamma, \mu, \lambda, \eta)$ be the Lagrangian relating the primal and dual variables. It is derived from the primal cost function defined in Eq. (3) along with the covering constraint, running intersection property, the edge and the clique constraints defined in Eq. (4), Eq. (7), Eq. (8) and Eq. (9) respectively. The Lagrangian can be computed from the dual variables $(\gamma, \mu, \lambda, \eta)$ as follows:

$$
\begin{aligned}
& \mathcal{L}(\tau, \rho, \gamma, \mu, \lambda, \eta) \\
= & \sum_{C \in \mathcal{D}} H(C)\tau(C) - \sum_{(C,D) \in \mathcal{E}} H(C \cap D)\rho(C, D) \\
+ & \sum_{i \in V} \gamma_i \Big(1 - \sum_{C \in \mathcal{D}} 1_{i \in C} \tau(C) \Big) + \sum_{i \in V} \mu_i \Big(\sum_{(C,D) \in \mathcal{E}} 1_{i \in (C \cap D)} \rho(C, D) - \sum_{C \in \mathcal{D}} 1_{i \in C} \tau(C) + 1 \Big) \\
+ & \sum_{C \in \mathcal{D}} \sum_{(C,D) \in \mathcal{E}} \lambda_{CD} \Big(\rho(C, D) - \tau(C) \Big) + \sum_{C \in \mathcal{D}} \eta_C \Big(\tau(C) - \sum_{(C,D) \in \mathcal{E}} \rho(C, D) \Big) \\
= & \sum_{C \in \mathcal{D}} \Big(H(C) - \sum_{i \in C} (\mu_i + \gamma_i) - \sum_{(C,D) \in \mathcal{E}} \lambda_{CD} + \eta_C \Big) \tau(C) \\
- & \sum_{(C,D) \in \mathcal{E}} \Big(H(C \cap D) - \sum_{i \in (C \cap D)} \mu_i - \lambda_{CD} - \lambda_{DC} + \eta_C + \eta_D \Big) \rho(C, D) + \sum_{i \in V} (\mu_i + \gamma_i),
\end{aligned}
\tag{15}
$$

with the following *dual constraints* on the Lagrange multipliers

$$
\begin{aligned}
& \forall i \in V, && \gamma_i \geq 0, \\
& \forall C \in \mathcal{D}, \quad \forall (C, D) \in \mathcal{E}, && \lambda_{CD} \geq 0, \\
& \forall C \in \mathcal{D}, && \eta_C \geq 0.
\end{aligned}
\tag{16}
$$

We can now derive a dual optimization problem with $\mathcal{Q}(\gamma, \mu, \lambda, \eta)$ represent the dual cost function, which can be derived from the Lagrangian in Eq. (15). We use the the number of edges constraint, the number of cliques constraint, tree constraint and hyperforest constraint given by Eq. (5), Eq. (6), Eq. (10) and Eq. (11) respectively in deriving the dual as follows:

$$
\begin{aligned}
& \mathcal{Q}(\gamma,\mu,\lambda,\eta) \\
= & \inf_{\substack{\tau\in[0,1]^{\mathcal{D}} \\ \sum_{C\in\mathcal{D}}\tau(C)=n-k \\ \tau\in \text{ hyperforest polytope of } (V,\mathcal{D})}} \left(H(C)-\sum_{i\in C}(\mu_i+\gamma_i)-\sum_{(C,D)\in\mathcal{E}}\lambda_{CD}+\eta_C\right)\tau(C) \\
- & \sup_{\substack{\rho\in[0,1]^{\mathcal{E}} \\ \sum_{(C,D)\in\mathcal{E}}\rho(C,D)=n-k-1 \\ \rho\in\text{forest polytope of } (\mathcal{D},\mathcal{E})}} \sum_{(C,D)\in\mathcal{E}} \left(H(C\cap D)-\sum_{i\in(C\cap D)}\mu_i-\lambda_{CD}-\lambda_{DC}+\eta_C+\eta_D\right)\rho(C,D) \\
+ & \sum_{i\in V}(\mu_i+\gamma_i). \qquad (17)
\end{aligned}
$$

It is decomposed in three parts defined in Eq. (19), Eq. (20) and Eq. (21) respectively :

$$
\mathcal{Q}(\gamma,\mu,\lambda,\eta) = q_1(\gamma,\mu,\lambda,\eta)+q_2(\gamma,\mu,\lambda,\eta)+q_3(\gamma,\mu,\lambda,\eta), \qquad (18)
$$

where

$$
q_1(\gamma,\mu,\lambda,\eta) = \inf_{\substack{\tau\in[0,1]^{\mathcal{D}} \\ \sum_{C\in\mathcal{D}}\tau(C)=n-k \\ \tau\in \text{ hyperforest polytope of } (V,\mathcal{D})}} \sum_{C\in\mathcal{D}} \left(H(C)-\sum_{i\in C}(\mu_i+\gamma_i)-\sum_{(C,D)\in\mathcal{E}}\lambda_{CD}+\eta_C\right)\tau(C) \qquad (19)
$$

$$
q_2(\gamma,\mu,\lambda,\eta) = - \sup_{\substack{\rho\in[0,1]^{\mathcal{E}} \\ \sum_{(C,D)\in\mathcal{E}}\rho(C,D)=n-k-1 \\ \rho\in \text{ forest polytope of } (\mathcal{D},\mathcal{E})}} \sum_{(C,D)\in\mathcal{E}} \left(H(C\cap D)-\sum_{i\in(C\cap D)}\mu_i-\lambda_{CD}-\lambda_{DC}+\eta_C+\eta_D\right)\rho(C,D). \qquad (20)
$$

$$
q_3(\gamma,\mu,\lambda,\eta) = \sum_{i\in V}(\mu_i+\gamma_i). \qquad (21)
$$

Therefore, the dual optimization problem using the dual cost function defined in Eq. (17) and the dual constraints defined in Eq. (16) is given by

$$
\max \mathcal{Q}(\gamma,\mu,\lambda,\eta) \text{ subject to } \begin{cases} \forall i\in V,\gamma_i\geq 0, \\ \forall C\in\mathcal{D},\forall(C,D)\in\mathcal{E},\lambda_{CD}\geq 0, \\ \forall C\in\mathcal{D},\eta_C\geq 0. \end{cases} \qquad (22)
$$

Proposition 2 *If $k=1$, the convex relaxation in Eq. (14) is equivalent to Eq. (12).*

Proof If $k=1$, all the cliques in the clique space contain only 2 vertices, i.e., $\forall C\in\mathcal{D},|C|=2$ and the number of elements in the feasible edges is only 1, i.e., $\forall(C,D)\in\mathcal{E},|C\cap D|=1$.

Solving the convex relaxation defined in Eq. (14) is equivalent to solving the dual defined in Eq. (22). On solving the dual variables, the optimal dual solution is given by

$$
\begin{aligned}
& \forall i\in V,\mu_i = H(\{i\}), \\
& \forall i\in V,\gamma_i = 0, \\
& \forall C\in\mathcal{D},\forall(C,D)\in\mathcal{E},\lambda_{CD}=0, \\
& \forall C\in\mathcal{D},\eta_C=0,
\end{aligned} \qquad (23)
$$

where $H(\{i\}) = -\hat{p}_i(x_i)\log(\hat{p}_i(x_i))$.

The optimal solution to the dual problem is given by

$$\begin{aligned}\mathcal{Q}^*(\gamma,\mu,\lambda,\eta) &= \inf_{\substack{\tau\in[0,1]^{\mathcal{D}} \\ \sum_{C\in\mathcal{D}}\tau(C)=n-k \\ \tau\in \text{ hyperforest polytope of } (V,\mathcal{D})}} \sum_{C\in\mathcal{D}}\Big(H(C)-\sum_{i\in C}H(\{i\})\Big)\tau(C)+\sum_{i\in V}H(\{i\}) \\ &= \inf_{\substack{\tau\in[0,1]^{\mathcal{D}} \\ \sum_{C\in\mathcal{D}}\tau(C)=n-k \\ \tau\in \text{ hyperforest polytope of } (V,\mathcal{D})}} -I(C).\tau(C)+\sum_{i\in V}H(\{i\}), \qquad (24)\end{aligned}$$

where $\forall C\in\mathcal{D}, I(C)=\sum_{i\in C}H(\{i\})-H(C)$, which defines the mutual information of the elements in the clique, i.e., an edge if $k=1$. The constraints in Eq. (24) define a spanning tree polytope [4] and the optimal solution is a maximal information spanning tree, which is given by Chow-Liu trees [1]. They also form the optimal solution to the non-convex primal optimization defined in Eq. (12). ■

Approximate Greedy Primal Solution. We describe an algorithm to project from the average of a sequence of fractional primary infeasible solutions, estimated during the iterations of projective supergradient, to an integral primary feasible solution. "AddClique" adds all the edges of a clique to the adjacency matrix. "checkGraphDecomposability" checks if the maximal cardinality search is a perfect elimination ordering. For decomposable graphs the maximal cardinality search yields a perfect elimination ordering [2]. We refer to this as *decomposability test* in this paper. "getNumberConnectedComponents" gives the number of connected components in the graph using breadth-first search. Note that the projection only uses the average clique selection function, $\hat{\tau}_m$, to obtain the primary feasible solutions, τ_m. The corresponding edge selection, ρ_m, can be estimated from clique selection, τ_m, by selecting the edges between consecutive cliques of the perfect sequence of selected cliques [3]. The time complexity of the projection algorithm is $O(n^{k+2})$. This is due to decomposability test with run time complexity $O(n^{k+1})$, that is performed on adding $O(n)$ cliques.

Algorithm 1 Approximate Greedy Primal Solution

Input: primal infeasible sequence τ^t of *Projected Supergradient algorithm*, treewidth k, number of Vertices n, set of cliques $\mathcal{D}$ and integer m such that $0<m\leq T$
Output: approximate discrete primal feasible solution τ_m after m iterations of Algorithm 1
Initialize Adjacency Matrix $Adj = zeros(n,n)$, $\hat{\tau}_m = \frac{1}{m}\sum_{t=0}^{m}\tau^t$ and $\tau_m = zeros(length(\hat{\tau}_m))$
$order$ = Sorted indices in the descending order $\hat{\tau}_m$
repeat
  Initialize $decomposable = false$, $treewidth = 0$, $numConnectedComponents = 0$, $i = 1$
  update $TestAdj$ = AddClique(Adj, $\mathcal{D}(order(i))$)
  update $[decomposable, treewidth]$ = checkGraphDecomposability($TestAdj$)
  if $decomposable$ = true **and** $treewidth \leq k$ **then**
    update $Adj = TestAdj$
    update $\tau_m(Order(i)) = 1$
  end if
  $[numConnectedComponents]$ = getNumberConnectedComponents($TestAdj$)
  update $i = i+1$
until $decomposable = true$, $treewidth = k$, $numConnectedComponents = 1$, i = length($order$)

Generating Decomposable Covariance Matrices. In order to generate a covariance matrix of a multivariate Gaussian representing a graph, we generate a random matrix and project this onto the graph to get a covariance matrix which represents the graph. If the projection is performed onto decomposable graphs, the resultant covariance matrices are called *decomposable covariance matrices*.

A random positive definite covariance matrix, Σ' is generated as follows:

$$\Sigma' = \frac{d}{d'} Z Z^\top + (1 - \frac{d}{d'}) I, \tag{25}$$

where Z is a random matrix of dimensions $n \times d'$, I is the n-dimensional identity matrix and d is a parameter to determine the correlations between the nodes of the graph, which takes values in $\{0, d'\}$. In our experiments, we choose d' to be 128. We have tight correlations between the nodes with higher values of d.

The random positive definite covariance matrix, Σ', which is generated using Eq. (25) is projected onto a decomposable graph G as follows:

$$(\Sigma)^{-1} = \sum_{C \in \mathcal{C}(G)} [(\Sigma'_C)^{-1}]_n - \sum_{(C,D) \in \mathcal{T}(G)} [(\Sigma'_{C \cap D})^{-1}]_n, \tag{26}$$

where the operator $[(\Sigma'_X)^{-1}]_n$ gives an $n \times n$ matrix whose columns and rows representing the set $X \subseteq V$ are filled by $(\Sigma'_X)^{-1}$ and the rest of the elements of the matrix are filled with *zeroes*. The matrix, Σ, thus generated represents the covariance matrix of a multivariate Gaussian on a decomposable graph, G.

The projection ensures the following relationship between the random positive definite matrix, Σ' and the projected covariance matrix Σ:

$$\begin{aligned} \Sigma(i,j) &= \Sigma'(i,j) \text{ if } A(i,j) = 1 \text{ or } i = j, \\ \Sigma^{-1}(i,j) &= 0 \text{ if } A(i,j) = 0. \end{aligned} \tag{27}$$

where A is the adjacency matrix of the decomposable graph G onto which Σ' was projected.

The entropy of a multivariate Gaussian with a covariance matrix, Σ, is given by $\frac{1}{2}\log(2\pi e)^n|\Sigma|$, where $|\Sigma|$ denotes the determinant of the covariance matrix. However, for Gaussian distribution that is factored in $G \in \mathcal{G}$:

$$|\Sigma| = \frac{\prod_{C \in \mathcal{C}(G)} |\Sigma_C|}{\prod_{((C,D) \in \mathcal{T}(G)} |\Sigma_{C \cap D}|}, \tag{28}$$

where Σ_X is the sub-matrix of the covariance matrix whose rows and columns belong to the set $X \subseteq V$. Therefore, for any multivariate decomposable Gaussian graphical model, G:

$$\begin{aligned} H(G) &= \frac{1}{2}\log((2\pi e)^n|\Sigma|) \\ &= \frac{1}{2}(\sum_{C \in \mathcal{C}(G)} \log((2\pi e)^n|\Sigma_C|) - \sum_{(C,D) \in \mathcal{T}(G)} \log((2\pi e)^n|\Sigma_{C \cap D}|)) \\ &= \sum_{C \in \mathcal{C}(G)} H(C) - \sum_{(C,D) \in \mathcal{T}(G)} H(C \cap D). \end{aligned} \tag{29}$$

Note that the entropy of any graph, G, is independent of the mean of the normal distribution, hence we consider only the covariance matrix.

References

[1] C. I. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. Inf. Theory*, 14:462–467, 1968.

[2] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. North Holland, 2004.

[3] S. L. Lauritzen. *Graphical Models*. Oxford University Press, Oxford, 1996.

[4] A. Schrijver. *Combinatorial optimization: Polyhedra and efficiency*. Springer, 2004.