Column generation for exact Bayesian network learning: Work in Progress

James Cussens, University of York

CoCoMile 2012

Encoding graphs with binary 'family' variables

- Suppose there are $|V|$ fully-observed discrete variables in some dataset. We want to learn a BN (with $|V|$ vertices) from this data.

Encoding graphs with binary 'family' variables

- Suppose there are $|V|$ fully-observed discrete variables in some dataset. We want to learn a BN (with $|V|$ vertices) from this data.
- Can encode any graph by creating a binary variable $I(W \rightarrow u)$ for each BN variable $u \in V$ and each candidate parent set W.
- Each $I(W \rightarrow u)$ has a *local score* $c'(u, W)$.

Encoding graphs with binary 'family' variables

- Suppose there are $|V|$ fully-observed discrete variables in some dataset. We want to learn a BN (with $|V|$ vertices) from this data.
- Can encode any graph by creating a binary variable $I(W \rightarrow u)$ for each BN variable $u \in V$ and each candidate parent set W.
- Each $I(W \rightarrow u)$ has a *local score* $c'(u, W)$.
- Big problem already: That could be a lot of $I(W \rightarrow u)$ variables.

Encoding graphs with binary 'family' variables

- Suppose there are $|V|$ fully-observed discrete variables in some dataset. We want to learn a BN (with $|V|$ vertices) from this data.
- Can encode any graph by creating a binary variable $I(W \rightarrow u)$ for each BN variable $u \in V$ and each candidate parent set W.
- Each $I(W \rightarrow u)$ has a *local score* $c'(u, W)$.
- Big problem already: That could be a lot of $I(W \rightarrow u)$ variables.
- What might save us: Only $|V|$ of these $I(W \rightarrow u)$ variables will be non-zero in any solution.

BN learning as constrained optimisation

$$\begin{array}{l}\text{Instantiate the } I(W \to u) \text{ to maximise:}\\ \sum_{u,W} c'(u,W) I(W \to u)\\ \text{subject to the } I(W \to u) \text{ representing a DAG.}\end{array} \tag{1}$$

Set $c(u,W) = -c'(u,W)$ and consider

$$\begin{array}{l}\text{Instantiate the } I(W \to u) \text{ to minimise:}\\ \sum_{u,W} c(u,W) I(W \to u)\\ \text{subject to the } I(W \to u) \text{ representing a DAG.}\end{array}$$

An integer linear programming approach

$$\forall u \in V : \sum_{W} I(W \to u) = 1 \tag{2}$$

$$\text{Where } C \subseteq V : \sum_{u \in C} \sum_{W : W \cap C = \emptyset} I(W \to u) \geq 1 \tag{3}$$

Cluster constraints (3) added 'on the fly' as cutting planes.
Introduced in [1].

Slack variable representation

$$\sum_{u \in C} \sum_{W : W \cap C = \emptyset} I(W \to u) \geq 1 \tag{4}$$

$$-w_C + \sum_{u \in C} \sum_{W : W \cap C = \emptyset} I(W \to u) = 1 \tag{5}$$

where $w_C \geq 0$. Let n be the number of $I(W \to u)$ variables then can represent the feasible region using:

$$\mathbf{Ax} = \mathbf{b} \tag{6}$$

where $\mathbf{A}$ is a $(|V| + |\mathcal{C}|) \times (n + |\mathcal{C}|)$ matrix.

Dictionary representation of an empty graph solution

Let ζ be the variable for the objective function and $\bar{\zeta}$ a constant which is the score of the empty graph.

$$
\begin{aligned}
\zeta &= \bar{\zeta} + \sum_{u,W:W\neq\emptyset} [c(u,W) - c(u,\emptyset)] I(W \to u) && (7) \\
I(\emptyset \to u) &= 1 - \sum_{u,W:W\neq\emptyset} I(W \to u) && (8) \\
w_C &= |C| - 1 - \sum_{u\in C} \sum_{W:W\cap C\neq\emptyset} I(W \to u) && (9)
\end{aligned}
$$

Basic and non-basic variables

- LHS: The $I(\emptyset \to u)$ variables and the slack variables are (currently) *basic* and may be positive.
- RHS: The $I(W \to u)$ variables for $W \neq \emptyset$ are (currently) non-basic and have value zero.

$$\zeta = \bar{\zeta} + \sum_{u,W:W\neq\emptyset} [c(u,W) - c(u,\emptyset)]I(W \to u) \quad (10)$$

$$I(\emptyset \to u) = 1 - \sum_{u,W:W\neq\emptyset} I(W \to u) \quad (11)$$

$$w_C = |C| - 1 - \sum_{u\in C} \sum_{W:W\cap C\neq\emptyset} I(W \to u) \quad (12)$$

Choose a non-basic variable with *negative reduced cost* to bring into the basis.

Column generation

- Column generation = variable generation
- *It is not necessary to explicitly represent non-basic variables*.
- Only create them when they are to enter the basis.

Computing reduced costs

For any basis:

- $\mathbf{x} = (\mathbf{x_B}, \mathbf{x_D})$
- $\mathbf{c} = (\mathbf{c_B}, \mathbf{c_D})$
- $\mathbf{B}$ is the (square) submatrix of the original $\mathbf{A}$ matrix whose columns correspond to $\mathbf{x_B}$. $\mathbf{D}$ is the (non-square) matrix formed from the remaining (non-basic) columns.
- Compute dual values: $\lambda^T = \mathbf{c_B}^T \mathbf{B}^{-1}$.
- Compute reduced costs: $\mathbf{r_D} = \mathbf{c_D} - \lambda^T \mathbf{D}$.

What we need

- To compute the reduced cost of a potential new variable we need:
- its objective coefficient value and
- its coefficient for each original linear constraint (= row of $\mathbf{A}$).
- (Note that a row is added to $\mathbf{A}$ each time a cutting plane is added.)

An ILP for variable creation

- A new variable $I(W \rightarrow u)$ is determined by a choice of the child u and also the parents W.
- Let $I_{\mathrm{ch}}(u)$ indicate that u is chosen as the child and let $I_{\mathrm{pa}}(u)$ represent that u is chosen as a parent.

$$\sum_{u \in V} I_{\mathrm{ch}}(u) = 1 \tag{13}$$

$$\forall u \in V : I_{\mathrm{ch}}(u) + I_{\mathrm{pa}}(u) \leq 1 \tag{14}$$

Cluster constraints

Create a variable x_C for each $C \in \mathcal{C}$ indicating whether the new variable is involved in the constraint for C. We have:

$$
\begin{aligned}
x_C &\geq \sum_{u \in C} I_{\mathrm{ch}}(u) - \sum_{u \in C} I_{\mathrm{pa}}(u) && (15) \\
\sum_{u \in C} I_{\mathrm{ch}}(u) &\geq x_C && (16) \\
1 - \sum_{u \in C} I_{\mathrm{pa}}(u) &\geq x_C && (17)
\end{aligned}
$$

Computing the reduced cost of the new variable

- Let λ_C be the dual value corresponding to the constraint for cluster C.
- Let λ_u be the dual value for the convexity constraint (2) for variable u.

The reduced cost for a new variable $I(W \rightarrow u)$ is then:

$$c(u, W) - \sum_{u} \lambda_u I_{\mathrm{ch}}(u) - \sum_{C \in \mathcal{C}} \lambda_C x_C \qquad (18)$$

Hmmm, how to get $c(u, W)$?

Proposed strategy

- View $c(u, W)$ as a real-valued variable.

Proposed strategy

- View $c(u, W)$ as a real-valued variable.
- Or rather a variable which is a lower bound on this value.
- This will lead to over-optimistic generation of new variables.
- Once a new variable is proposed probably worth the effort to consult the data to compute $c(u, W)$ exactly.

Tommi Jaakkola, David Sontag, Amir Globerson, and Marina Meila.
Learning Bayesian network structure using LP relaxations.
In *Proceedings of 13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, volume 9, pages 358–365, 2010.
Journal of Machine Learning Research Workshop and Conference Proceedings.