Networks

The Sparse Candidate Algorithm

Outline

- Objective for Bayesian structure learning task
- The Sparse Candidate Algorithm
- Mutual information and KL-divergence
- Efficiency of the algorithm

Structure Learning task objective

- We wish to maximize the following score

constant (depends on D)

$$\text{score}(G:D) = \log \Pr(G \mid D)$$
$$= \log \Pr(D \mid G) + \log \Pr(G) + C$$

log probability of data D given graph G

log prior probability of graph G

- This score can be expressed as a sum of easily computable scores of individual vertices because of the following:
  - factorization of the likelihood via the network
  - parameter independence
  - conjugate priors allowing for closed-form expressions

$$\text{score}(G:D) = \sum_i Score(X_i, Parents(X_i) : D)$$

Bayesian Network Search: The *Sparse Candidate* Algorithm [Friedman et al., *UAI* 1999]

Graph with no edges

Given: data set D, initial network B_0, parameter k

Loop for $n = 1, 2, \ldots$ until convergence

Restrict

Based on D and B_{n-1}, select for each variable X_i a set C_i^n ($|C_i^n| \leq k$) of candidate parents.
This defines a directed graph $H_n = (\mathcal{X}, E)$, where $E = \{X_j \to X_i | \forall i, j, X_j \in C_i^n\}$.
(Note that H_n is usually cyclic.)

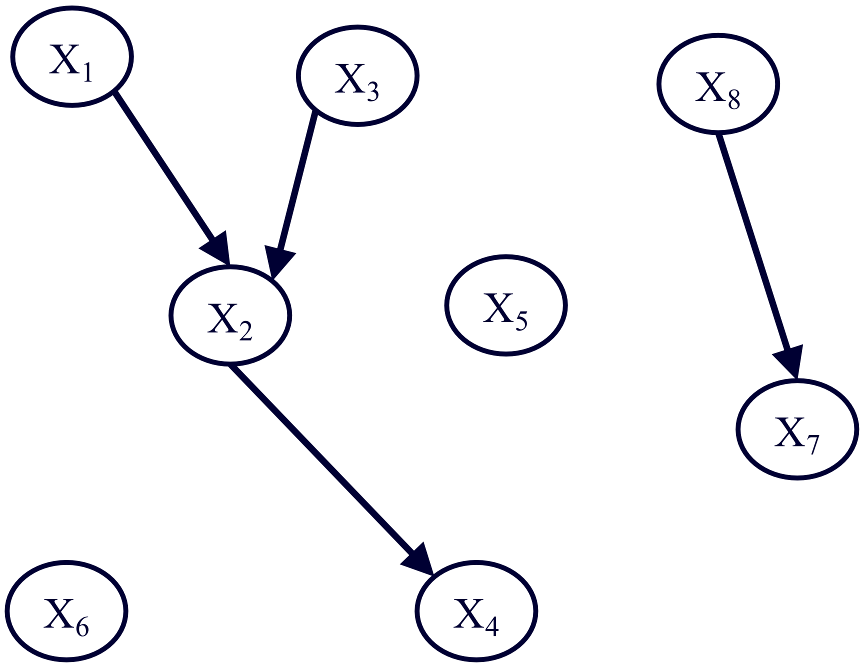
Maximize

Find network $B_n = \langle G_n, \Theta_n \rangle$ maximizing Score$(B_n \mid D)$ among networks that satisfy $G_n \subset H_n$ (i.e., $\forall X_i, \mathbf{Pa}^{G_n}(X_i) \subseteq C_i^n$,).

acyclic

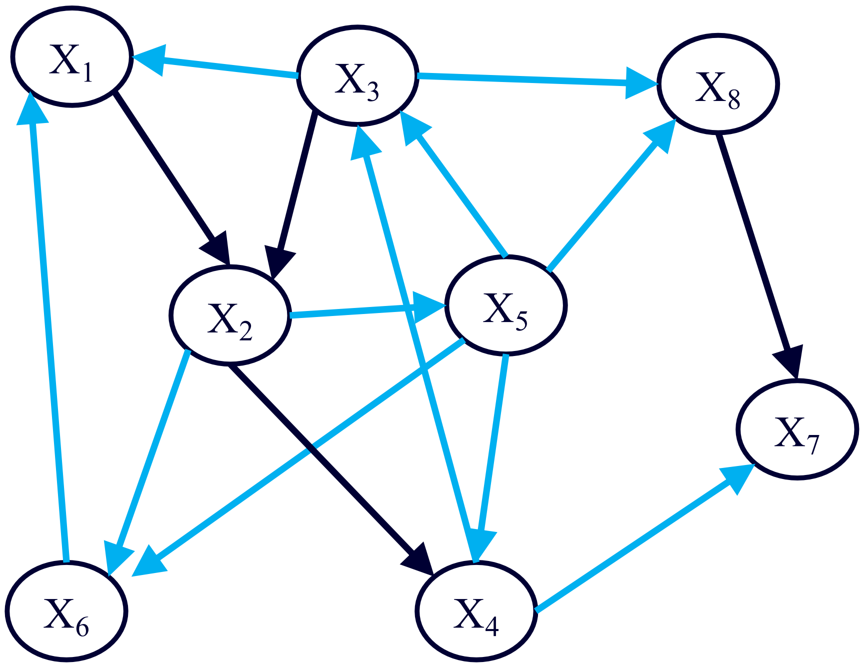
Return B_n

Sparse Candidate – Current Network



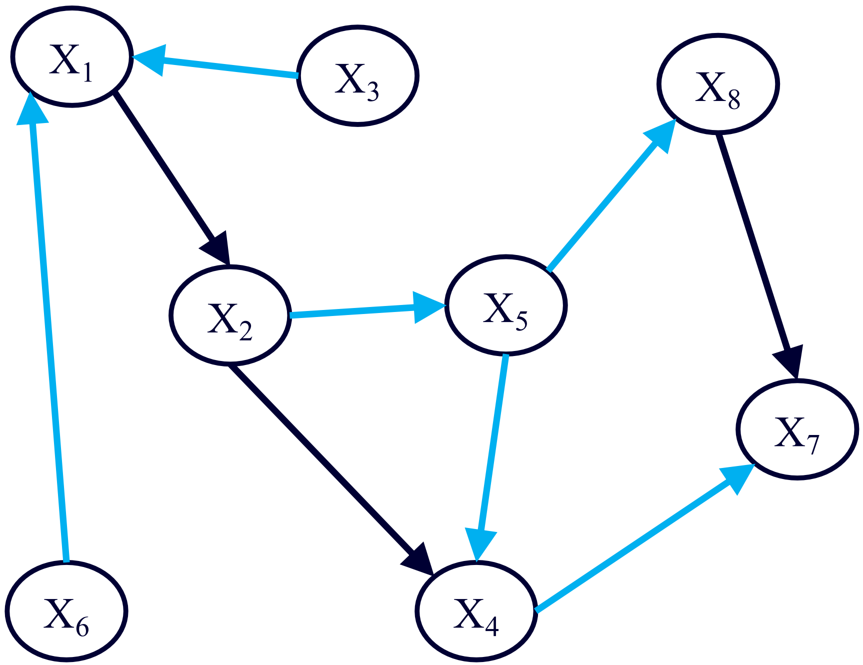
Each iteration, n, starts with network structure, B_{n-1} found in the previous iteration

Sparse Candidate – Restrict



Up to k (2 in this example) candidate parents are selected for each variable

Sparse Candidate – Maximize



From the set of all candidate parents, a high-scoring Bayesian network if selected

The Restrict Step In Sparse Candidate

- to identify candidate parents in the <u>first</u> iteration, can compute the *mutual information* between pairs of variables

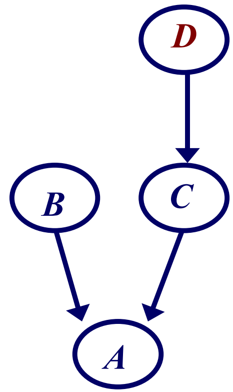
$$I(X,Y) = \sum_{x,y} \hat{P}(x,y) \log \frac{\hat{P}(x,y)}{\hat{P}(x)\hat{P}(y)}$$

- where $\hat{P}$ denotes the probabilities estimated from the data set

How dependent two RVs are

The Restrict Step In Sparse Candidate

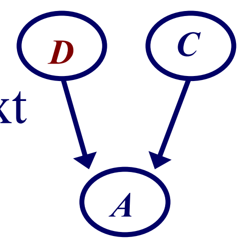
- suppose true network structure is:



- We're selecting two candidate parents for A and $I(A,C) > I(A,D) > I(A,B)$

if $k = 2$

- the candidate parents for A would then be C and D; how could we get B as a candidate parent on the next iteration?



The Restrict Step In Sparse Candidate

- *Kullback-Leibler* (KL) *divergence* provides a distance measure between two distributions, *P* and *Q*

$$D_{KL}(P(X) \| Q(X)) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$

- mutual information can be thought of as the KL divergence between the distributions

$$\hat{P}(X,Y)$$

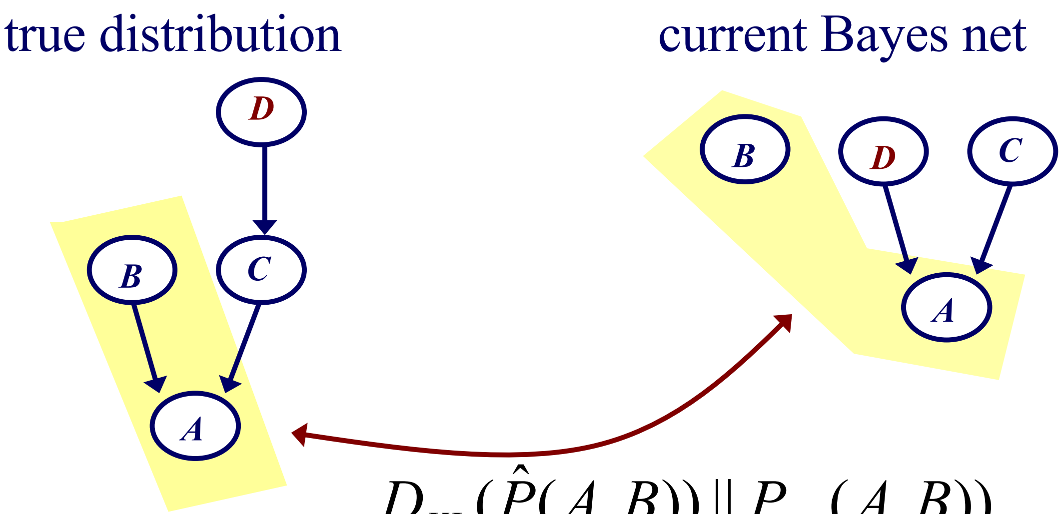
and

$$Q(X,Y) = \hat{P}(X)\,\hat{P}(Y)$$ (assumes *X* and *Y* are independent)

The Restrict Step In Sparse Candidate

- we can use KL to assess the discrepancy between the network's estimate $P_{net}(X, Y)$ and the empirical estimate

$$M(X,Y) = D_{KL}(\hat{P}(X,Y)) \,||\, P_{net}(X,Y))$$



$$D_{KL}(\hat{P}(A,B)) \,||\, P_{net}(A,B))$$

The Restrict Step in Sparse Candidate

Input:

- Data set $D = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$,
- A network B_n,
- a score
- parameter k.

Output: For each variable X_i a set of candidate parents C_i of size k.

Loop for each X_i $i = 1, \ldots, n$

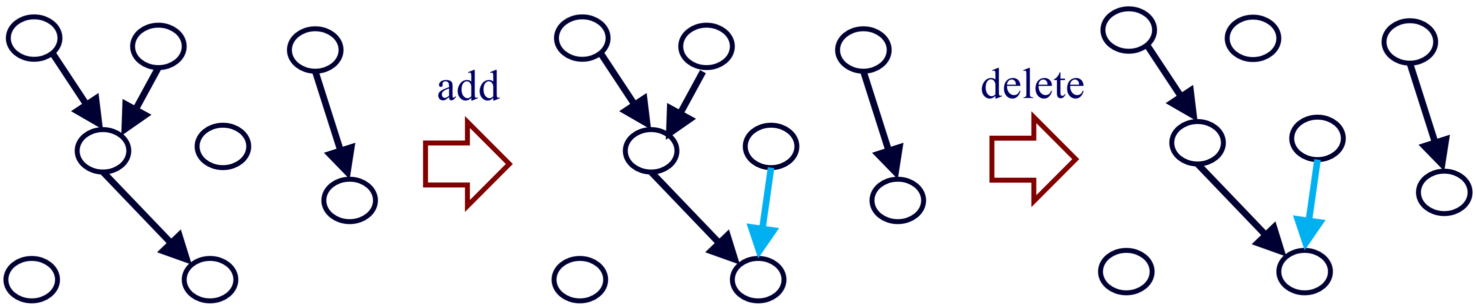
- Calculate $\mathrm{M}(X_i, X_j)$ for all $X_j \neq X_i$ such that $X_j \notin \mathbf{Pa}(X_i)$
- Choose $x_1, \ldots, x_{k-l}$ with highest ranking , where $l = |\mathbf{Pa}(X_i)|$.
- Set $C_i = \underbrace{\mathbf{Pa}(X_i)} \cup \{x_1, \ldots, x_{k-l}\}$

Return $\{C_i\}$

important to ensure monotonic improvement

The Maximize Step in Sparse Candidate

- hill-climbing search with *add-edge*, *delete-edge*, *reverse-edge* operators
- test to ensure that cycles aren't introduced into the graph



Efficiency of Sparse Candidate

	possible parent sets for each node	changes scored on first iteration of search	changes scored on subsequent iterations
ordinary greedy search	$O(2^n)$	$O(n^2)$	$O(n)$
greedy search w/at most *k* parents	$O\left(\binom{n}{k}\right)$	$O(n^2)$	$O(n)$
Sparse Candidate	$O(2^k)$	$O(kn)$	$O(k)$

Summary

- Sparse candidate algorithm is a heuristic algorithm for finding a Bayesian network with maximum score
- It uses a “restrict” step to limit the set of edges that are considered during a second “maximize” hill-climbing step
- The restrict step uses mutual information and KL-divergence to select candidate edges