Bayesian optimization of the PC algorithm for learning Gaussian Bayesian networks

Irene Córdoba, Eduardo C. Garrido-Merchán, Daniel Hernández-Lobato, Concha Bielza, Pedro Larrañaga.

Presented by: Eduardo C. Garrido-Merchán.

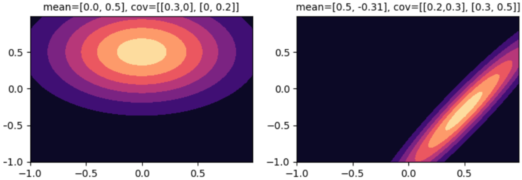
Universidad Politécnica de Madrid, Universidad Autónoma de Madrid.
October 24th.

Index

- Introduction.
- The PC algorithm.
- Evaluation of the learned structure.
- Experiments.
- Conclusions and further work.
- Questions.

Introduction

- **Bayesian Networks (BN)** serves as a **compact representation** between variables in a domain.
- **Conditional Independences** are encoded by missing edges in a directed acyclic graph **(DAG)**.
- They yield a **Modular Factorization of the Joint Probability Distribution** over the data.
- **Gaussian Bayesian Networks are Gaussian Multivariate Distributions**.



Structure and Parameter Learning

- **Gaussian Bayesian Network** fitting to the data includes:
- **Structure Learning**: Recovering the graph structure. (Combinatorial Space Search)
- **Parameter Learning**: Fitting the numerical quantities of the model.
- We will focus on **Structure Learning**, in particular, on **optimizing the PC algorithm**.
- The **PC algorithm** determines absent edges in the DAG, using *statistical tests* and a *significance level*.

The combinatorial space

- Depending on the nodes number, reconstructing a BN involves searching in a **huge space**.

d	Number of DAGs with d nodes
1	1
2	3
3	25
4	543
5	29281
6	3781503
7	1138779265
8	783702329343
9	1213442454842881
10	4175098976430598143
11	31603459396418917607425
12	521939651343829405020504063
13	18676600744432035186664816926721
14	1439428141044398334941790719839535103
15	237725265553410354992180218286376719253505
16	83756670773733320287699303047996412235223138303
17	62707921196923889899446452602494921906963551482675201
18	99421195322159515895228914592354524516555026878588305014783
19	332771901227107591736177573311261125883583076258421902583546773505

- The number of DAGs depending on the number d of nodes, taken from `http://oeis.org/A003024` [OEIS Foundation Inc., 2017].
- We do not know gradients to search in this space and the evaluation is costly...
- **Bayesian Optimization combined with a well thought objective** arises as an **ideal solution**!

Gaussian Bayesian Networks and the PC algorithm

- We want to *reconstruct* the skeleton of a **GBN** from data optimizing the *PC algorithm*.
- The *PC algorithm* first estimates the skeleton and then orientates it.
- Starts with the *complete graph* and in a *backward stepwise elimination* fashion it **removes edges**.
- For every node X_i from the graph, it looks every neighbor of it X_j and test $X_i \perp\!\!\!\perp X_j | \mathbf{X}_C$.
- If the test succeds, the edge is removed.
- If the DAG is big, the PC algorithm is computationally very expensive and **in order to select its hyperparameters, BO is a good solution**.

The PC algorithm

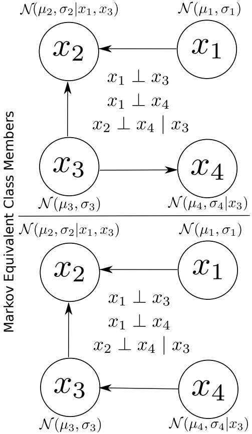
Algorithm 1 The PC algorithm in its population version

Input: Conditional independence information about $\boldsymbol{X} = (X_1, \ldots, X_p)$
Output: Skeleton of the Gaussian Bayesian network

$G \leftarrow$ complete undirected graph on $\{1, \ldots, p\}$
$l \leftarrow -1$
repeat
  $l \leftarrow l + 1$
  repeat
    Select i such that $(i, j) \in E$ and $|\mathrm{ne}(i) \setminus \{j\}| \geq l$
    repeat
      Choose new $C \subseteq \mathrm{ne}(i) \setminus \{j\}$ with $|C| = l$
      if $X_i \perp\!\!\!\perp X_j \mid \boldsymbol{X}_C$ **then**
        $E \leftarrow E \setminus \{(i, j), (j, i)\}$
      end if
    until (i, j) has been deleted or all neighbor subsets of size l have been tested
  until All $(i, j) \in E$ such that $|\mathrm{ne}(i) \setminus \{j\}| \geq l$ have been tested
until $|\mathrm{ne}(i) \setminus \{j\}| < l$ for all $(i, j) \in E$

Optimizing the PC algorithm: Evaluating the Quality of the Learned Structure

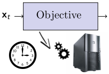
- *Different size networks* and *number of neighbours*.
- Retrieve **optimum significance level and statistical test for the PC algorithm**.
- From a number of samples from a **BN**, we want to obtain through the PC algorithm a **Markov equivalent BN**.



We achieve this by optimizing the **normalized Structural Hamming Distance metric**: $\frac{SHD}{p(p-1)/2}$

An Optimization Problem

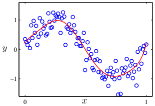
- The normalized SHD objective function is very expensive to evaluate.



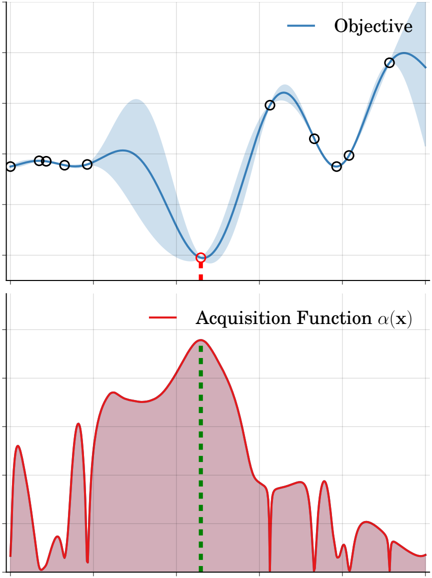
- We do not have gradients: The objective is a black-box.

- The evaluation can be noisy.

$$y = f(\mathbf{x}) + \epsilon$$

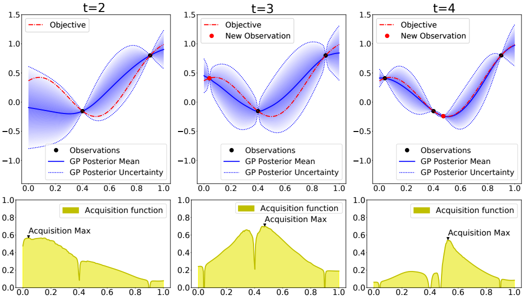


Bayesian Optimization



1. Get initial sample.
2. Fit a model to the data:
$$p(y|\mathbf{x}, \mathcal{D}_n) \,.$$
3. Select data collection strategy:
$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)] \,.$$
4. Optimize acquisition function $\alpha(\mathbf{x})$.
5. Collect data and update model.
6. Repeat!

Bayesian Optimization



for $t = 1, 2, 3, \ldots, max\_steps$ **do**

1: Find the next point to evaluate by optimizing the acquisition function:

$$\mathbf{x}_t = \arg\max_{\mathbf{x}} \quad \alpha(\mathbf{x}|\mathcal{D}_{1:t-1}).$$

2: Evaluate the black-box objective $f(\cdot)$ at $\mathbf{x}_t$: $y_t = f(\mathbf{x}_t) + \epsilon_t$.

3: Augment the observed data $\mathcal{D}_{1:t} = \mathcal{D}_{1:t-1} \bigcup \{\mathbf{x}_t, y_t\}$.

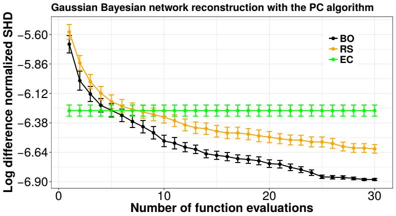
4: Update the Gaussian process model using $\mathcal{D}_{1:t}$.

end

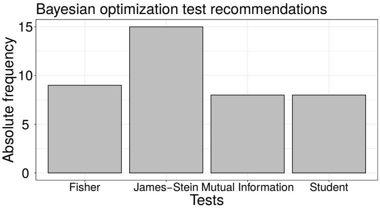
Result: Optimize the mean of the Gaussian process to find the solution.

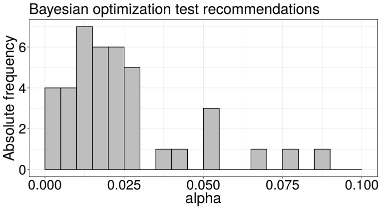
Experiments

- **Hyperparameters:** $\alpha \in [-5, 1]$ log space.
- **Statistical test:** Fisher Z transform, Student's T, χ^2, James-Stein.
- We used *Spearmint* for BO and *bnlearn* for the PC algorithm.
- We used **PES** over a set of 32 different **GBN** learning scenarios.
- We create 40 replicas of each experiment.



Experiments





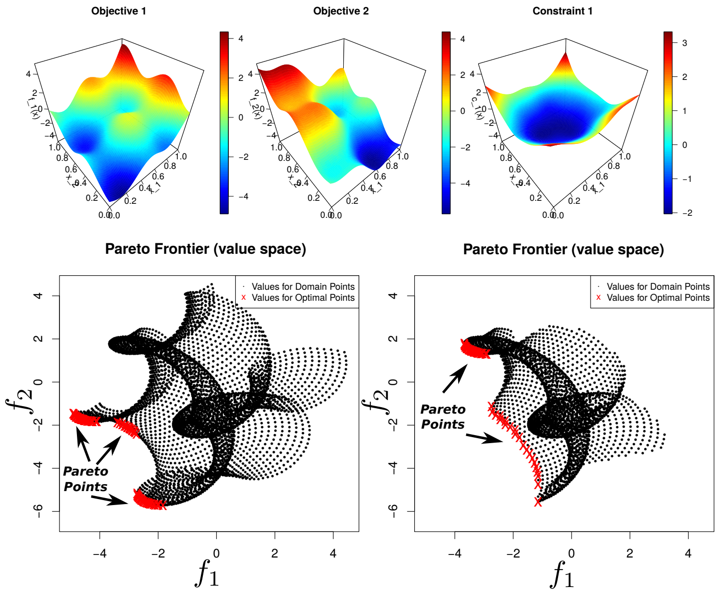
Conclusions

- We used **BO** for selecting the optimal parameters of the **PC algorithm** for structure recovery in **BNs**.
- Expert suggestion is *outperformed* in a small number of iterations. surprising result of the *statistical test*.
- We have shown the importance over the selection of the *statistical test* in the considered scenarios.
- Related literature only consider the importance over the *significance level*.

Further work

- **Mix Bayesian Optimization with Regression** to optimize both Structure and Parameter learning.
- Optimize **another objectives**: Measures that do not rely on the true graph structure like **network scores**.
- Consider **constraints**: BN shape, number of neighbours...
- Consider **other Graphical Models** apart from BNs.
- Consider **real data** scenarios.
- Consider **other BO or optimization techniques** for this problem.
- Apply **Bayesian Combinatorial optimization** techniques.

Further work



Questions

Thank you for your attention!