Laboratory of biological data mining

Giacomo Fantoni
Telegram: @GiacomoFantoni
Github: https://github.com/giacThePhantom/LaboratoryOfBiologicalDataMining

November 20, 2021

Contents

Chapter 1

Introduction

Data mining and data analysis are not the same thing: the former deals with data that pre-exists with respect to the research question, whereas the latter deals with data that is generated and collected in order to answer some research question.

1.1 Type of data

Data is canonically represented via tables. It implies the existence of a mapping between facts of the world and symbols. The measurement is a mapping between facts in the world and elements of sets equipped with some mathematical structure.

1.1.1 Categorical data

In categorical data the set is finite and it has no structure. The only operation that can be done on it is to distinguish the elements of the set.

1.1.2 Ordered categorical data

In ordered categorical data the set has an order: its element are in a mathematical relationship with the properties of an order:

- Reflexivity.
- Antisymmetry.
- Transitivity.

1.1.3 Discrete data

In discrete data the set is a subset of the natural numbers. Operations permitted are sum and difference.

1.1.4 Continuous data

In continuous data the set is an interval of the real numbers. Some scales have an absolute zero. The operations permitted are sum and difference. Division makes sense only if the scale is absolute.

1.2 Metadata

Metadata is data about the data. For example it contains informations about its origin, the method, time, format, source and owner.

1.3 Interventional data and observational data

Interventional data and observational data are distinguished on the basis of the intervention on the system. The former is generated by experimental procedures and the latter by pure observation/

Chapter 2

PC algorithm

2.1 Estimating high-dimensional directed acyclic graphs with the PC-algorithm

2.1.1 Introduction

Graphical models are a popular probabilistic tool to analyse and visualize conditional independence relationships between random variables. The major building blocks of these models are nodes (the random variables) and edges (conditional dependence). The structure of conditional independence among the random variables can be explored using the Markov properties. The estimation of a DAG from data is difficult due to the enormous size of the space of DAGs. The PC-algorithm is an alternative to greedy or structurally restricted approaches. It starts from a complete, undirected graph and deletes recursively edges based on conditional independence decisions. This yields an undirected graph that can be partially directed and further extended to represent the underlying DAG. This algorithm runs in the worst case in exponential time, but if the true underlying DAG is sparse, it is reduced to a polynomial runtime. Here there is a focus on estimating the equivalence class and the skeleton of DAGs corresopnding to multivariate Gaussian distributions in high-dimensional context, or where the number of nodes p may be much larger than the sample size n.

2.1.2 Finding the equivalence class of a DAG

Let $G = (V, E)$ a graph. The set of vertices V corresponds to the components of a random vector $\vec{X} \in \mathbb{R}^p$. A probability distribution P on $\mathbb{R}^p$ is said to be faithful with respect to G if conditional independences of the distribution can be inferred from d-separation in G. Consider a random vector $\vec{X} \sim P$. Faithfulness of P with respect to G means that for any $i, j \in V$ with $i \neq j$ and any set $s \subseteq V$:

$$\vec{X}^{(i)} \wedge \vec{X}^{(j)} \text{ are conditionally independent given } \{\vec{X}^{(r)} | r \in s\} \Leftrightarrow i \wedge y \text{ are d-separated by } s$$

Faithfulness is ruling out some classes of probability distributions. The skeleton of a DAG is the undirected graph obtained from G by substituting undirected edges for directed ones. A v-structure in a DAG is an ordered triple of node such that $(i, j) \in G \wedge (k, j) \in G \wedge (i, k) \notin G$. Two DAGs are equivalent is and only is they have the same skeleton and v-structures. If P is faithful with

respect to a DAG G there is an edge between node i and j in the skeleton of DAG G is and only if $\forall s \subseteq V \backslash \{i,j\}, \vec{X}^{(i)} \wedge \vec{X}^{(j)}$ are conditionally dependent given $\{\vec{X}^{(r)}, r \in s\}$. If P is faithful with respect to a DAG G the skeleton of the DAG is a subset to the conditional independence graph corresponding to P. Every edge in the skeleton indicates some strong dependence which cannot be explained by accounting for other variables.

2.1.2.1 PC-algorithm for finding the skeleton

The PC-algorithm betters from a naive strategies that checks for conditional independences given all subsets. For the variance PC-stable of this algorithm the deletion of the arc is postponed to the change of l so the result does not depend on the order of the variables and it is parallelizable.

2.1.2.1.1 Population version In the population version of the PC-algorithm perfect knowledge about all necessary conditional independence relations is assumed available.

: $PC_{pop}(V)$

l = -1
C = $\tilde{C}$
repeat
  l += 1
  repeat
    Select a new ordered pair of nodes i, j that are adjacent in C such that $|adj(C,i)\backslash\{j\}|gel$
    repeat
      Choose new $k \subseteq adj(C,i)\backslash\{j\}$ with $|k| = l$
      if $i \wedge j$ *are conditionally independent given* k **then**
        Delete edge i, j
        Denote this new graph by C
        Save k in $S(i,j)$ and $S(j,i)$
    until *edge* i, j *is deleted or all* $k \subseteq adj(C,i)\backslash\{j\}$ *with* $|k| = l$ *have been chosen*
  until *all ordered pairs of adjacent variables* i *and* j *such that* $|adj(C,i)\backslash\{j\} \geq l$ *and* $k \subseteq adj(C,i)\backslash\{j\}$ *with* $|k| = l$ *have been tested for conditional independence*
until *for each ordered pair of adjacend nodes* $i, j : |adj(C,i)\backslash\{j\}| < l$
return Estimated skeleton C, separation sets S

The maximal value of l is denoted m_{reach} and depends on the underlying distribution. Considering a DAG G and assume that the distribution P is faithful to G. Denote the maximal number of neighbours by $q = \max_{1 \leq j \leq p} |adj(G,j)|$. Then the PC_{pop} algorithm construct the true skeleton of the DAG. Moreover $m_{reach} \in \{q-1, q\}$.

2.1.2.1.2 Sample version For finite sample there is a need to estimate conditional independencies. Assuming faithful models (conditional independence relations correspond to d-separations) in the Gaussian case conditional independences can be inferred from partial correlations. Assume that distribution P of the random vector $\vec{X}$ is a multivariate normal. For $i \neq \in \{1, \ldots, p\}, k \subseteq \{1, \ldots, p\} \backslash \{i,j\}$, denote $\rho_{i,j|k}$ the partial correlation between $\vec{X}^{(i)}$ and $\vec{X}^{(j)}$ given $\{\vec{X}^{(r)}, r \in k\}$ then $\rho_{i,j|k} = 0$ if and only if $\vec{X}^{(i)}$ and $\vec{X}^{(j)}$ are conditionally independent given $\{\vec{X}^{(r)}, r \in k\}$. The partial correlations can be estimated and the sample partial correlation $\hat{\rho}_{i,j|k}$ can be calculated, for some $h \in k$:

$$\rho_{i,j|k} = \frac{\rho_{i,j|k\setminus h} - \rho_{i,h|k\setminus h}\rho_{j,h|k\setminus h}}{\sqrt{(1-\rho^2_{i,h|k\setminus h})(1-\rho^2_{j,h|k\setminus h})}}$$

For testing whether a partial correlation is zero or not Fisher's z-transform is applied:

$$Z(i,j|k) = \frac{1}{2}\log(\frac{1=\hat{\rho}_{i,j|k}}{1-\hat{\rho}_{i,j|k}})$$

The null hypothesis is rejected $H_o(i,j|k) : \rho_{i,j|k} = 0$ against the two sided alternative $H_A(i,j|K) : \rho_{i,j|k} \neq 0$ if $\sqrt{n-|k|-3}Z(i,j|k) > \Phi^{-1}(1-\frac{\alpha}{2})$, where Φ denotes the cdf of $\mathcal{N}(0,1)$. The sample version of the PC-algorithm is the same of the population version, with the if statement replaced by $\sqrt{n-|k|-3}Z(i,j|k) > \Phi^{-1}(1-\frac{\alpha}{2})$. This algorithm yields a data-dependent $\hat{m}_{reach,n}$. The only tuning parameter is α, the significance level for testing partial correlations. This algorithm is asymptotically consistent even if p is much larger than n but the DAG is sparse. For the genehome project, linear correlations between genes are computed using the Pearson coefficient:

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

2.1.2.2 Extending the skeleton to the equivalence class

While finding the skeleton the separations sets that made edges drop out were recorded by S. This is essential for extending the skeleton to the equivalence class.

: `Extending_to_CPDAG(`G_{skel}, S`)`

for all *pairs of non adjacent* i, j *with common neighbour* k **do**
  if $k \notin S(i,j)$ **then**
    Replace $i - k - j \in G_{skel}$ with $i \to k \leftarrow j$

In the resulting PDAG try to orient as many undirected edges as possible by repeated application of:

R1 orient $j - k$ into $j \to k$ whenever there is $i \to j$ such that i and k are non-adjacent.

R2 orient $i - j$ into $i \to j$ whenever there is a chain $i \to k \to j$.

R3 orient $i - j$ into $i \to j$ whenever there are two chains $i - k \to j$ and $i - l \to k$ such that k and l are non-adjacent.

R4 orient $i - j$ into $i \to j$ whenever there are two chains $i - k \to l$ and $k \to l \to j$ such that k and l are non-adjacent

return CPDAG G

2.2 A computing system for discovering causal relationships among human genes to improve drug repositioning

The genehome project aims to expand gene networks using transcriptomic datasets. For human data the objective is to provide a public resource to navigate and combine results by expanding each single human transcript. The platform hosted the NES^2RA algorithm. Starting from a local gene

network LGN based on previous biological knowledge, its expansion consists in a set of genes and a list of interactions which describe putative causal relationships with the genes in the LGN. The expansion is calculated on observational gene expression data organized in a coherent normalized data matrix. To overcome the problem of unique elaborations OneGenE has been developed. The list of gene expansions is calculated for each gene in an organism by systematically running single-gene NES^2RA expansions with fixed parameters and then combine them afterwards to simulate LGN expansions. The expansions of the gene networks is based on the transcriptomic dataset provided by the FANTOM project. Their data comes from sequencing of RNA extracted from different samples of human tissues and cell lines and contains expression profiles of 201802 gene isoforms. Drug repositioning is an alternative approach for the discovery of new therapeutic opportunities for already approved medicines. This method, which relies on previous knowledge, speeds up the approval procedure of the drug regulators and can represent a valuable approach.

2.2.1 Method

OneGenE is a method to compute ranked candidate gene lists that expands known local gene networks given gene expression data. It is based on the systematic and iterative application of the skeleton function of the PC algorithm on subsets of the input data. Candidate expansion lists are pre-computed for each gene of the target organism. Secondly a set of transcript of interest LGN is provided as input and the intermediate results are aggregated.

: `Pre-computation-step(`I, D, A, S, E`)`

L $= \emptyset$
foreach $g \in S$ **do**
  foreach $\theta = (\alpha, d, i) \in A \times D \times I$ **do**
    %NES^2RA ranking procedure call
    $\%l_{g,\theta} = RP(S, g, E, 1, i, d, \alpha$, or:
    foreach $j \leq i$ **do**
      Randomly generate a minimal collection of subsets of dimension d of S such that g is in every subset and each transcript is in at least one subset
    foreach *subset* **do**
      Run the PC-skeleton on the expression data E restricted to the transcripts of the subset and generate a network
    foreach γ *adjacent to* g *in the network* **do**
      %Compute absolute frequency
      f_γ = numbers of networks where γ and g are adjacent
      %Compute relative frequency
      $f'_\gamma = \frac{f_\gamma}{\text{numbers of subsets that contains } \gamma}$
    $l_{g,\theta}$ = genes ordered with respect to f'_γ
    $L = L \cup l_{g,\theta}$
return L

2.2.1.1 Data and input

The algorithm starts with an $n \times m$ gene expression data matrix E, where $n = |S|$ is the number of transcripts S and m is the number of samples and a set of parametertuples $\Theta = \{\theta\} = \{(\alpha, d, i) | \alpha \in A, d \in D, i \in I\}$ where A, D and I are the sets of alpha values, tile sizes and the number of iterations.

2.2.1.2 Nested loop

For each transcript g in S, p instances of PC-IM are executed, where $p = |\Theta| = |A||D||I|$. The internal loop receive as input a single gene g with probability vbector $\Pi = 1$. The internal loop comprises the subsetting of the transcripts, the run of PC-skeleton on each subset and the computation of absolute frequencies, relative frequencies and the corresponding order of the candidates list.6

2.2.1.3 Pre-computation result

In OneGenE the ranking aggregation is postponed. Each PC-IM returns a candidate expansion list $l_{g,\theta}$ for each tuple of parameters corresponding to the set of lists resulting from the algorithm. The candidate lists are stored and the data is ready to be queried.

2.2.1.4 Ranking or list aggregation

Let S_{LGN} be the set of transcripts in an input LGN and $l_{g,\theta}$ the candidate expansion list of g with parameters θ. The final candidate expansion list is obtained combining $\mathcal{L} = \{l_{g,\theta} | g \in S_{LGN}, \theta \in \Theta\}$ by means of a ranking aggregator. High relative frequency provides evidence of a putative direct causal relationship. Ranks are instead useful for comparing list and prioritization.

```
: List_Aggregation_Complete(f'_min, K, L)
while true do
    L_temp = ∅
    User selects T set of transcript
    foreach l_{t,θ} ∈ L such that t ∈ T do
        L_temp = L_temp ∪ {l_{t,θ}}
    Case frequency: L_temp =fre(L_temp, f'_min)
    Case top k: L_temp = top(L_temp, K)
    Case top variable k_t: L_temp = top(L_temp, K)
    return List_Aggregation(L_temp)
```

2.2.1.5 Data and running paramters

The human transcriptome data have been obtained from the FANTOM5 project. It was generated using single molecule CAGE or cap analysis gene expressione. Normalized expression values are extimated as transcripts per milion TPM. The data has been filtered removing unknown transcripts and with absent or low expression values among almost all samples. The single gene NES^2RA expansion where submitted with a tile size of 1000, 1000 iterations and $\alpha = 0.05$.

2.2.2 Validation

To evaluate the biological pertinence of the single-gene NES^2RA expansion obtained by OneGenE where used gene expansions involving genes of medical relevance for neural motor dideases and hematopoietic tumors. For each expansion, the top scoring 250 transcripts where considered. OneGenE expansions where benchmarked against a simple Pearson correlation analysis. Starting from the same seed transcript, the top 250 correlated transcripts were considered and compared to the 250 transcripts identified by OneGenE. To quantify the overlap among the expansions obtained from the same transcript the Jaccard index was calculated: the number of items shared between the two sets divided by the total number of items in both sets. OneGenE expansions are largely pppulated

by distinct genes with respect to the correlation approach. To evaluate the biological pertinence, a functional enrichment was performed and the single-gene expansions consistently achieves higher biological pertinence than the correlation approach using the fraction of pertinent enrichments. Lastly using known protein-protein interaction in STRING, for each of the expansion the list of gene was compared with the list of interactions in STRING. Based on the overlap between the genes and annotated interactions odds ratio values were calculated with outstanding results.

2.2.3 Prostate cancer

22 genes expanded as single-gene by NES^2RA on two transcriptomic datasets. First analyse direct interactions within input genes and represent as graphs. Focusing on two networks and aggregating the expansion list og the genes belonging to it. A comparison with STRING and functional enrichment analyses allowed to understand nature and composition of those networks. After filtering genes alredy known to be related to prostate cancer, query against Gene Drug interaction database identified novel targets for this disease that can support drug repositioning. The functional relationship between the input genes obtained is studied by their mutual interaction: the pair (x, y) of input genes is defined as the presence of gene x into the expansion list of y and viceversa. After a comparison with STRING functional involvement of the networks and the gene composition enrichment analyses with KEGG pathways and gene anthology biological process categories. Finally after filtering the networks for genes already known DGIbd database of gene-drug interaction was queried and retrieved some drugs involved. After that target that may be cytotoxic where removed.

2.2.4 Coronary artery disease

The OneGenE approach was used to identify novel putative drug target for coronary artery disease CAD, considering genes genetically associated with it. The list of those genes has been obtained from open targets. In order to retain the genes most related the expansions list where trimmed containing only the first $N_l = \max\{5, 1 - R_l + 1\}$, where R_l is the rank based on genetic association derived from open targets. The lists of isoforms were aggregated summing the relative frequencies computed by NES^2RA. Then the list of isoforms was ranked and converted into a ranked list of genes. ToppGene was used to perform enrichment analysis against the biological processes of the gene ontology. To quantify the overlap between the ranked list og genes genetically associated with CAD and the ranked expansion lists obtained from NES^2RA, they used the weighted jaccard similarity: $WJS(\rho, \sigma) = \frac{\sum_{i=1}^{n} \min(\rho_i, \sigma_i)}{\sum_{i=1}^{n} \max(\rho_i, \sigma_i)}$, where ρ_i and σ_i are the weights corresponding to the same item i, the weight of a feature i in a ranked list ρ is computed as $len(\rho) - rank(i) + 1$. These scores depend on the length of the list, so they are not comparable. To make them comparable a permutation approach was used to estimate a set of score distributions. For each length 2000 random list of genes were generated and the WJS was computed and the score distribution associated to that length generated. Then these distributions where used to compute empirical p-values for multiple hypothesis testing.

2.3 NESRA

Before genehome the scientific pipeline consisted of:

- Organism choice.
- Finding a suitable gene expression dataset.
- Dataset filtering and imputing.
- Choose target LGN.
- PC-IM parameters test.

NESRA is a network expansion by variable subsetting and ranking aggregation. It systematically and iteratively apply subsetting varying parameters and the list is then aggregated.

```
: NESRA(I, D, A, k)
%Where I os the set of values of number of iterations, D is the set of
  values of the subset dimension, A the set of values of the significance
  level and k the maximum lenght of the list
L = ∅
foreach α ∈ A do
    foreach d ∈ D do
        foreach i ∈ I do
            L = L∪Ranking_Procedure(S, S_LGN, E, i, d, α)
L = Top(L, k)
return Ranking_Aggregation(L)
```

Its successor NES^2RA models, using a probability vector, the confidence of the presence of the genes belonging to the local gene network.