Learning a partial correlation graph using only few covariance queries

Vasiliki Velona

Einstein Institute of Mathematics,
Hebrew University of Jerusalem

based on joint work with Gábor Lugosi, Jakub Truszkowski, Piotr Zwiernik

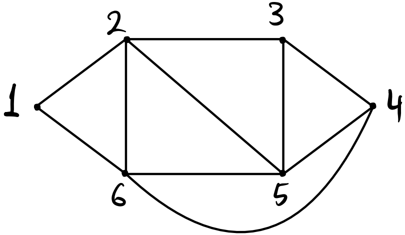
Gaussian graphical models

- Let $X = (X_1, \ldots, X_n)$ be a Gaussian random vector and Σ be its covariance matrix.
- Let K be the inverse covariance matrix and $(K_{ij})_{1 \leq i,j \leq n}$ be its entries.
- K encodes conditional independence relations:

$$K_{ij} = 0 \quad \Longleftrightarrow \quad X_i \perp\!\!\!\perp X_j \mid X_{[n] \setminus \{i,j\}} \ ,$$

where $X_i \perp\!\!\!\perp X_j \mid X_{[n] \setminus \{i,j\}}$ denotes that X_i is conditionally independent of X_j given $X_{[n] \setminus \{i,j\}}$.

Gaussian graphical models

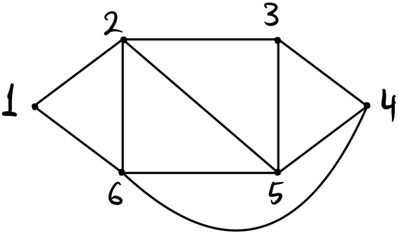


$$X_1 \perp\!\!\!\perp X_4 \mid X_2 X_3 X_5 X_6$$

$$X_3 \perp\!\!\!\perp X_6 \mid X_1 X_2 X_4 X_5$$

- Call the set of Gaussian vectors that satisfy the same independence relations of the above type *Gaussian graphical model* - the primary motivation for this work.
- In general, graph separation relations imply conditional independence (*global Markov property*, Hammersley-Clifford theorem).

Gaussian graphical models



$X_1 \perp\!\!\!\perp X_4 \mid X_2 X_6$

$X_3 \perp\!\!\!\perp X_6 \mid X_2 X_5 X_4$

and so on...

- Call the set of Gaussian vectors that satisfy the same independence relations of the above type *Gaussian graphical model* - the primary motivation for this work.
- In general, graph separation relations imply conditional independence (*global Markov property*, Hammersley-Clifford theorem).

Partial correlation graphs

We are interested in learning the graph defined by the zeros of K, that is, a graph with n vertices where

An edge ij exists if $K_{ij} \neq 0$.

This is called the *partial correlation graph*.

Partial correlation graphs

Partial correlation graphs: An edge ij exists if $K_{ij} \neq 0$.

- In the Gaussian case, learning the partial correlation graph corresponds to learning conditional independence relations.
- Similar situation in nonparanormal distributions (Liu, Lafferty, Wasserman; 2009) or discrete distributions for suitably augmented covariance matrix (Loh, Wainwright; 2013).
- Useful interpretations in other settings, such as elliptical distributions (Rossel, Zwiernik; 2021).
- Used in applications implicitly assuming normality, even if not the case.
- general relationship between conditional independence and the structure of the inverse covariance matrix not clear.

Structure recovery with covariance queries

> *Problem:* Given access to entries of Σ, learn which entries of Σ^{-1} are non-zero (the partial correlation graph), by asking only for a small fraction of all the entries of Σ.

Call the above problem structure recovery or reconstruction. For simplicity, first work at population level:

- Assume access to entries σ_{ij} of the covariance matrix Σ, through queries to a covariance oracle.
- The covariance oracle takes a pair $i, j \in [n]$ as input and outputs the corresponding entry σ_{ij} of matrix Σ.

Structure recovery with covariance queries

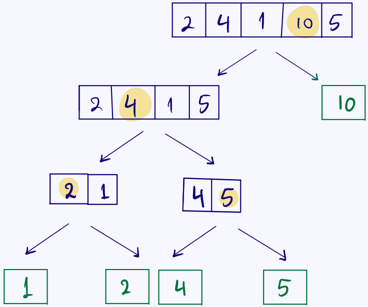
Problem: Given access to entries of the covariance matrix Σ, learn which entries of Σ^{-1} are non-zero, by asking only for a small fraction of all the entries of Σ.

- Small = subquadratic or quasi-linear.
- Aim in applications where n^2 is prohibitive to store - orthogonal setting to standard litterature.

- With noisiless covariance oracle, the problem is equivalent to inverting a symmetric positive definite matrix seeing as few of its entries as possible. (deal with noisy entries later)

Remember the quicksort algorithm

- Initial idea comes from quicksort algorithm: finding a good pivot leads to sub-quadratic complexity $\mathcal{O}(n \log n)$.
- Common strategy in *divide-and-conquer* algorithms.



- ▶ We take this idea to structure recovery for graphs resembling a tree in some way.

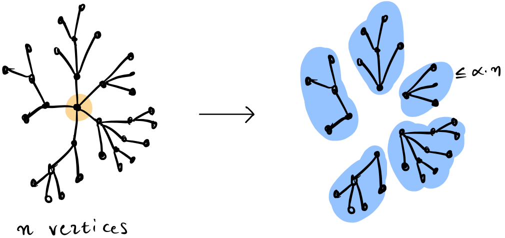
Divide-and-conquer for tree-structures

Finding a good pivot leads to sub-quadratic complexity.

In our case:

- Pivot: *central* vertex or set of vertices S.
- Having selected a pivot, recover the connected components in $G \setminus S$, then recurse in each one of them.

- We need logarithmic recursion depth to guarantee quasi-linear time (the components must shrink by a constant factor).

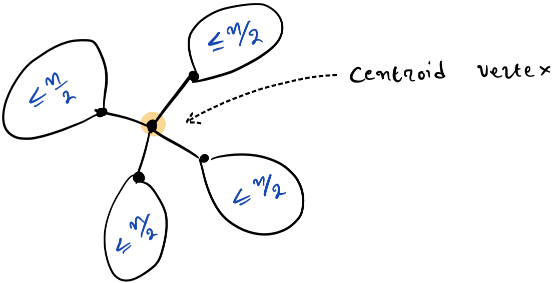


The centroid vertex

- In the case of trees one can use the *centroid* vertex, defined as

$$\arg\min_{v \in G} \max_{C \in \mathcal{C}^v} |C| \;,$$

where $\mathcal{C}^v$ is the set of connected components of $G \setminus v$. Then the largest component has at most half the vertices (shrinking factor $1/2$).



The surrogate-centroid vertex

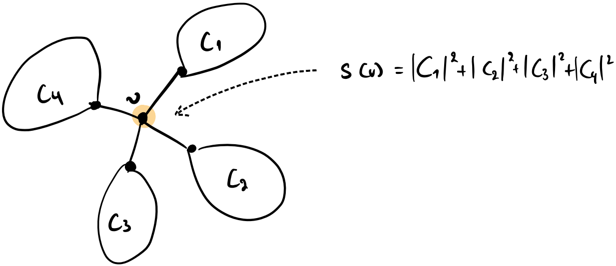
- The centroid seems hard to estimate; we instead use the *s*-central vertex defined as

$$\arg\min_v \sum_{C \in \mathcal{C}^v} |C|^2 \, .$$

- The *s*-central vertex also splits the graph nicely.
- Actually,

$$s(v) \le c(v) \le \sqrt{s(v)} \, .$$

and $optimal(s) \le optimal(c)$.



A convenient decomposition

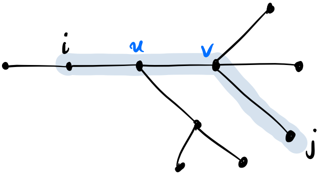
To estimate the s-central vertex, we use the following property:

Theorem (folklore)

For $i, j \in V$, the normalized entries $\rho_{ij} = \sigma_{ij}/\sqrt{\sigma_{ii}\sigma_{jj}}$ satisfy the product formula

$$\rho_{ij} \;=\; \prod_{uv \in \overline{ij}} \rho_{uv},$$

where $\overline{ij}$ denotes the unique path between i and j in the tree.



$\rho_{ij} = \rho_{iu} \cdot \rho_{uv} \cdot \rho_{vj}$

A convenient decomposition

Then for three vertices u, v, w we can ask queries of the form: *Does u separate v and w?* by computing the determinant of the minor $\Sigma_{vu,uw}$.

- If $\rho_{vw} = \rho_{vu}\rho_{wu}$, then *yes*.

> *Separation relation $\leftrightarrow$ compute determinant of a minor.*

Find the s-central vertex:

- For vertex v, pick κ pairs and estimate $\hat{s}(v)$ from them.
- One needs only $\mathcal{O}\left(\log\left(\frac{n}{\epsilon}\right)\right)$ pairs for success probability at least $1-\epsilon$, using Hoeffding's inequality.

Recovering the connected components

The connected components after deleting central vertex w are recovered as follows:

1. Sort $|\rho_{uw}|$ for $u \in V \setminus \{w\}$ in decreasing order.
2. For every vertex v,
   - if v is separated from w by an already discovered neighbour u of w, then put it in the component where u is.
   - Otherwise create a new component for v and identify v as a neighbour of w.

- Time complexity is of order $\mathcal{O}(n \log n + n\kappa + nd)$ and the query complexity is of order $\mathcal{O}(n\kappa + nd)$, where d is the maximum degree.

Result for tree-recovery

Algorithm: Do the previous recursively, until all the edges are recovered.

Theorem (Lugosi, Truszkowski, V., Zwiernik; 2021)

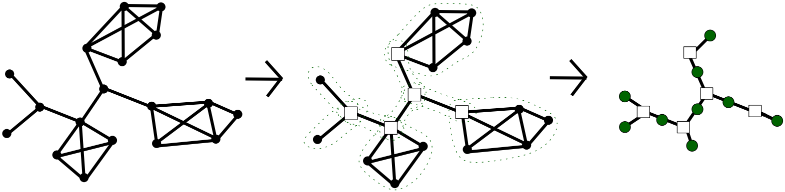
Assume graph with maximum degree $\leq d$. *With probability at least* $1 - \epsilon$, *the algorithm for tree reconstruction requires time and queries of the order*

$$\mathcal{O}\left(n \log(n) \max\left\{\log\left(\frac{n}{\epsilon}\right), d\right\}\right).$$

- Our algorithm's performance is tight up to logarithmic factors. The dependence on d is essential.

Recovery of tree-like graphs

- Tree-like graphs: graphs with small 2-connected components.
- The block-cut tree of a graph:



Recovery of tree-like graphs

Theorem (Lugosi, Truszkowski, V., Zwiernik; 2021)

Assume graph with largest 2-connected component of size $\leq b$ and maximum degree of the block-cut tree $\leq d$. With probability at least $1-\epsilon$, one recovers the graph in time and queries

$$\mathcal{O}_{\epsilon,d,b}(n \log^2 n) \ .$$

- We show this by proving that there exists a vertex s.t., after deleting it, the biggest component is at most $\left(1-\frac{1}{2d}\right) n$ (when $n > db$). Then we can use this vertex as a good pivot on which we recurse.

Can we stretch this idea further?

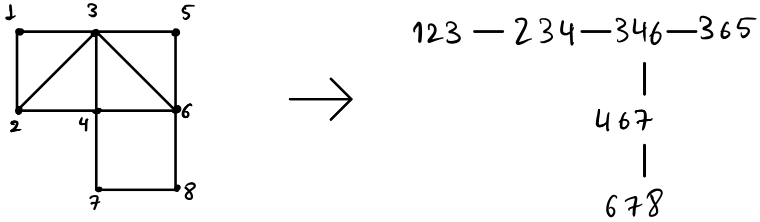
The treewidth of a graph

A *tree decomposition* of a graph $G(V, E)$ is a tree T with vertices $B_1, \ldots, B_m$, where $B_i \subseteq V$ satisfy

1. The union of all sets B_i equals V.
2. If B_i and B_j both contain v, then all vertices B_k of T in the unique path between B_i and B_j contain v as well.
3. For every edge uv in G, there is B_i that contains both u and v.

- The *width* of a tree decomposition is the size of its largest set B_i minus one.
- The *treewidth* of a graph G, denoted $tw(G)$, is the minimum width among all possible tree decompositions of G.

An example



Balanced separators

We use a generalised centrality notion. For any set $S \subset V$ we write

$$c(S) = \frac{1}{|V \setminus S|} \max_{C \in \mathcal{C}^S} |C| \; ,$$

It is known that every graph with bounded treewidth has a small balanced separator:

Theorem (folklore)

If $tw(G) \leq k$ then G has a separator S such that $|S| \leq k+1$ and

$$c(S) \leq \frac{1}{2} \cdot \frac{|V| - k}{|V| - (k+1)} \; .$$

Separation and rank

Let $\mathcal{M}(G)$ be the set of covariance matrices such that

$$ij \notin G \Rightarrow \sigma_{ij} = 0 \; .$$

Theorem (Sullivant, Talaska, Draisma; 2010)

For a generic matrix in $\mathcal{M}(G)$,

$$\mathrm{rank}(\Sigma_{A,B}) = \min\{|S| : S \text{ separates } A \text{ and } B\} \; .$$

Algorithmic consequence: For vertex sets A, B, a vertex v lies in some minimum size separator of them if and only if $\mathrm{rank}(\Sigma_{Av,Bv}) = \mathrm{rank}(\Sigma_{A,B})$.

Finding a minimal separator

Let vertex sets A, B. One can use the previous observation to find a minimal separator S of A,B:

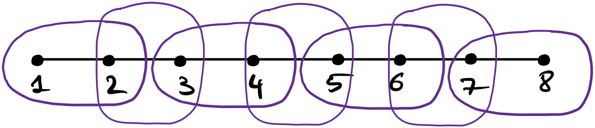
- Start with $S = \emptyset$. For all $v \in V$, if $\mathrm{rank}(\Sigma_{ASv,BSv}) = \mathrm{rank}(\Sigma_{A,B})$, then put v in S.

- If A,B are picked at random, how to guarantee that S gives a balanced split in the whole graph?

Vapnik-Chervonenkis dimension

- Let V be a set. $W \subset V$ is *shattered* by a family of sets U if $\{W \cap R : R \in U\}$ is the set of all subsets of W.
- The *VC-dimension* of U, denoted by $VC(U)$, is the maximal size of a set shattered by U.

◦ Example:



The VC dimension of this family of sets is 2.

δ-samples and VC dimension

A set $W \subseteq V$ is a *δ-sample* for a set-family $\mathcal{F}_k$ if for all sets $C \in \mathcal{F}_k$,

$$\frac{|C|}{|V|} - \delta \leq \frac{|W \cap C|}{|W|} \leq \frac{|C|}{|V|} + \delta\ .$$

- Vapnik-Chervonenkis inequality: Assume $\mathrm{V}C(\mathcal{F}_k) = r$. A set W obtained by sampling m vertices from V uniformly at random, with replacement, is a δ-sample of $\mathcal{F}_k$ with probability at least $1 - \tau$ if

$$m \geq \max\left(\frac{10r}{\delta^2}\log\left(\frac{8r}{\delta^2}\right), \frac{2}{\delta^2}\log\left(\frac{2}{\tau}\right)\right)\ .$$

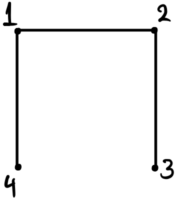
A set-family of interest

Theorem (Feige, Mahdian; 2006)

For fixed k, let $\mathcal{F}_k$ be the set-family that contains all connected components after removing a set S of at most k vertices and their complements (if $G \setminus S$ is disconnected). Then

$$\mathrm{VC}(\mathcal{F}_k) \leq 11 \cdot k \ .$$

An example:



$$\mathcal{F}_1 = \left\{ \begin{matrix} \{1,4\} & \{2,3\} \\ \{3\} & \{4\} \end{matrix} \right\}$$

Finding a balanced separator

Algorithm: Take a (small) random sample W. For all partitions A, B of W compute $rank(\Sigma_{A,B})$. There will be a partition such that $r := rank(\Sigma_{A,B}) \leq tw + 1$. Pick the most balanced partition.

There exists a small balanced separator (by bounded treewidth) and we can find a balanced partition of a random sample that corresponds to a small balanced separator in the full graph (by δ-sample properties).

Summing up the algorithm

1. Take a random sample and pick the most balanced partition A, B that is separated by few vertices ($\leq \textit{treewidth} + 1$).
2. Find a minimal separator of A, B.
3. Find the connected components after removing S:
   A vertex u belongs to the connected component of v if $\mathrm{rank}(\Sigma_{uS,vS}) = |S| + 1$.
4. Recurse in each connected component, using the conditional distribution on S.

General result for noise-less regime

Theorem (Lugosi, Truszkowski, V., Zwiernik; 2021)

Assume G connected graph with treewidth $\leq k$ and maximum degree $\leq d$. Then, with probability at least $1 - \frac{1}{n^8}$, the query complexity of our reconstruction algorithm is of the order

$$\mathcal{O}\left((2^{\mathcal{O}(k \log k)} + dk \log n)k^2 n \log^3 n\right),$$

and the time complexity is of the order

$$\mathcal{O}\left((2^{\mathcal{O}(k \log k)} + dk \log n)k^3 n \log^4 n\right).$$

Learning the graph with noisy oracle

We now assume a noisy covariance oracle that, when queried for the (i,j)-th entry of Σ, returns a value $\widehat{\sigma}_{ij}$ satisfying

$$\max_{ij} |\widehat{\sigma}_{ij} - \sigma_{ij}| < \epsilon$$

for some $\epsilon \in (0,1)$.

Under assumptions such as

$$\delta \leq |\sigma_{ij}| \leq \gamma \qquad \text{for all } ij \in E \ ,$$

for $0 < \delta < \gamma < 1$, we show that our (tree) algorithm works in this regime as well.

Learning the graph with noisy oracle

- In the noisy case the recovery guarantees crucially depend on the diameter D. Under our assumptions, $|\sigma_{ij}| \leq \gamma^{d(i,j)}$ where $d(i,j)$ is the distance of vertex i and vertex j in the tree. This value is *indistinguishable from zero* by the noisy covariance oracle unless $d(i,j) < \log(1/\epsilon)/\log(1/\gamma)$.
- One can construct such an oracle. For instance

$$N \geq 32\left(\frac{\kappa}{\epsilon}\right)^2 \log\frac{n}{\eta}$$

samples are enough with probability at least $1-\eta$, if the fourth moment is bounded by κ for all i (Lugosi, Mendelson; 2019).

Remarks, further questions

- the matrix itself is recovered not just edges.
- costly to store the entire sample covariance matrix; online algorithms
- for large classes of graphs, the structure of the corresponding partial correlation graphs can be determined much faster than even computing the empirical covariance matrix.
- approach orthogonal to standard algorithms; perhaps relevant to numerical linear algebra

- details of noisy oracle in general case.
- necessity of treewidth and maximum degree (what about other parameters such as fragmentation)
- general relationship between conditional independence and the structure of the inverse covariance matrix.

Thank you!