Unfolding latent tree structures using 4th order tensors

Ishteva, Park, Song

Arthur Gretton's notes

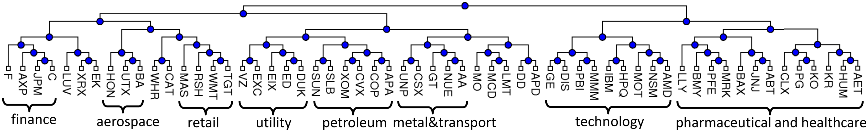
June 7, 2013

Problem setup

We are given:

- d observed (leaf) variables with n states each,
- hidden variables of k (unknown) states each (k can be different for different hidden variables: notational convenience)
- an assumed binary tree: each hidden variable has exactly two children.

Goal: recover the tree.



First step: four leaves, two latent variables

How do we connect four leaves, x_1, x_2, x_3, x_4, with two latent variables, g, h?
Assume the true structure is:

$$P(x_1, x_2, x_3, x_4) = \sum_{g,h} P(x_1|h)P(x_2|h)P(g,h)P(x_3|g)P(x_4|g).$$

The joint probability can be concisely written

$$P(x_1, x_2, x_3, x_4) = \langle \mathcal{T}_1, \mathcal{T}_2 \rangle_3$$

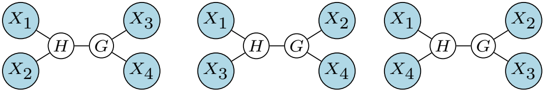
where

$$\mathcal{T}_1 = \mathcal{I} \times_1 P_{1|H} \times_2 P_{2|H}$$
$$\mathcal{T}_2 = \mathcal{I} \times_1 P_{3|G} \times_2 P_{4|G} \times_3 P_{HG}$$

and $\mathcal{I}$ is the unit 3-tensor (size $k \times k \times k$).

Joint probability table is 4th order tensor

There are three possibilities:



(b) $\{\{1,2\},\{3,4\}\}$ $\{\{1,3\},\{2,4\}\}$ $\{\{1,4\},\{2,3\}\}$

The following reshapings group the variables such that variables sharing a latent factor are either in the rows, or in the columns:

$$A = \text{reshape}(\mathcal{P}, n^2, n^2);$$
$$B = \text{reshape}(\text{permute}(\mathcal{P}, [1,3,2,4]), n^2, n^2);$$
$$C = \text{reshape}(\text{permute}(\mathcal{P}, [1,4,2,3]), n^2, n^2).$$

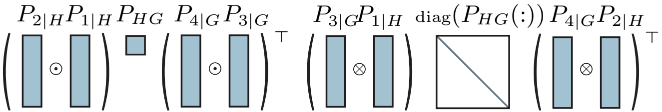
Unfoldings of the fourth order tensor

The following equations give the linear algebraic expressions for these unfoldings:

$$A = \left(P_{2|H} \odot P_{1|H}\right)\ P_{HG}\ \left(P_{4|G} \odot P_{3|G}\right)^{\top}, \tag{5}$$

$$B = \left(P_{3|G} \otimes P_{1|H}\right) \operatorname{diag}(P_{HG}(:)) \left(P_{4|G} \otimes P_{2|H}\right)^{\top}, \tag{6}$$

$$C = \left(P_{4|G} \otimes P_{1|H}\right) \operatorname{diag}(P_{HG}(:)) \left(P_{3|G} \otimes P_{2|H}\right)^{\top}. \tag{7}$$



Note that:

- $\operatorname{rank}(A) = \operatorname{rank}(P_{GH}) = k$
- $\operatorname{rank}(B) = \operatorname{rank}(C) = \operatorname{nnz}(P_{GH})$ (number of non-zero entries).

Thus, generally speaking, $\operatorname{rank}(A) \ll \operatorname{rank}(B) = \operatorname{rank}(C)$.

Nuclear norm proxy for rank

Instead of rank, use nuclear norm

$$\|M\|_* = \sum_{i=1}^{n} \sigma_i(M)$$

where $\sigma_i(M)$ is ith singular value. From *Fazel et al. (2001)*: best convex lower bound of the rank over the unit ball of matrices
$M \; : \; \|M\|_F = \sigma_1(M) \leq 1.$

Nuclear norm proxy for rank

Instead of rank, use nuclear norm

$$\|M\|_* = \sum_{i=1}^{n} \sigma_i(M)$$

where $\sigma_i(M)$ is ith singular value. From *Fazel et al. (2001)*: best convex lower bound of the rank over the unit ball of matrices $M : \|M\|_F = \sigma_1(M) \leq 1$.

Algorithm 1 $i^* = \text{Quartet}(X_1, X_2, X_3, X_4)$

1: Estimate $\widehat{\mathcal{P}}(X_1, X_2, X_3, X_4)$ from a set of m *i.i.d.* samples $\{(x_1^l, x_2^l, x_3^l, x_4^l)\}_{l=1}^m$.

2: Unfold $\widehat{\mathcal{P}}$ in three different ways into matrices $\widehat{A}$, $\widehat{B}$ and $\widehat{C}$, and compute their nuclear norms

$$a_1 = \|\widehat{A}\|_*, \ a_2 = \|\widehat{B}\|_* \text{ and } a_3 = \|\widehat{C}\|_*.$$

3: Return $i^* = \text{argmin}_{i \in \{1,2,3\}} a_i$.

Dependence interpretation

Dependence interpretation:

- A encodes the dependence between pair $\{1, 2\}$ and pair $\{3, 4\}$,
- $\|A\|_*$ is the strength of this dependence
- Given the graph structure, $\{1, 2\}$ and $\{3, 4\}$ are weakly dependent, but $\{1, 3\}$ and $\{2, 4\}$ are strongly dependent.

When is recovery possible?

Given that we use the proxy $\|M\|_*$ for $\mathrm{rank}(M)$, when can we recover the structure?

- G, H independent, and $P_{GH} = P_G P_H^\top$. Then

$$A_\perp = (P_{2|H} \odot P_{1|H})\ P_H P_G^\top\ (P_{4|G} \odot P_{3|G})^\top = P_{12}(:)\ P_{34}(:)^\top, \qquad (11)$$

$$B_\perp = (P_{3|G} \otimes P_{1|H})(\mathrm{diag}(P_G) \otimes \mathrm{diag}(P_H))(P_{4|G} \otimes P_{2|H})^\top = P_{34} \otimes P_{12}, \qquad (12)$$

hence $\mathrm{rank}(A_\perp) = 1 \ll \mathrm{rank}(B_\perp)$.

- $G = H$ (deterministic relation), then indeterminate.

Conditions for quartet recovery

Define

$$\theta := \min\{\|B_\perp\|_* - \|A_\perp\|_*, \|C_\perp\|_* - \|A_\perp\|_*\}$$
$$\Delta := P_{GH} - P_G P_H^\top$$

Lemma 4 *If* $\|\Delta\|_F \leq \frac{\theta}{k^2+k}$, *the minimum of* $\|A\|_*$, $\|B\|_*$ *and* $\|C\|_*$ *will reveal the correct quartet relation.*

Conditions for quartet recovery

Define

$$\theta := \min\{\|B_\perp\|_* - \|A_\perp\|_*, \|C_\perp\|_* - \|A_\perp\|_*\}$$
$$\Delta := P_{GH} - P_G P_H^\top$$

Lemma 4 *If $\|\Delta\|_F \leq \frac{\theta}{k^2+k}$, the minimum of $\|A\|_*$, $\|B\|_*$ and $\|C\|_*$ will reveal the correct quartet relation.*

When we compute probabilty tables from m observations, and defining $\alpha = \min\{\|B\|_* - \|A\|_*, \|C\|_* - \|A\|_*\}$

Lemma 5 *With probability $1-8e^{-\frac{1}{32}m\alpha^2}$, Algorithm 1 returns the correct quartet relation.*

Tree recovery algorithm

The quartet test may be used to recover trees:

Algorithm 2 $\mathcal{T} = \text{BuildTree}(X_1, \ldots, X_d)$

1: Connect any 4 variables X_1, X_2, X_3, X_4 with 2 latent variables in a tree $\mathcal{T}$ using Algorithm 1.
2: **for** $i = 4, 5, \ldots, d-1$ **do** {insert $(i+1)$-th leaf X_{i+1}}
3: Choose root R that splits $\mathcal{T}$ into sub-trees $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3$ of roughly equal size.
4: Choose any triplet $(X_{i_1}, X_{i_2}, X_{i_3})$ of leaves from different sub-trees.
5: Test which sub-tree should X_{i+1} be joined to: $i^* \leftarrow \text{Quartet}(X_{i+1}, X_{i_1}, X_{i_2}, X_{i_3})$.
6: Repeat recursively from step 3 with $\mathcal{T} := \mathcal{T}_{i^*}$. This will eventually reduce to a tree with a single leaf. Join X_{i+1} to it via hidden variable.
7: **end for**