Estimating the Graph of a Probabilistic Model

Lesson 7 : Lab Session

Advanced Machine Learning, CentraleSupelec

Teacher's Assistant: Omar CHEHAB

Professors : Emilie CHOUZENOUX, Frederic PASCAL

CentraleSupélec | université PARIS-SACLAY

General Information

- **Assignment :** alone or in pairs, you will code the algorithms you learnt in 'scikit-learn formalism', and apply them to images and text.

- **Due :** the 5 lab assignments for lessons 3-7 are due a week from when they are given, at aml.centralesupelec.2020@gmail.com

- **Grading** : each assignment is worth 4 points — your 4 best labs out of the 5 will be retained and will count for half of your final grade.

- **Questions :** questions or feedback are welcome after class or by email at l-emir-omar.chehab@inria.fr

Graphical Model

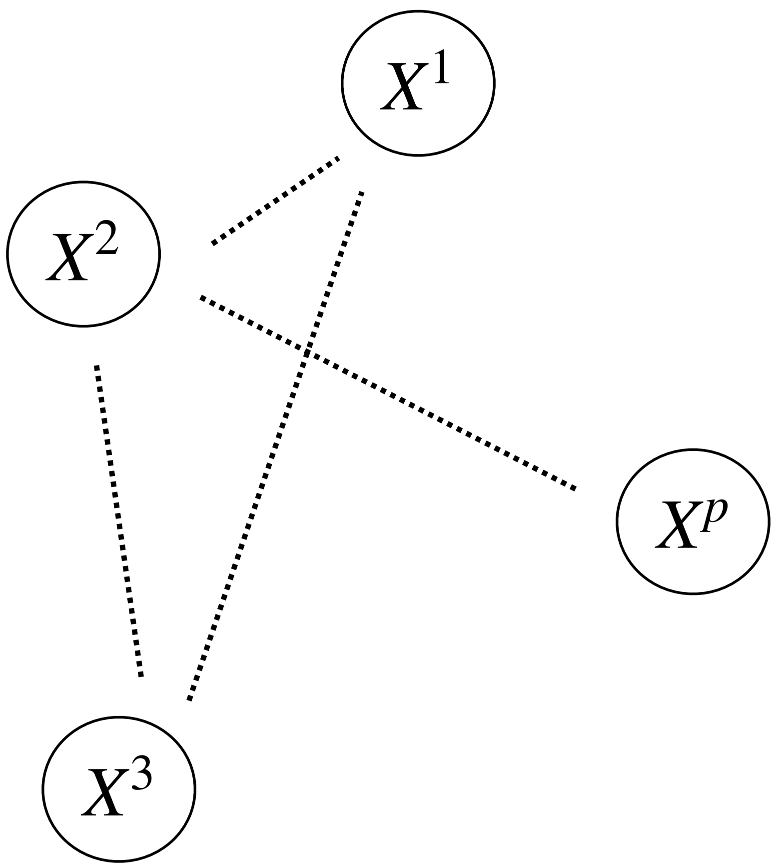
$X = \begin{pmatrix} X^1 \\ \cdots \\ X^p \end{pmatrix}$

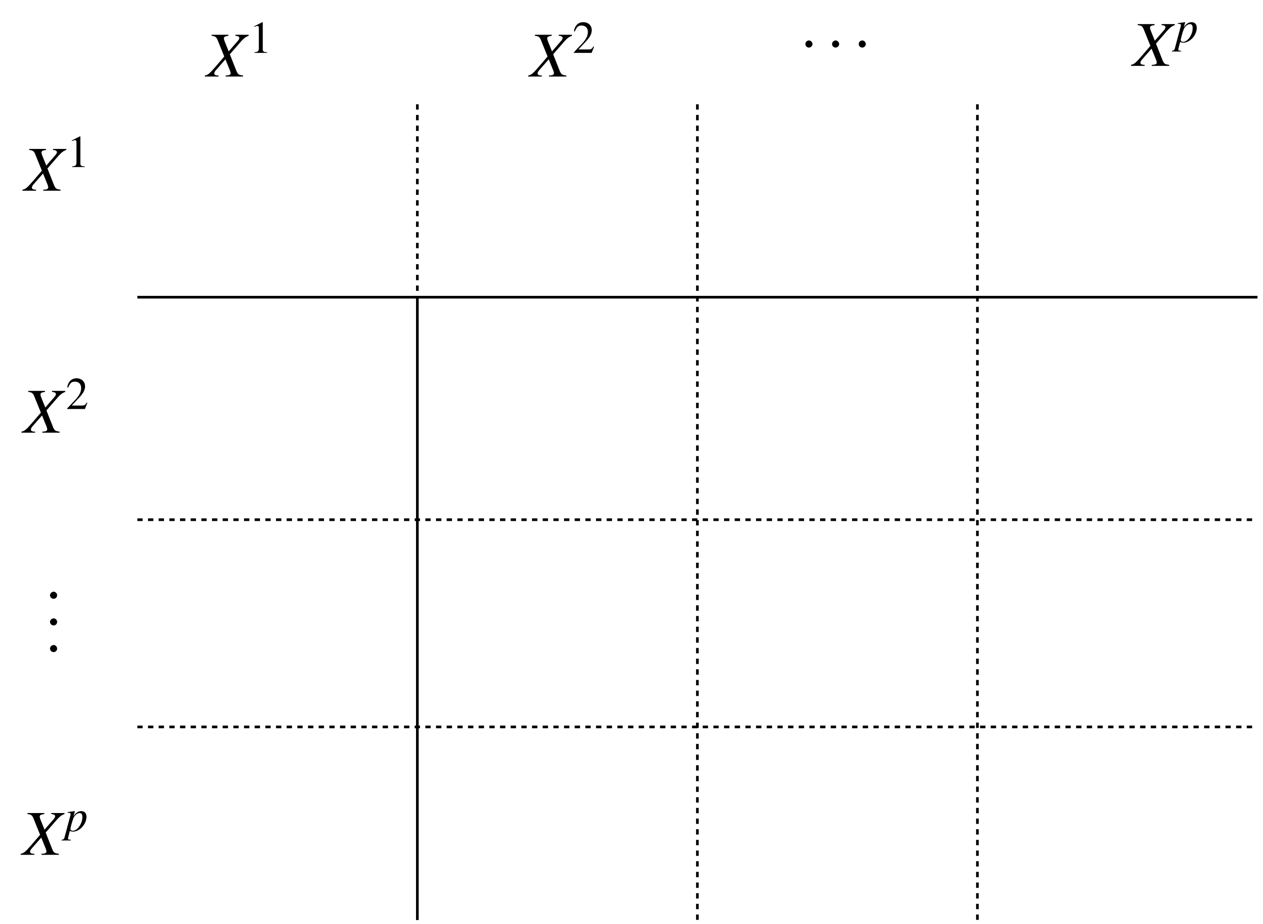
no edge (i, j) $\iff$ X^i and X^j conditionally **in**dependent   hard to test!

$\iff$ X^i and X^j have zero inverse correlation   easy to test!

if $X \sim \mathcal{N}(\mu, \Sigma)$

$\Lambda = \Sigma^{-1}$ precision matrix





Estimating the Graph

Classic : learn the *whole* precision matrix via Maximum-Likelihood

$$\min_{\Lambda \succ 0} - \log\det\Lambda + Tr(S\Lambda) \quad \underset{\text{closed-form}}{\Rightarrow} \quad \hat{\Lambda} = S^{-1} = \left(\frac{1}{n}\sum_{i=1}^{n}\left(x_i - \hat{\mu}\right)\left(x_i - \hat{\mu}\right)^T\right)^{-1} \quad \text{where } \hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

inverse empirical covariance

empirical mean

not great when $n << p$: reduce #parameters via a sparsity constraint

GLASSO : learn the *whole* precision matrix via **sparse** Maximum-Likelihood

[Fiedman et al., 2008]

$$\min_{\Lambda \succ 0} - \log\det\Lambda + Tr(S\Lambda) + \lambda\|\Lambda\|_1 \quad \underset{\text{ADMM algo}}{\Rightarrow}$$

> 0

reformulate

$$\min_{\Lambda} - \log\det\Lambda + Tr(S\Lambda) + 1_{\Lambda \succ 0} + \lambda\|Y\|_1$$

$$Y = \Lambda$$

$$T_k \leftarrow Y_{k-1} - Z_{k-1} - \frac{1}{\nu}S$$

intermediate step for computations

$\nu > 0$ weight of the penalization $Y = \Lambda$

$$\Lambda_k \leftarrow \frac{1}{2}\operatorname{diag}\left(\left\{\lambda_i(T_k) + \sqrt{\lambda_i(T_k)^2 + \frac{4}{\nu}}\right\}\right)$$

primal variable

$$Y_k \leftarrow \mathrm{ST}_{(\lambda/\nu)}\left(\Lambda_k + Z_{k-1}\right)$$

relaxed primal variable

$$Z_k \leftarrow Z_{k-1} + \nu\left(\Lambda_k - Y_k\right)$$

dual variable

Nodewise Reg. : learn the *support* of the precision matrix via **sparse** linear regressions

[Meinshausen et al., 2006]

predict each variable X^j using the others $\{X^k, k \neq j\}$ with LASSO $\rightarrow \hat{\beta}^j \in \mathbb{R}^{p-1}$ weights of the regression…

then deduce the support : $\Lambda_{jk} \neq 0 \iff \hat{\beta}^j_k \quad AND \quad \hat{\beta}^k_j$ are non-zero …used as edge-activators

(also try OR)

Assignment: plan

1. Graphical LASSO (*your own code*)

2. Node-wise Regression (*your own code*)

3. Application : learn graph of simulated data (*your own code*)