Multi-componets Data, Signal and Image Processing for Biological and Medical Applications

Ali Mohammad-Djafari

Laboratoire des Signaux et Systèmes
UMR 8506 CNRS - CS - Univ Paris Sud
CentraleSupélec, Gif-sur-Yvette.

djafari@lss.supelec.fr
http://djafari.free.fr

January 6, 2017

Summary 3: Data redundancy, Dimensionality Reduction, ...

- Redundancy and structure
- Dimentionality Reduction
- PCA and ICA
- PPCA and its extensions
- Stationarity / non-stationarity
- Discriminant Analysis (DA)
- Classification and Clustering
- Mixture Models
- Factor Analysis
- Blind Sources Separation

Dimension reduction, PCA, Factor Analysis, ICA

- M variables $g_1, \cdots, g_M$ are observed. They are redundant. Can we express them with $N \leq M$ factors $f_1, \cdots, f_N$? How many factors (Principal Components, Independent Components) can describe the observed data?
- Each variable is observed T times. To index them, we use t, but this does not forcibly means time. So, we have $\{g_1(t), \cdots, g_M(t), t = 1, \cdots, T\}$.
- We may represent these data either as a vector or a matrix:

$$\boldsymbol{g}(t) = \begin{bmatrix} g_1(t) \\ g_2(t) \\ \vdots \\ g_M(t) \end{bmatrix}, t = 1, .., T \text{ or } \boldsymbol{G} = \begin{bmatrix} g_1(1) & g_1(1) & \cdots & g_1(T) \\ g_2(1) & g_2(1) & \cdots & g_1(T) \\ \vdots & & & \\ g_M(1) & g_M(1) & \cdots & g_1(T) \end{bmatrix}$$

Dimension reduction, PCA, Factor Analysis, ICA

- If we define the $N < M$ factors as

$$\boldsymbol{f}(t) = \begin{bmatrix} f_1(t) \\ f_2(t) \\ \vdots \\ f_N(t) \end{bmatrix}, t = 1, .., T \text{ or } \boldsymbol{F} = \begin{bmatrix} f_1(1) & f_1(1) & \cdots & f_1(T) \\ f_2(1) & g_2(1) & \cdots & f_1(T) \\ \vdots & & & \\ f_M(1) & g_M(1) & \cdots & f_1(T) \end{bmatrix}$$

  where we assume that each factor is a linear combination of data

  $f_j(t) = \sum_{i=1}^{M} b_{ji} g_i(t)$ or inversely $g_i(t) = \sum_{j=1}^{N} a_{ij} f_j(t)$.

- Now, if we define the matrices $\boldsymbol{B} = \{b_{ji}\}$ or $\boldsymbol{A} = \{a_{ij}\}$ we can write

$$\boldsymbol{f}(t) = \boldsymbol{B}\boldsymbol{g}(t) \text{ or } \boldsymbol{g}(t) = \boldsymbol{A}\boldsymbol{f}(t)$$

- $\boldsymbol{B}$ is called Loading matrix and $\boldsymbol{A}$ is called mixing matrix.
- Ideal case is then $\boldsymbol{B} = \boldsymbol{A}^{-1}$. But this is not interesting, because we want to have $N < M$.
- We may accept some errors: $\boldsymbol{g}(t) = \boldsymbol{A}\boldsymbol{f}(t) + \boldsymbol{\epsilon}(t)$

Dimension reduction, PCA, Factor Analysis

- M variables $\boldsymbol{g}(t)$ are observed. They are redundant. Can we express them with $N \leq M$ factors $\boldsymbol{f}$?
  How many factors (Principal Components, Independent Components) can describe the observed data?

$$\boldsymbol{f}(t) = \boldsymbol{B}\boldsymbol{g}(t) \text{ or } \boldsymbol{g}(t) = \boldsymbol{A}\boldsymbol{f}(t)$$

  or still

$$\boldsymbol{F} = \boldsymbol{B}\boldsymbol{G} \text{ or } \boldsymbol{G} = \boldsymbol{A}\boldsymbol{F}$$

- We assume all the variables to be centred.
- PCA uses the second order statistics

$$\text{cov}[\boldsymbol{g}] = \boldsymbol{A}\text{cov}[\boldsymbol{f}]\boldsymbol{A}^t$$

- We want principal Components (PC) to be non-correlated: cov$[\boldsymbol{f}]$ be a diagonal matrix.

$$\text{cov}[\boldsymbol{f}] = \text{diag}\,[v_1, \cdots, v_N]$$

- One solution: Estimate cov$[\boldsymbol{g}] = \sum_{t=1}^{T} \boldsymbol{g}(t)\boldsymbol{g}'(t)$ and use it.

Dimension reduction, PCA Algorithms

- Forward model

$$\boldsymbol{g}(t) = \boldsymbol{A}\boldsymbol{f}(t) \text{ or } \boldsymbol{G} = \boldsymbol{A}\boldsymbol{F}$$

- Classical PCA algorithm:
  - Estimate $\text{cov}[\boldsymbol{g}] = \sum_{t=1}^{T} \boldsymbol{g}(t)\boldsymbol{g}'(t)$
  - Hoping that it is positive Definite, compute its SVD: $\text{cov}[\boldsymbol{g}] = \boldsymbol{A}\boldsymbol{V}\boldsymbol{A}^t$
  - Identify $\widetilde{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{V}^{1/2}$ and compute $\widehat{\boldsymbol{f}} = \boldsymbol{V}^{-1/2}\boldsymbol{A}^t\boldsymbol{g}$
  - The number of factors is the number of non-zero singular values.
  - The data can be retrieved exactly by

$$\widehat{\boldsymbol{g}} = \widehat{\boldsymbol{A}}\widehat{\boldsymbol{f}} = \boldsymbol{A}\boldsymbol{V}^{1/2}\boldsymbol{V}^{-1/2}\boldsymbol{A}^t\boldsymbol{g} = \boldsymbol{A}\boldsymbol{A}^t\boldsymbol{g} = \boldsymbol{g}$$

  - if we keep K SV, we make an error which can be used as criterion to determine the number of factors

$$E = \frac{\|\widehat{\boldsymbol{g}} - \boldsymbol{g}\|^2}{\|\boldsymbol{g}\|^2}$$

Dimension reduction: PPCA

- Forward model

$$\boldsymbol{g}(t) = \boldsymbol{A}\boldsymbol{f}(t) + \boldsymbol{\epsilon}(t) \text{ or } \boldsymbol{G} = \boldsymbol{A}\boldsymbol{F} + \boldsymbol{E}$$

- Probabilistic PCA:

$$\text{cov}[\boldsymbol{g}] = \boldsymbol{A}\text{cov}[\boldsymbol{f}]\boldsymbol{A}^t + \text{cov}[\boldsymbol{\epsilon}]$$

- Estimate $\text{cov}[\boldsymbol{g}] = \sum_{t=1}^{T} \boldsymbol{g}(t)\boldsymbol{g}'(t)$
- Hoping that it is positive Definite, compute its SVD: $\text{cov}[\boldsymbol{g}] = \boldsymbol{A}\boldsymbol{V}\boldsymbol{A}^t$
- Keep the SV which are greater than v_ϵ and Identify $\widehat{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{V}^{1/2}$ and compute $\widehat{\boldsymbol{f}} = \boldsymbol{V}^{-1/2}\boldsymbol{A}^t\boldsymbol{g}$
- The number of factors is the number of singular values which are greater than v_ϵ.
- The main difficulty is to estimate v_ϵ.

Application on a set of data

- Genes expressions Time series data in two organs:

Colon:

Clock:	Rev, Per2, Bmal1
Metabolism:	CE2, Top1, UGT, DBP
CC:	Wee1, Ccna2, Ccnb2
Apoptose:	Bcl2, Mdm2, Bax, P53

Liver:

Clock:	Rev, Per2, Bmal1
Metabolism:	CE2, Top1, UGT, DBP
CC:	Wee1, P21
Apoptose:	Bcl2, Mdm2, Bax, P53

Physiology:

Temperature, Activity

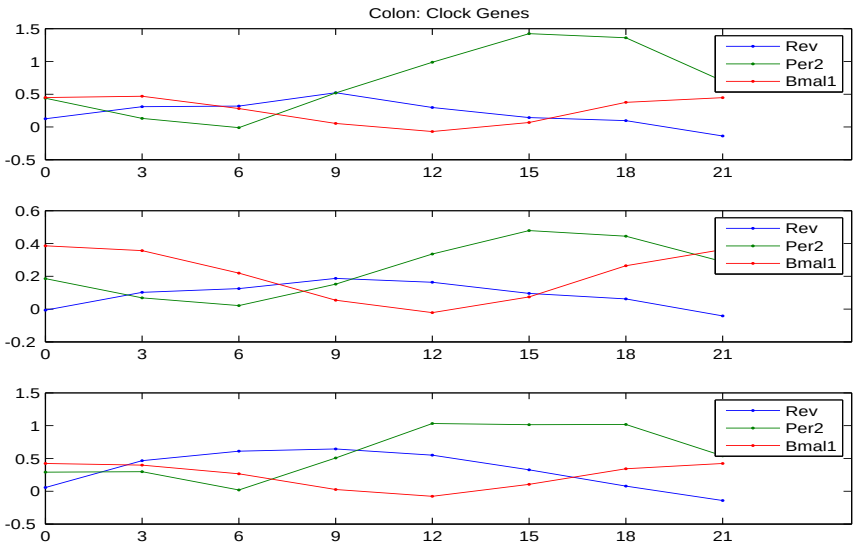
Cortico, Melato

- The data are obtained via the COSINOR Model

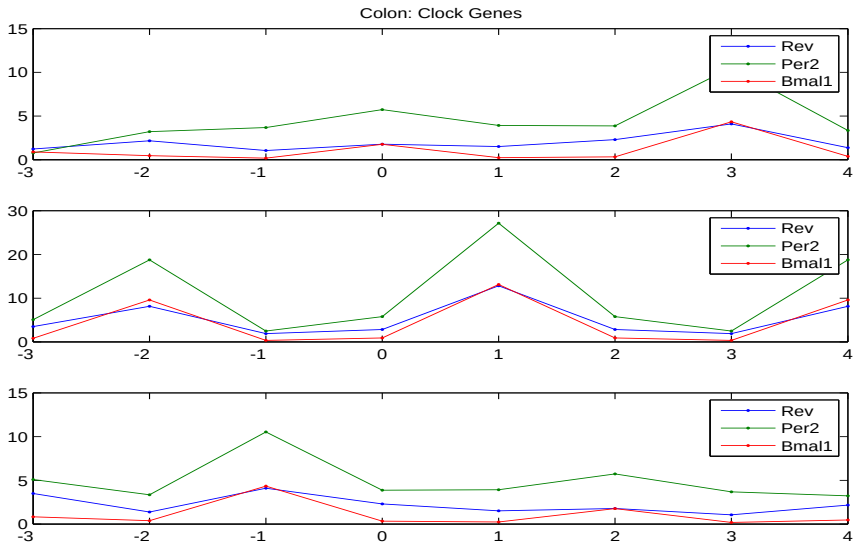
$$g(t) = M + \sum_{k=1}^{3} A_k \cos(k\omega_0 t + \phi_k), \omega_0 = \frac{2\pi}{24}, t = 0, 3, 6, .., 21$$

Genes Clock

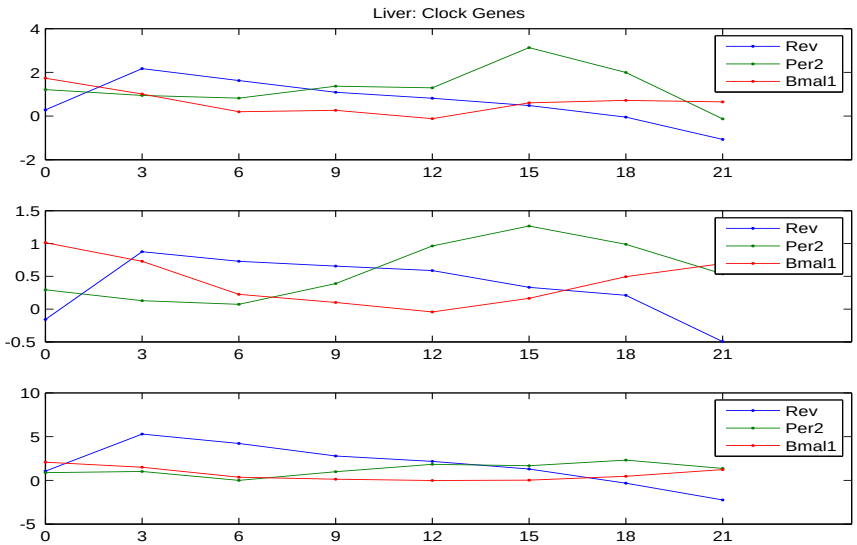
Colon: Time Series



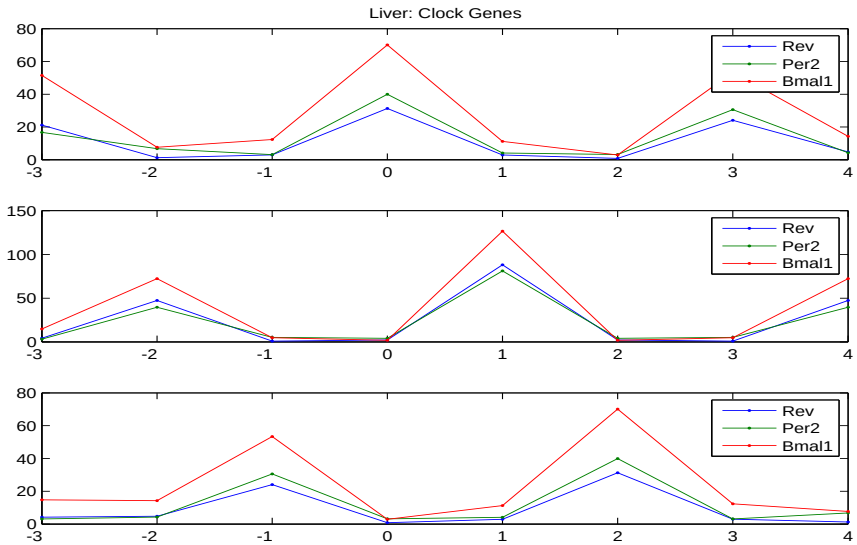
Fourier Transform



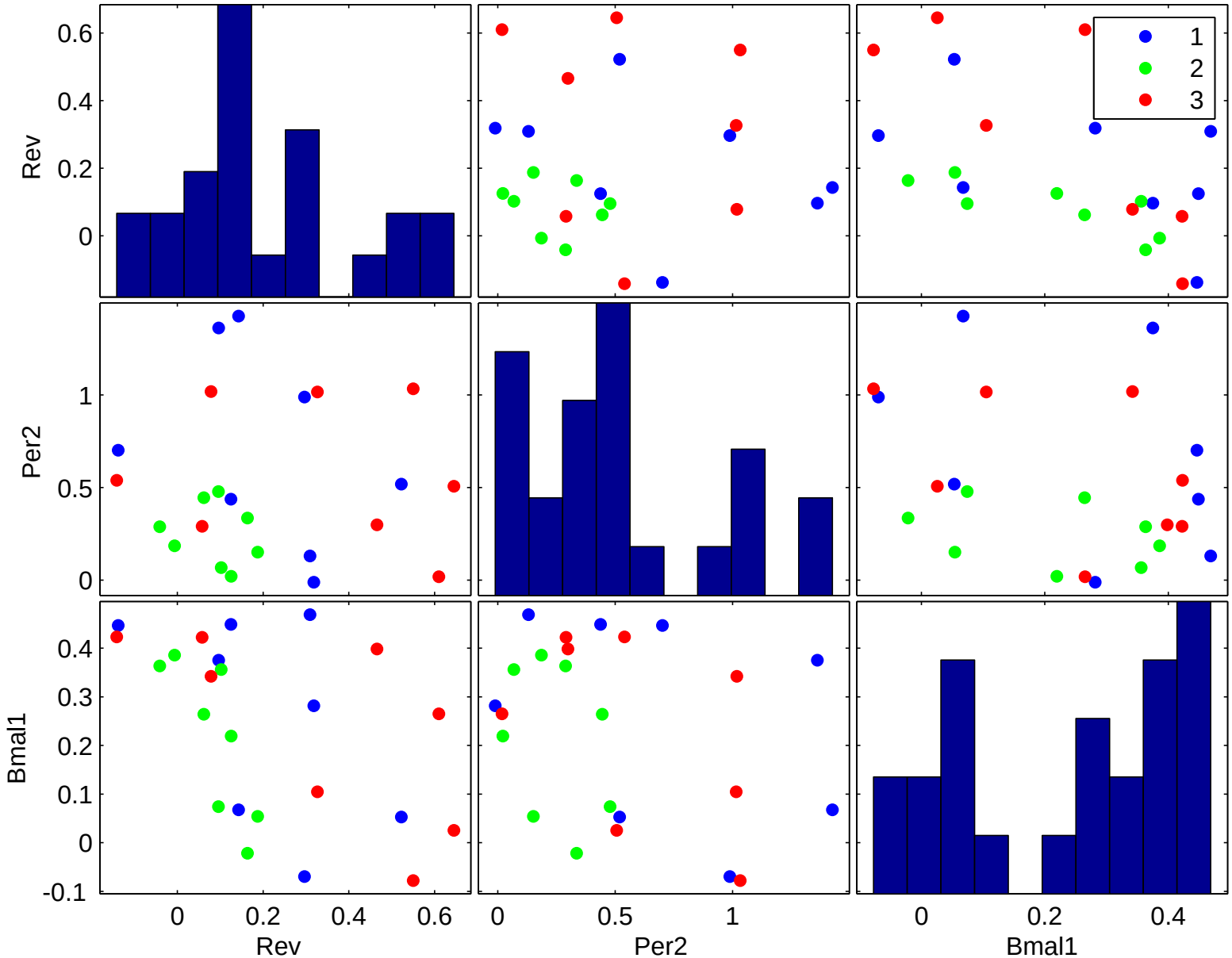
Liver: Time Series



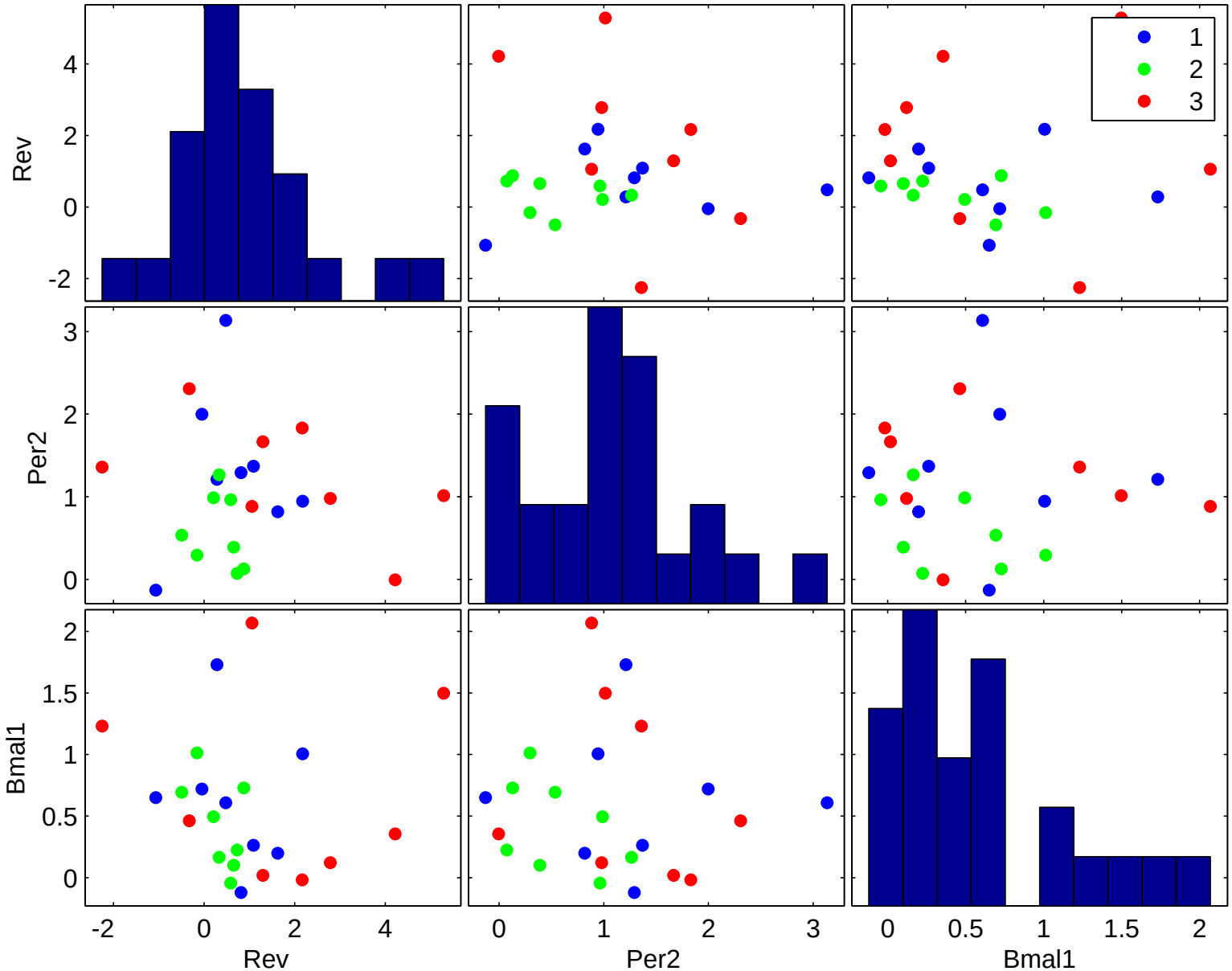
Fourier Transform



Genes Colon Time series: Clock

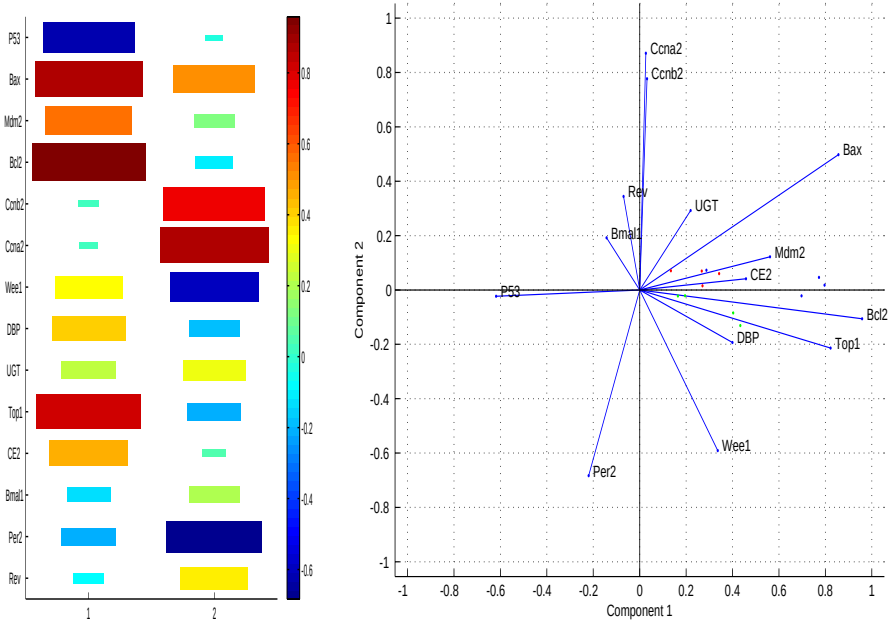


Genes Liver Time series: Clock

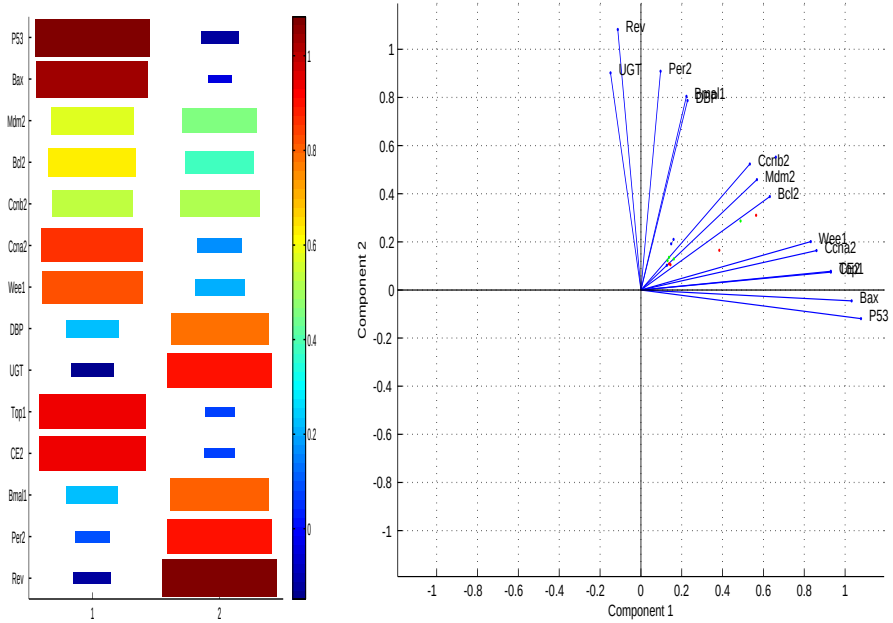


Factor Analysis: 2 factors: Colon

Time series



FT Amplitudes



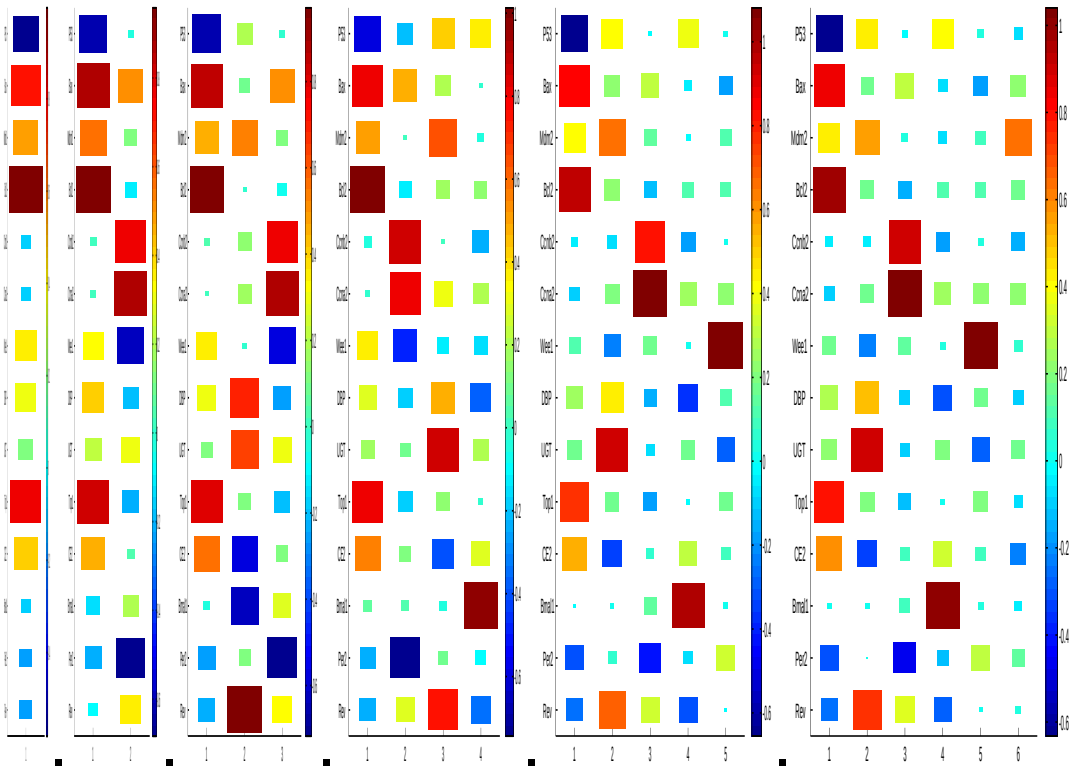
How to determine the number of factors

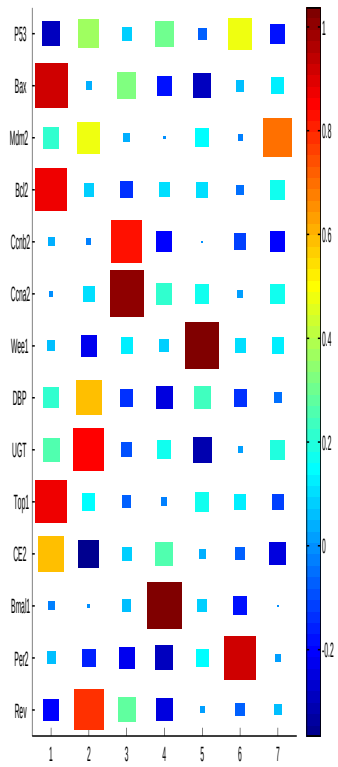
- When N is given:

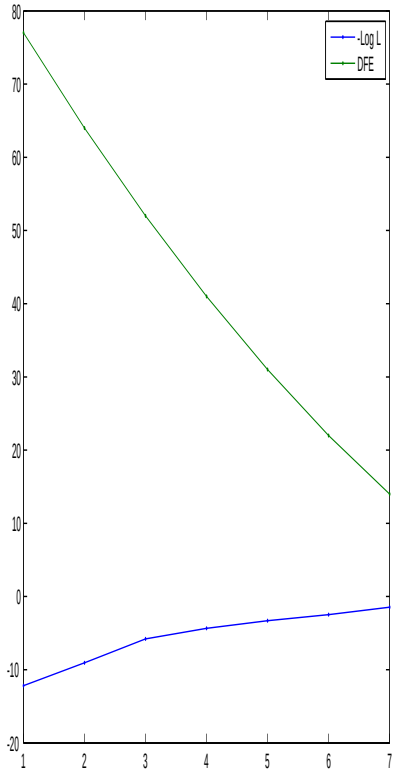
$$p(\boldsymbol{A}, \boldsymbol{f}|\boldsymbol{g}) \propto p(\boldsymbol{g}|\boldsymbol{A}, \boldsymbol{f})\, p(\boldsymbol{A})\, p(\boldsymbol{f})$$

  Different choices for $p(\boldsymbol{A})$ and $p(\boldsymbol{f})$ and
  Different methods to estimate both $\boldsymbol{A}$ and $\boldsymbol{f}$:
  JMAP, EM, Variational Bayesian Approximation
- When N is not known:
  - Model selection
  - Bayesian or Maximum likelihood methods
  - To determine the number of factors we do the analyze with different N factors and use two criteria:
  - -log likelihood $-\ln p(\boldsymbol{g}|\boldsymbol{A}, N)$ of the observations and
  - DFE: Degrees of freedom error $(N-M)^2 - (N+M))/2$ related to AIC or BIC model selection criteria.

Factor Analysis: Time series, colon







Dimension reduction: ML PPCA

- Forward model

$$\boldsymbol{g}(t) = \boldsymbol{A}\boldsymbol{f}(t) + \epsilon(t) \text{ or } \boldsymbol{G} = \boldsymbol{A}\boldsymbol{F} + \boldsymbol{E}$$

- ML PPCA:

$$\boldsymbol{\Sigma}_g = \text{cov}[\boldsymbol{g}] = \boldsymbol{A}\text{cov}[\boldsymbol{f}]\boldsymbol{A}^t + v_\epsilon \boldsymbol{I} = \boldsymbol{A}\boldsymbol{V}_f\boldsymbol{A}^t + v_\epsilon \boldsymbol{I}$$

$$\boldsymbol{\Sigma}_g = \boldsymbol{A}\text{diag}\left[\boldsymbol{v}_f\right]\boldsymbol{A}^t + v_\epsilon \boldsymbol{I}$$

- Likelihood

$$p(\boldsymbol{g}|\boldsymbol{A}, \boldsymbol{v}_f, v_\epsilon) = \mathcal{N}(\boldsymbol{g}|0, \boldsymbol{\Sigma}_g)$$

$$p(\boldsymbol{g}|\boldsymbol{A}, \boldsymbol{v}_f, v_\epsilon) \propto \det(\boldsymbol{\Sigma}_g)^{-1/2} \exp\left[-\frac{1}{2v_\epsilon}\sum_t \|\boldsymbol{g}(t) - \boldsymbol{A}\boldsymbol{f}(t)\|^2\right]$$

- Alternate maximization of the -log likelihood

$$\mathcal{L}(\boldsymbol{A}, \boldsymbol{v}_f, v_\epsilon) = \frac{1}{2}\ln\det(\boldsymbol{A}\text{diag}\left[\boldsymbol{v}_f\right]\boldsymbol{A}^t + v_\epsilon \boldsymbol{I}) + \frac{1}{2v_\epsilon}\sum_t \|\boldsymbol{g}(t) - \boldsymbol{A}\boldsymbol{f}(t)\|^2$$

with respect to its arguments can give the desired solution.

Dimension reduction, PCA, Factor Analysis, ICA

- M variables $\boldsymbol{g}(t)$ are observed. They are redundant. Can we express them with $N \leq M$ factors $\boldsymbol{f}$?
  How many factors (Principal Components, Independent Components) can describe the observed data?

$$\begin{cases} g_i(t) = \sum_{j=1}^{N} a_{ij} f_j(t) + \epsilon_i(t) \\ \boldsymbol{g}(t) = \boldsymbol{A}\boldsymbol{f}(t) + \boldsymbol{\epsilon}(t) \end{cases} \quad \begin{cases} \boldsymbol{A} : (M \times N) \text{ Loading matrix }, N \leq M \\ \boldsymbol{f}(\text{t}) \ : \ \textit{factors}, \textit{sources} \end{cases}$$

- How to find both $\boldsymbol{A}$ and factors $\boldsymbol{f}(t)$?

Three cases:

- $\boldsymbol{A}$ known, find $\boldsymbol{f}$
- $\boldsymbol{f}$ known, find $\boldsymbol{A}$
- $\boldsymbol{A}$ and $\boldsymbol{f}$ both uknown

Case 1: A known, find f

$$g(t) = Af(t) + \epsilon(t)$$

- First, assume the data iid. So $g = Af + \epsilon$.
- Second, assume no noise. So $g = Af$
- Ideal case $M = N$ and A invertible: $\widehat{f} = A^{-1}g$
- $M < N$: $Af = g$ has infinite number of solutions. Minimum Norme (MN) Solution:

$$\widehat{f} = \arg \min_{\{f: Af = g\}} \left\{ \|f\|^2 \right\}$$

Lagrangian technique: $AA^t f = A^t g$
If A has full rank: rank $(A) = \min\{M, N\}$

$$\widehat{f} = A^t [AA^t]^{-1} g$$

Case 1: $\boldsymbol{A}$ known, find $\boldsymbol{f}$: LS

- $M > N$: $\boldsymbol{A}\boldsymbol{f} = \boldsymbol{g}$ may not have any solution.
  Least Square (LS) Solution:

$$\widehat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} \left\{ \|\boldsymbol{g} - \boldsymbol{A}\boldsymbol{f}\|^2 \right\}$$

  Gradient of $J(\boldsymbol{f}) = \|\boldsymbol{g} - \boldsymbol{A}\boldsymbol{f}\|^2$ is :
  $-2\boldsymbol{A}^t(\boldsymbol{g} - \boldsymbol{A}\boldsymbol{f}) = 0 \longrightarrow \boldsymbol{A}^t\boldsymbol{A}\boldsymbol{f} = \boldsymbol{A}^t\boldsymbol{g}$.
- If $\boldsymbol{A}$ has full rank: rank $(\boldsymbol{A}) = \min\{M, N\}$:

$$\widehat{\boldsymbol{f}} = [\boldsymbol{A}^t\boldsymbol{A}]^{-1}\boldsymbol{A}^t\boldsymbol{g}$$

- General case ?
  - Truncated Singular Value Decomposition (TSVD)
  - Regularization
  - Bayesian

Case 1: A known, find f: TSVD

- SVD: $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^t, \quad \boldsymbol{U}\boldsymbol{U}^t = \boldsymbol{I}, \quad \boldsymbol{V}^t\boldsymbol{V} = \boldsymbol{I}, \quad \boldsymbol{\Lambda}$ diagonal.
- Right SVD: $\boldsymbol{A}\boldsymbol{A}^t\boldsymbol{v}_k = \lambda_k \boldsymbol{v}_k$
- Left SVD: $\boldsymbol{A}^t\boldsymbol{A}\boldsymbol{u}_k = \lambda_k \boldsymbol{u}_k$
- Full rank

$$\widehat{\boldsymbol{f}} = \boldsymbol{V}\boldsymbol{S}^{-1}\boldsymbol{U}^t\boldsymbol{g} = \sum_k \boldsymbol{v}_k \frac{< \boldsymbol{g}, \boldsymbol{u}_k >}{s_k}$$

- Rank deficient rank $(\boldsymbol{A}) = K < \min\{M, N\}$

$$\widehat{\boldsymbol{f}} = \boldsymbol{V}\boldsymbol{S}^{+}\boldsymbol{U}^t\boldsymbol{g} = \sum_k^K \boldsymbol{v}_k \frac{< \boldsymbol{g}, \boldsymbol{u}_k >}{s_k}$$

- Main difficulty: How to decide which singular values are near to zero.

Case 1: A known, find f: Regularization

- MN: Minimize $\|f\|^2$ subject to $Af = g$,
- LS: Minimize $\|g - Af\|^2$
- MN + LS:

$$\widehat{f} = \arg\min_{f} \{J(f)\}$$

with $J(f) = \|g - Af\|^2 + \lambda\|f\|^2$

- Gradient of $J(f)$ is :

$$-2A^t(g - Af) - 2\lambda I = 0 \longrightarrow (A^tA + \lambda I)f = A^tg$$

$$\widehat{f} = [A^tA + \lambda I]^{-1}A^tg$$

- $\lambda \longrightarrow 0$ LS.
- When $M < N$, we may also obtain

$$\widehat{f} = A^t[\mu AA^t + I]^{-1}g, \quad \text{with } \mu = \frac{1}{\lambda}.$$

Case 1: A known, find f: Bayesian inference

$$g = Af + \epsilon$$

- Prior knowledge on ϵ:

$$\epsilon \sim \mathcal{N}(\epsilon|0, v_\epsilon I) \longrightarrow p(g|f, A) = \mathcal{N}(g|Af, v_\epsilon I) \propto \exp\left[\frac{1}{2v_\epsilon}\|g - Af\|^2\right]$$

- Simple prior models for f: $p(f|\alpha) \propto \exp\left[-\frac{1}{2v_f}\|f\|^2\right]$
- Expression of the posterior law:

$$p(f|g, A) \propto p(g|f, A)\, p(f) \propto \exp\left[-\frac{1}{2v_\epsilon}J(f)\right]$$

$$\text{with} \qquad J(f) = \|g - Af\|^2 + \lambda\|f\|^2, \quad \lambda = v_\epsilon/v_f$$

- Link between MAP estimation and regularization

$$\widehat{f} = \arg\max_f \{p(f|g, A)\} = \arg\min_f \{J(f)\}$$

- Solution: $\widehat{f} = (A'A + \lambda I)^{-1} A'g$

Bayesian inference for sources $\boldsymbol{f}$ when $\boldsymbol{A}$ is known

- More general prior model $p(\boldsymbol{f}) \propto \exp\left[-\frac{\alpha}{2}\Omega(\boldsymbol{f})\right]$
- MAP:

$$J(\boldsymbol{f}) = \|\boldsymbol{g} - \boldsymbol{A}\boldsymbol{f}\|^2 + \lambda\Omega(\boldsymbol{f}), \quad \lambda = \nu_\epsilon\alpha$$

$$\boxed{p(\boldsymbol{f}|\boldsymbol{\theta}, \boldsymbol{g})} \longrightarrow \boxed{\begin{array}{c}\text{Optimization of} \\ J(\boldsymbol{f}) = \|\boldsymbol{g} - \boldsymbol{A}\boldsymbol{f}\|^2 + \lambda\Omega(\boldsymbol{f})\end{array}} \longrightarrow \widehat{\boldsymbol{f}}$$

- Different priors=Different expressions for $\Omega(\boldsymbol{f})$
- Solution can be obtained using appropriate optimization algorithm.

MAP estimation with sparsity enforcing priors

- Gaussian: $\Omega(\boldsymbol{f}) = \|\boldsymbol{f}\|^2 = \sum_j |f_j|^2$

$$J(\boldsymbol{f}) = \|\boldsymbol{g} - \boldsymbol{A}\boldsymbol{f}\|^2 + \lambda\|\boldsymbol{f}\|^2 \longrightarrow \widehat{\boldsymbol{f}} = [\boldsymbol{A}'\boldsymbol{A} + \lambda\boldsymbol{I}]^{-1}\boldsymbol{A}'\boldsymbol{g}$$

- Generalized Gaussian:

$$\Omega(\boldsymbol{f}) = \gamma \sum_j |f_j|^\beta)$$

- Student-t model:

$$\Omega(\boldsymbol{f}) = \frac{\nu+1}{2} \sum_j \log\left(1 + f_j^2/\nu\right)$$

- Elastic Net model:

$$\Omega(\boldsymbol{f}) = \sum_j \left[\gamma_1 |f_j| + \gamma_2 f_j^2\right]$$

For an extended list of such sparsity enforcing priors see:
A. Mohammad-Djafari, "Bayesian approach with prior models which enforce sparsity in signal and image processing," EURASIP Journal on Advances in Signal Processing, vol. Special issue on Sparse Signal Processing, 2012.

Case 1: A known, find f: Regularization

- More general case:

$$\widehat{f} = \arg\min_{f} \{J(f)\}$$

with $J(f) = \|g - Af\|^2 + \lambda\|Df\|^2$

- Gradient of $J(f)$ is :

$$-2A^t(g - Af) - 2D^tD = 0 \longrightarrow (A^tA + \lambda D^tD)f = A^tg$$

$$\widehat{f} = [A^tA + \lambda D^tD]^{-1}A^tg$$

- L1 Regularization:

$$J(f) = \|g - Af\|_2^2 + \lambda\|f\|_1$$

with $\|f\|_1 = \sum_j |f_j|$.

- Still more general

$$J(f) = \Delta_1(g, Af) + \lambda\Delta_2(f, f_0))$$

Case2: f known, find A

- $g = Af + \epsilon$ is linear in f and in A.

$$g = Af \longrightarrow g_i = \sum_j a_{ij} f_j \longrightarrow g = \sum_j a_{*j} f_j$$

$$\begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} f_1 & 0 & f_2 & 0 \\ 0 & f_1 & 0 & f_2 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{21} \\ a_{12} \\ a_{22} \end{bmatrix}$$

$$g = Af = Fa \quad \text{with} \quad F = f \odot I, \quad a = \text{vec}(A)$$

- A known, estimation of f: $g = Af + \epsilon$
- f known, estimation of A: $g = Fa + \epsilon$
- Joint estimation of f and A: $g = Af + \epsilon = Fa + \epsilon$

Cases 1 and 2: Regularization

$$\boldsymbol{g} = \boldsymbol{A}\boldsymbol{f} \quad \text{or} \quad \boldsymbol{g} = \boldsymbol{A}\boldsymbol{f} = \boldsymbol{F}\boldsymbol{a}$$

- $\boldsymbol{A}$ known, find $\boldsymbol{f}$

$$\widehat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} \left\{ \|\boldsymbol{g} - \boldsymbol{A}\boldsymbol{f}\|^2 + \lambda \|\boldsymbol{f}\|^2 \right\} = (\boldsymbol{A}'\boldsymbol{A} + \lambda \boldsymbol{I})^{-1} \boldsymbol{A}'\boldsymbol{g}$$

- $\boldsymbol{f}$ known, find $\boldsymbol{A}$

$$\widehat{\boldsymbol{A}} = \arg\min_{\boldsymbol{A}} \left\{ \|\boldsymbol{g} - \boldsymbol{A}\boldsymbol{f}\|^2 + \lambda \|\boldsymbol{A}\|^2 \right\} = \boldsymbol{g}\boldsymbol{f}'(\boldsymbol{f}\boldsymbol{f}' + \lambda \boldsymbol{I})^{-1}$$

- Both $\boldsymbol{A}$ and $\boldsymbol{f}$ are unknown

$$(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{A}}) = \arg\min_{(\boldsymbol{f}, \boldsymbol{A})} \left\{ \|\boldsymbol{g} - \boldsymbol{A}\boldsymbol{f}\|^2 + \lambda_1 \|\boldsymbol{f}\|^2 + \lambda_2 \|\boldsymbol{A}\|^2 \right\}$$

Alternate optimisation ?

Estimation of $\boldsymbol{A}$ when the sources $\boldsymbol{f}$ are known

Source separation is a bilinear model:

$$\boldsymbol{g} = \boldsymbol{A}\boldsymbol{f} = \boldsymbol{F}\boldsymbol{a}$$

$$\begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} f_1 & 0 & f_2 & 0 \\ 0 & f_1 & 0 & f_2 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{21} \\ a_{12} \\ a_{22} \end{bmatrix}$$

$$\boldsymbol{F} = \boldsymbol{f} \odot \boldsymbol{I}, \quad \boldsymbol{a} = \text{vec}(\boldsymbol{A})$$

- Problem is more ill-posed (underdetermined).
- We need absolutely to impose constraintes on elements or the structure of $\boldsymbol{A}$, for example:
  - Positivity of the elements
  - Toeplitz or TBT structure
  - Symmetry
  - Sparsity
- The same Bayesian approach then can be applied.

Estimation of A when the sources f are known

$$g = Af + \epsilon = Fa + \epsilon$$

- Prior on noise:

$$p(g|f, A) = \mathcal{N}(g|Af, v_\epsilon I) \propto \exp\left[\frac{1}{2v_\epsilon}\|g - Af\|^2\right]$$
$$\propto \exp\left[\frac{1}{2v_\epsilon}\|g - Fa\|^2\right]$$

- Simple prior models for a:

$$p(A|\alpha) \propto \exp\left[-\alpha\|a\|^2\right] \propto \exp\left[-\alpha\|A\|^2\right]$$

- Expression of the posterior law:

$$p(A|g, f) \propto p(g|f, A)\, p(A) \propto \exp\left[-J(A)\right]$$

$$\text{with} \qquad J(A) = \frac{1}{2v_\epsilon}\|g - Af\|^2 + \alpha\|A\|^2$$

- MAP estimation:

$$\widehat{a} = (F'F + \lambda I)^{-1} F'g \leftrightarrow \widehat{A} = gf'(ff' + \lambda I)^{-1}$$

$$g(t) = Af(t) + \epsilon(t) \longrightarrow \widehat{A} = \left(\sum_t g(t)f'(t)\right)\left(\sum_t f(t)f'(t) + \lambda I\right)^{-1}$$

Case 3: both f and A unknown

- Both A and f are unknown:

$$(\widehat{f}, \widehat{A}) = \arg\min_{(f,A)} \left\{ \|g - Af\|^2 + \lambda_1 \|f\|^2 + \lambda_2 \|A\|^2 \right\}$$

- Undeterminations:
  - Permutation: $AP, P'f$
  - Scale: $kA, \frac{1}{k}f$
- Alternate optimisation

$$\begin{cases} \widehat{f} = \arg\min_f \left\{ \|g - Af\|^2 + \lambda_1 \|f\|^2 \right\} = (A'A + \lambda_1 I)^{-1} A'g \\ \widehat{A} = \arg\min_A \left\{ \|g - Af\|^2 + \lambda_2 \|A\|^2 \right\} = gf'(ff' + \lambda_2 I) \end{cases}$$

- Importance of initialization and other constraintes such as positivity
  - Non-negative Matrix decomposition

Both $\boldsymbol{A}$ and $\boldsymbol{f}$ unknown: Bayesian approach

Three main steps:

1. Assigning priors: $p(\boldsymbol{f})$ and $p(\boldsymbol{A})$
2. Obtaining the expressions of the joint posterior:
$$p(\boldsymbol{f}, \boldsymbol{A}|\boldsymbol{g}) \propto p(\boldsymbol{g}|\boldsymbol{f}, \boldsymbol{A})\, p(\boldsymbol{f})\, p(\boldsymbol{A})$$
3. Doing the computations:
   - Joint optimization of $p(\boldsymbol{f}, \boldsymbol{A}|\boldsymbol{g})$;
   - MCMC Gibbs sampling methods which need generation of samples from the conditionals $p(\boldsymbol{f}|\boldsymbol{A}, \boldsymbol{g})$ and $p(\boldsymbol{A}|\boldsymbol{f}, \boldsymbol{g})$;
   - Marginalisation and EM algorithm:

$$p(\boldsymbol{A}|\boldsymbol{g}) = \int p(\boldsymbol{f}, \boldsymbol{A}|\boldsymbol{g})\, \mathrm{d}\boldsymbol{f} \longrightarrow \widehat{\boldsymbol{A}} \longrightarrow p(\boldsymbol{f}|\widehat{\boldsymbol{A}}, \boldsymbol{g}) \longrightarrow \widehat{\boldsymbol{f}}$$

   - Bayesian Variational Approximation (BVA) methods which approximate $p(\boldsymbol{f}, \boldsymbol{A}|\boldsymbol{g})$ by a separable one $q(\boldsymbol{f}, \boldsymbol{A}) = q_1(\boldsymbol{f})q_2(\boldsymbol{A})$ and then using them for the estimation of $\boldsymbol{f}$ and $\boldsymbol{A}$.

Bayesian source separation: both $\boldsymbol{A}$ and $\boldsymbol{f}$ unknown

$$p(\boldsymbol{f}, \boldsymbol{A}|\boldsymbol{g}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3) = \frac{p(\boldsymbol{g}|\boldsymbol{f}, \boldsymbol{A}, \boldsymbol{\theta}_1)\, p(\boldsymbol{f}|\boldsymbol{\theta}_2)\, p(\boldsymbol{A}|\boldsymbol{\theta}_3)}{p(\boldsymbol{g}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3)}$$

- Joint estimation (JMAP):
  $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{A}}) = \arg\max_{(\boldsymbol{f}, \boldsymbol{A})} \{p(\boldsymbol{f}, \boldsymbol{A}|\boldsymbol{g}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3)\}$
- JMAP with Gaussian priors:

$$(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{A}}) = \arg\min_{(\boldsymbol{f}, \boldsymbol{A})} \left\{ \|\boldsymbol{g} - \boldsymbol{A}\boldsymbol{f}\|^2 + \lambda_1 \|\boldsymbol{f}\|^2 + \lambda_2 \|\boldsymbol{A}\|^2 \right\}$$

- Permutation and scale ideterminations:
  needs good choices for priors
- Alternate optimisation

$$\begin{cases} \widehat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} \left\{ \|\boldsymbol{g} - \boldsymbol{A}\boldsymbol{f}\|^2 + \lambda_1 \|\boldsymbol{f}\|^2 \right\} = (\boldsymbol{A}'\boldsymbol{A} + \lambda_1 \boldsymbol{I})^{-1} \boldsymbol{A}'\boldsymbol{g} \\ \widehat{\boldsymbol{A}} = \arg\min_{\boldsymbol{A}} \left\{ \|\boldsymbol{g} - \boldsymbol{A}\boldsymbol{f}\|^2 + \lambda_2 \|\boldsymbol{A}\|^2 \right\} = \boldsymbol{g}\boldsymbol{f}'(\boldsymbol{f}\boldsymbol{f}' + \lambda_2 \boldsymbol{I}) \end{cases}$$

- Importance of initialization and constraintes such as positivity

Joint MAP Estimation of A and f with Gaussian priors

$$g(t) = A\, f(t) + \epsilon(t), \quad t = 1, \cdots, T, \quad iid$$

$$\begin{aligned} p(f_{1..T}, A | g_{1..T}) &\propto p(g_{1..T}|A, f_{1..T}, v_\epsilon)\, p(f_{1..T})\, p(A|A_0, V_0) \\ &\propto \prod_t p(g(t)|A, f(t), v_\epsilon)\, p(f(t)|z(t))\, p(A|A_0, V_0) \end{aligned}$$

Joint MAP: Alternate optimization

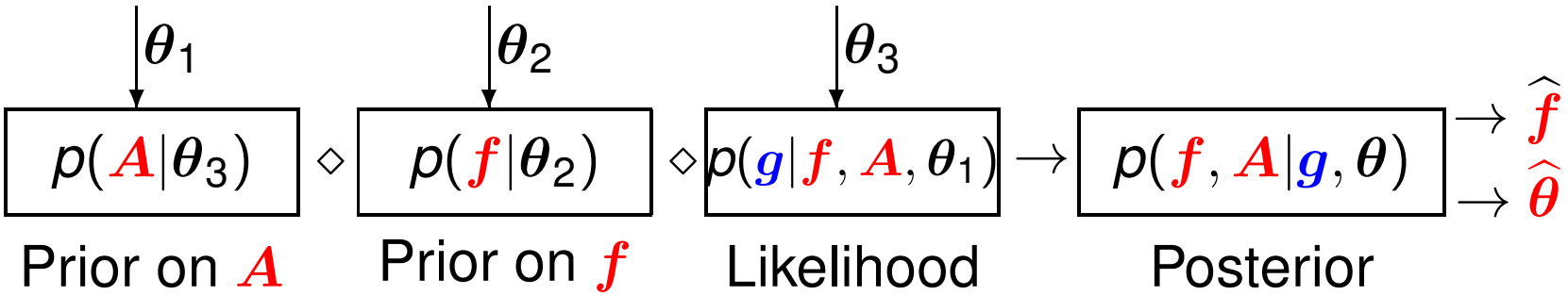
$$\begin{cases} \widehat{f}(t) = (\widehat{A}'\widehat{A} + \lambda_f I)^{-1} \widehat{A}' g(t), & \lambda_f = v_\epsilon / v_f \\ \widehat{A} = \sum_t g(t)\widehat{f}'(t) \left(\sum_t \widehat{f}(t)\widehat{f}'(t) + \lambda_a I \right)^{-1} & \lambda_a = v_\epsilon / v_a \end{cases}$$

Alternate optimization Algorithm:

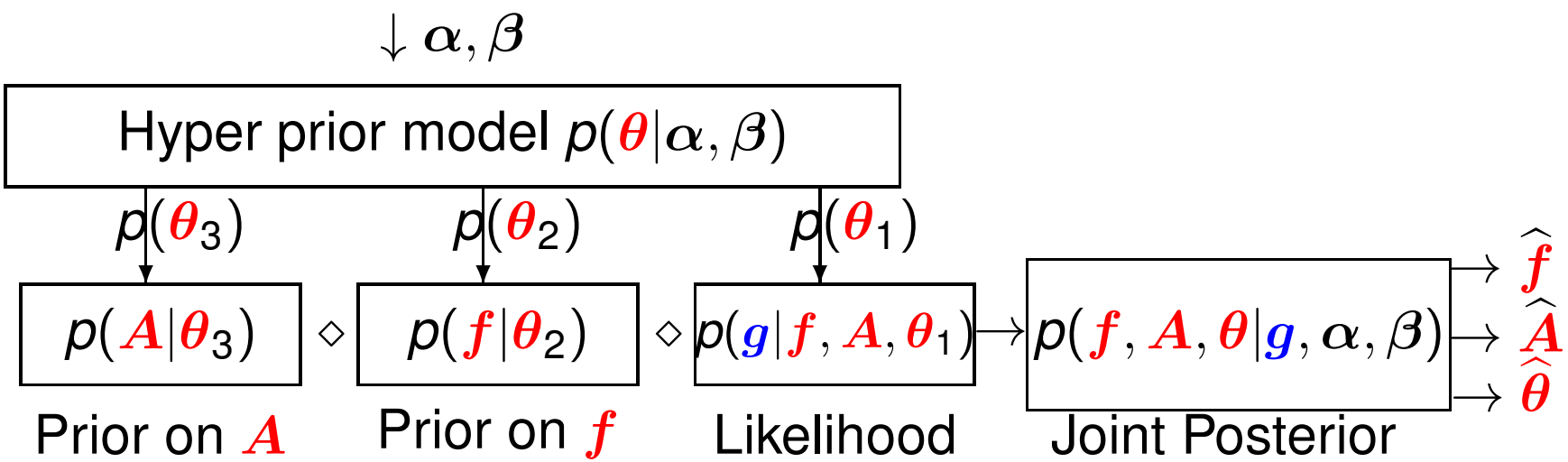
$$\begin{array}{ccccc} A^{(0)} & \longrightarrow & \widehat{A} & \longrightarrow & \boxed{\left(\widehat{A}'\widehat{A} + \lambda_f I\right)^{-1} \widehat{A}' g} \longrightarrow \widehat{f}(t) \\ & & \uparrow & & \downarrow \\ & & \widehat{A} & \longleftarrow & \boxed{\left(\sum_t g(t)\widehat{f}'(t)\right) \left(\sum_t \widehat{f}(t)\widehat{f}'(t) + \lambda_a I\right)^{-1}} \longleftarrow \widehat{f}(t) \end{array}$$

Summary of Bayesian estimation with different levels

- Simple Bayesian Model and Estimation



- Full Bayesian Model and Hyperparameter Estimation scheme



Summary of Bayesian estimation with different levels

- Marginalization over f

$$\boxed{p(f, A|g)} \longrightarrow \boxed{p(A|g)} \longrightarrow \widehat{A} \rightarrow \boxed{p(f|\widehat{A}, g)} \rightarrow \widehat{f}$$

Joint Posterior Marginalize over f

- Marginalization over A

$$\boxed{p(f, A|g)} \longrightarrow \boxed{p(f|g)} \longrightarrow \widehat{f} \rightarrow \boxed{p(A|\widehat{f}, g)} \rightarrow \widehat{A}$$

Joint Posterior Marginalize over A

- Joint MAP

$$\boxed{p(f, A|g)} \longrightarrow \boxed{\begin{array}{l} \widetilde{f} = \arg\max_f \left\{p(f, \widetilde{A}|g)\right\} \\ \widetilde{A} = \arg\max_A \left\{p(\widetilde{f}, A|g)\right\} \end{array}} \begin{array}{l} \longrightarrow \widehat{A} \\ \longrightarrow \widehat{f} \end{array}$$

Joint Posterior Alternate optimization

- Other solutions:
  - MCMC for exploring the joint posterior law and computing mean values
  - Variational Bayesian Approximation (VBA)

Variational Bayesian Approximation

- Main idea: Approximate the joint pdf $p(\boldsymbol{A}, \boldsymbol{f}|\boldsymbol{g})$ difficult to handle by a simpler one. For example a separable one $q(\boldsymbol{A}, \boldsymbol{f}|\boldsymbol{g}) = q_1(\boldsymbol{A})\, q_2(\boldsymbol{f})$ or even $q_1(\boldsymbol{A}) \prod_j q_{2j}(\boldsymbol{f}_j)$.
- Criterion: minimize

$$\mathrm{KL}(q|p) = \int q \ln \frac{q}{p} = -H(q_1) - H(q_2) - < \ln p(\boldsymbol{A}, \boldsymbol{f}|\boldsymbol{g}) >_q$$

- Solution obtained by alternate optimization:

$$\begin{cases} q_1(\boldsymbol{f}) \propto \exp\left[< \ln p(\boldsymbol{f}, \boldsymbol{A}|\boldsymbol{g}) >_{q_2(\boldsymbol{A})}\right] \\ q_2(\boldsymbol{A}) \propto \exp\left[< \ln p(\boldsymbol{f}, \boldsymbol{A}|\boldsymbol{g}) >_{q_1(\boldsymbol{f})}\right] \end{cases}$$

- Use $q_1(\boldsymbol{f})$ for infering on $\boldsymbol{f}$ and $q_2(\boldsymbol{A})$ for infering on $\boldsymbol{A}$

VBA and links with JMAP and EM

Three possibilities:

- $q_1(\boldsymbol{f}) = \delta(\boldsymbol{f} - \widetilde{\boldsymbol{f}})$ and $q_2(\boldsymbol{A}) = \delta(\boldsymbol{A} - \widetilde{\boldsymbol{A}}) \longrightarrow$ JMAP

$$\left\{\begin{array}{l} q_1(\boldsymbol{f}) \propto p(\boldsymbol{f}, \widetilde{\boldsymbol{A}}|\boldsymbol{g}) \\ q_2(\boldsymbol{A}) \propto p(\widetilde{\boldsymbol{f}}, \boldsymbol{A}|\boldsymbol{g}) \end{array}\right. \longrightarrow \left\{\begin{array}{l} \widetilde{\boldsymbol{f}} = \arg\max_{\boldsymbol{f}} \left\{p(\boldsymbol{f}, \widetilde{\boldsymbol{A}}|\boldsymbol{g})\right\} \\ \widetilde{\boldsymbol{A}} = \arg\max_{\boldsymbol{A}} \left\{p(\widetilde{\boldsymbol{f}}, \boldsymbol{A}|\boldsymbol{g})\right\} \end{array}\right.$$

- $q_1(\boldsymbol{f})$ free and $q_2(\boldsymbol{A}) = \delta(\boldsymbol{A} - \widetilde{\boldsymbol{A}}) \longrightarrow$ EM

$$\left\{\begin{array}{l} q_1(\boldsymbol{f}) \propto \exp\left[\langle \ln p(\boldsymbol{f}, \boldsymbol{A}|\boldsymbol{g})\rangle_{q_2(\boldsymbol{A})}\right] \propto p(\boldsymbol{f}, \widetilde{\boldsymbol{A}}|\boldsymbol{g}) = p(\boldsymbol{f}|\widetilde{\boldsymbol{A}}, \boldsymbol{g}) \\ q_2(\boldsymbol{A}) \propto \exp\left[\langle \ln p(\boldsymbol{f}, \boldsymbol{A}|\boldsymbol{g})\rangle_{q_1(\boldsymbol{f})}\right] \propto \exp\left[Q(\boldsymbol{A}, \widetilde{\boldsymbol{A}})\right] \end{array}\right.$$

$$\longrightarrow \text{EM} \left\{\begin{array}{l} Q(\boldsymbol{A}, \widetilde{\boldsymbol{A}}) = \langle \ln p(\boldsymbol{f}, \boldsymbol{A}|\boldsymbol{g})\rangle_{p(\boldsymbol{f}|\widetilde{\boldsymbol{A}}, \boldsymbol{g})} \\ \widetilde{\boldsymbol{A}} = \arg\max_{\boldsymbol{A}} \left\{Q(\boldsymbol{A}, \widetilde{\boldsymbol{A}})\right\} \end{array}\right.$$

- $q_1(\boldsymbol{f})$ and $q_2(\boldsymbol{A})$ free forms:
  Affordable if exponential families and conjugate priors

JMAP, EM and VBA

JMAP Alternate optimization Algorithm:

$$\begin{array}{ccccccc} \boldsymbol{A}^{(0)} & \longrightarrow & \widetilde{\boldsymbol{A}} & \longrightarrow & \boxed{\widetilde{\boldsymbol{f}} = \arg\max_{\boldsymbol{f}} \left\{ p(\boldsymbol{f}, \widetilde{\boldsymbol{A}} | \boldsymbol{g}) \right\}} & \longrightarrow \widetilde{\boldsymbol{f}} \longrightarrow \widehat{\boldsymbol{f}} \\ & & \uparrow & & & & \downarrow \\ \widehat{\boldsymbol{A}} & \longleftarrow & \widetilde{\boldsymbol{A}} & \longleftarrow & \boxed{\widetilde{\boldsymbol{A}} = \arg\max_{\boldsymbol{A}} \left\{ p(\widetilde{\boldsymbol{f}}, \boldsymbol{A} | \boldsymbol{g}) \right\}} & \longleftarrow \widetilde{\boldsymbol{f}} \end{array}$$

EM:

$$\begin{array}{ccccccc} \boldsymbol{A}^{(0)} & \longrightarrow & \widetilde{\boldsymbol{A}} & \longrightarrow & \boxed{q_1(\boldsymbol{f}) = p(\boldsymbol{f} | \widetilde{\boldsymbol{A}}, \boldsymbol{g})} & \longrightarrow q_1(\boldsymbol{f}) \longrightarrow \widehat{\boldsymbol{f}} \\ & & \uparrow & & & & \downarrow \\ \widehat{\boldsymbol{A}} & \longleftarrow & \widetilde{\boldsymbol{A}} & \longleftarrow & \boxed{\begin{array}{c} Q(\boldsymbol{A}, \widetilde{\boldsymbol{A}}) = \langle \ln p(\boldsymbol{f}, \boldsymbol{A} | \boldsymbol{g}) \rangle_{q_1(\boldsymbol{f})} \\ \widetilde{\boldsymbol{A}} = \arg\max_{\boldsymbol{A}} \left\{ Q(\boldsymbol{A}, \widetilde{\boldsymbol{A}}) \right\} \end{array}} & \longleftarrow q_1(\boldsymbol{f}) \end{array}$$

VBA:

$$\begin{array}{ccccccc} \boldsymbol{A}^{(0)} & \longrightarrow & q_2(\boldsymbol{A}) & \longrightarrow & \boxed{q_1(\boldsymbol{f}) \propto \exp\left[\langle \ln p(\boldsymbol{f}, \boldsymbol{A} | \boldsymbol{g}) \rangle_{q_2(\boldsymbol{A})} \right]} & \longrightarrow q_1(\boldsymbol{f}) \longrightarrow \widehat{\boldsymbol{f}} \\ & & \uparrow & & & & \downarrow \\ \widehat{\boldsymbol{A}} & \longleftarrow & q_2(\boldsymbol{A}) & \longleftarrow & \boxed{q_2(\boldsymbol{A}) \propto \exp\left[\langle \ln p(\boldsymbol{f}, \boldsymbol{A} | \boldsymbol{g}) \rangle_{q_1(\boldsymbol{f})} \right]} & \longleftarrow q_1(\boldsymbol{f}) \end{array}$$

Summary of Bayesian estimation with different levels

- Marginalization over f

$$\boxed{p(f, A|g)} \longrightarrow \boxed{p(A|g)} \longrightarrow \widehat{A} \rightarrow \boxed{p(f|\widehat{A}, g)} \rightarrow \widehat{f}$$

Joint Posterior Marginalize over f

- Marginalization over A

$$\boxed{p(f, A|g)} \longrightarrow \boxed{p(f|g)} \longrightarrow \widehat{f} \rightarrow \boxed{p(A|\widehat{f}, g)} \rightarrow \widehat{A}$$

Joint Posterior Marginalize over A

- Joint MAP

$$\boxed{p(f, A|g)} \longrightarrow \boxed{\begin{array}{l} \widetilde{f} = \arg\max_f \left\{p(f, \widetilde{A}|g)\right\} \\ \widetilde{A} = \arg\max_A \left\{p(\widetilde{f}, A|g)\right\} \end{array}} \begin{array}{l} \longrightarrow \widehat{A} \\ \longrightarrow \widehat{f} \end{array}$$

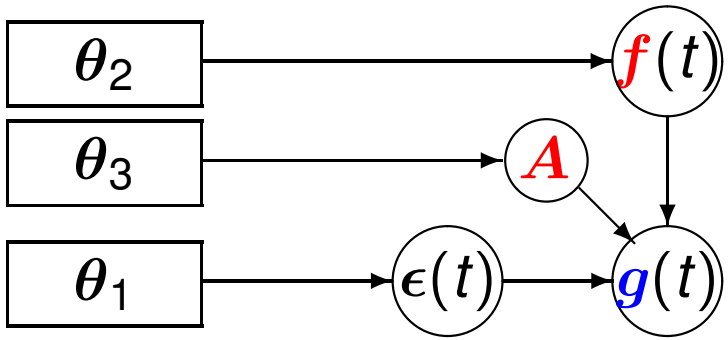
Joint Posterior Alternate optimization

- VBA

$$\boxed{p(f, A|g)} \longrightarrow \boxed{\begin{array}{l} q_1(f) \propto \exp\left[\langle \ln p(f, A|g)\rangle_{q_2(A)}\right] \\ q_2(A) \propto \exp\left[\langle \ln p(f, A|g)\rangle_{q_1(f)}\right] \end{array}} \begin{array}{l} \longrightarrow q_2(A) \longrightarrow \widehat{A} \\ \longrightarrow q_1(f) \longrightarrow \widehat{f} \end{array}$$

Joint Posterior VB Approximation

General case and Link with other methods



$$p(f(t)|\theta_2)$$
$$p(A|\theta_3) \qquad \rightarrow p(f(t), A|g(t), \theta)$$
$$p(g(t)|A, f(t), \theta_1)$$

$$p(f_{1..T}, A|g_{1..T}) \propto p(g_{1..T}|A, f_{1..T}, \theta_1)\, p(f_{1..T}|\theta_2)\, p(A|\theta_3)$$

Different scenarios:

- IID (strict stationnarity) $g(t) = A\, f(t) + \epsilon(t) \rightarrow g = A\, f + \epsilon$
- Second order stationnarity (Gaussian):
  All the probability laws are Gaussian
- Non Gaussian but IID: ICA
- Non-Gaussian, Time dependent, ...

Gaussian white case: PCA, MNF, PMF and NMF

- White and Gaussian signals $\boldsymbol{f}(t), \epsilon(t) \longrightarrow \boldsymbol{g}(t)$:

$$\boldsymbol{g}(t) = \boldsymbol{A}\,\boldsymbol{f}(t) + \boldsymbol{\epsilon}(t) \longrightarrow \boldsymbol{g} = \boldsymbol{A}\,\boldsymbol{f} + \boldsymbol{\epsilon}$$

- Likelihood:

$$p(\boldsymbol{g}|\boldsymbol{A}, \boldsymbol{f}) = \mathcal{N}(\boldsymbol{g}|\boldsymbol{A}\boldsymbol{f}, \boldsymbol{\Sigma}_\epsilon), \qquad p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}|0, \boldsymbol{\Sigma}_f)$$

$$\longrightarrow p(\boldsymbol{g}|\boldsymbol{A}) = \mathcal{N}(\boldsymbol{g}|0, \boldsymbol{A}\boldsymbol{\Sigma}_f\boldsymbol{A}' + \boldsymbol{\Sigma}_\epsilon)$$

- PCA : Estimate $\boldsymbol{\Sigma}_g$ by $\frac{1}{T}\sum_t \boldsymbol{g}(t)\boldsymbol{g}'(t)$, svd and keep all the non-zero svd: $\boldsymbol{\Sigma}_g = \boldsymbol{A}\boldsymbol{\Sigma}_f\boldsymbol{A}'$
- Minimum Norm Factorization (MNF) : Estimate $\boldsymbol{\Sigma}_g$, svd and keep all svd $\geq \sigma_\epsilon$: $\boldsymbol{\Sigma}_g = \boldsymbol{A}\boldsymbol{\Sigma}_f\boldsymbol{A}' + \boldsymbol{\Sigma}_\epsilon$
- Positive Matrix Factorization (MNF) : Decompose $\boldsymbol{\Sigma}_g$ in positive definite matrices [Paatero & Tapper, 94]
- Non-negative Matrix Factorization (NMF) : Decompose $\boldsymbol{\Sigma}_g$ in Non-negative definite matrices [Lee & Seung,99]

Non Gaussian white case: ICA

- White Non Gaussian signals and Exact model (no noise):

$$\boldsymbol{f}(t) \longrightarrow \boldsymbol{g}(t) \longrightarrow \boldsymbol{y}(t) = \boldsymbol{A}^{-1}\boldsymbol{g}(t) \longrightarrow \boldsymbol{y}(t) = \boldsymbol{B}\boldsymbol{g}(t)$$

- ICA: Find $\boldsymbol{B}$ in such a way that the components of $\boldsymbol{y}$ be the most independent
- Different measures of independencies:
  Entropy:

$$H(\boldsymbol{y}) = -\int p(y_i) \ln p(y_i)\, \mathrm{d}y_i$$

  Relative entropy or Kullback-Leibler divergence:

$$\mathrm{KL}(p(\boldsymbol{y}) : \prod_i p(y_i)) = \int p(y_i) \ln \frac{\prod_i p(y_i)}{p(\boldsymbol{y}} \, \mathrm{d}y_i$$

- Different choices and approximations for $p(y_i) \longrightarrow$ contrast functions, cumulants basis criteria

Non Gaussian white case: Maximum Likelihood

- White Non Gaussian signals (Accounting for noise)

$$\boldsymbol{g}(t) = \boldsymbol{A}\,\boldsymbol{f}(t) + \boldsymbol{\epsilon}(t) \longrightarrow \boldsymbol{g} = \boldsymbol{A}\,\boldsymbol{f} + \boldsymbol{\epsilon}$$

- Likelihood:

$$p(\boldsymbol{g}|\boldsymbol{A}, \boldsymbol{\Sigma}_\epsilon) = \int p(\boldsymbol{g}|\boldsymbol{A}, \boldsymbol{f}, \boldsymbol{\Sigma}_\epsilon) p(\boldsymbol{f})\, \mathrm{d}\boldsymbol{f}$$

- ICA (Maximum Likelihood) :

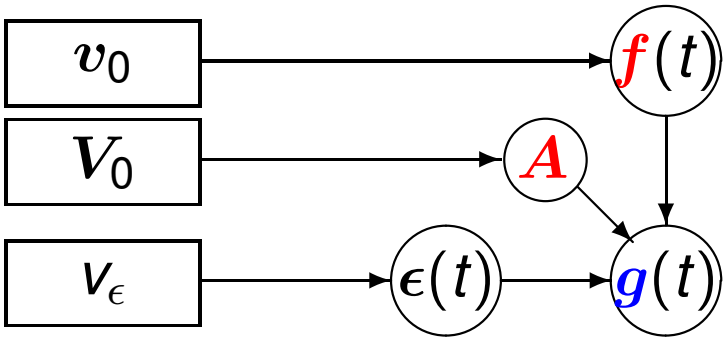
$$\widehat{\boldsymbol{\theta}} = (\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{\Sigma}_\epsilon}) = \arg\max_{\boldsymbol{\theta}} \{p(\boldsymbol{g}|\boldsymbol{\theta}\}$$

- EM iterative algorithm :

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}') &= \mathrm{E}\left\{\ln p(\boldsymbol{g}, \boldsymbol{f}, \boldsymbol{\theta}|\boldsymbol{\theta}'\right\} \\ \boldsymbol{\theta}' &= \arg\max_{\boldsymbol{\theta}} \left\{Q(\boldsymbol{\theta}, \boldsymbol{\theta}')\right\} \end{aligned}$$

-

General Gaussian case: Joint Estimation of A and f



$$p(f_j(t)|v_{0j}) = \mathcal{N}(f_j(t|0, v_{0j})$$
$$p(f(t)|v_0) \propto \exp\left[-\frac{1}{2}\sum_j f_j^2(t)/v_{0j}\right]$$
$$p(A_{ij}|0, V_{0ij}) = \mathcal{N}(A_{ij}|0, V_{0ij})$$
$$p(A|0, V_0) = \mathcal{N}(A|0, V_0)$$
$$p(g(t)|A, f(t), v_\epsilon) = \mathcal{N}(Af(t), v_\epsilon I)$$

$$\begin{aligned} p(f_{1..T}, A|g_{1..T}) \propto & \quad p(g_{1..T}|A, f_{1..T}, v_\epsilon)\, p(f_{1..T}|v_0)\, p(A|0, V_0) \\ \propto & \quad \prod_t p(g(t)|A, f(t), v_\epsilon)\, p(f(t)|v_0)\, p(A|0, V_0) \end{aligned}$$

$$p(f(t)|g_{1..T}, A, v_\epsilon, v_0, V_0) = \mathcal{N}(f(t)|\widehat{f}(t), \widehat{\Sigma})$$
$$p(A|g_{1..T}, f_{1..T}, v_\epsilon, v_0, V_0) = \mathcal{N}(A|\widehat{A}, \widehat{V})$$

Two approaches:

- Alternate joint MAP (JMAP) estimation
- Bayesian Variational Approximation

Joint Estimation of $\boldsymbol{A}$ and $\boldsymbol{f}$: Alternate JMAP

Some simplification:

$\boldsymbol{v}_0 = [v_f, .., v_f]'$, All sources a priori same variance v_f

$\boldsymbol{v}_\epsilon = [v_\epsilon, .., v_\epsilon]'$, All noise terms a priori same variance v_ϵ

$\boldsymbol{A}_0 = 0, \quad \boldsymbol{V}_0 = v_a \boldsymbol{I}$

$$p(\boldsymbol{f}(t)|\boldsymbol{g}(t), \boldsymbol{A}, v_\epsilon, \boldsymbol{v}_0) = \mathcal{N}(\boldsymbol{f}(t)|\widehat{\boldsymbol{f}}(t), \widehat{\boldsymbol{\Sigma}})$$

$$\begin{cases} \widehat{\boldsymbol{\Sigma}} = (\boldsymbol{A}'\boldsymbol{A} + \lambda_f \boldsymbol{I})^{-1} \\ \widehat{\boldsymbol{f}}(t) = (\boldsymbol{A}'\boldsymbol{A} + \lambda_f \boldsymbol{I})^{-1} \boldsymbol{A}'\boldsymbol{g}(t), \quad \lambda_f = v_\epsilon / v_f \end{cases}$$

$$p(\boldsymbol{A}|\boldsymbol{g}(t), \boldsymbol{f}(t), v_\epsilon, \boldsymbol{A}_0, \boldsymbol{V}_0) = \mathcal{N}(\boldsymbol{A}|\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{V}})$$

$$\begin{cases} \widehat{\boldsymbol{V}} = (\boldsymbol{F}'\boldsymbol{F} + \lambda_f \boldsymbol{I})^{-1} \\ \widehat{\boldsymbol{A}} = \left(\sum_t \boldsymbol{g}(t)\boldsymbol{f}'(t)\right)\left(\sum_t \boldsymbol{f}(t)\boldsymbol{f}'(t) + \lambda_a \boldsymbol{I}\right)^{-1}, \quad \lambda_a = v_\epsilon / v_a \end{cases}$$

JMAP:

$$\boldsymbol{A}^{(0)} \longrightarrow \widehat{\boldsymbol{A}} \longrightarrow \boxed{\left(\widehat{\boldsymbol{A}}'\widehat{\boldsymbol{A}} + \lambda_f \boldsymbol{I}\right)^{-1} \widehat{\boldsymbol{A}}'\boldsymbol{g}} \longrightarrow \widehat{\boldsymbol{f}}(t)$$

$$\uparrow \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \downarrow$$

$$\widehat{\boldsymbol{A}} \longleftarrow \boxed{\left(\sum_t \boldsymbol{g}(t)\widehat{\boldsymbol{f}}'(t)\right)\left(\sum_t \widehat{\boldsymbol{f}}(t)\widehat{\boldsymbol{f}}'(t) + \lambda_a \boldsymbol{I}\right)^{-1}} \longleftarrow \widehat{\boldsymbol{f}}(t)$$

Joint Estimation: Variational Bayesian Approximation

$$p(\boldsymbol{f}(t), \boldsymbol{A}|\boldsymbol{g}(t)) \longrightarrow q_1(\boldsymbol{f}(t))\, q_2(\boldsymbol{A})$$

$$q_1(\boldsymbol{f}(t)|\boldsymbol{g}(t), \widetilde{\boldsymbol{A}}, v_\epsilon, \boldsymbol{v}_0, \boldsymbol{V}_0) = \mathcal{N}(\boldsymbol{f}(t)|\widetilde{\boldsymbol{f}}(t), \widetilde{\boldsymbol{\Sigma}})$$

$$\begin{cases} \widetilde{\boldsymbol{\Sigma}} = (\widetilde{\boldsymbol{A}}'\widetilde{\boldsymbol{A}} + \lambda_f \widetilde{\boldsymbol{V}})^{-1} \\ \widetilde{\boldsymbol{f}}(t) = (\widetilde{\boldsymbol{A}}'\widetilde{\boldsymbol{A}} + \lambda_f \widetilde{\boldsymbol{V}})^{-1} \widetilde{\boldsymbol{A}}' \boldsymbol{g}(t), \quad \lambda_f = v_\epsilon / v_f \end{cases}$$

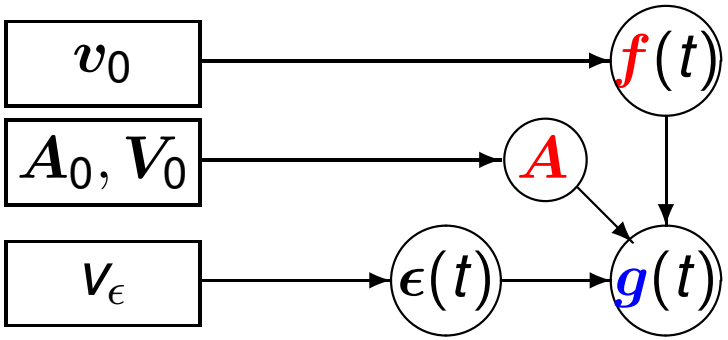
$$q_2(\boldsymbol{A}|\boldsymbol{g}(t), \widetilde{\boldsymbol{f}}(t), v_\epsilon, \boldsymbol{A}_0, \boldsymbol{V}_0) = \mathcal{N}(\boldsymbol{A}|\widetilde{\boldsymbol{A}}, \widetilde{\boldsymbol{V}})$$

$$\begin{cases} \widetilde{\boldsymbol{V}} = (\widetilde{\boldsymbol{F}}'\widetilde{\boldsymbol{F}} + \lambda_f \widetilde{\boldsymbol{\Sigma}})^{-1} \\ \widetilde{\boldsymbol{A}} = \sum_t \boldsymbol{g}(t)\widetilde{\boldsymbol{f}}'(t) \left(\sum_t \widetilde{\boldsymbol{f}}(t)\widetilde{\boldsymbol{f}}'(t) + \lambda_a \widetilde{\boldsymbol{\Sigma}} \right)^{-1}, \quad \lambda_a = v_\epsilon / v_a \end{cases}$$

$$\begin{array}{ccccc} \boldsymbol{A}^{(0)} \longrightarrow \widehat{\boldsymbol{A}} \longrightarrow & \boxed{\widehat{\boldsymbol{f}}(t) = \left(\widehat{\boldsymbol{A}}'\widehat{\boldsymbol{A}} + \lambda_f \widehat{\boldsymbol{V}}\right)^{-1} \widehat{\boldsymbol{A}}' \boldsymbol{g}(t)} & \longrightarrow & \widehat{\boldsymbol{f}}(t) \\ \boldsymbol{V}^{(0)} \longrightarrow \widehat{\boldsymbol{V}} \longrightarrow & \boxed{\widehat{\boldsymbol{\Sigma}} = (\boldsymbol{A}'\boldsymbol{A} + \lambda_f \widehat{\boldsymbol{V}})^{-1}} & \longrightarrow & \widehat{\boldsymbol{\Sigma}} \\ \Uparrow & & & \Downarrow \\ \widehat{\boldsymbol{A}} \longleftarrow & \boxed{\widehat{\boldsymbol{A}} = \sum_t \boldsymbol{g}(t)\widehat{\boldsymbol{f}}'(t) \left(\sum_t \widehat{\boldsymbol{f}}(t)\widehat{\boldsymbol{f}}'(t) + \lambda_a \widehat{\boldsymbol{\Sigma}} \right)^{-1}} & \longleftarrow & \widehat{\boldsymbol{f}}(t) \\ \widehat{\boldsymbol{V}} \longleftarrow & \boxed{\widehat{\boldsymbol{V}} = (\boldsymbol{F}'\boldsymbol{F} + \lambda_f \widehat{\boldsymbol{\Sigma}})^{-1}} & \longleftarrow & \widehat{\boldsymbol{\Sigma}} \end{array}$$

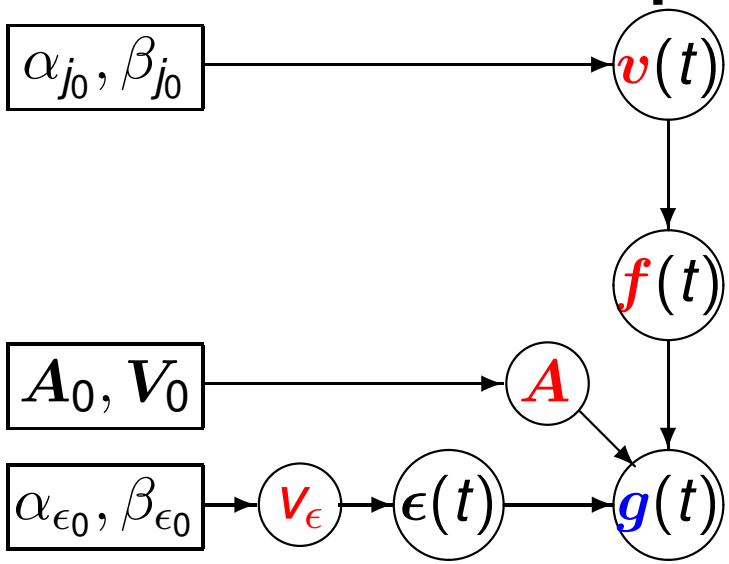
Other more complexe models

Gaussian iid:



$$p(f_j(t)|v_{0j}) = \mathcal{N}(0, v_{0j})$$
$$p(\boldsymbol{f}(t)|\boldsymbol{v}_0) \propto \exp\left[-\frac{1}{2}\sum_j f_j^2(t)/v_{0j}\right]$$
$$p(A_{ij}|A_{0ij}, V_{0ij}) = \mathcal{N}(A_{0ij}, V_{0ij})$$
$$p(\boldsymbol{A}|\boldsymbol{A}_0, \boldsymbol{V}_0) = \mathcal{N}(\boldsymbol{A}_0, \boldsymbol{V}_0)$$
$$p(\boldsymbol{g}(t)|\boldsymbol{A}, \boldsymbol{f}(t), v_\epsilon) = \mathcal{N}(\boldsymbol{A}\boldsymbol{f}(t), v_\epsilon \boldsymbol{I})$$

Variance modulated prior model inducing sparsity:

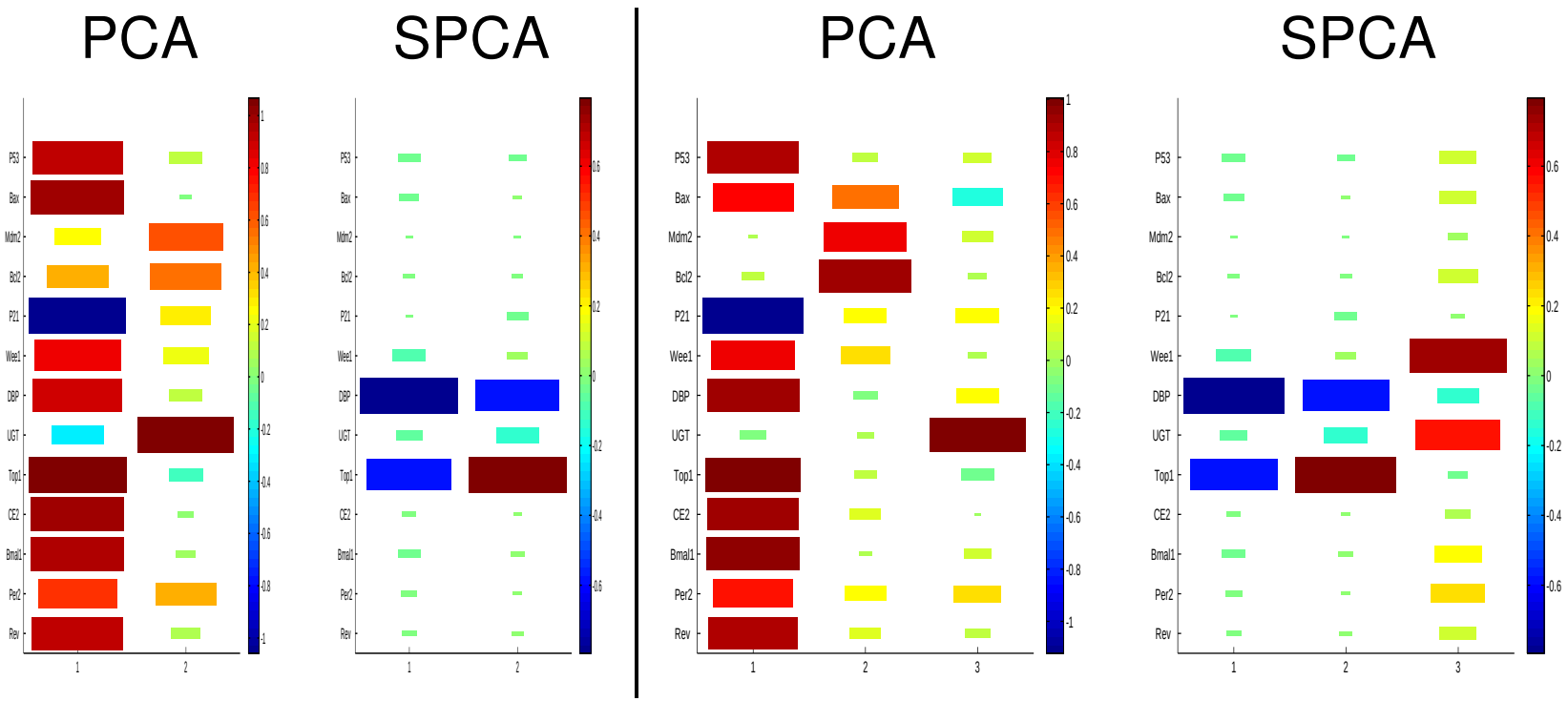


$$p(v_j(t)|\alpha_{j_0}, \beta_{j_0}) = \mathcal{IG}(\alpha_{j_0}, \beta_{j_0})$$
$$p(f_j(t)|v_j(t)) = \mathcal{N}(0, v_j(t))$$
$$p(\boldsymbol{f}(t)|\boldsymbol{v}(t)) \propto \exp\left[-\sum_j f_j^2(t)/v_j(t)\right]$$
$$p(\boldsymbol{A}|\boldsymbol{A}_0, \boldsymbol{V}_0) = \mathcal{N}(\boldsymbol{A}_0, \boldsymbol{V}_0)$$
$$p(\boldsymbol{g}(t)|\boldsymbol{A}, \boldsymbol{f}(t), v_\epsilon) = \mathcal{N}(\boldsymbol{A}\boldsymbol{f}(t), v_\epsilon \boldsymbol{I})$$
$$p(v_\epsilon|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) = \mathcal{IG}(\alpha_{\epsilon_0}, \beta_{\epsilon_0})$$

$$p(\boldsymbol{f}_{1..T}, \boldsymbol{A}, \boldsymbol{v}_{1..T}, v_\epsilon|\boldsymbol{g}_{1..T}) \propto \prod_t p(\boldsymbol{g}(t)|\boldsymbol{A}, \boldsymbol{f}(t), v_\epsilon)\, p(\boldsymbol{f}(t)|\boldsymbol{v}(t))$$
$$\prod_t \prod_j p(v_j(t)|\alpha_{j_0}, \beta_{j_0})\, p(\boldsymbol{A}|\boldsymbol{A}_0, \boldsymbol{V}_0)$$

Sparse PCA

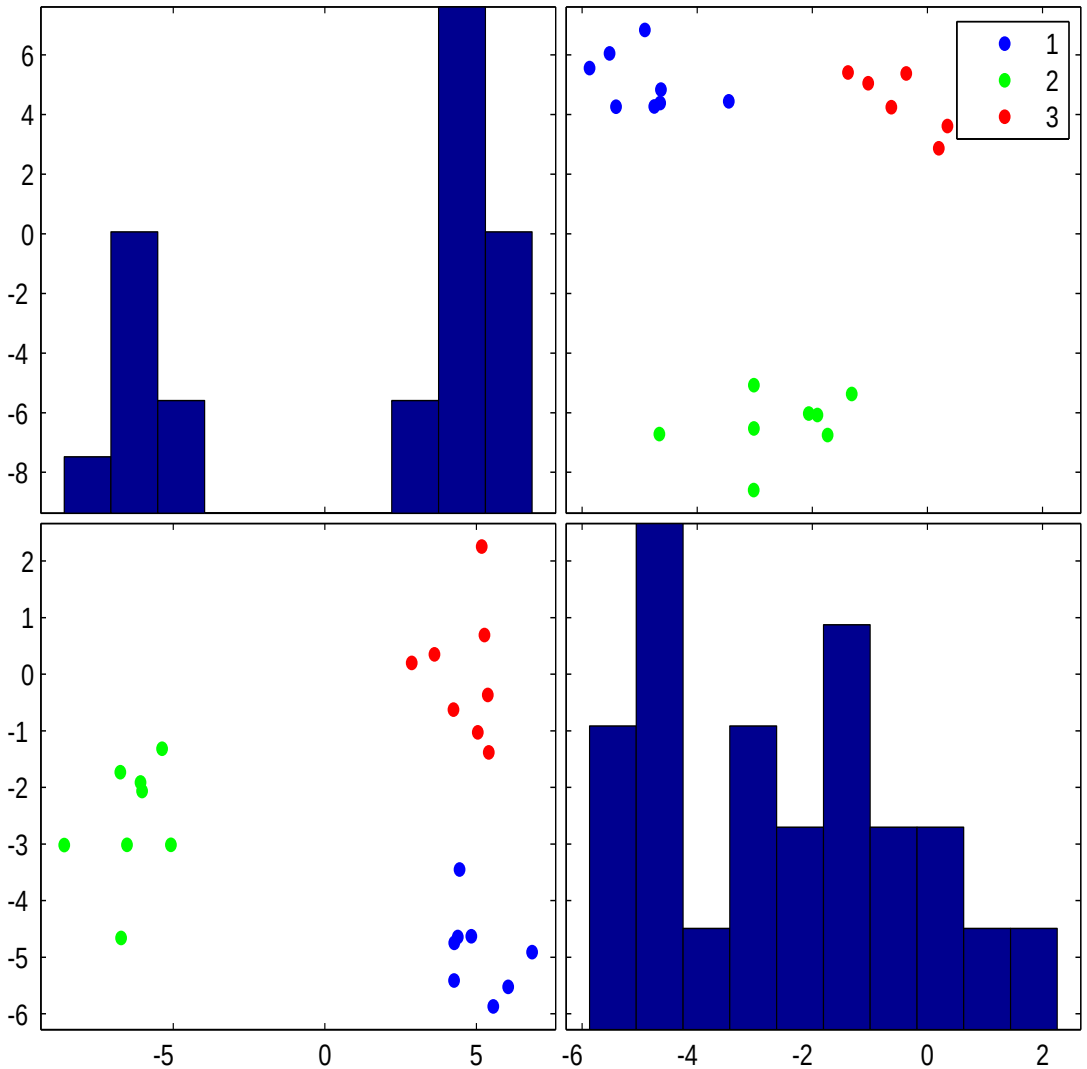
- In classical PCA, FA and ICA, one looks to obtain principal (uncorrelated or independent) components.
- In Sparse PCA or FA, one looks for the loading matrix A with sparsest components.
- This can be imposed via the prior $p(A)$. This leads to least variables selections.

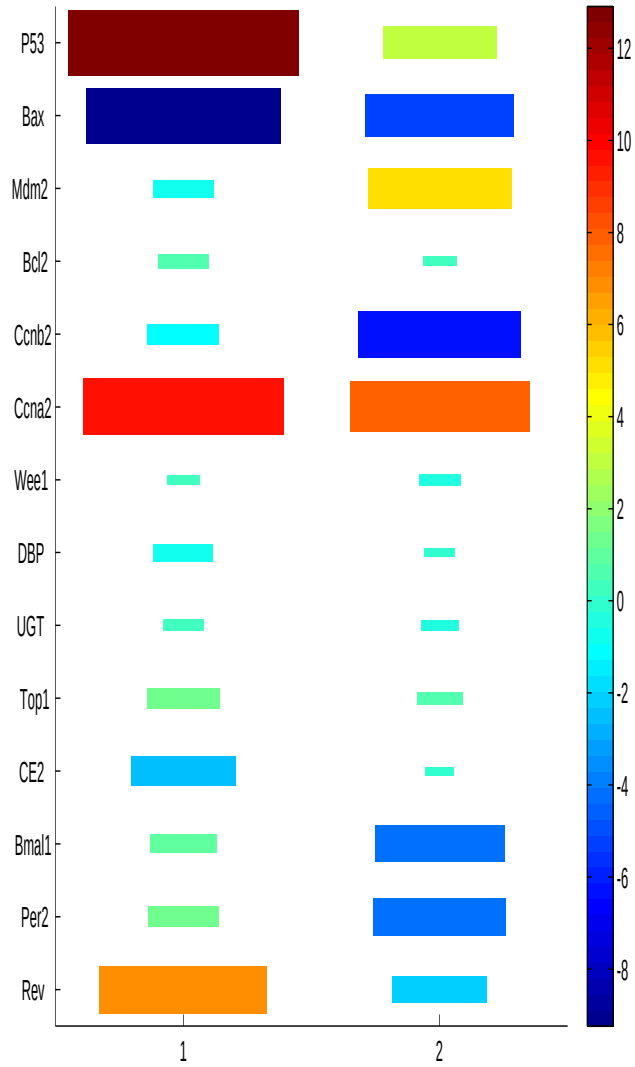


Discriminant Analysis

- When we have data and classes, the question to answer is:
  What are the most discriminant factors?
- There are many variants:
  - Linear Discriminant Analysis (LDA),
  - Quadratic Discriminant Analysis (QDA),
  - Exponential Discriminant Analysis (EDA),
  - Regularized LDA (RLDA), ...
- One can also ask for Sparsest Linear Discriminant factors (SLDA)
- Deterministic point of view (Geometrical distances)
- Probabilistic point of view (Mixture densities)
- Mixture of Gaussians models:
  Each classe is modelled by a Gaussian pdf

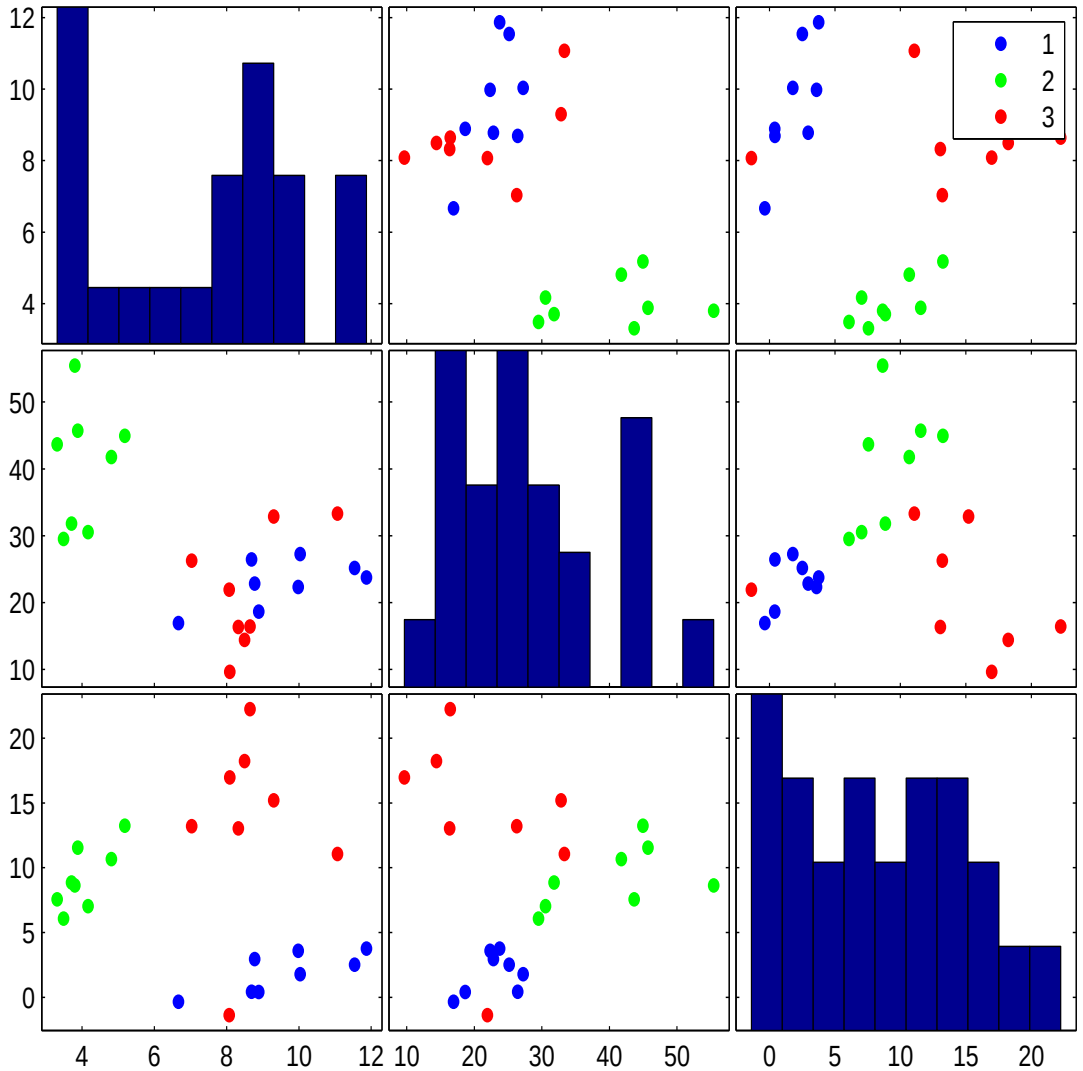
Discriminant Analysis: Time series, Colon

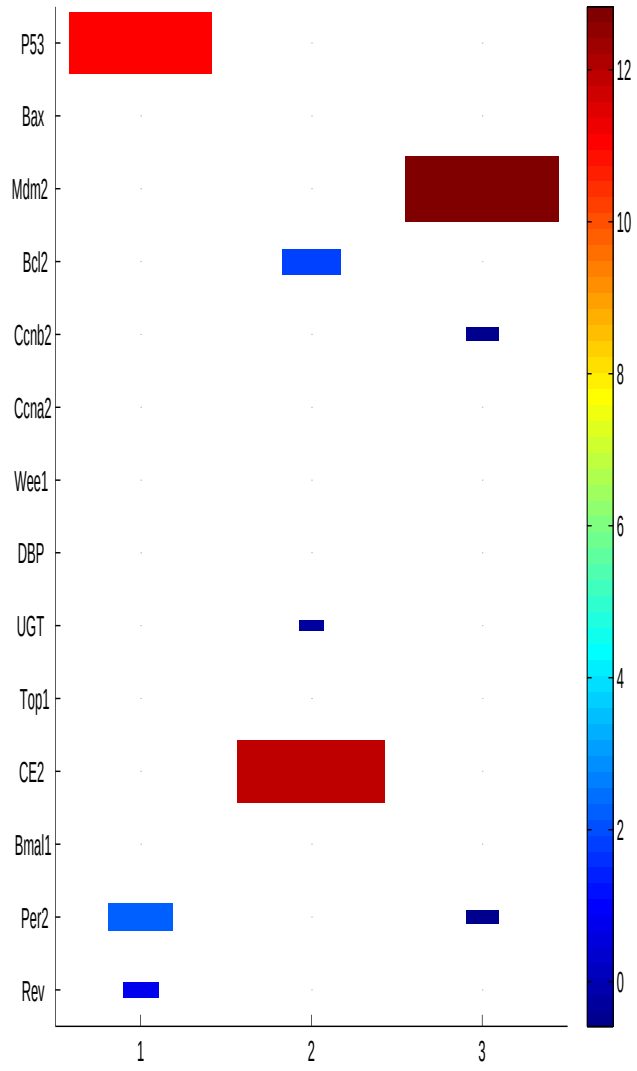




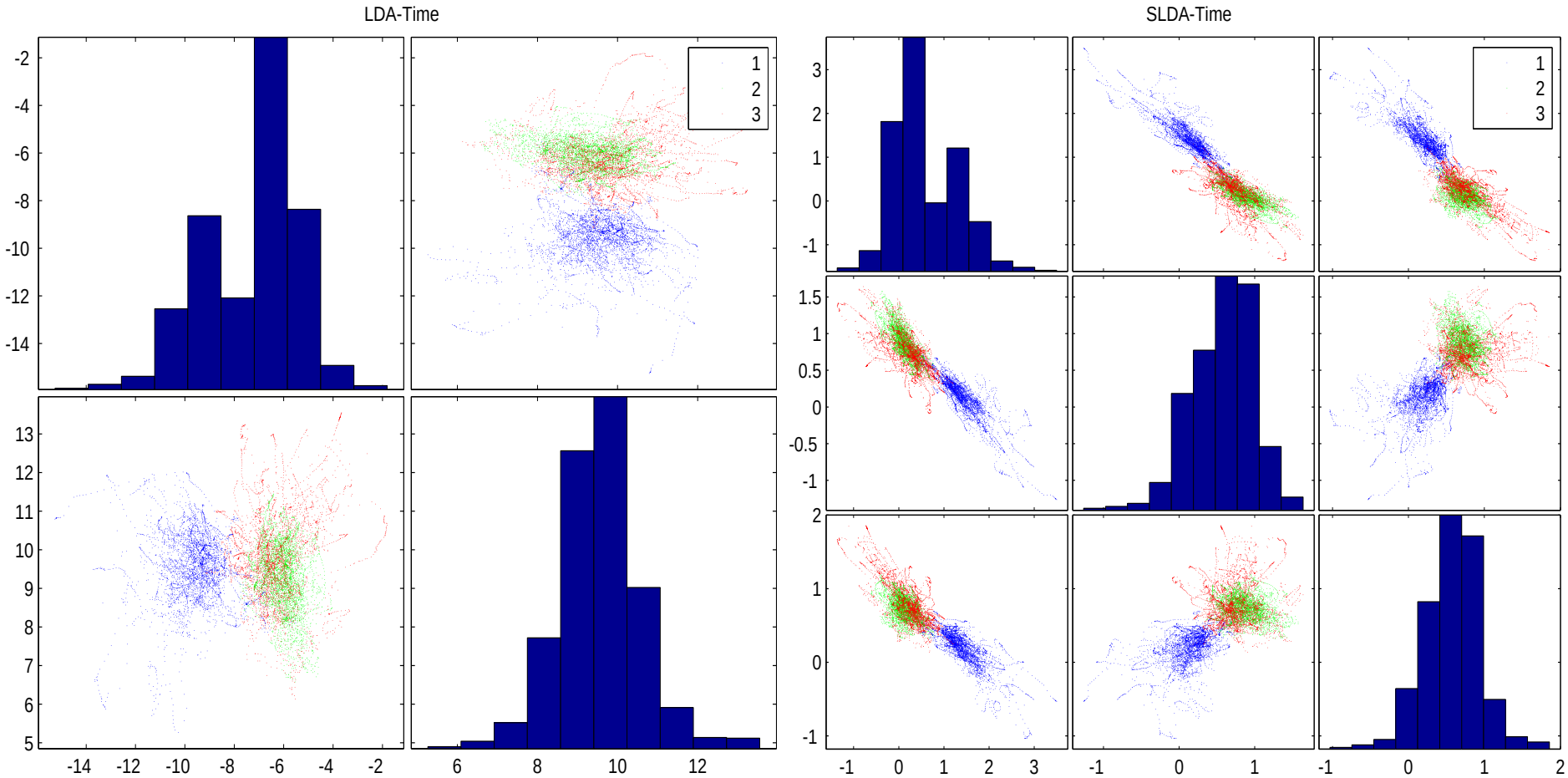
Sparse Discriminant Analysis: Time series, colon

What are the sparsest discriminant factors?





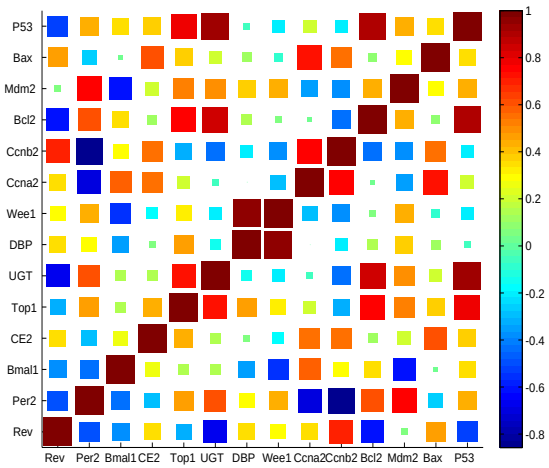
LDA and SLDA study on time serie: 1:before, 2:during, 3:



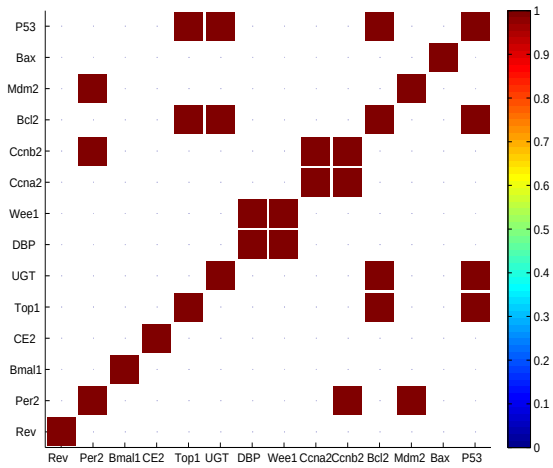
Dependancy graphs

- The main objective here is to show the dependencies between variables
- Three different measures can be used: Pearson ρ, Spearman ρ_s and Kendall τ
- In this study we used ρ_s
- A table of 2 by 2 mutual ρ_s are computed and used in different forms: Hinton, Adjacency table and Graphical network representation

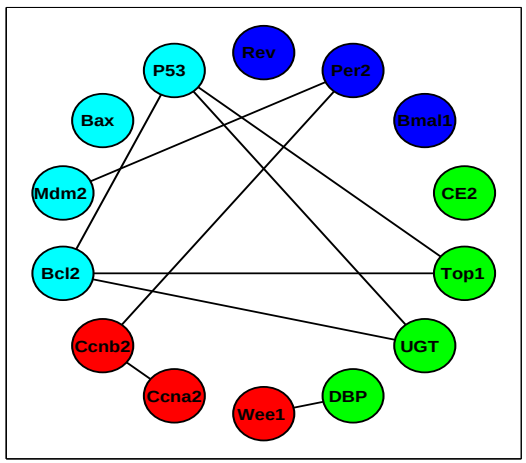
Hinton



Adjacency

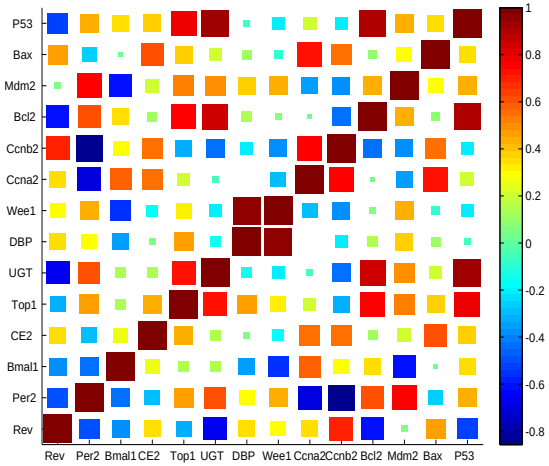


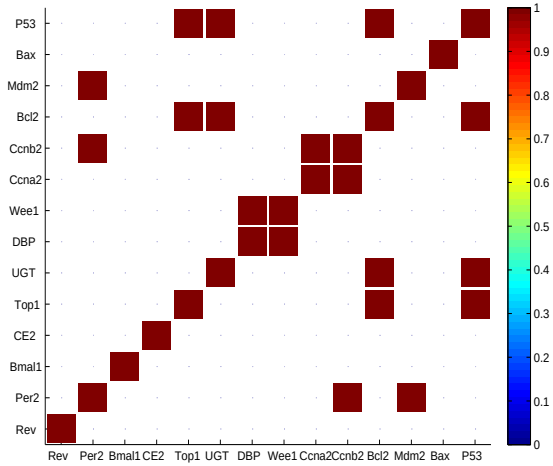
Network

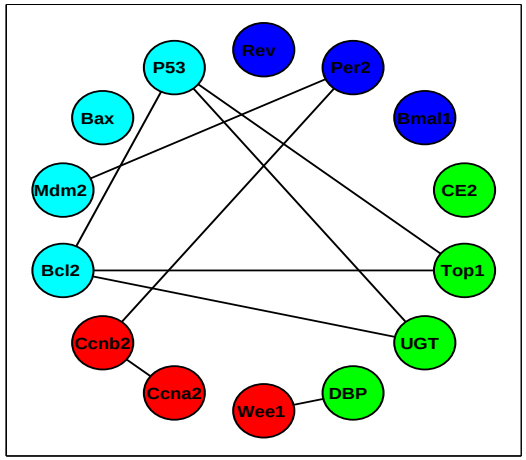


Graph of Dependancies: Colon, Class 1

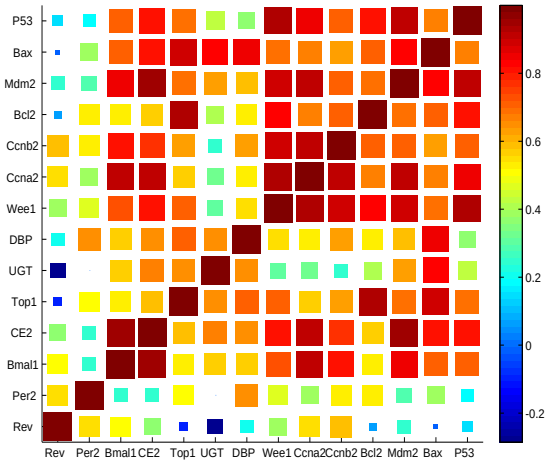
Time series

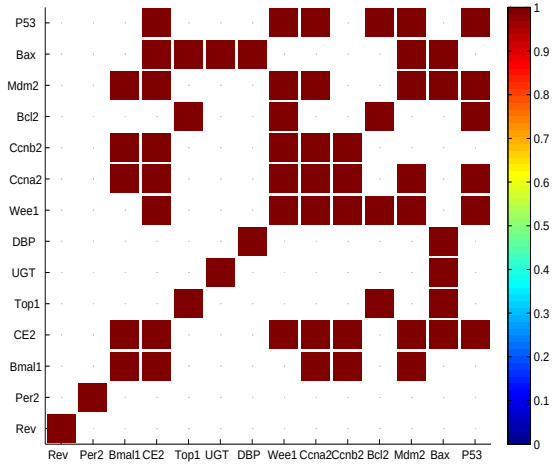


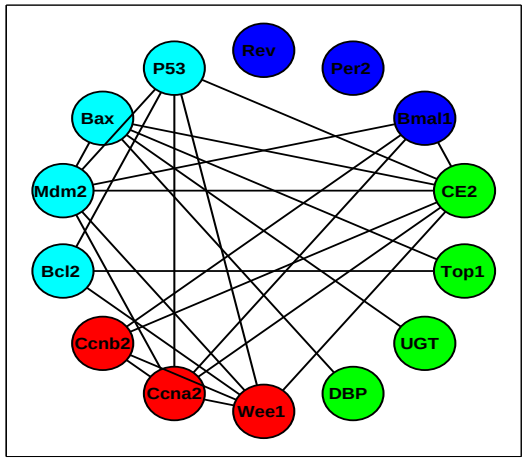




FT amplitudes



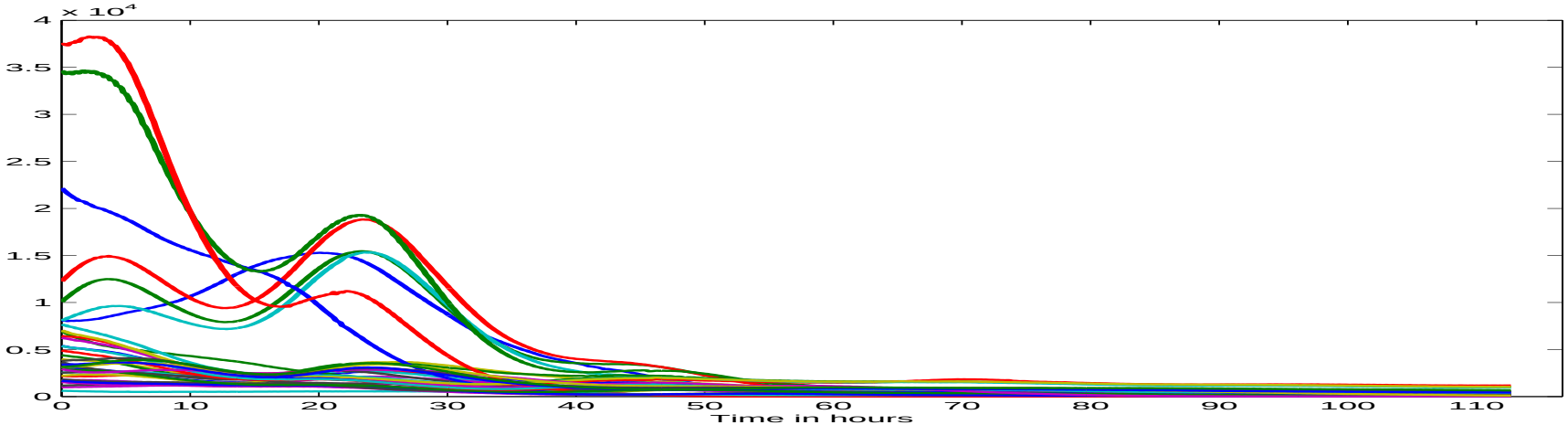


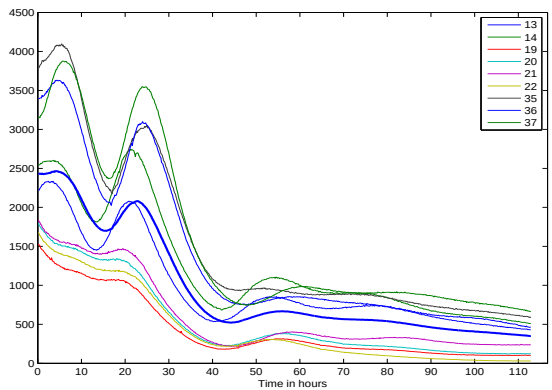


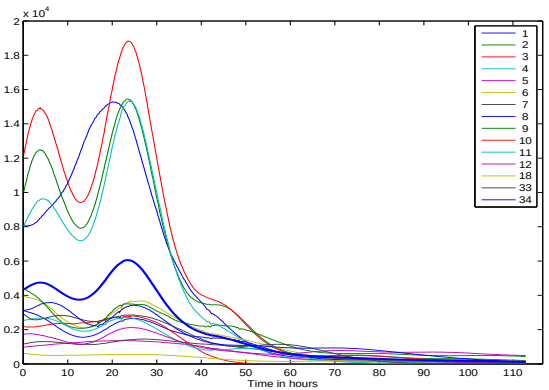
Classification tools

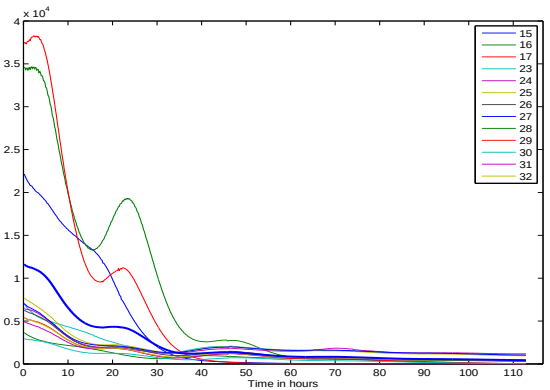
- Supervised classification
  - K nearest neighbors methods
  - Needs Training sets data
  - Must be careful to measure the performances of the classification on a different set of data (Test set)
- Unsupervised classification
  - Mixture models
  - Expectation-Maximization methods
  - Bayesian versions of EM
  - Bayesian Variational Approximation (VBA)

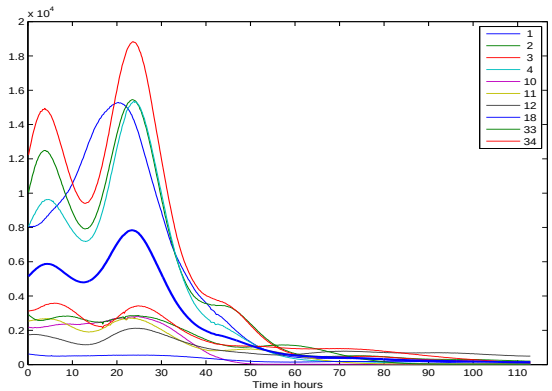
Classification tools

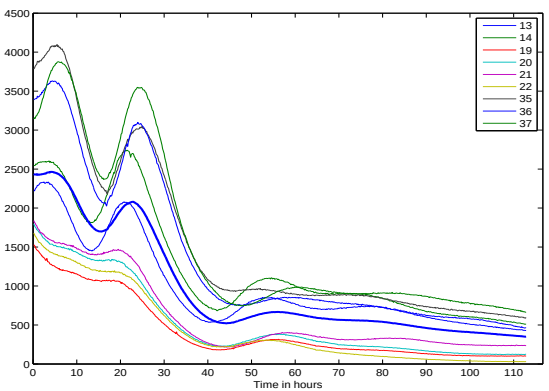












Contents

Mixture models

- General mixture model

$$p(\boldsymbol{x}|a, \Theta, K) = \sum_{k=1}^{K} a_k \, p_k(\boldsymbol{x}_k|\theta_k), \quad 0 < a_k < 1$$

- Same family $p_k(\boldsymbol{x}_k|\theta_k) = p(\boldsymbol{x}_k|\theta_k), \ \forall k$
- Gaussian $p(\boldsymbol{x}_k|\theta_k) = \mathcal{N}(\boldsymbol{x}_k|\mu_k, \boldsymbol{\Sigma}_k)$ with $\theta_k = (\mu_k, \boldsymbol{\Sigma}_k)$
- Data $\boldsymbol{X} = \{\boldsymbol{x}_n, n = 1, \cdots, N\}$ where each element $\boldsymbol{x}_n$ can be in one of these classes c_n.
- $a_k = p(c_n = k)$, $\boldsymbol{a} = \{a_k, k = 1, \cdots, K\}$, $\Theta = \{\theta_k, k = 1, \cdots, K\}$

$$p(\boldsymbol{X}_n, c_n = k|\boldsymbol{a}, \boldsymbol{\theta}) = \prod_{n=1}^{N} p(\boldsymbol{x}_n, c_n = k|\boldsymbol{a}, \boldsymbol{\theta}).$$

Different problems

- Training:
  Given a set of (training) data $\boldsymbol{X}$ and classes $\boldsymbol{c}$, estimate the parameters $\boldsymbol{a}$ and $\boldsymbol{\Theta}$.
- Supervised classification:
  Given a sample $\boldsymbol{x}_m$ and the parameters K, $\boldsymbol{a}$ and $\boldsymbol{\Theta}$ determine its class

$$k^* = \arg\max_k \{p(c_m = k|\boldsymbol{x}_m, \boldsymbol{a}, \boldsymbol{\Theta}, K)\}.$$

- Semi-supervised classification (Proportions are not known):
  Given sample $\boldsymbol{x}_m$ and the parameters K and $\boldsymbol{\Theta}$, determine its class

$$k^* = \arg\max_k \{p(c_m = k|\boldsymbol{x}_m, \boldsymbol{\Theta}, K)\}.$$

- Clustering or unsupervised classification (Number of classes K is not known):
  Given a set of data $\boldsymbol{X}$, determine K and $\boldsymbol{c}$.

Training

- Given a set of (training) data $\boldsymbol{X}$ and classes $\boldsymbol{c}$, estimate the parameters $\boldsymbol{a}$ and $\boldsymbol{\Theta}$.
- Maximum Likelihood (ML):

$$(\widehat{\boldsymbol{a}}, \widehat{\boldsymbol{\Theta}}) = \arg\max_{(\boldsymbol{a}, \boldsymbol{\Theta})} \{p(\boldsymbol{X}, \boldsymbol{c}|\boldsymbol{a}, \boldsymbol{\Theta}, K)\}.$$

- Bayesian: Assign priors $p(\boldsymbol{a}|K)$ and $p(\boldsymbol{\Theta}|K) = \prod_{k=1}^{K} p(\boldsymbol{\theta}_k)$ and write the expression of the joint posterior laws:

$$p(\boldsymbol{a}, \boldsymbol{\Theta}|\boldsymbol{X}, \boldsymbol{c}, K) = \frac{p(\boldsymbol{X}, \boldsymbol{c}|\boldsymbol{a}, \boldsymbol{\Theta}, K)\, p(\boldsymbol{a}|K)\, p(\boldsymbol{\Theta}|K)}{p(\boldsymbol{X}, \boldsymbol{c}|K)}$$

where

$$p(\boldsymbol{X}, \boldsymbol{c}|K) = \iint p(\boldsymbol{X}, \boldsymbol{c}|\boldsymbol{a}, \boldsymbol{\Theta}|K) p(\boldsymbol{a}|K)\, p(\boldsymbol{\Theta}|K)\, \mathrm{d}\boldsymbol{a}\, \mathrm{d}\boldsymbol{\Theta}$$

- Infer on $\boldsymbol{a}$ and $\boldsymbol{\Theta}$ either as the Maximum A Posteriori (MAP) or Posterior Mean (PM).

Supervised classification

- Given a sample $\boldsymbol{x}_m$ and the parameters K, $\boldsymbol{a}$ and $\boldsymbol{\Theta}$ determine

$$p(c_m = k|\boldsymbol{x}_m, \boldsymbol{a}, \boldsymbol{\Theta}, K) = \frac{p(\boldsymbol{x}_m, c_m = k|\boldsymbol{a}, \boldsymbol{\Theta}, K)}{p(\boldsymbol{x}_m|\boldsymbol{a}, \boldsymbol{\Theta}, K)}$$

where $p(\boldsymbol{x}_m, c_m = k|\boldsymbol{a}, \boldsymbol{\Theta}, K) = a_k p(\boldsymbol{x}_m|\boldsymbol{\theta}_k)$ and

$$p(\boldsymbol{x}_m|\boldsymbol{a}, \boldsymbol{\Theta}, K) = \sum_{k=1}^{K} a_k \, p(\boldsymbol{x}_m|\boldsymbol{\theta}_k)$$

- Best class k^*:

$$k^* = \arg\max_k \{p(c_m = k|\boldsymbol{x}_m, \boldsymbol{a}, \boldsymbol{\Theta}, K)\}$$

Semi-supervised classification

- Given sample $\boldsymbol{x}_m$ and the parameters K and $\boldsymbol{\Theta}$ (not the proportions $\boldsymbol{a}$), determine the probabilities

$$p(c_m = k|\boldsymbol{x}_m, \boldsymbol{\Theta}, K) = \frac{p(\boldsymbol{x}_m, c_m = k|\boldsymbol{\Theta}, K)}{p(\boldsymbol{x}_m|\boldsymbol{\Theta}, K)}$$

where

$$p(\boldsymbol{x}_m, c_m = k|\boldsymbol{\Theta}, K) = \int p(\boldsymbol{x}_m, c_m = k|\boldsymbol{a}, \boldsymbol{\Theta}, K)p(\boldsymbol{a}|K)\, \mathrm{d}\boldsymbol{a}$$

and

$$p(\boldsymbol{x}_m|\boldsymbol{\Theta}, K) = \sum_{k=1}^{K} p(\boldsymbol{x}_m, c_m = k|\boldsymbol{\Theta}, K)$$

- Best class k^*, for example the MAP solution:

$$k^* = \arg\max_k \{p(c_m = k|\boldsymbol{x}_m, \boldsymbol{\Theta}, K)\}.$$

Clustering or non-supervised classification

- Given a set of data $\boldsymbol{X}$, determine K and $\boldsymbol{c}$.
- Determination of the number of classes:

$$p(K = L|\boldsymbol{X}) = \frac{p(\boldsymbol{X}, K = L)}{p(\boldsymbol{X})} = \frac{p(\boldsymbol{X}|K = L)\, p(K = L)}{p(\boldsymbol{X})}$$

and

$$p(\boldsymbol{X}) = \sum_{L=1}^{L_0} p(K = L)\, p(\boldsymbol{X}|K = L),$$

where L_0 is the a priori maximum number of classes and

$$p(\boldsymbol{X}|K = L) = \int \int \prod_{n} \prod_{k=1}^{L} a_k p(\boldsymbol{x}_n, c_n = k|\boldsymbol{\theta}_k) p(\boldsymbol{a}|K)\, p(\boldsymbol{\Theta}|K)\, \mathrm{d}\boldsymbol{a}$$

- When K and $\boldsymbol{c}$ are determined, we can also determine the characteristics of those classes $\boldsymbol{a}$ and $\boldsymbol{\Theta}$.

Mixture of Student-t model

- Student-t and its Infinite Gaussian Scaled Model (IGSM):

$$\mathcal{T}(\boldsymbol{x}|\nu,\boldsymbol{\mu},\boldsymbol{\Sigma}) = \int_0^\infty \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},z^{-1}\boldsymbol{\Sigma})\,\mathcal{G}(z|\frac{\nu}{2},\frac{\nu}{2})\,\mathrm{d}z$$

where

$$\begin{aligned}\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) &= |2\pi\boldsymbol{\Sigma}|^{-\frac{1}{2}}\exp\left[-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]\\ &= |2\pi\boldsymbol{\Sigma}|^{-\frac{1}{2}}\exp\left[-\tfrac{1}{2}\mathrm{Tr}\left\{(\boldsymbol{x}-\boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})'\right\}\right]\end{aligned}$$

and

$$\mathcal{G}(z|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}z^{\alpha-1}\exp\left[-\beta z\right].$$

- Mixture of Student-t:

$$p(\boldsymbol{x}|\{\nu_k,a_k,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k,k=1,\cdots,K\},K) = \sum_{k=1}^{K} a_k\,\mathcal{T}(\boldsymbol{x}_n|\nu_k,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k).$$

Mixture of Student-t model

- Introducing z_{nk}, $z_k = \{z_{nk}, n = 1, \cdots, N\}$, $\boldsymbol{Z} = \{z_{nk}\}$, $\boldsymbol{c} = \{c_n, n = 1, \cdots, N\}$, $\boldsymbol{\theta}_k = \{\nu_k, \boldsymbol{a}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_k, k = 1, \cdots, K\}$
- Assigning the priors $p(\boldsymbol{\Theta}) = \prod_k p(\boldsymbol{\theta}_k)$, we can write:

$$p(\boldsymbol{X}, \boldsymbol{c}, \boldsymbol{Z}, \boldsymbol{\Theta}|K) = \prod_n \prod_k a_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, z_{n,k}^{-1}\boldsymbol{\Sigma}_k)\, \mathcal{G}(z_{nk}|\tfrac{\nu_k}{2}, \tfrac{\nu_k}{2})\, p(\boldsymbol{\theta}_k$$

- Joint posterior law:

$$p(\boldsymbol{c}, \boldsymbol{Z}, \boldsymbol{\Theta}|\boldsymbol{X}, K) = \frac{p(\boldsymbol{X}, \boldsymbol{c}, \boldsymbol{Z}, \boldsymbol{\Theta}|K)}{p(\boldsymbol{X}|K)}.$$

- The main task now is to propose some approximations to it in such a way that we can use it easily in all the above mentioned tasks of classification or clustering.

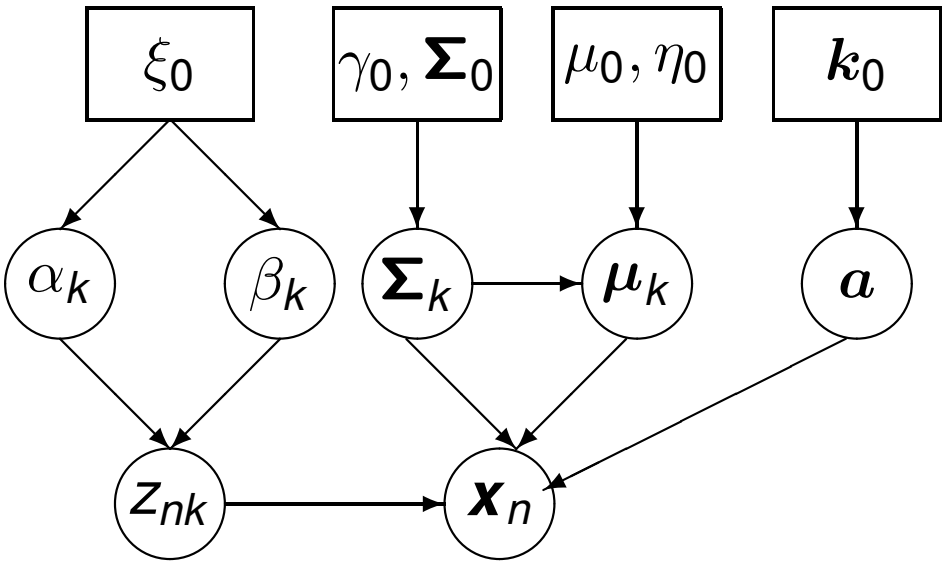
Variational Bayesian Approximation (VBA)

- Main idea: to propose easy computational approximation $q(\boldsymbol{c}, \boldsymbol{Z}, \boldsymbol{\Theta})$ for $p(\boldsymbol{c}, \boldsymbol{Z}, \boldsymbol{\Theta}|\boldsymbol{X}, K)$.
- Criterion: KL$(q : p)$
- Interestingly, by noting that
  $p(\boldsymbol{c}, \boldsymbol{Z}, \boldsymbol{\Theta}|\boldsymbol{X}, K) = p(\boldsymbol{X}, \boldsymbol{c}, \boldsymbol{Z}, \boldsymbol{\Theta}|K)/p(\boldsymbol{X}|K)$
  we have:
  $$\text{KL}(q : p) = -\mathcal{F}(q) + \ln p(\boldsymbol{X}|K)$$
  where
  $$\mathcal{F}(q) = \langle -\ln p(\boldsymbol{X}, \boldsymbol{c}, \boldsymbol{Z}, \boldsymbol{\Theta}|K)\rangle_q$$
  is called free energy of q and we have the following properties:
  – Maximizing $\mathcal{F}(q)$ or minimizing KL$(q : p)$ are equivalent and both give un upper bound to the evidence of the model $\ln p(\boldsymbol{X}|K)$.
  – When the optimum q^* is obtained, $\mathcal{F}(q^*)$ can be used as a criterion for model selection.

VBA: choosing the good families

- Using KL($q : p$) has the very interesting property that using q to compute the means we obtain the same values if we have used p (Conservation of the means).
- Unfortunately, this is not the case for variances or other moments.
- If p is in the exponential family, then choosing appropriate conjugate priors, the structure of q will be the same and we can obtain appropriate fast optimization algorithms.

Hierarchical graphical model



Graphical representation of the model.

VBA for mixture of Student-t

- In our case, noting that

$$p(\boldsymbol{X}, \boldsymbol{c}, \boldsymbol{Z}, \boldsymbol{\Theta}|K) = \prod_n \prod_k p(\boldsymbol{x}_n, c_n, z_{nk}|a_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \nu_k) \prod_k [p(\alpha_k)\, p(\beta_k)\, p(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k)\, p(\boldsymbol{\Sigma}_k)]$$

with

$$p(\boldsymbol{x}_n, c_n, z_{nk}|a_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \nu_k) = \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, z_{n,k}^{-1}\boldsymbol{\Sigma}_k)\, \mathcal{G}(z_{nk}|\alpha_k, \beta_k)$$

is separable, in one side for $[\boldsymbol{c}, \boldsymbol{Z}]$ and in other size in components of $\boldsymbol{\Theta}$, we propose to use

$$q(\boldsymbol{c}, \boldsymbol{Z}, \boldsymbol{\Theta}) = q(\boldsymbol{c}, \boldsymbol{Z})\, q(\boldsymbol{\Theta}).$$

VBA for mixture of Student-t

- With this decomposition, the expression of the Kullback-Leibler divergence becomes:

$$\mathrm{KL}(q_1(\boldsymbol{c},\boldsymbol{Z})q_2(\boldsymbol{\Theta}) : p(\boldsymbol{c},\boldsymbol{Z},\boldsymbol{\Theta}|\boldsymbol{X},K) =$$
$$\sum_{\boldsymbol{c}}\int\int q_1(\boldsymbol{c},\boldsymbol{Z})q_2(\boldsymbol{\Theta})\ln\frac{q_1(\boldsymbol{c},\boldsymbol{Z})q_2(\boldsymbol{\Theta})}{p(\boldsymbol{c},\boldsymbol{Z},\boldsymbol{\Theta}|\boldsymbol{X},K)}\,\mathrm{d}\boldsymbol{\Theta}\,\mathrm{d}\boldsymbol{Z}$$

- The expression of the Free energy becomes:

$$\mathcal{F}(q_1(\boldsymbol{c},\boldsymbol{Z})q_2(\boldsymbol{\Theta})) = \sum_{\boldsymbol{c}}\int\int q_1(\boldsymbol{c},\boldsymbol{Z})q_2(\boldsymbol{\Theta})\ln\frac{p(\boldsymbol{X},\boldsymbol{c},\boldsymbol{Z}|\boldsymbol{\Theta},K)p(\boldsymbol{\Theta}|K)}{q_1(\boldsymbol{c},\boldsymbol{Z})q_2(\boldsymbol{\Theta})}\,\mathrm{d}\boldsymbol{\Theta}$$

Proposed VBA for Mixture of Student-t priors model

- Using a generalized Student-t obtained by replacing $\mathcal{G}(z_{n,k}|\frac{\nu_k}{2},\frac{\nu_k}{2})$ by $\mathcal{G}(z_{n,k}|\alpha_k,\beta_k)$ it will be easier to propose conjugate priors for α_k,β_k than for ν_k.

$$p(\boldsymbol{x}_n, c_n = k, z_{nk}|a_k,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k,\alpha_k,\beta_k,K) = a_k\,\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, z_{n,k}^{-1}\boldsymbol{\Sigma}_k)\,\mathcal{G}(z_{n,k}|\alpha_k,\beta_k$$

- In the following, noting by $\boldsymbol{\Theta} = \{(a_k,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k,\alpha_k,\beta_k), k = 1,\cdots,K\}$, we propose to use the factorized prior laws:

$$p(\boldsymbol{\Theta}) = p(\boldsymbol{a})\sum_k [p(\alpha_k)\,p(\beta_k)\,p(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k)\,p(\boldsymbol{\Sigma}_k)]$$

with the following components:

$$\begin{cases} p(\boldsymbol{a}) = \mathcal{D}(\boldsymbol{a}|\boldsymbol{k}_0), \quad \boldsymbol{k}_0 = [k_0,\cdots,k_0] = k_0\mathbf{1} \\ p(\alpha_k) = \mathcal{E}(\alpha_k|\zeta_0) = \mathcal{G}(\alpha_k|1,\zeta_0) \\ p(\beta_k) = \mathcal{E}(\beta_k|\zeta_0) = \mathcal{G}(\alpha_k|1,\zeta_0) \\ p(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k) = \mathcal{N}(\boldsymbol{\mu}_k|\mu_0\mathbf{1},\eta_0^{-1}\boldsymbol{\Sigma}_k) \\ p(\boldsymbol{\Sigma}_k) = \mathcal{IW}(\boldsymbol{\Sigma}_k|\gamma_0,\gamma_0\Sigma_0) \end{cases}$$

Proposed VBA for Mixture of Student-t priors model

where

$$\mathcal{D}(\boldsymbol{a}|\boldsymbol{k}) = \frac{\Gamma(\sum_l k_k)}{\prod_l \Gamma(k_l)} \prod_l a_l^{k_l - 1}$$

is the Dirichlet pdf,

$$\mathcal{E}(t|\zeta_0) = \zeta_0 \exp\left[-\zeta_0 t\right]$$

is the Exponential pdf,

$$\mathcal{G}(t|a,b) = \frac{b^a}{\Gamma(a)} t^{a-1} \exp\left[-bt\right]$$

is the Gamma pdf and

$$\mathcal{IW}(\boldsymbol{\Sigma}|\gamma,\gamma\boldsymbol{\Delta}) = \frac{|\frac{1}{2}\boldsymbol{\Delta}|^{\gamma/2} \exp\left[-\frac{1}{2}\mathrm{Tr}\left\{\boldsymbol{\Delta}\boldsymbol{\Sigma}^{-1}\right\}\right]}{\Gamma_D(\gamma/2)|\boldsymbol{\Sigma}|^{\frac{\gamma+D+1}{2}}}.$$

is the inverse Wishart pdf.

With these prior laws and the likelihood: joint posterior law:

$$p_k(\boldsymbol{c}, \boldsymbol{Z}, \boldsymbol{\Theta}|\boldsymbol{X}) = \frac{p(\boldsymbol{X}, \boldsymbol{c}, \boldsymbol{Z}, \boldsymbol{\Theta})}{p(\boldsymbol{X})}.$$

Expressions of q

$$q(\boldsymbol{c}, \boldsymbol{Z}, \boldsymbol{\Theta}) = q(\boldsymbol{c}, \boldsymbol{Z})\, q(\boldsymbol{\Theta}) = \prod_n \prod_k [q(c_n = k|z_{nk})\, q(z_{nk})]$$
$$\prod_k [q(\alpha_k)\, q(\beta_k)\, q(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k)\, q(\boldsymbol{\Sigma}_k)]\, q(\boldsymbol{a}).$$

with:

$$\begin{cases} q(\boldsymbol{a}) = \mathcal{D}(\boldsymbol{a}|\tilde{\boldsymbol{k}}), \quad \tilde{\boldsymbol{k}} = [\tilde{k}_1, \cdots, \tilde{k}_K] \\ q(\alpha_k) = \mathcal{G}(\alpha_k|\tilde{\zeta}_k, \tilde{\eta}_k) \\ q(\beta_k) = \mathcal{G}(\beta_k|\tilde{\zeta}_k, \tilde{\eta}_k) \\ q(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k) = \mathcal{N}(\boldsymbol{\mu}_k|\tilde{\boldsymbol{\mu}}, \tilde{\eta}^{-1}\boldsymbol{\Sigma}_k) \\ q(\boldsymbol{\Sigma}_k) = \mathcal{IW}(\boldsymbol{\Sigma}_k|\tilde{\gamma}, \tilde{\gamma}\tilde{\Sigma}) \end{cases}$$

With these choices, we have

$$\mathcal{F}(q(\boldsymbol{c}, \boldsymbol{Z}, \boldsymbol{\Theta})) = \langle \ln p(\boldsymbol{X}, \boldsymbol{c}, \boldsymbol{Z}, \boldsymbol{\Theta}|K) \rangle_{q(\boldsymbol{c}, \boldsymbol{Z}, \boldsymbol{\Theta})} = \prod_k \prod_n \mathcal{F}_{1_{kn}} + \prod_k \mathcal{F}_{2_k}$$

$$\mathcal{F}_{1_{kn}} = \langle \ln p(\boldsymbol{x}_n, c_n, z_{nk}, \boldsymbol{\theta}_k) \rangle_{q(c_n=k|z_{nk})q(z_{nk})}$$
$$\mathcal{F}_{2_k} = \langle \ln p(\boldsymbol{x}_n, c_n, z_{nk}, \boldsymbol{\theta}_k) \rangle_{q(\boldsymbol{\theta}_k)}$$

VBA Algorithm step

Expressions of the updating expressions of the tilded parameters are obtained by following three steps:

- E step: Optimizing $\mathcal{F}$ with respect to $q(\boldsymbol{c}, \boldsymbol{Z})$ when keeping $q(\boldsymbol{\Theta})$ fixed, we obtain the expression of $q(c_n = k|z_{nk}) = \tilde{a}_k$, $q(z_{nk}) = \mathcal{G}(z_{nk}|\widetilde{\alpha}_k, \widetilde{\beta}_k)$.
- M step: Optimizing $\mathcal{F}$ with respect to $q(\boldsymbol{\Theta})$ when keeping $q(\boldsymbol{c}, \boldsymbol{Z})$ fixed, we obtain the expression of $q(\boldsymbol{a}) = \mathcal{D}(\boldsymbol{a}|\tilde{\boldsymbol{k}})$, $\quad \tilde{\boldsymbol{k}} = [\tilde{k}_1, \cdots, \tilde{k}_K]$, $q(\alpha_k) = \mathcal{G}(\alpha_k|\tilde{\zeta}_k, \tilde{\eta}_k)$, $q(\beta_k) = \mathcal{G}(\beta_k|\tilde{\zeta}_k, \tilde{\eta}_k)$, $q(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k) = \mathcal{N}(\boldsymbol{\mu}_k|\widetilde{\boldsymbol{\mu}}, \tilde{\eta}^{-1}\boldsymbol{\Sigma}_k)$, and $q(\boldsymbol{\Sigma}_k) = \mathcal{IW}(\boldsymbol{\Sigma}_k|\tilde{\gamma}, \tilde{\gamma}\tilde{\Sigma})$, which gives the updating algorithm for the corresponding tilded parameters.
- $\mathcal{F}$ evaluation: After each E step and M step, we can also evaluate the expression of $\mathcal{F}(q)$ which can be used for stopping rule of the iterative algorithm.
- Final value of $\mathcal{F}(q)$ for each value of K, noted $\mathcal{F}_k$, can be used as a criterion for model selection, i.e.; the determination of the number of clusters.

Conclusions

- Clustering and classification of a set of data are between the most important tasks in statistical researches for many applications such as data mining in biology.
- Mixture models and in particular Mixture of Gaussians are classical models for these tasks.
- We proposed to use a mixture of generalised Student-t distribution model for the data via a hierarchical graphical model.
- To obtain fast algorithms and be able to handle large data sets, we used conjugate priors everywhere it was possible.
- The proposed algorithm has been used for clustering, classification and discriminant analysis of some biological data (Cancer research related), but in this paper, we only presented the main algorithm.