An introduction to Computational Anatomy

Stéphanie Allassonnière

Faculté de médecine, Université Paris Descartes

September 2018

Outline

1. Introduction to Computational Anatomy
2. Registration technics
3. Statistical analysis of the deformations
4. Bayesian Modelling for template estimation

Outline

Goals

To design mathematical methods and algorithms to model and analyse the anatomy

- Characterise anatomical shapes :

Goals

To design mathematical methods and algorithms to model and analyse the anatomy

- Characterise anatomical shapes :
  Estimate representative organs within groups (anatomical invariants)

Goals

To design mathematical methods and algorithms to model and analyse the anatomy

- Characterise anatomical shapes :
  Estimate representative organs within groups (anatomical invariants)
- Analysis of populations :

Goals

To design mathematical methods and algorithms to model and analyse the anatomy

- Characterise anatomical shapes :
  Estimate representative organs within groups (anatomical invariants)
- Analysis of populations :
  Establish "normal" variability

Goals

To design mathematical methods and algorithms to model and analyse the anatomy

- Characterise anatomical shapes :
  Estimate representative organs within groups (anatomical invariants)
- Analysis of populations :
  Establish "normal" variability
- Classification / Discrimination :

Goals

To design mathematical methods and algorithms to model and analyse the anatomy

- Characterise anatomical shapes :
  Estimate representative organs within groups (anatomical invariants)
- Analysis of populations :
  Establish "normal" variability
- Classification / Discrimination :
  Classify pathologies from structural deviations

Goals

To design mathematical methods and algorithms to model and analyse the anatomy

- Characterise anatomical shapes :
  Estimate representative organs within groups (anatomical invariants)
- Analysis of populations :
  Establish "normal" variability
- Classification / Discrimination :
  Classify pathologies from structural deviations
- Learn the temporal evolution (growth, evolution of a disease) :

Goals

To design mathematical methods and algorithms to model and analyse the anatomy

- Characterise anatomical shapes :
  Estimate representative organs within groups (anatomical invariants)
- Analysis of populations :
  Establish "normal" variability
- Classification / Discrimination :
  Classify pathologies from structural deviations
- Learn the temporal evolution (growth, evolution of a disease) :
  Model organ development across time

Goals

To design mathematical methods and algorithms to model and analyse the anatomy

- Characterise anatomical shapes :
  Estimate representative organs within groups (anatomical invariants)
- Analysis of populations :
  Establish "normal" variability
- Classification / Discrimination :
  Classify pathologies from structural deviations
- Learn the temporal evolution (growth, evolution of a disease) :
  Model organ development across time
- Segmentation, prediction, help therapy :

Goals

To design mathematical methods and algorithms to model and analyse the anatomy

- Characterise anatomical shapes :
  Estimate representative organs within groups (anatomical invariants)
- Analysis of populations :
  Establish “normal” variability
- Classification / Discrimination :
  Classify pathologies from structural deviations
- Learn the temporal evolution (growth, evolution of a disease) :
  Model organ development across time
- Segmentation, prediction, help therapy :
  Build prior knowledge to simulate new anatomies, segment areas or organs in new patient, predict from the shape of an organ the evolution of a disease (through clinical variables for example)

Mathematical tools

- Characterise anatomical shapes :
- Analysis of populations :
- Classification / Discrimination :
- Learn the temporal evolution (growth, evolution of a disease) :
- Segmentation, prediction, help therapy :

Mathematical tools

- Characterise anatomical shapes : Geometry
- Analysis of populations :
- Classification / Discrimination :
- Learn the temporal evolution (growth, evolution of a disease) :
- Segmentation, prediction, help therapy :

Mathematical tools

- Characterise anatomical shapes : Geometry
- Analysis of populations : Statistical Modelling and differential geometry
- Classification / Discrimination :
- Learn the temporal evolution (growth, evolution of a disease) :
- Segmentation, prediction, help therapy :

Mathematical tools

- Characterise anatomical shapes : Geometry
- Analysis of populations : Statistical Modelling and differential geometry
- Classification / Discrimination : Statistical analysis
- Learn the temporal evolution (growth, evolution of a disease) :
- Segmentation, prediction, help therapy :

Mathematical tools

- Characterise anatomical shapes : Geometry
- Analysis of populations : Statistical Modelling and differential geometry
- Classification / Discrimination : Statistical analysis
- Learn the temporal evolution (growth, evolution of a disease) : Statistical learning
- Segmentation, prediction, help therapy :

Mathematical tools

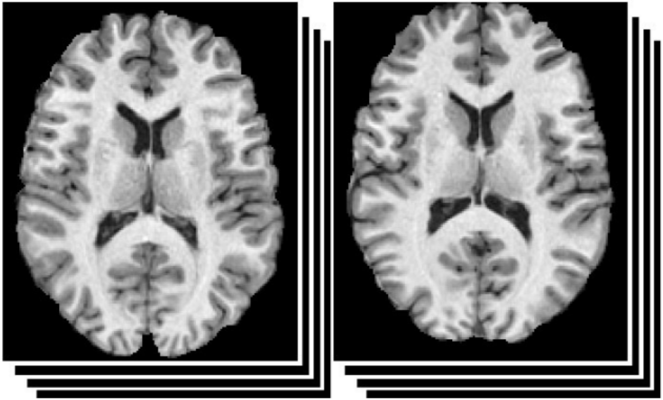
- Characterise anatomical shapes : Geometry
- Analysis of populations : Statistical Modelling and differential geometry
- Classification / Discrimination : Statistical analysis
- Learn the temporal evolution (growth, evolution of a disease) : Statistical learning
- Segmentation, prediction, help therapy : Functional analysis

Mathematical tools

- Characterise anatomical shapes : Geometry
- Analysis of populations : Statistical Modelling and differential geometry
- Classification / Discrimination : Statistical analysis
- Learn the temporal evolution (growth, evolution of a disease) : Statistical learning
- Segmentation, prediction, help therapy : Functional analysis
- Etc...

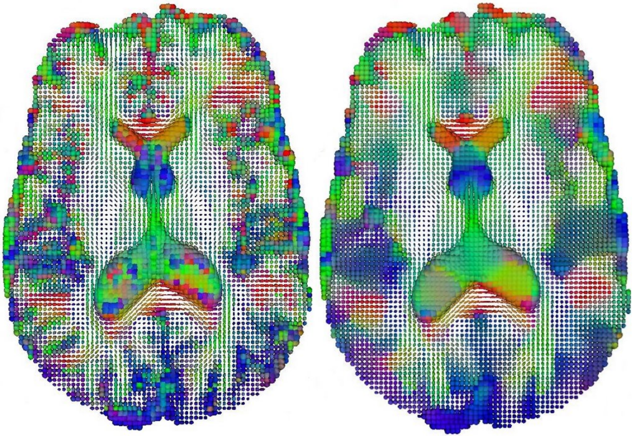
Kinds of data

- Images (grey level, tensors, etc)



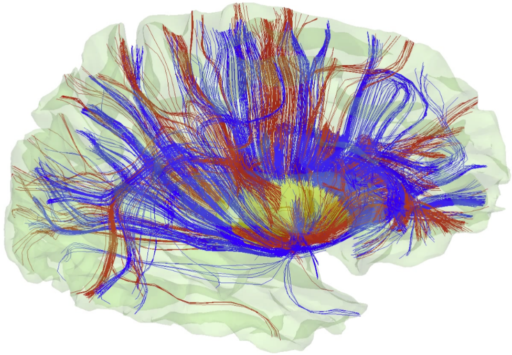
Kinds of data

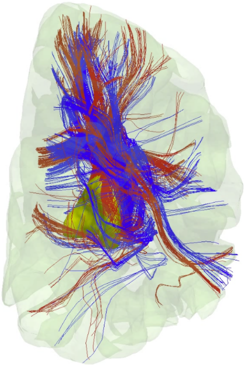
- Images (grey level, tensors, etc)



Kinds of data

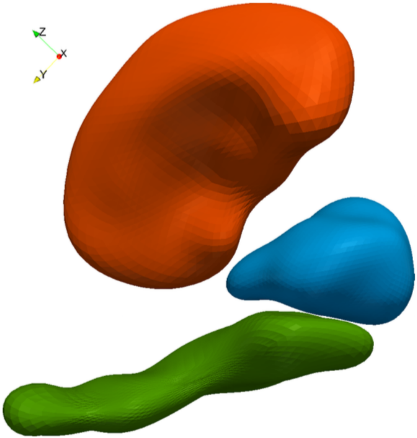
- Anatomical landmarks : anatomical points, fibres, gyri, etc





Kinds of data

- Meshed surfaces (with point correspondence or not)



Kinds of data

- Clinical variables related to the data (age, diagnosis, physiological parameters, etc)

Alzheimer Disease Assessment Scale—Cognitive Subscale (ADAS-cog) 11-Item

	Score range
Memory and new learning	**0 - 35**
Word recall *(mean number of words not recalled)*	0 - 10
Orientation *(one point for each incorrect response)*	0 - 8
Word recognition *(mean number of incorrect responses)*	0 - 12
Remembering test instructions	0 - 5
Language	**0 - 25**
Commands	0 - 5
Spoken language ability	0 - 5
Naming objects/fingers	0 - 5
Word-finding difficulty	0 - 5
Comprehension	0 - 5
Praxis	**0 - 10**
Constructional praxis	0 - 5
Ideational praxis	0 - 5
Total	**0 - 70**

Increasing scores indicate worsening cognitive function.
Rosen WG, et al. *Am J Psychiatry.* 1984;141:1356-1364.

NEED : compare these elements in a mathematical way

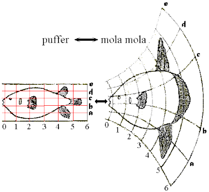
$\longrightarrow$ Computational anatomy is the correct setting

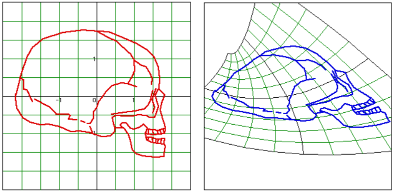
$\longrightarrow$ Requires models of the data

Deformable Template Model (Genander, '80)

The idea of Deformable Template Model

- Compare two observations via the quantification of the deformation from one to the other (D'Arcy Thompson, 1917)

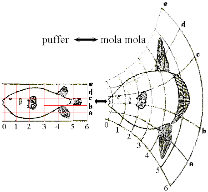


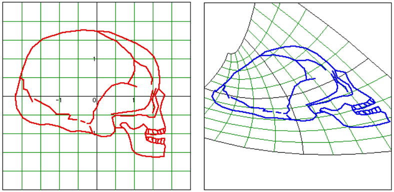


- Each element of a population is a smooth deformation of a template

The idea of Deformable Template Model

- Compare two observations via the quantification of the deformation from one to the other (D'Arcy Thompson, 1917)



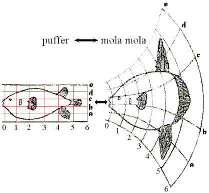


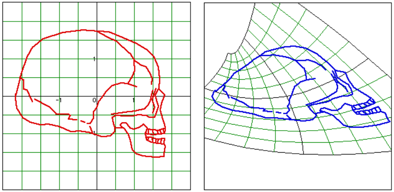
Registration

- Each element of a population is a smooth deformation of a template

The idea of Deformable Template Model

- Compare two observations via the quantification of the deformation from one to the other (D'Arcy Thompson, 1917)





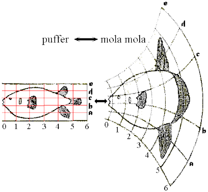
Registration

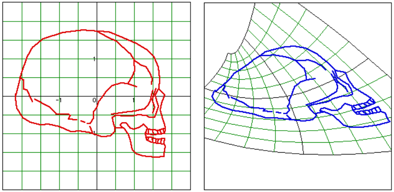
- Each element of a population is a smooth deformation of a template

Template estimation

The idea of Deformable Template Model

- Compare two observations via the quantification of the deformation from one to the other (D'Arcy Thompson, 1917)





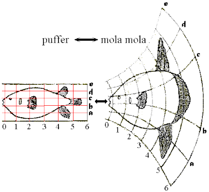
Registration

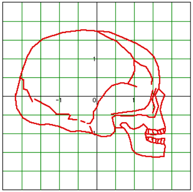
- Each element of a population is a smooth deformation of a template

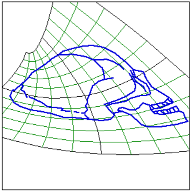
Template estimation / Mean

The idea of Deformable Template Model

- Compare two observations via the quantification of the deformation from one to the other (D'Arcy Thompson, 1917)







Registration / Variance

- Each element of a population is a smooth deformation of a template

Template estimation / Mean

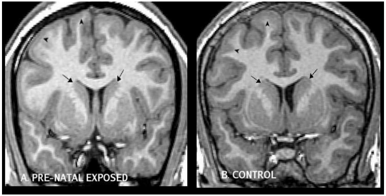
Outline

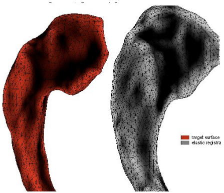
1. Introduction to Computational Anatomy
2. Registration technics

3. Statistical analysis of the deformations
4. Bayesian Modelling for template estimation

Outline

The Registration issue

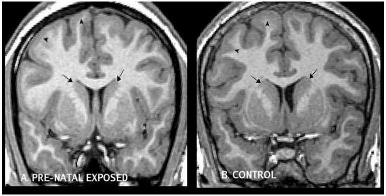


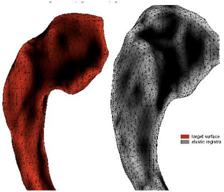


- Deformable template mathematical model : I_0 and I_1 two data :

$$I_1 \simeq \phi \, . \, I_O$$

The Registration issue



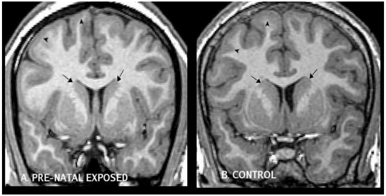


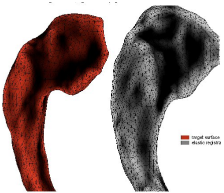
- Deformable template mathematical model : I_0 and I_1 two data :

$$I_1 \simeq \phi \ . \ I_O$$

- How to quantify the difference between the two objects ?

The Registration issue



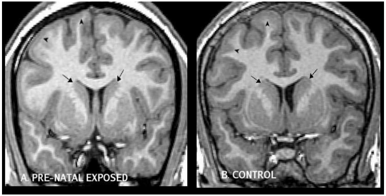


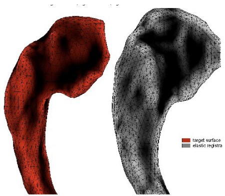
- Deformable template mathematical model : I_0 and I_1 two data :

$$I_1 \simeq \phi \ . \ I_O$$

- How to quantify the difference between the two objects ?
- What kind of deformations ?

The Registration issue





- Deformable template mathematical model : I_0 and I_1 two data :

$$I_1 \simeq \phi \,.\, I_O$$

- How to quantify the difference between the two objects ?
- What kind of deformations ?
- How to apply the deformation to an object ?

Object distances

Difference between two objects : (Same for the following solutions)

- *Images* : L^2 difference of the two functions

$$\|I_1 - \phi \,.\, I_0\|_2^2$$

- *Landmarks* : Sum of the Euclidean distance between points :

$$\sum_{1 \leq i \leq N} \|x_i^1 - \phi \,.\, x_i^0\|_2^2$$

- *Unlabeled landmark set, meshes, fibers* : Requires to embed the objects into a mathematical space where “addition, mean” and other mathematical operations are stable (see Jean) !

Rigid body or affine registration

- Translation, rotation, scaling. Finite dimensional deformation but very restrictive.

Rigid body or affine registration

- Translation, rotation, scaling. Finite dimensional deformation but very restrictive.
- First improvement : ϕ = Affine transformation

Rigid body or affine registration

- Translation, rotation, scaling. Finite dimensional deformation but very restrictive.
- First improvement : $\phi =$ Affine transformation
- Application of the deformation to :

  Images : $\phi \,.\, I_0 \;=\; I_0 \circ \phi^{-1}$

  $\rightarrow$ Image support is deformed, grey levels are transported

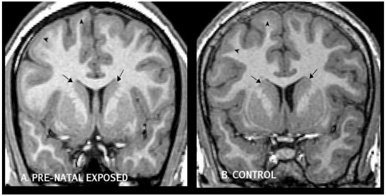
  Landmarks : $\phi \,.\, (x_i)_{1 \leq i \leq N} \;= (\phi(x_i))_{1 \leq i \leq N}$

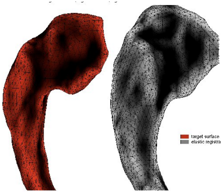
  Others : see Jean !

Rigid body registration

- Easy parametrisation of $\phi \rightarrow$ Fast computations

- Restricted deformations





No affine transformation to match these objects : Need for non linear deformations

Linearised deformations

- Same difference between two objects

Linearised deformations

- Same difference between two objects
- *Idea* : each pixel/voxel/point has its own movement

$$\phi = Id + v$$

where $v \in V$ space of smooth vector fields typically Reproducing Kernel Hilbert Space (RKHS)

Linearised deformations

- Same difference between two objects
- *Idea* : each pixel/voxel/point has its own movement

$$\phi = Id + v$$

where $v \in V$ space of smooth vector fields typically Reproducing Kernel Hilbert Space (RKHS)

- Application of the deformation to :

Images : $\phi \, . \, I_0 \;=\; I_0(Id - v)$
$\rightarrow$ Image support is deformed by $Id - v \simeq \phi^{-1}$,
grey levels are transported

Landmarks : $\phi \, . \, (x_i)_{1 \leq i \leq N} \;= (x_i + v(x_i))_{1 \leq i \leq N}$

Linearised deformations

- ϕ only depends on v $\rightarrow$ Explicit parametrisation
- Relevant for small deformations (where $Id - v \simeq \phi^{-1}$ is valid)

- No invertibility guaranteed : overlaps may appear $\rightarrow$ some tissue may disappear

Linearised deformations

- ϕ only depends on v $\rightarrow$ Explicit parametrisation
- Relevant for small deformations (where $Id - v \simeq \phi^{-1}$ is valid)

- No invertibility guaranteed : overlaps may appear $\rightarrow$ some tissue may disappear
  - Need of a diffeomorphic condition on ϕ

LDDMM framework

Let $\Omega \in \mathbb{R}^d$ be an open set.
The framework defines

- a class of **objects** $\mathcal{O}$ (eg : images $I : \Omega \to \mathbb{R}$, landmarks $(x_i)_{1 \leq i \leq N}$,...)
- a class of **diffeomorphic deformations** $\phi : \Omega \to \Omega$, $\mathcal{D}$
- a specific **group action**

LDDMM framework : The group of diffeomorphisms

- At each time step the deformation is a “small deformation” :

$$\phi_{t_i+\Delta t_i} = Id + v_{t_i}$$

LDDMM framework : The group of diffeomorphisms

- At each time step the deformation is a "small deformation" :

$$\phi_{t_i+\Delta t_i} = Id + v_{t_i}$$

- Final deformation $\phi_1 = \phi_{t_T} \circ \ldots \circ \phi_{t_1} \circ \phi_0$ $(\sum_{i=1}^{T} \Delta t_i = 1)$
- $\phi_0 = Id$ (no deformation to start with)

LDDMM framework : The group of diffeomorphisms

- At each time step the deformation is a “small deformation” :

$$\phi_{t_i+\Delta t_i} = Id + v_{t_i}$$

- Final deformation $\phi_1 = \phi_{t_T} \circ ... \circ \phi_{t_1} \circ \phi_0$ $(\sum_{i=1}^{T} \Delta t_i = 1)$
- $\phi_0 = Id$ (no deformation to start with)
- When $\Delta t \to 0$, ϕ_t solution of :

$$\text{The flow equation : } \begin{cases} \frac{d\phi_t^v}{dt} & = & v_t \circ \phi_t^v \\ \phi_0^v & = & Id \end{cases}$$

LDDMM framework : The group of diffeomorphisms

If $v \in V$, with V "admissible space", then :

- **Existence** and **uniqueness** of the solution of the flow equation

LDDMM framework : The group of diffeomorphisms

If $v \in V$, with V "admissible space", then :

- **Existence** and **uniqueness** of the solution of the flow equation
- The set $\mathcal{D} = \{\phi_1^v,\ v \in L^2([0,1], V)\}$ is a **subgroup of diffeomorphisms** on Ω

LDDMM framework : The group of diffeomorphisms

If $v \in V$, with V "admissible space", then :

- **Existence** and **uniqueness** of the solution of the flow equation
- The set $\mathcal{D} = \{\phi_1^v,\ v \in L^2([0,1], V)\}$ is a **subgroup of diffeomorphisms** on Ω
- $\mathcal{D}$ is equipped with a **right-invariant metric** : $d(\phi, \psi) = d(Id, \psi \circ \phi^{-1})$

LDDMM framework : The group of diffeomorphisms

If $v \in V$, with V "admissible space", then :

- **Existence** and **uniqueness** of the solution of the flow equation
- The set $\mathcal{D} = \{\phi_1^v,\ v \in L^2([0,1], V)\}$ is a **subgroup of diffeomorphisms** on Ω
- $\mathcal{D}$ is equipped with a **right-invariant metric** : $d(\phi, \psi) = d(Id, \psi \circ \phi^{-1})$
- It defines a **group action** : $\phi_1^v\ .\ I = I'$, $\phi_1^v \in \mathcal{D}$ and $I, I' \in \mathcal{O}$

LDDMM framework : The group of diffeomorphisms

If $v \in V$, with V "admissible space", then :

- **Existence** and **uniqueness** of the solution of the flow equation
- The set $\mathcal{D} = \{\phi_1^v,\ v \in L^2([0,1], V)\}$ is a **subgroup of diffeomorphisms** on Ω
- $\mathcal{D}$ is equipped with a **right-invariant metric** : $d(\phi, \psi) = d(Id, \psi \circ \phi^{-1})$
- It defines a **group action** : $\phi_1^v \,.\, I = I'$, $\phi_1^v \in \mathcal{D}$ and $I, I' \in \mathcal{O}$
- and a **distance** between two objects O_0 and O_1 is computed via the group action :

$$d(O_0, O_1) = \inf_{v_t \in V, \phi_1^v(O_0) = O_1} d(Id, \phi_1^v)$$

LDDMM framework : The group of diffeomorphisms

- Existence, uniqueness and regularity (diffeomorphism) of the flow ϕ^v guaranteed for $(v_t)_{t\in[0,1]}$ such as

$$\int_0^1 \|v_t\|_V^2 dt < \infty \ .$$

- Then : $d(O_0, O_1) = \inf_{v_t \in V, \phi_1^v(O_0)=O_1} \{\int_0^1 ||v_t||_V^2 dt\}$

LDDMM framework : The Matching problem

How to match one object onto another ?

- Group action $\Rightarrow$ a way to apply the deformation and **stay in** $\mathcal{O}$

LDDMM framework : The Matching problem

How to match one object onto another ?

- Group action $\Rightarrow$ a way to apply the deformation and **stay in** $\mathcal{O}$

$$+$$

- A distance between **diffeomorphic** objects

LDDMM framework : The Matching problem

How to match one object onto another ?

- Group action $\Rightarrow$ a way to apply the deformation and **stay in** $\mathcal{O}$

$$+$$

- A distance between **diffeomorphic** objects

$$+$$

- A **similarity term between objects**

LDDMM framework : The Matching problem

How to match one object onto another ?

- Group action $\Rightarrow$ a way to apply the deformation and **stay in** $\mathcal{O}$

$$+$$

- A distance between **diffeomorphic** objects : deformation cost

$$+$$

- A **similarity term between objects** : data attachment term

$$=$$

An energy to minimise :

$$E(v) = \underbrace{\frac{1}{2}\int_0^1 \|v_t\|_V^2 dt}_{\text{deformation cost}} + \underbrace{\frac{1}{2\sigma^2}}_{\text{tradeoff parameter}} \times \underbrace{|I_1 - I_0 \circ (\phi_1^v)^{-1}|^2}_{\text{data attachment term}}$$

LDDMM framework : From Hamiltonian to Shooting

The Hamiltonian formulation :

Let V be a RKHS which kernel K_V : (example for N landmarks)

- $\exists\ (p_i(t))_{1 \leq i \leq N} \in \mathbb{R}^n$ such that $v_t(x) = \sum_{i=1}^{N} K_V(x_i(t), x) p_i(t)$

LDDMM framework : From Hamiltonian to Shooting

The Hamiltonian formulation :

Let V be a RKHS which kernel K_V : (example for N landmarks)

- $\exists\ (p_i(t))_{1\leq i\leq N} \in \mathbb{R}^n$ such that $v_t(x) = \sum_{i=1}^{N} K_V(x_i(t), x)p_i(t)$
- $(p_i(t))_{1\leq i\leq N}$ = momenta of the landmarks at time t

LDDMM framework : From Hamiltonian to Shooting

The Hamiltonian formulation :

Let V be a RKHS which kernel K_V : (example for N landmarks)

- $\exists\ (p_i(t))_{1\leq i\leq N} \in \mathbb{R}^n$ such that $v_t(x) = \sum_{i=1}^{N} K_V(x_i(t), x)p_i(t)$
- $(p_i(t))_{1\leq i\leq N}$ = momenta of the landmarks at time t
- Let $H(x, p) = \frac{1}{2}\langle p, K(x)p\rangle$ = Hamiltonian

LDDMM framework : From Hamiltonian to Shooting

The Hamiltonian formulation :

Let V be a RKHS which kernel K_V : (example for N landmarks)

- $\exists \ (p_i(t))_{1 \leq i \leq N} \in \mathbb{R}^n$ such that $v_t(x) = \sum_{i=1}^{N} K_V(x_i(t), x) p_i(t)$
- $(p_i(t))_{1 \leq i \leq N} =$ momenta of the landmarks at time t
- Let $H(x, p) = \frac{1}{2}\langle p, K(x)p \rangle =$ Hamiltonian

Then

LDDMM framework : From Hamiltonian to Shooting

The Hamiltonian formulation :

Let V be a RKHS which kernel K_V : (example for N landmarks)

- $\exists \; (p_i(t))_{1\leq i\leq N} \in \mathbb{R}^n$ such that $v_t(x) = \sum_{i=1}^{N} K_V(x_i(t), x)p_i(t)$
- $(p_i(t))_{1\leq i\leq N}$ = momenta of the landmarks at time t
- Let $H(x,p) = \frac{1}{2}\langle p, K(x)p\rangle$ = Hamiltonian

Then

$$\begin{cases} \frac{dx(t)}{dt} = K_V(x(t))p(t) = \frac{\partial H}{\partial p}(x,p) \\ \frac{dp(t)}{dt} = -\frac{1}{2}\nabla_{x(t)} K_V(p(t),p(t)) = -\frac{\partial H}{\partial x}(x,p) \end{cases}$$

LDDMM framework : From Hamiltonian to Shooting

The Hamiltonian formulation :

Let V be a RKHS which kernel K_V : (example for N landmarks)

- $\exists\ (p_i(t))_{1 \leq i \leq N} \in \mathbb{R}^n$ such that $v_t(x) = \sum_{i=1}^{N} K_V(x_i(t), x) p_i(t)$
- $(p_i(t))_{1 \leq i \leq N}$ = momenta of the landmarks at time t
- Let $H(x, p) = \frac{1}{2}\langle p, K(x)p \rangle$ = Hamiltonian

Then

$$\begin{cases} \frac{dx(t)}{dt} & = & K_V(x(t))p(t) & = & \frac{\partial H}{\partial p}(x,p) \\ \frac{dp(t)}{dt} & = & -\frac{1}{2}\nabla_{x(t)} K_V(p(t), p(t)) & = & -\frac{\partial H}{\partial x}(x,p) \end{cases}$$

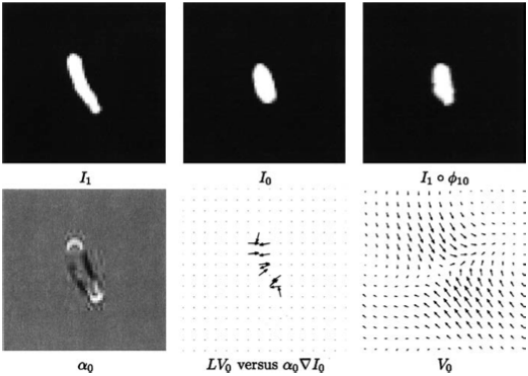
- Hamiltonian constant on the geodesics

LDDMM framework : From Hamiltonian to Shooting

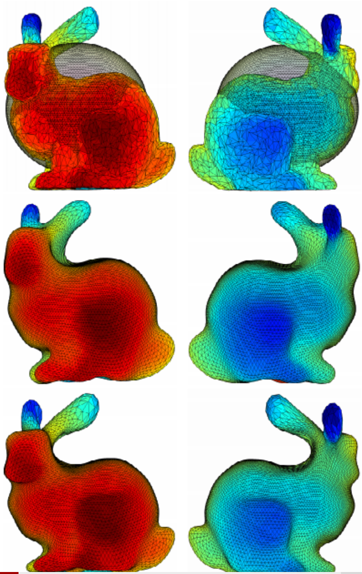
Consequences :

$\Rightarrow$ Parametrisation of the problem by **only** x_0 (known) and p_0

- For images : the momentum supported by the gradient of the image



LDDMM framework : Example of deformations



So now ?

Can do statistics on either the objects or the deformations !
For example :

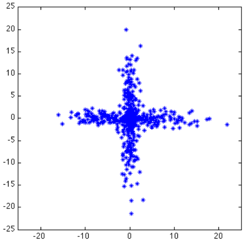
- Global shape analysis
- Local deformation pattern detection

Outline

Outline

1. Introduction to Computational Anatomy
2. Registration technics
3. Statistical analysis of the deformations
   - Noisy ICA general model
   - Gaussian Graphical Models
4. Bayesian Modelling for template estimation

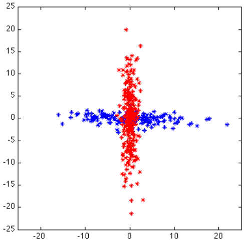
Noisy ICA : Goal of the decomposition :



Data point cloud

- Data : one point cloud
- Goal : explain these data

Noisy ICA : Goal of the decomposition :

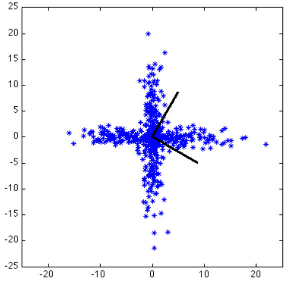


Data point cloud
Mixture of two Gaussian distributions

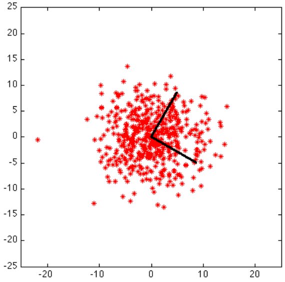
- Data : one point cloud
- Goal : explain these data
- How : Extracting the **"sources"** which had generated the data

Principal Component Analysis (PCA)

- **Orthogonal** direction of maximum variance :
- Assumes a **Gaussian** distribution : $X = \mu + \Sigma^{1/2}\varepsilon$, $\varepsilon \sim \mathcal{N}(0, Id)$.



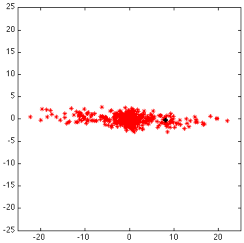
Result from PCA (black lines)



Resampling from the model

Independent Component Analysis (ICA)

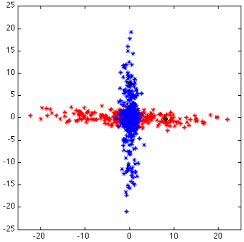
- **Interpretation** of the data ($\neq$ Description)
- Finds **sources** which may have generated the cloud
- Accounts for **non Gaussian** distributions (more flexible model)



First estimated source

Independent Component Analysis (ICA)

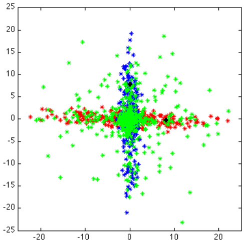
- **Interpretation** of the data ($\neq$ Description)
- Finds **sources** which may have generated the cloud
- Accounts for **non Gaussian** distributions (more flexible model)



First estimated source
Second estimated source

Independent Component Analysis (ICA)

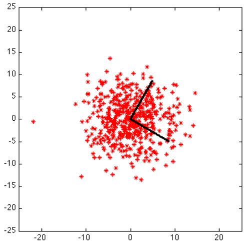
- **Interpretation** of the data ($\neq$ Description)
- Finds **sources** which may have generated the cloud
- Accounts for **non Gaussian** distributions (more flexible model)

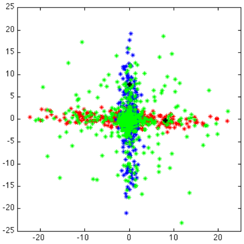


First estimated source
Second estimated source
Resampling form the model

Difference between PCA and ICA

PCA	**ICA**
Maximum variance axes	Source separation
Geometrical (**orthogonal** axis)	Statistical (**source points** in the plane)
Description of data	**Explanation** and **interpretation**
Gaussian distribution **only**	**Many** other possible distributions e.g. mixtures, continuous or discrete (see later)





General hierarchical model :

$$X_i = A\boldsymbol{\beta}_i + \sigma\varepsilon_i$$

$*$ Observations : X_1^n in $(\mathbb{R}^d)^n$

$*$ Source matrix : A called decomposition matrix

$*$ Gaussian noise : $\sigma\varepsilon_i$

$*$ Independent components : $\boldsymbol{\beta}_i \in \mathbb{R}^p$, $p << d$

$\rightarrow$ random vector with **independent** coordinates

General hierarchical model :

$$X_i = A\boldsymbol{\beta}_i + \sigma\varepsilon_i$$

* Observations : X_1^n in $(\mathbb{R}^d)^n$
* Source matrix : A called decomposition matrix
* Gaussian noise : $\sigma\varepsilon_i$
* Independent components : $\boldsymbol{\beta}_i \in \mathbb{R}^p$, $p << d$

$\rightarrow$ random vector with **independent** coordinates

General hierarchical model :

$$X_i = A\boldsymbol{\beta}_i + \sigma\varepsilon_i$$

* Observations : X_1^n in $(\mathbb{R}^d)^n$
* Source matrix : A called decomposition matrix
* Gaussian noise : $\sigma\varepsilon_i$
* Independent components : $\boldsymbol{\beta}_i \in \mathbb{R}^p$, $p << d$

$\rightarrow$ random vector with **independent** coordinates

General hierarchical model :

$$X_i = A\boldsymbol{\beta}_i + \sigma\varepsilon_i$$

$*$ Observations : X_1^n in $(\mathbb{R}^d)^n$
$*$ Source matrix : A called decomposition matrix
$*$ Gaussian noise : $\sigma\varepsilon_i$
$*$ Independent components : $\boldsymbol{\beta}_i \in \mathbb{R}^p$, $p << d$
$\rightarrow$ random vector with **independent** coordinates

General hierarchical model :

$$X_i = A\boldsymbol{\beta}_i + \sigma\varepsilon_i$$

* Observations : X_1^n in $(\mathbb{R}^d)^n$
* Source matrix : A called decomposition matrix
* Gaussian noise : $\sigma\varepsilon_i$
* Independent components : $\boldsymbol{\beta}_i \in \mathbb{R}^p$, $p << d$

$\rightarrow$ random vector with **independent** coordinates

General hierarchical model :

$$X_i = A\boldsymbol{\beta}_i + \sigma\varepsilon_i$$

- $*$ Observations : X_1^n in $(\mathbb{R}^d)^n$
- $*$ Source matrix : A called decomposition matrix
- $*$ Gaussian noise : $\sigma\varepsilon_i$
- $*$ Independent components : $\boldsymbol{\beta}_i \in \mathbb{R}^p$, $p << d$

$\rightarrow$ random vector with **independent** coordinates

β_1^n **hidden variables**.

General hierarchical model :

$X_i = A\beta_i + \sigma\varepsilon_i$

$*$ Observations : X_1^n in $(\mathbb{R}^d)^n$
$*$ Source matrix : A called decomposition matrix
$*$ Gaussian noise : $\sigma\varepsilon_i$
$*$ Independent components : $\beta_i \in \mathbb{R}^p$, $p << d$
$\rightarrow$ random vector with **independent** coordinates
β_1^n **hidden variables**.

- Model : for all images X_i, $1 \leq i \leq n$

$$\begin{cases} \beta_{i,j} & \sim \quad \nu_\eta \mid \eta \,, \forall\, 1 \leq j \leq p \\ X_i & \sim \quad \mathcal{N}(A\beta_i, \sigma^2 Id) \ \mid A,\ \sigma^2,\ \beta_i \,. \end{cases}$$

- **Various choices of the distribution ν_η on the independent components**

Various examples of distributions :

Independent Factor analysis

$$\begin{cases} \beta_{i,j} & \sim \quad \nu_\eta \mid \eta, \forall\, 1 \le j \le p \\ X_i & \sim \quad \mathcal{N}(A\beta_i, \sigma^2 Id) \ \mid A,\ \sigma^2,\ \beta_i \end{cases}$$

- For identifiability, ν_η cannot be Gaussian
- ν_η is a **mixture of K 1D Gaussian distributions** $\mathcal{N}(m_k, 1),\ k = 1, .., K$ with weights $(w_k)_{1 \le k \le K}$.
- $\eta = (m_k, w_k)_{1 \le k \le K}$
- $\theta = (A, \sigma^2, (m_k, w_k)_{1 \le k \le K})$

Various examples of distributions :

Continuous distributions

$$\begin{cases} \boldsymbol{\beta}_{i,j} & \sim & \nu_\eta \mid \eta, \forall\, 1 \leq j \leq p \\ X_i & \sim & \mathcal{N}(A\boldsymbol{\beta}_i, \sigma^2 Id) \ \mid A,\ \sigma^2,\ \boldsymbol{\beta}_i \end{cases}$$

- ν_η is either :
  - **Logistic** $\mathcal{L}og(1/2)$,
  - **Laplacian**,
  - **Exponentially scaled Gaussian(EG)** : $\beta_i^j = s_i^j Y_i^j$ where $\boldsymbol{Y} \sim \mathcal{N}(\boldsymbol{\mu}, Id)$ and $\boldsymbol{\mu} = (\mu, \ldots, \mu)$; $s_i^1, \ldots, s_i^p$ are independent $\mathcal{E}xp(1)$, also independent from $\boldsymbol{Y}$ (sub-exponential tail)
- $\eta = \emptyset$ or μ
- $\theta = (A, \sigma^2)$ or $\theta = (A, \sigma^2, \mu)$

Various examples of distributions :

Discrete distributions

$$\begin{cases} \boldsymbol{\beta}_{i,j} & \sim \quad \nu_\eta \mid \eta \,, \forall\, 1 \leq j \leq p \\ X_i & \sim \quad \mathcal{N}(A\boldsymbol{\beta}_i, \sigma^2 Id) \;\mid A,\; \sigma^2,\; \boldsymbol{\beta}_i \end{cases}$$

Idea : Introduce a switch to cancel some of the decomposition vectors

$\rightarrow$ Either binary ("on/off") or ternary (activate, inhibit, remove)

Various examples of distributions :

Discrete distributions

$$\begin{cases} \boldsymbol{\beta}_{i,j} & \sim & \nu_\eta \mid \eta, \forall\, 1 \leq j \leq p \\ X_i & \sim & \mathcal{N}(A\boldsymbol{\beta}_i, \sigma^2 Id) \ \mid A,\ \sigma^2,\ \boldsymbol{\beta}_i \end{cases}$$

Idea : Introduce a switch to cancel some of the decomposition vectors

$\rightarrow$ Either binary ("on/off") or ternary (activate, inhibit, remove)

- **Bernoulli-censored Gaussian (BG)** : $\beta^j = b^j Y^j$ with $b^j \sim \mathcal{B}(\alpha)$, $\boldsymbol{Y}$ is a Gaussian vector with distribution $\mathcal{N}(\boldsymbol{\mu}, Id)$.
- **Exponentially scaled Bernoulli-censored Gaussian (EBG)** : mix of EG and BG
- **Exponentially-scaled ternary distribution (ET)** : $\beta^j = s^j Y^j$, where $s^1, \ldots, s^p$ are i.i.d. $\mathcal{E}xp(1)$. $\gamma = P(Y^j = -1) = P(Y^j = 1)$, providing a symmetric distribution for the components of $\boldsymbol{Y}$.

- $\theta = (A, \sigma^2, \mu, \alpha, \gamma)$

ML Estimator :

- Parameters θ are estimated by maximum likelihood :

$$\hat{\theta} = \arg\max P(X;\theta)$$

- β_1^n unobserved random variables + Maximise a likelihood
  $\rightarrow$ **EM algorithm** (Expectation - Maximisation)

ML Estimator :

- Parameters θ are estimated by maximum likelihood :

$$\hat{\theta} = \arg\max P(X; \theta)$$

- β_1^n unobserved random variables + Maximise a likelihood
  $\rightarrow$ **EM algorithm** (Expectation - Maximisation)

  Iteration k of the algorithm :

  E Step : Compute the posterior distribution of β_i given X_i and θ_l : $\nu_{i,k}(d\beta_i)$

  M Step : Parameter update :

$$\theta_{k+1} = \arg\max_{\theta} \mathbb{E}_{\nu_{1,k}^n} \left[\log q(\beta_1^n, X_1^n; \theta)|\right].$$

ML Estimator :

- Parameters θ are estimated by maximum likelihood :

$$\hat{\theta} = \arg\max P(X;\theta)$$

- β_1^n unobserved random variables + Maximise a likelihood

  $\rightarrow$ **EM algorithm** (Expectation - Maximisation)

  Iteration k of the algorithm :

  E Step : Compute the posterior distribution of β_i given X_i and θ_l : $\nu_{i,k}(d\beta_i)$

  M Step : Parameter update :

$$\theta_{k+1} = \arg\max_{\theta} \mathbb{E}_{\nu_{1,k}^n}\left[\log q(\beta_1^n, X_1^n; \theta)\right].$$

BUT : E step not computationally tractable !

Others :

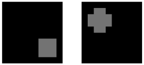
FastICA :

- Employs a fixed point algorithm to minimise the mutual information between the coordinates of $A^{-1}X$

Particle filtering within EM :

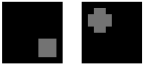
- Approximates the posterior distribution using particle filtering

Experiments

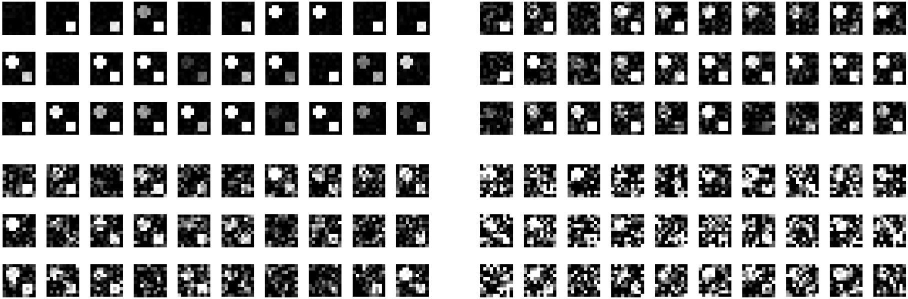


Two source images

Experiments



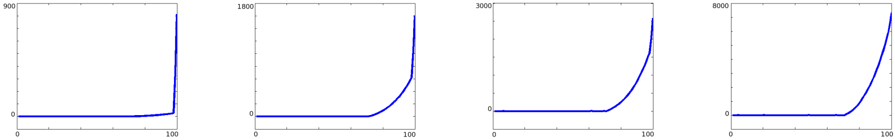
Two source images



Samples of the four training sets different level of noise. From left to right and top to bottom : $\sigma = 0.1,\ 0.5,\ 0.8,\ 1.5$

Results of the PCA decomposition

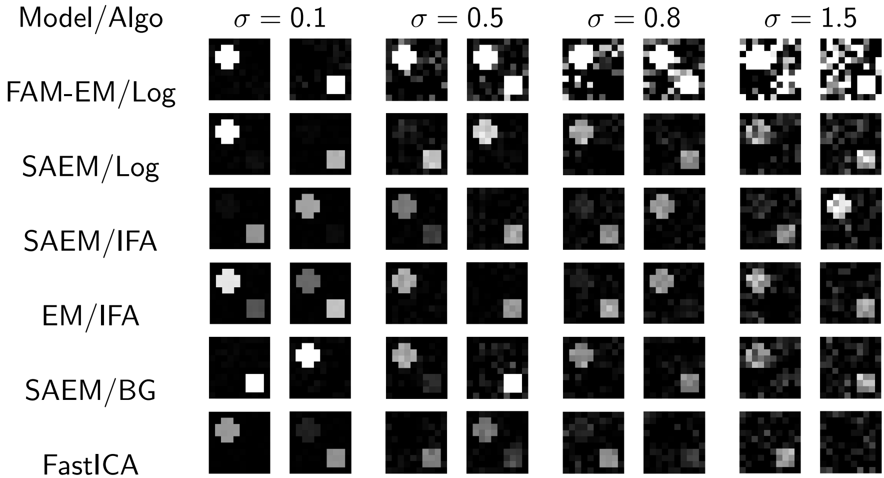
$\sigma = 0.1$ $\sigma = 0.5$ $\sigma = 0.8$ $\sigma = 1.5$



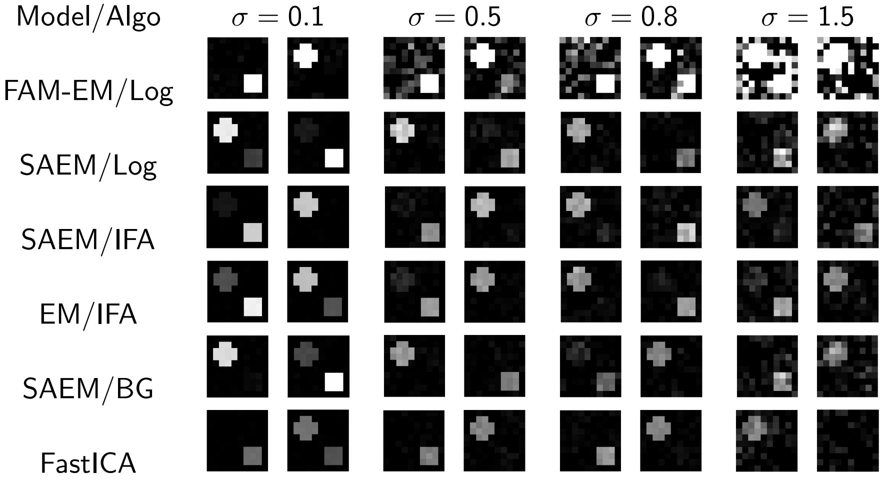
Cumulative eigen values of the PCA decomposition

Two first Principal Components (orthogonal images).

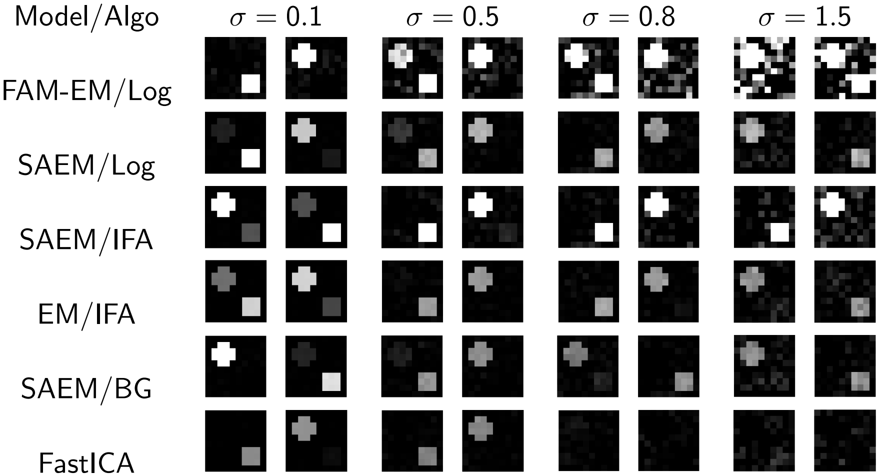
Comparison : 30 images per training set



Comparison : 50 images per training set



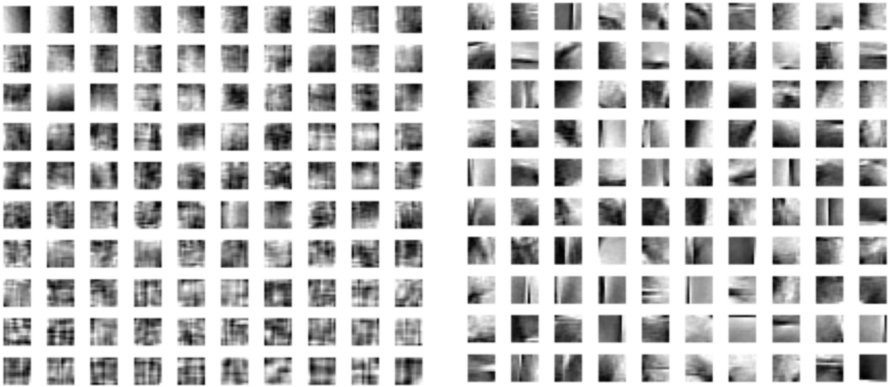
Comparison : 100 images per training set



Patches of faces from Caltech101 database

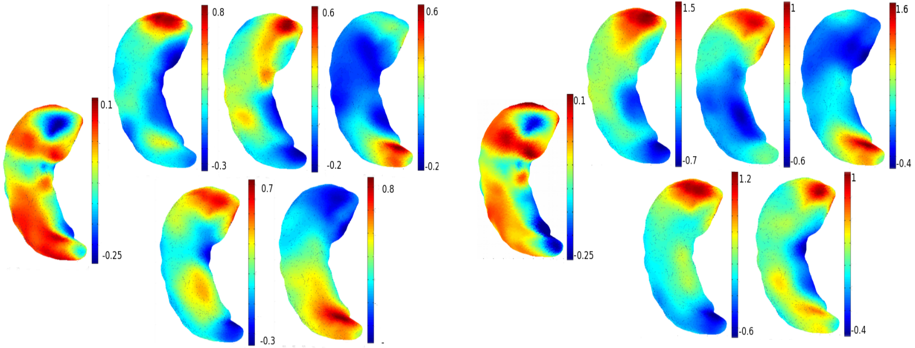
100 random images picked from the 10, 000 *images used as the training set. These images are patches extracted from the face images of the Caltech101 data base. Each image is a grey level image of size* 13×13.

Patches of faces from Caltech101 database



100 *decomposition vectors from 2 models. Left : Log-ICA. Right : BG-ICA.*

101 hippocampus deformations (3 populations : Ctrl - Mild AD - AD)



Mean and five decomposition vectors estimated with L-ICA (left) and ET-ICA (right). Each image has its own colorbar to highlight the major patterns.

101 hippocampus deformations (3 populations : Ctrl - Mild AD - AD)

	Ctrl/AD		Ctrl/mild AD	
Model	L-ICA	BG-ICA	L-ICA	BG-ICA
Mean	0.31×10^{-3}	0.33×10^{-3}	9.0×10^{-3}	1.09×10^{-2}
Std dev.	0.16×10^{-3}	0.25×10^{-3}	3.8×10^{-3}	$4.6\ 7.6 \times 10^{-3}$

TABLE – Mean and standard deviation of the p-values for the two models with the decomposition vectors. Means and standard deviations are computed over 50 runs to separate the Controls from the AD group (left columns) and to separate the Controls from the mild AD group (right columns). PCA p-values : 0.3×10^{-3} and 7.7×10^{-3} using 95% of the cumulative variance.

What more can we look for ?

- Relations between regions of the brain or an anatomical structure

What more can we look for ?

- Relations between regions of the brain or an anatomical structure
- Such as : **correlation patterns**

What more can we look for ?

- Relations between regions of the brain or an anatomical structure
- Such as : **correlation patterns**
- *Example : computation of the correlation matrix using a PCA decomposition*

What more can we look for ?

- Relations between regions of the brain or an anatomical structure
- Such as : **correlation patterns**
- *Example : computation of the correlation matrix using a PCA decomposition*
- **Then** *highlight the most important by thresholding*

What more can we look for ?

- Relations between regions of the brain or an anatomical structure
- Such as : **correlation patterns**
- *Example : computation of the correlation matrix using a PCA decomposition*
- **Then** *highlight the most important by thresholding*

However :

- Correlations describe the global statistical dependencies between variables

What more can we look for ?

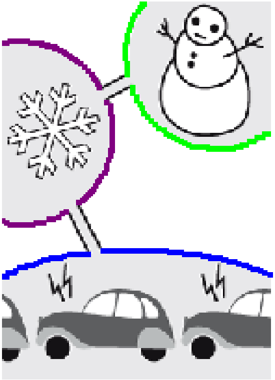
- Relations between regions of the brain or an anatomical structure
- Such as : **correlation patterns**
- *Example : computation of the correlation matrix using a PCA decomposition*
- **Then** *highlight the most important by thresholding*

However :

- Correlations describe the global statistical dependencies between variables

= **both direct and indirect interactions**

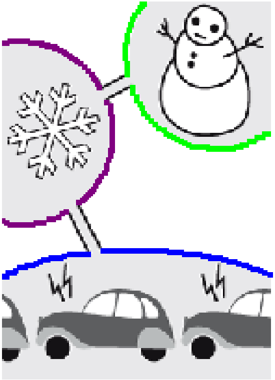
Correlation vs **Conditional** Correlation



Under Gaussian assumption

- Traffic jam intensity **correlated to** Number of snowmen in town **due to snowstorm**.

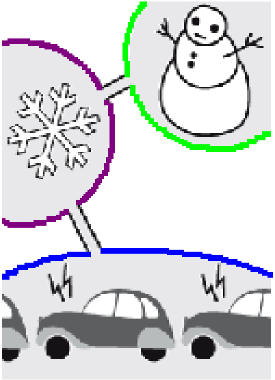
Correlation vs **Conditional** Correlation



Under Gaussian assumption

- Traffic jam intensity **correlated to** Number of snowmen in town **due to snowstorm**.
- **But** conditionally on Snow, Number of snowmen is **independent** of Traffic jams

Correlation vs **Conditional** Correlation



Under Gaussian assumption

- Traffic jam intensity **correlated to** Number of snowmen in town **due to snowstorm**.
- **But** conditionally on Snow, Number of snowmen is **independent** of Traffic jams

  $\Leftrightarrow$ No edge between Traffic jam and Snowmen random variables

Why do we need sparsity ?

- Only **few** of these direct interaction are important

Why do we need sparsity ?

- Only **few** of these direct interaction are important
- No post processing

Why do we need sparsity ?

- Only **few** of these direct interaction are important
- No post processing
- Introduce sparsity into the modelling

Why do we need sparsity ?

- Only **few** of these direct interaction are important
- No post processing
- Introduce sparsity into the modelling

- + high-dimension-low-sample-size paradigm

Why do we need sparsity ?

- Only **few** of these direct interaction are important
- No post processing
- Introduce sparsity into the modelling

- + high-dimension-low-sample-size paradigm
- Although the underlying real graph is **not sparse**, perform a **sparse estimation of its structure**

Why do we need sparsity ?

- Only **few** of these direct interaction are important
- No post processing
- Introduce sparsity into the modelling

- $+$ high-dimension-low-sample-size paradigm
- Although the underlying real graph is **not sparse**, perform a **sparse estimation of its structure**

Significant edges only appear and estimation more stable

The GGM statistical Model

Consider

- p **points** on a given shape = **nodes of the graph**

The GGM statistical Model

Consider

- p **points** on a given shape = **nodes of the graph**
- On these points, we observe n **random responses**

The GGM statistical Model

Consider

- p **points** on a given shape = **nodes of the graph**
- On these points, we observe n **random responses**
- The p nodes of the graph are thus identified to p **random variables** denoted $\boldsymbol{X} = (X_1, ..., X_p)$

The GGM statistical Model

Consider

- p **points** on a given shape = **nodes of the graph**
- On these points, we observe n **random responses**
- The p nodes of the graph are thus identified to p **random variables** denoted $\boldsymbol{X} = (X_1, ..., X_p)$
- $\boldsymbol{X}$ assumed to be distributed as a **multivariate Gaussian** $\mathcal{N}_p(0, \Sigma)$

The GGM statistical Model

Consider

- p **points** on a given shape = **nodes of the graph**
- On these points, we observe n **random responses**
- The p nodes of the graph are thus identified to p **random variables** denoted $\boldsymbol{X} = (X_1, ..., X_p)$
- $\boldsymbol{X}$ assumed to be distributed as a **multivariate Gaussian** $\mathcal{N}_p(0, \Sigma)$

The graph G_Σ of conditional dependencies is defined as follows :

The GGM statistical Model

Consider

- p **points** on a given shape = **nodes of the graph**
- On these points, we observe n **random responses**
- The p nodes of the graph are thus identified to p **random variables** denoted $\boldsymbol{X} = (X_1, ..., X_p)$
- $\boldsymbol{X}$ assumed to be distributed as a **multivariate Gaussian** $\mathcal{N}_p(0, \Sigma)$

The graph G_Σ of conditional dependencies is defined as follows :

- There **exists an edge** between nodes a and b **if and only if** the variables X_a and X_b are **dependent given all the remaining variables**

The GGM statistical Model

Consider

- p **points** on a given shape = **nodes of the graph**
- On these points, we observe n **random responses**
- The p nodes of the graph are thus identified to p **random variables** denoted $\boldsymbol{X} = (X_1, ..., X_p)$
- $\boldsymbol{X}$ assumed to be distributed as a **multivariate Gaussian** $\mathcal{N}_p(0, \Sigma)$

The graph G_Σ of conditional dependencies is defined as follows :

- There **exists an edge** between nodes a and b **if and only if** the variables X_a and X_b are **dependent given all the remaining variables**
- Conditional correlations given by non-zero entries of Σ^{-1}

Introducing a neighbourhood prior

- The neighbouring points of the graph are **very likely** to be conditionally correlated.

Introducing a neighbourhood prior

- The neighbouring points of the graph are **very likely** to be conditionally correlated.
- Information known a priori $\longrightarrow$ Into the statistical model

Introducing a neighbourhood prior

- The neighbouring points of the graph are **very likely** to be conditionally correlated.
- Information known a priori $\longrightarrow$ Into the statistical model
- **New data** : Neighbouring graph G_0

Introducing a neighbourhood prior

- The neighbouring points of the graph are **very likely** to be conditionally correlated.
- Information known a priori $\longrightarrow$ Into the statistical model
- **New data** : Neighbouring graph G_0
- There are correlations between these points

Introducing a neighbourhood prior

- The neighbouring points of the graph are **very likely** to be conditionally correlated.
- Information known a priori $\longrightarrow$ Into the statistical model
- **New data** : Neighbouring graph G_0
- There are correlations between these points
- But **we are not estimating them** rather looking for the other ones = Long distance conditional correlations

Introducing a neighbourhood prior

- The neighbouring points of the graph are **very likely** to be conditionally correlated.
- Information known a priori $\longrightarrow$ Into the statistical model
- **New data** : Neighbouring graph G_0
- There are correlations between these points
- But **we are not estimating them** rather looking for the other ones = Long distance conditional correlations
- **Our idea :** Do the estimation in the **orthogonal space of** G_0

$$X_a - X_{m_a}(X_{m_a}^T X_{m_a})^{-1} X_{m_a}^T X_a \,, \tag{1}$$

Introducing a neighbourhood prior

- The neighbouring points of the graph are **very likely** to be conditionally correlated.
- Information known a priori $\longrightarrow$ Into the statistical model
- **New data** : Neighbouring graph G_0
- There are correlations between these points
- But **we are not estimating them** rather looking for the other ones = Long distance conditional correlations
- **Our idea :** Do the estimation in the **orthogonal space of** G_0

$$X_a - X_{m_a}(X_{m_a}^T X_{m_a})^{-1} X_{m_a}^T X_a \,, \tag{1}$$

- May be too strong constrain (numerically un-invertible) $\longrightarrow$ alleviate through :

$$X_a - X_{m_a}(X_{m_a}^T X_{m_a} + \gamma_0 Id)^{-1} X_{m_a}^T X_a \,. \tag{2}$$

No prior

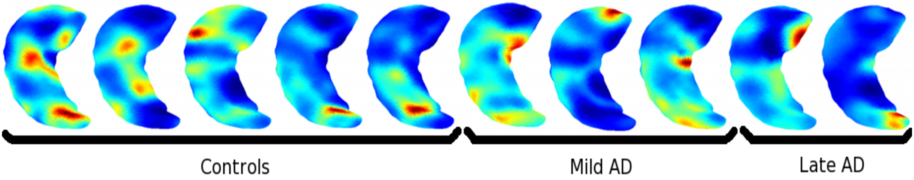
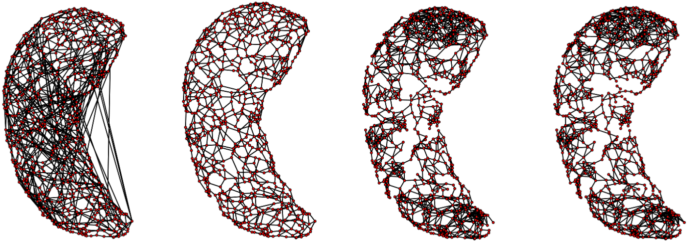


FIGURE – Examples of the training set. The colour depends on the intensity of the Jacobian of the deformation. Blue means a contraction and red dilatation. The intensity itself is not important but rather its relative value with respect to the others.

No prior



Lasso + Or   Lasso + And   Enet + Or   Enet + And

With prior

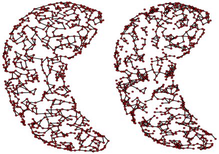
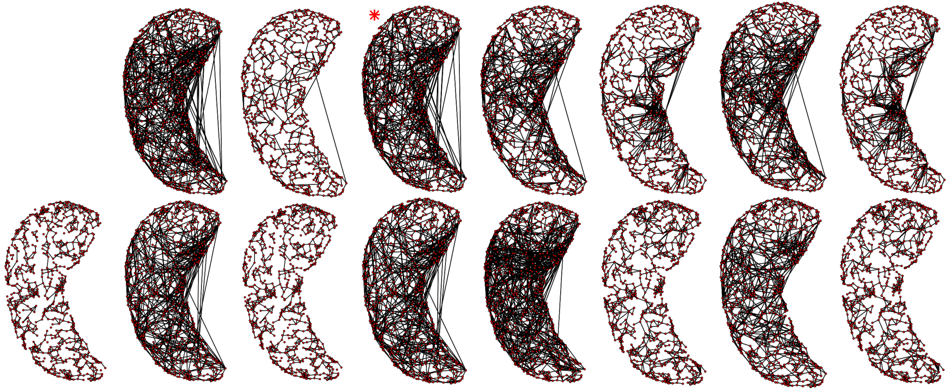


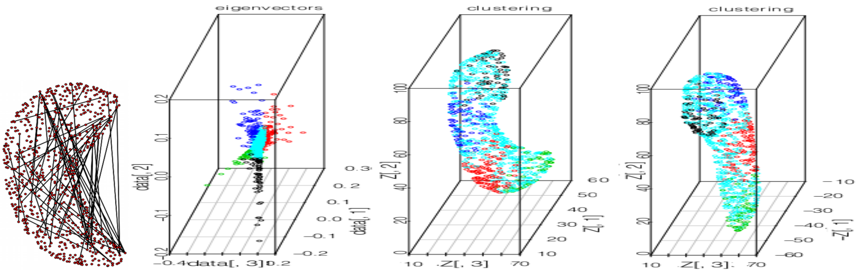
FIGURE – Two examples of neighbourhood graphs we used. Left : 3nearest-neighbour graph. Right : neighbours have Euclidean distance below a given threshold.

Note : User's choice : you can add **connection that you know**.

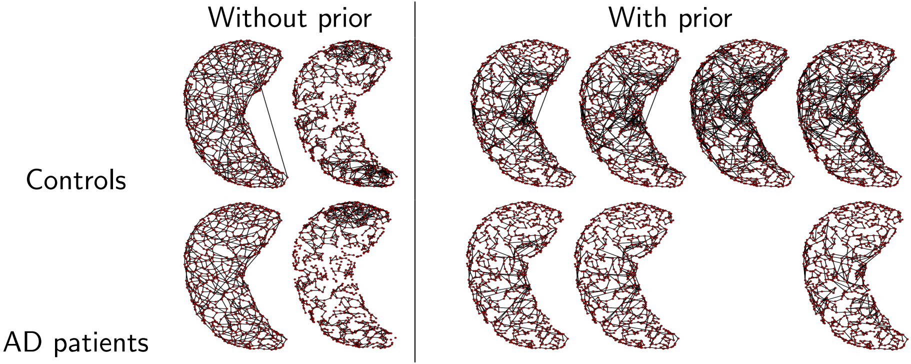


LASSO	LASSO	LASSO	LASSO	Enet	Enet	Enet	Enet
Orth	Orth	Ridge	Ridge	Orth	Orth	Ridge	Ridge
Or	And	Or	And	Or	And	Or	And

Clustering of the Shape : using spectral clustering



Population comparison



Outline

1. Introduction to Computational Anatomy
2. Registration technics
3. Statistical analysis of the deformations
4. Bayesian Modelling for template estimation

Outline

What are the mean and the variability

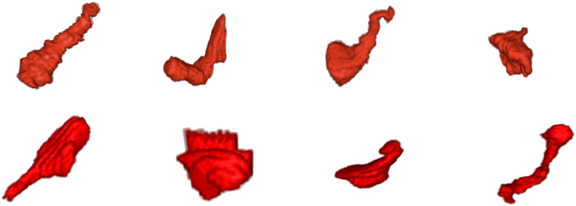
- Matching depends on the **template** I_0

What are the mean and the variability

- Matching depends on the **template** I_0
- What is a good template ?

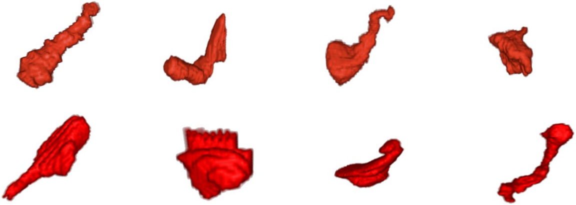
What are the mean and the variability

- Matching depends on the **template** I_0
- What is a good template ?



What are the mean and the variability

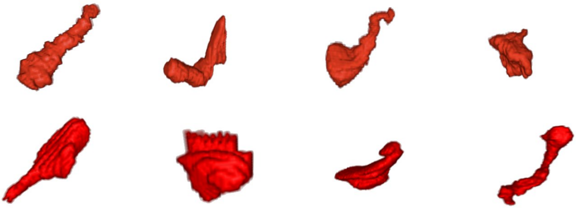
- Matching depends on the **template** I_0
- What is a good template ?



- One of them ?

What are the mean and the variability

- Matching depends on the **template** I_0
- What is a good template ?



- One of them ?
- Which one and why this one ?

What are the mean and the variability

Other question (related)

- Let $(O_i)_{1\leq i\leq n}$ be a homogeneous population (control or AD, or autistic, etc)

What are the mean and the variability

Other question (related)

- Let $(O_i)_{1 \leq i \leq n}$ be a homogeneous population (control or AD, or autistic, etc)
- Problem : O_i lives in a **manifold**

What are the mean and the variability

Other question (related)

- Let $(O_i)_{1\leq i\leq n}$ be a homogeneous population (control or AD, or autistic, etc)
- Problem : O_i lives in a **manifold**
- Question : how to compute a **mean** ? And population **normal variability** ?



BME Template model

- Population of n grey level images y_1^n

BME Template model

- Population of n grey level images y_1^n
- Observation model :

BME Template model

- Population of n grey level images y_1^n
- Observation model :
  - Each observation y belongs to an **unknown** component t

BME Template model

- Population of n grey level images y_1^n
- Observation model :
  - Each observation y belongs to an **unknown** component t
  - Conditional on the image membership to component t,

BME Template model

- Population of n grey level images y_1^n
- Observation model :
  - Each observation y belongs to an **unknown** component t
  - Conditional on the image membership to component t,
    - $\exists\ z : \mathbb{R}^3 \to \mathbb{R}^3$ an **unobserved** deformation field

BME Template model

- Population of n grey level images y_1^n
- Observation model :
  - Each observation y belongs to an **unknown** component t
  - Conditional on the image membership to component t,
    - $\exists\ z : \mathbb{R}^3 \to \mathbb{R}^3$ an **unobserved** deformation field
    - a **continuously defined template** $I_t : \mathbb{R}^3 \to \mathbb{R}$

BME Template model

- Population of n grey level images y_1^n
- Observation model :
  - Each observation y belongs to an **unknown** component t
  - Conditional on the image membership to component t,
    - $\exists\ z : \mathbb{R}^3 \to \mathbb{R}^3$ an **unobserved** deformation field
    - a **continuously defined template** $I_t : \mathbb{R}^3 \to \mathbb{R}$
    - a Gaussian centred white noise ϵ of variance σ_t^2

BME Template model

- Population of n grey level images y_1^n
- Observation model :
  - Each observation y belongs to an **unknown** component t
  - Conditional on the image membership to component t,
    - $\exists\ z : \mathbb{R}^3 \to \mathbb{R}^3$ an **unobserved** deformation field
    - a **continuously defined template** $I_t : \mathbb{R}^3 \to \mathbb{R}$
    - a Gaussian centred white noise ϵ of variance σ_t^2

  such that

$$y(s) = I_t(x_s - z(x_s)) + \epsilon_t(s) = z\ .\ I_t(s) + \epsilon(s)\,,$$

BME Template model

- Template and deformation model :

BME Template model

- Template and deformation model :
  - $I_t \in V_p$ RKHS with kernel K_p
  - Spline model : Given $(p_k)_{1\leq k\leq k_p}$, $\exists\ \alpha_t \in \mathbb{R}^{k_p}$ such that :

$$I_t(x) = \mathbf{K_p}\alpha_t(x), = \sum_{k=1}^{k_p} K_p(x, p_k)\alpha_t(k)$$

BME Template model

- Template and deformation model :
  - $I_t \in V_p$ RKHS with kernel K_p
  - Spline model : Given $(p_k)_{1 \leq k \leq k_p}$, $\exists\ \alpha_t \in \mathbb{R}^{k_p}$ such that :

$$I_t(x) = \mathbf{K_p}\alpha_t(x),\ = \sum_{k=1}^{k_p} K_p(x, p_k)\alpha_t(k)$$

Same spline model for the deformation :

BME Template model

- Template and deformation model :
  - $I_t \in V_p$ RKHS with kernel K_p
  - Spline model : Given $(p_k)_{1\leq k\leq k_p}$, $\exists\ \alpha_t \in \mathbb{R}^{k_p}$ such that :

$$I_t(x) = \mathbf{K_p}\alpha_t(x),\ = \sum_{k=1}^{k_p} K_p(x, p_k)\alpha_t(k)$$

    Same spline model for the deformation :
  - $z \in V_g$ RKHS with kernel K_g
  - Given $(g_k)_{1\leq k\leq k_g}$ $\exists\ (\beta^{(1)}, \beta^{(2)}) \in \mathbb{R}^{k_g} \times \mathbb{R}^{k_g}$ such that :

$$z(x) = (\mathbf{K_g}\beta)(x) = \sum_{k=1}^{k_g} K_g(x, g_k)(\beta^{(1)}(k), \beta^{(2)}(k)).$$

BME Template model

- Hierarchical generative model :

$$
\begin{cases}
\rho \sim \nu_\rho, \\
\theta = (\alpha_t, \sigma_t^2, \Gamma_g^t)_{1 \leq t \leq T} \sim \otimes_{t=1}^{T} (\nu_p \otimes \nu_g) \\
\tau_1^n \sim \otimes_{i=1}^{n} \sum_{t=1}^{T} \rho_t \delta_t \mid \rho \\
\beta_1^n \sim \otimes_{i=1}^{n} \mathcal{N}(0, \Gamma_g^{\tau_i}) \mid \tau_1^n \\
y_1^n \sim \otimes_{i=1}^{n} \mathcal{N}(z_{\beta_i} I_{\alpha_{\tau_i}}, \sigma_{\tau_i}^2 Id_\Lambda) \mid \beta_1^n, \ \tau_1^n
\end{cases}
$$

BME Template model

- Hierarchical generative model :

weights prior

$$\begin{cases} \rho \sim \nu_\rho, \\ \theta = (\alpha_t, \sigma_t^2, \Gamma_g^t)_{1 \leq t \leq T} \sim \otimes_{t=1}^{T} (\nu_p \otimes \nu_g) \\ \tau_1^n \sim \otimes_{i=1}^{n} \sum_{t=1}^{T} \rho_t \delta_t \mid \rho \\ \beta_1^n \sim \otimes_{i=1}^{n} \mathcal{N}(0, \Gamma_g^{\tau_i}) | \ \tau_1^n \\ y_1^n \sim \otimes_{i=1}^{n} \mathcal{N}(z_{\beta_i} I_{\alpha_{\tau_i}}, \sigma_{\tau_i}^2 Id_\Lambda) \mid \beta_1^n, \ \tau_1^n \end{cases}$$

BME Template model

- Hierarchical generative model :

$$\begin{cases} \rho \sim \nu_\rho, \\ \theta = (\alpha_t, \sigma_t^2, \Gamma_g^t)_{1 \le t \le T} \sim \otimes_{t=1}^{T} (\nu_p \otimes \nu_g) \quad \text{parameters priors} \\ \tau_1^n \sim \otimes_{i=1}^{n} \sum_{t=1}^{T} \rho_t \delta_t \mid \rho \\ \beta_1^n \sim \otimes_{i=1}^{n} \mathcal{N}(0, \Gamma_g^{\tau_i}) \mid \tau_1^n \\ y_1^n \sim \otimes_{i=1}^{n} \mathcal{N}(z_{\beta_i} I_{\alpha_{\tau_i}}, \sigma_{\tau_i}^2 Id_\Lambda) \mid \beta_1^n,\ \tau_1^n \end{cases}$$

BME Template model

- Hierarchical generative model :

$$\begin{cases} \rho \sim \nu_\rho, \\ \theta = (\alpha_t, \sigma_t^2, \Gamma_g^t)_{1 \le t \le T} \sim \otimes_{t=1}^T (\nu_p \otimes \nu_g) \\ \tau_1^n \sim \otimes_{i=1}^n \sum_{t=1}^T \rho_t \delta_t \mid \rho \\ \beta_1^n \sim \otimes_{i=1}^n \mathcal{N}(0, \Gamma_g^{\tau_i}) \mid \tau_1^n \\ y_1^n \sim \otimes_{i=1}^n \mathcal{N}(z_{\beta_i} I_{\alpha_{\tau_i}}, \sigma_{\tau_i}^2 Id_\Lambda) \mid \beta_1^n,\ \tau_1^n \end{cases}$$

pick labels for images

BME Template model

- Hierarchical generative model :

$$\begin{cases} \rho \sim \nu_\rho, \\ \theta = (\alpha_t, \sigma_t^2, \Gamma_g^t)_{1\leq t \leq T} \sim \otimes_{t=1}^{T}(\nu_p \otimes \nu_g) \\ \tau_1^n \sim \otimes_{i=1}^{n} \sum_{t=1}^{T} \rho_t \delta_t \mid \rho \\ \beta_1^n \sim \otimes_{i=1}^{n} \mathcal{N}(0, \Gamma_g^{\tau_i}) \mid \tau_1^n \\ y_1^n \sim \otimes_{i=1}^{n} \mathcal{N}(z_{\beta_i} I_{\alpha_{\tau_i}}, \sigma_{\tau_i}^2 Id_\Lambda) \mid \beta_1^n,\ \tau_1^n \end{cases}$$

draw deformations from τ_1^n

BME Template model

- Hierarchical generative model :

$$\begin{cases} \rho \sim \nu_\rho, \\ \theta = (\alpha_t, \sigma_t^2, \Gamma_g^t)_{1 \le t \le T} \sim \otimes_{t=1}^T (\nu_p \otimes \nu_g) \\ \tau_1^n \sim \otimes_{i=1}^n \sum_{t=1}^T \rho_t \delta_t \mid \rho \\ \beta_1^n \sim \otimes_{i=1}^n \mathcal{N}(0, \Gamma_g^{\tau_i}) \mid \tau_1^n \\ y_1^n \sim \otimes_{i=1}^n \mathcal{N}(z_{\beta_i} I_{\alpha_{\tau_i}}, \sigma_{\tau_i}^2 Id_\Lambda) \mid \beta_1^n,\ \tau_1^n \end{cases}$$

draw images from β_1^n and θ_τ

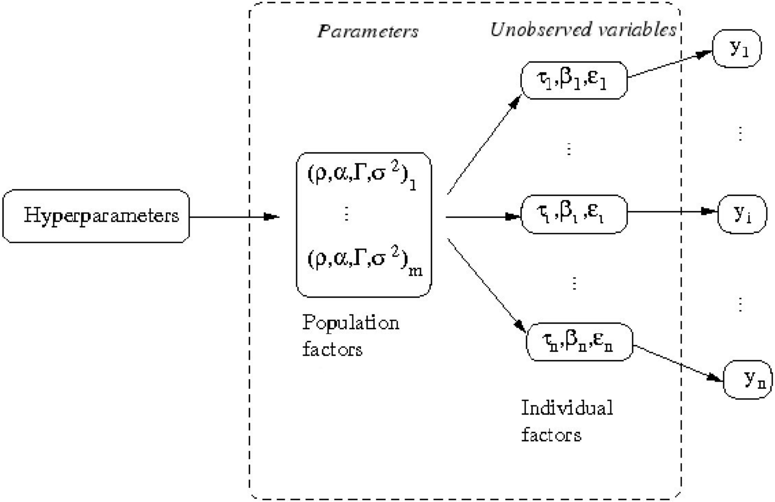
BME Template model

- Hierarchical generative model :

$$\begin{cases} \rho \sim \nu_\rho, \\ \theta = (\alpha_t, \sigma_t^2, \Gamma_g^t)_{1 \leq t \leq T} \sim \otimes_{t=1}^{T} (\nu_p \otimes \nu_g) \\ \tau_1^n \sim \otimes_{i=1}^{n} \sum_{t=1}^{T} \rho_t \delta_t \mid \rho \\ \beta_1^n \sim \otimes_{i=1}^{n} \mathcal{N}(0, \Gamma_g^{\tau_i}) \mid \tau_1^n \\ y_1^n \sim \otimes_{i=1}^{n} \mathcal{N}(z_{\beta_i} I_{\alpha_{\tau_i}}, \sigma_{\tau_i}^2 Id_\Lambda) \mid \beta_1^n, \ \tau_1^n \end{cases}$$

- + weakly informative priors

BME Template model



How to learn the parameters ? the MAP Estimator :

Parameters θ are estimated by maximum posterior likelihood :

$$\hat{\theta} = \arg\max P(\theta|y)$$

where $\theta \in \Theta = \{ (\alpha, \sigma^2, \Gamma_g) | \alpha \in \mathbb{R}^{k_p},\ \sigma^2 > 0,\ \Gamma_g \in \mathcal{S}ym^+_{2k_g,*}(\mathbb{R}) \}$.
$\mathcal{S}ym^+_{2k_g,*}(\mathbb{R})$ is the set of positive definite symmetric matrices.

Let $\Theta_* = \{ \theta_* \in \Theta \mid E_P(\log q(y|\theta_*)) = \sup_{\theta \in \Theta} E_P(\log q(y|\theta)) \}$ where P denotes the distribution governing the observations.

How to do in practice ?

Since β_1^n are unobserved variables, a natural approach to reach the MAP estimator is the **EM algorithm**.

Iteration l of the algorithm :
E Step : Compute the posterior law on $\beta_i, i = 1, \ldots, n$.

M Step : Parameter update :

$$\theta_{l+1} = \arg\max_{\theta} E\left[\log q(\theta, \beta_1^n, y_1^n) | y_1^n, \theta_l\right].$$

BUT : the E step is not tractable !

E step : First solution proposed :

Fast approximation with modes :

- **E Step :**
  $\nu_{i,k}(d\beta_i) \simeq \delta_{\beta_i^*}$, $\forall i = 1, \ldots, n$. β_i^* maximise the conditional distribution on β with the current parameters :

$$\beta_i^* = \arg\max_{\beta} \log q(\beta | X_i; \theta_k)$$

- **M Step :** Parameter update : uses the "completed observations"

$$\theta_{k+1} = \arg\max_{\theta} \log q((\beta^*)_1^n, X_1^n; \theta).$$

Details of the maximisation step :

Geometry :

$$\theta_{g,l+1} = \Gamma_{g,l+1} = \frac{1}{n + a_g}(n[\beta\beta^t]_l + a_g\Sigma_g).$$

where

$$[\beta\beta^t]_l = \frac{1}{n}\sum_{i=1}^{n}\int \beta\beta^t \nu_{l,i}(\beta) d\beta,$$

is the empirical covariance matrix with respect to the posterior density function.

$\rightarrow$ Importance of the prior !

Details of the maximisation step :

Photometry :

$$\begin{cases} \alpha = \left(n \left[\left(K_p^\beta \right)^t K_p^\beta \right]_I + \sigma^2 \Sigma_p^{-1} \right)^{-1} \left(n \left[\left(K_p^\beta \right)^t Y \right]_I + \sigma^2 \Sigma_p^{-1} \mu_p \right) \\ \sigma^2 = \frac{1}{n|\Lambda| + a_p} \left(n \left([Y^t Y]_I + \alpha^t \left[\left(K_p^\beta \right)^t K_p^\beta \right]_I \alpha \right. \right. \\ \qquad \left. \left. -2\alpha^t \left[\left(K_p^\beta \right)^t Y \right]_I \right) + a_p \sigma_0^2 \right) . \end{cases}$$

E step : First solution proposed : Fast approximation with modes :

- $\nu^*_{i,l}(d\beta_i) = \delta_{\beta^*_i}$, β^*_i maximise the conditional distribution on β for each component :

$$\beta^*_{i,\tau} = \arg\max_{\beta} \log q(\beta|\alpha_{l,\tau}, \sigma_{l,\tau}, \Gamma_{g,l,\tau}, y_i, \tau) =$$

$$\arg\min_{\beta} \left\{ \frac{1}{2}\beta^t (\Gamma_{g,l,\tau})^{-1}\beta + \frac{1}{2\sigma^2_{l,\tau}} |y_i - K^\beta_p \alpha_{l,\tau}|^2 \right\} ,$$

- Maximise the conditional distribution on τ given the β^*_i.
- Pick β^*_{i,τ^*}.

Advantages and drawbacks :

- Computation of β_i^* : **standard gradient descent**.
- Reduce the EM algorithm to an iterative maximisation of the joint density.

- Highly sensitive to noise (see experiments)

Our solution : **MCMC- Stochastic Approximation EM** algorithm :

Iteration $k \to k+1$ of the algorithm :

- **Simulation step** : $\beta^{k+1} \sim \Pi_{\theta_l}(\beta^l, \cdot)$
  where $\Pi_{\theta_k}(\beta^k, \cdot)$ is a transition probability of a convergent Markov Chain having the posterior distribution as stationary distribution,
- **Stochastic approximation** :

$$Q_{k+1}(\theta) = Q_k(\theta) + \Delta_k[\log q(\boldsymbol{X}, \beta^{k+1}; \theta) - Q_k(\theta)]$$

  where (Δ_k) is a decreasing sequence of positive step-sizes.
- **Maximisation step** : $\theta_{k+1} = \arg\max Q_{k+1}(\theta)$

[$*$] $\Pi_{\theta_k}(\beta^l, \cdot)$ given by different samplers.

Our solution : **MCMC- Stochastic Approximation EM** algorithm :

Iteration $k \to k+1$ of the algorithm :

- **Simulation step** : $\beta^{k+1} \sim \Pi_{\theta_l}(\beta^l, \cdot)$
  where $\Pi_{\theta_k}(\beta^k, \cdot)$ is a transition probability of a convergent Markov Chain having the posterior distribution as stationary distribution,
- **Stochastic approximation** : Continuous function : need to be stored

$$Q_{k+1}(\theta) = Q_k(\theta) + \Delta_k[\log q(\boldsymbol{X}, \beta^{k+1}; \theta) - Q_k(\theta)]$$

  where (Δ_k) is a decreasing sequence of positive step-sizes.
- **Maximisation step** : $\theta_{k+1} = \arg\max Q_{k+1}(\theta)$

[$*$] $\Pi_{\theta_k}(\beta^l, \cdot)$ given by different samplers.

Our solution : **MCMC- Stochastic Approximation EM** algorithm :

Iteration $k \to k+1$ of the algorithm :

- **Simulation step** : $\beta^{k+1} \sim \Pi_{\theta_l}(\beta^l, \cdot)$
  where $\Pi_{\theta_k}(\beta^k, \cdot)$ is a transition probability of a convergent Markov Chain having the posterior distribution as stationary distribution,
- **Stochastic approximation** : Continuous function : need to be stored

$$Q_{k+1}(\theta) = Q_k(\theta) + \Delta_k[\log q(\boldsymbol{X}, \beta^{k+1}; \theta) - Q_k(\theta)]$$

  where (Δ_k) is a decreasing sequence of positive step-sizes.
- **Maximisation step** : $\theta_{k+1} = \arg\max Q_{k+1}(\theta)$

[$*$] $\Pi_{\theta_k}(\beta^l, \cdot)$ given by different samplers.

Our solution : **MCMC- Stochastic Approximation EM** algorithm : (2) :

But !

- All our models belong to the Exponential family,

$$q(\boldsymbol{X}, \boldsymbol{\beta}^{k+1}; \theta) = \exp\left\{-\psi(\theta) + \langle S(\boldsymbol{X}, \boldsymbol{\beta}), \phi(\theta)\rangle\right\}$$

Our solution : **MCMC- Stochastic Approximation EM** algorithm : (2) :

But !

- All our models belong to the Exponential family,

$$q(\boldsymbol{X}, \boldsymbol{\beta}^{k+1}; \theta) = \exp\left\{-\psi(\theta) + \langle S(\boldsymbol{X}, \boldsymbol{\beta}), \phi(\theta)\rangle\right\}$$

- Stochastic approximation

$$Q_{k+1}(\theta) = Q_k(\theta) + \Delta_k[\log q(\boldsymbol{X}, \boldsymbol{\beta}^{k+1}; \theta) - Q_k(\theta)]$$

Our solution : **MCMC- Stochastic Approximation EM** algorithm : (2) :

But !

- All our models belong to the Exponential family,

$$q(\boldsymbol{X}, \beta^{k+1}; \theta) = \exp\left\{-\psi(\theta) + \langle S(\boldsymbol{X}, \beta), \phi(\theta)\rangle\right\}$$

- Stochastic approximation

$$s_{k+1} = s_k + \Delta_k \left(S(\boldsymbol{X}, \beta^{k+1}) - s_k\right)$$

Our solution : **MCMC- Stochastic Approximation EM** algorithm : (2) :

But !

- All our models belong to the Exponential family,

$$q(\boldsymbol{X}, \beta^{k+1}; \theta) = \exp\{-\psi(\theta) + \langle S(\boldsymbol{X}, \beta), \phi(\theta)\rangle\}$$

- Stochastic approximation

$$s_{k+1} = s_k + \Delta_k \left(S(\boldsymbol{X}, \beta^{k+1}) - s_k \right)$$

- It exists $\hat{\theta}$ (independent of k) such as

$$\theta_{k+1} = \hat{\theta}(s_{k+1}) \ .$$

Our solution : **MCMC- Stochastic Approximation EM** algorithm : (2) :

But !

- All our models belong to the Exponential family,

$$q(\boldsymbol{X}, \boldsymbol{\beta}^{k+1}; \theta) = \exp\left\{-\psi(\theta) + \langle S(\boldsymbol{X}, \boldsymbol{\beta}), \phi(\theta)\rangle\right\}$$

- Stochastic approximation

$$s_{k+1} = s_k + \Delta_k \left(S(\boldsymbol{X}, \boldsymbol{\beta}^{k+1}) - s_k \right)$$

- It exists $\hat{\theta}$ (independent of k) such as

$$\theta_{k+1} = \hat{\theta}(s_{k+1}) \ .$$

Very simple algorithm !

Theoretical Results :

With these models and algorithms we have proved some important asymptotic results :

Conditions :

- Smoothness of the model (classic conditions for convergence of stochastic approximation and EM)
- In case of AMALA sampler : condition for its geometric ergodicity

Theoretical Results :

With these models and algorithms we have proved some important asymptotic results :

Conditions :

- Smoothness of the model (classic conditions for convergence of stochastic approximation and EM)
- In case of AMALA sampler : condition for its geometric ergodicity

Results :

- Convergence of (s_k) a.s. towards critical point of mean field of the problem

Theoretical Results :

With these models and algorithms we have proved some important asymptotic results :

Conditions :

- Smoothness of the model (classic conditions for convergence of stochastic approximation and EM)
- In case of AMALA sampler : condition for its geometric ergodicity

Results :

- Convergence of (s_k) a.s. towards critical point of mean field of the problem
- Convergence of estimated parameters (θ_k) a.s. towards critical point of observed likelihood

Theoretical Results :

With these models and algorithms we have proved some important asymptotic results :

Conditions :

- Smoothness of the model (classic conditions for convergence of stochastic approximation and EM)
- In case of AMALA sampler : condition for its geometric ergodicity

Results :

- Convergence of (s_k) a.s. towards critical point of mean field of the problem
- Convergence of estimated parameters (θ_k) a.s. towards critical point of observed likelihood
- Central limit theorem for (θ_k)

Training sets

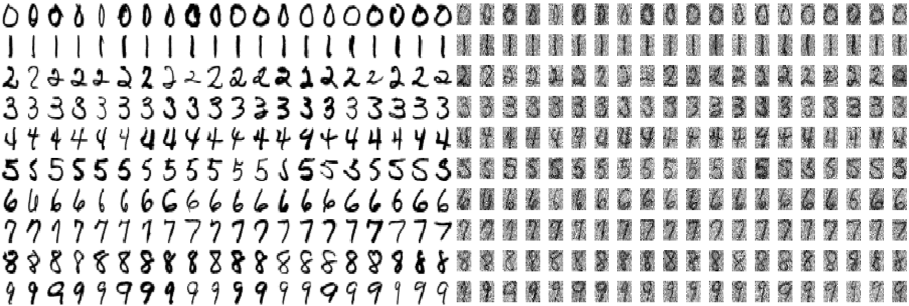


FIGURE – Left : Training set (inverse video). Right : Noisy training set (inverse video).

MCMC-SAEM algorithm :

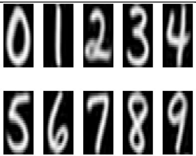
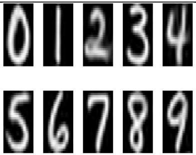
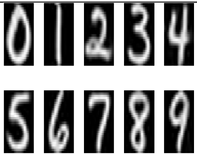
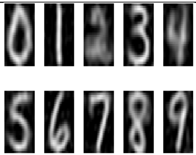
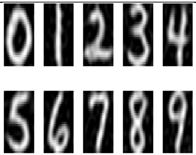
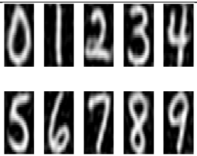
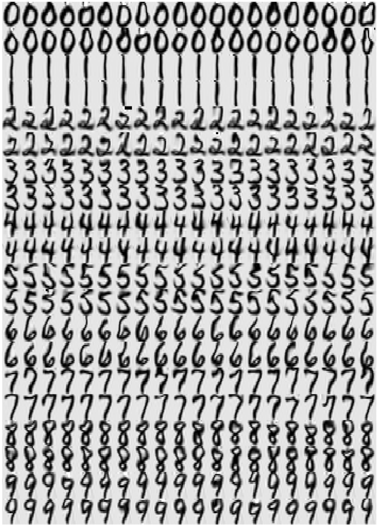
Algorithm/ Noise level	FAM-EM	H.G.-SAEM	AMALA-SAEM
No Noise			
Noisy of Variance 1			

FIGURE – Estimated templates using different algorithms and two level of noise. The training set includes 20 images per digit.

Estimated geometric variability



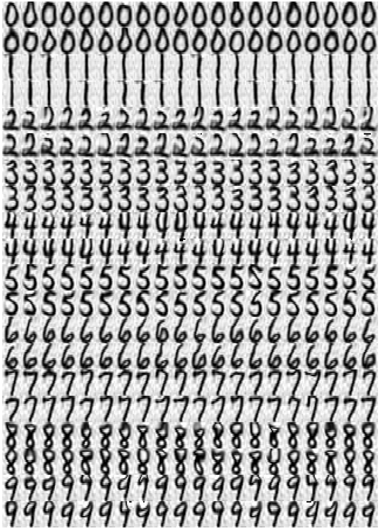


FIGURE – Synthetic samples generated with respect to the BME template model

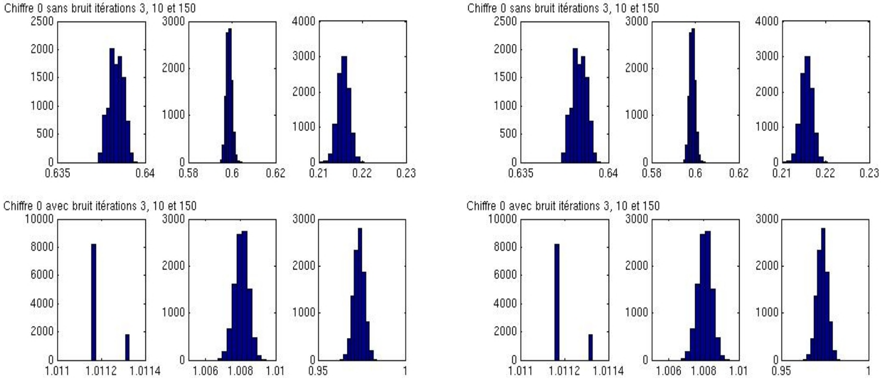


FIGURE – Evolution of the estimation of the noise variance along the SAEM iterations. Test of convergence towards the Gaussian distribution of the estimated parameters.

Classification rates :

Error rate	"EM-Mode" est.	SAEM-MCMC est.
Mode classifier	40.71	22.52
MCMC classifier	-	17.07

TABLE – Error rate with respect to the estimation and classification methods.

3D dendrite spines :

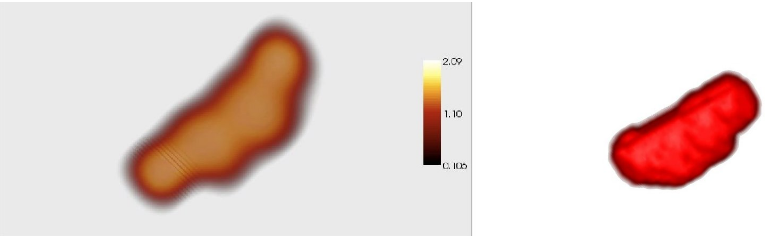


FIGURE – Estimated template with the one component model : Left : 3D representation of the grey level volume. Right : 3D representation of the thresholded volume.

3D dendrite spines :

FIGURE – Estimated templates of the two components with the 30 image training set : 3D representation after thresholding.

3D dendrite spines :

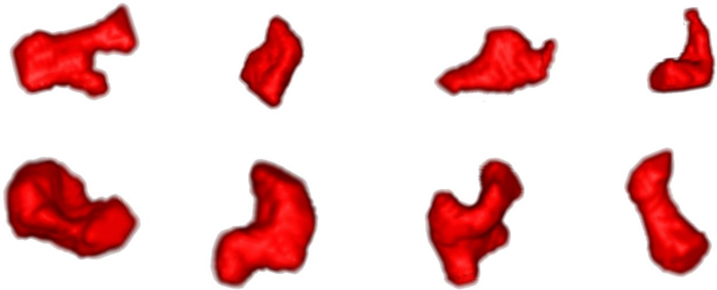


FIGURE – 3D view of eight synthetic data. The estimated template shown in Figure 7 is randomly deformation with respect to the estimated covariance matrix. The results are then thresholded in order to get a binary volume.

One step further ?

Models of longitudinal data !