Computational information geometry for audio signal processing
(Based on a tutorial presented at DAFx 2012)

Arnaud Dessein
Institut de Recherche et Coordination Acoustique/Musique
Equipe-Projet MuSync

February 10th 2012
Preface

- Information geometry is:
  - The marriage of *differential geometry*, *information theory*, and *statistical learning*.
  - Considering *probabilistic representations* as well-behaved *geometrical objects* with intuitive geometric properties.
Information geometry is:

- The marriage of *differential geometry*, *information theory*, and *statistical learning*.
- Considering *probabilistic representations* as well-behaved *geometrical objects* with intuitive geometric properties.

Historical remarks:

- Widely popularized among engineers by [Amari & Nagaoka, 2000].
- Actively researched and in application for *radar processing*, *image analysis*, *computational anatomy*, etc.
### Motivations

<table>
<thead>
<tr>
<th>Representation</th>
<th>Type</th>
<th>Data rate</th>
<th>Information quantity</th>
<th>Information Reduction: Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>p(x,y,z,t)</em></td>
<td>Physical</td>
<td>10-100 kHz</td>
<td>High information quantity</td>
<td>Information Generation: Synthesis</td>
</tr>
<tr>
<td>Control</td>
<td>Control</td>
<td>10Hz-1kHz</td>
<td>Low information quantity</td>
<td>Explicit representations</td>
</tr>
<tr>
<td>Signal</td>
<td>Symbolic</td>
<td>10-100 kHz</td>
<td>Implicit knowledge</td>
<td>Synthesis</td>
</tr>
<tr>
<td>Semantic</td>
<td>Semantic</td>
<td>&lt; 0.1 Hz</td>
<td></td>
<td>Analysis</td>
</tr>
</tbody>
</table>

**Figure:** Levels of representation of audio [Vinet, 2004], waveform and spectrogram representations.
Motivations

- Develop a comprehensive framework that allows to quantify, process and represent the information contained in audio signals, so as to fill in the gap between signal and symbolic representations [Cont et al., 2011].

**Figure:** Levels of representation of audio [Vinet, 2004], waveform and spectrogram representations.
Outline

1. Information geometry
2. Machine learning aspects
3. Framework for audio processing
4. Some applications
Outline

1. Information geometry
   - Background
   - Exponential families

2. Machine learning aspects

3. Framework for audio processing

4. Some applications
What is information geometry?

Statistical differentiable manifold.

Under certain assumptions, a parametric statistical model $S = \{ p_{\xi} : \xi \in \Xi \}$ of probability densities defined on $\mathcal{X}$ forms a differentiable manifold.
Information geometry
Machine learning aspects
Framework for audio processing
Some applications

What is information geometry?

Statistical differentiable manifold.

Under certain assumptions, a parametric statistical model $S = \{p_\xi : \xi \in \Xi\}$ of probability densities defined on $\mathcal{X}$ forms a differentiable manifold.

Example: $p_\xi(x) = \frac{1}{\sqrt{2\pi}\sigma^2} \exp \left\{ -\frac{(x - \mu)^2}{2\sigma^2} \right\}$ for all $x \in \mathcal{X} = \mathbb{R}$, with $\xi = [\mu, \sigma^2] \in \Xi = \mathbb{R} \times \mathbb{R}_{++}$.
What is information geometry?

**Statistical differentiable manifold.**

Under certain assumptions, a parametric statistical model \( S = \{ p_\xi : \xi \in \Xi \} \) of probability densities defined on \( \mathcal{X} \) forms a differentiable manifold.

- Example: \( p_\xi(x) = \frac{1}{\sqrt{2\pi}\sigma^2} \exp \left\{ -\frac{(x - \mu)^2}{2\sigma^2} \right\} \) for all \( x \in \mathcal{X} = \mathbb{R} \), with \( \xi = [\mu, \sigma^2] \in \Xi = \mathbb{R} \times \mathbb{R}_{++} \).

**Fisher information metric** [Rao, 1945, Chentsov, 1982].

Under certain assumptions, the Fisher information matrix defines the unique Riemannian metric \( g \) on \( S \): \( g_{ij}(\xi) = E_\xi[\partial_i \log p_\xi \partial_j \log p_\xi] \).
What is information geometry?

### Statistical differentiable manifold.

Under certain assumptions, a parametric statistical model \( S = \{p_\xi : \xi \in \Xi\} \) of probability densities defined on \( \mathcal{X} \) forms a differentiable manifold.

- Example: \( p_\xi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(x - \mu)^2}{2\sigma^2} \right\} \) for all \( x \in \mathcal{X} = \mathbb{R} \), with \( \xi = [\mu, \sigma^2] \in \Xi = \mathbb{R} \times \mathbb{R}_{++} \).

### Fisher information metric [Rao, 1945, Chentsov, 1982].

Under certain assumptions, the Fisher information matrix defines the unique Riemannian metric \( g \) on \( S \): \( g_{ij}(\xi) = E_\xi[\partial_i \log p_\xi \partial_j \log p_\xi] \).

### Affine connections [Chentsov, 1982, Amari & Nagaoka, 2000].

Under certain assumptions, the \( \alpha \)-connections \( \nabla^{(\alpha)} \) for \( \alpha \in \mathbb{R} \) are the unique affine connections on \( S \): \( \nabla^{(\alpha)} \partial_i \partial_j = \Gamma^{(\alpha)}_{ij,k}(\xi) \partial_k \) where

\[
\Gamma^{(\alpha)}_{ij,k}(\xi) = E_\xi \left[ (\partial_i \partial_j \log p_\xi + \frac{1-\alpha}{2} \partial_i \log p_\xi \partial_j \log p_\xi) (\partial_k \log p_\xi) \right].
\]
How to use information geometry from a computational viewpoint?

Exponential family [Darmois, 1935, Koopman, 1936, Pitman, 1936].

\[ p_\theta(x) = \exp \left( \theta^T T(x) - F(\theta) + C(x) \right) \quad \text{for all } x \in \mathcal{X}. \]

- \( \theta \): natural parameters in a convex open set \( \Theta \).
- \( F(\theta) \): log-normalizer, smooth strictly convex function on \( \Theta \).
- \( C(x) \): carrier measure, measurable function on \( \mathcal{X} \).
- \( T(x) \): sufficient statistic, measurable function on \( \mathcal{X} \).
How to use information geometry from a computational viewpoint?

Exponential family [Darmois, 1935, Koopman, 1936, Pitman, 1936].

\[ p_\theta(x) = \exp(\theta^T T(x) - F(\theta) + C(x)) \text{ for all } x \in \mathcal{X}. \]

**Figure:** A taxonomy of probability measures [Nielsen & Garcia, 2009].
How to use information geometry from a computational viewpoint?

**Exponential family** [Darmois, 1935, Koopman, 1936, Pitman, 1936].

\[ p_\theta(x) = \exp \left( \theta^T T(x) - F(\theta) + C(x) \right) \text{ for all } x \in \mathcal{X}. \]

- We consider a statistical manifold \( \mathcal{S} = \{ p_\theta : \theta \in \Theta \} \) equipped with \( g \) and the dual exponential and mixture connections \( \nabla^{(1)} \) and \( \nabla^{(-1)} \).

- \((\mathcal{S}, g, \nabla^{(1)}, \nabla^{(-1)})\) possesses two dual affine coordinate systems, natural parameters \( \theta \) and expectation parameters \( \eta = \nabla F(\theta) \).

- Dually flat geometry, Hessian structure, generated by the potential \( F \) together with its conjugate potential \( F^* \) defined by the Legendre-Fenchel transform: \( F^*(\eta) = \sup_{\theta \in \Theta} \theta^T \eta - F(\theta) \), which verifies \( \nabla F^* = (\nabla F)^{-1} \) so that \( \theta = \nabla F^*(\eta) \).

- Generalizes the self-dual Euclidean geometry, with notably two canonically associated Bregman divergences \( B_F \) and \( B_{F^*} \) instead of the self-dual Euclidean distance, but also dual geodesics, a generalized Pythagorean theorem and dual projections.
How to use information geometry from a computational viewpoint?

**Exponential family [Darmois, 1935, Koopman, 1936, Pitman, 1936].**

\[ p_\theta(x) = \exp(\theta^T T(x) - F(\theta) + C(x)) \text{ for all } x \in \mathcal{X}. \]

**Bregman divergence [Bregman, 1967].**

\[ B_G(\theta, \theta') = G(\theta) - G(\theta') - (\theta - \theta')^T \nabla G(\theta'). \]

- Canonical divergences of dually flat spaces, “bijection” with exponential families [Amari & Nagaoka, 2000, Banerjee et al., 2005]:
  \[ D_{KL}(p_\xi \parallel p_{\xi'}) = B_F(\theta' \parallel \theta) = B_F^*(\eta \parallel \eta'). \]

- No symmetry nor triangular inequality in general, but an information-theoretic interpretation.

  - Centroid computation and hard clustering (k-means).
  - Parameter estimation and soft clustering (expectation-maximization).
  - Proximity queries in ball trees (nearest-neighbors and range search).
How to use information geometry from a computational viewpoint?

**Exponential family** [Darmois, 1935, Koopman, 1936, Pitman, 1936].

\[ p_\theta(x) = \exp \left( \theta^T T(x) - F(\theta) + C(x) \right) \text{ for all } x \in \mathcal{X}. \]

**Figure:** Geometrical viewpoint [Nielsen & Nock, 2009].
Outline

1 Information geometry

2 Machine learning aspects
   - Centroid computation
   - Hard clustering
   - Soft clustering
   - Proximity queries
   - Change detection

3 Framework for audio processing

4 Some applications
How to compute the centroid of distributions?


\[
\hat{\theta} = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{j=1}^{n} B F(\theta \| \theta_j).
\]
How to compute the centroid of distributions?


- Right-sided centroid on natural parameters: \( \hat{\theta} = \arg \min_{\theta \in \Theta} \frac{1}{n} \sum_{j=1}^{n} B_F(\theta_j \parallel \theta) \).

- Non-convex optimization problem in general, but admits the following unique global solution: \( \hat{\theta} = \frac{1}{n} \sum_{j=1}^{n} \theta_j \).
How to compute the centroid of distributions?


- Right-sided centroid on natural parameters: \( \hat{\theta} = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{j=1}^{n} B_F(\theta_j \parallel \theta) \).

- Non-convex optimization problem in general, but admits the following unique global solution: \( \hat{\theta} = \frac{1}{n} \sum_{j=1}^{n} \theta_j \).

- Left-sided centroid on natural parameters: \( \hat{\theta} = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{j=1}^{n} B_F(\theta \parallel \theta_j) \).

- By duality: \( \hat{\theta} = (\nabla F)^{-1} \left( \frac{1}{n} \sum_{j=1}^{n} \nabla F(\theta_j) \right) \).
How to compute the centroid of distributions?


- Right-sided centroid on natural parameters: \( \hat{\theta} = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{j=1}^{n} \mathcal{B}_F(\theta_j \parallel \theta) \).

- Non-convex optimization problem in general, but admits the following unique global solution: \( \hat{\theta} = \frac{1}{n} \sum_{j=1}^{n} \theta_j \).

- Left-sided centroid on natural parameters: \( \hat{\theta} = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{j=1}^{n} \mathcal{B}_F(\theta \parallel \theta_j) \).

- By duality: \( \hat{\theta} = (\nabla F)^{-1} \left( \frac{1}{n} \sum_{j=1}^{n} \nabla F(\theta_j) \right) \).

- Observed point through sufficient statistics: \( \eta = T(x) \).

- Maximum likelihood estimator as a centroid: \( \hat{\eta} = \frac{1}{n} \sum_{j=1}^{n} \eta_j = \frac{1}{n} \sum_{j=1}^{n} T(x_j) \).
How to cluster observations or distributions?

- $k$-means with Bregman divergences [Banerjee et al., 2005].
How to cluster observations or distributions?

- *k*-means with Bregman divergences [Banerjee et al., 2005].

**Algorithm:**

1. Initialize the clusters.
2. Update the centroids of each cluster.
3. Assign each point to the cluster whose centroid is the closest to the point.
4. Repeat centroid update and point assignment until convergence.
How to cluster observations or distributions?

- $k$-means with Bregman divergences [Banerjee et al., 2005].
- Algorithm:
  1. Initialize the clusters.
  2. Update the centroids of each cluster.
  3. Assign each point to the cluster whose centroid is the closest to the point.
  4. Repeat centroid update and point assignment until convergence.
- Observations: on the observed points $\eta_j = T(x_j)$ with the corresponding Bregman divergences, estimate of the generating parameter in each cluster as a by-product.
How to cluster observations or distributions?

- $k$-means with Bregman divergences [Banerjee et al., 2005].

- Algorithm:
  1. Initialize the clusters.
  2. Update the centroids of each cluster.
  3. Assign each point to the cluster whose centroid is the closest to the point.
  4. Repeat centroid update and point assignment until convergence.

- Observations: on the observed points $\eta_j = T(x_j)$ with the corresponding Bregman divergences, estimate of the generating parameter in each cluster as a by-product.

- Distributions: on the parameters $\theta_j$ or $\eta_j$ with the corresponding Bregman divergences, equivalent to clustering the densities with the Kullback-Leibler divergence.
How to estimate the parameters of a mixture model?

- Expectation-maximization with exponential families [Banerjee et al., 2005].
How to estimate the parameters of a mixture model?

- Expectation-maximization with exponential families [Banerjee et al., 2005].
- Algorithm:
  1. Initialize the parameters and weights.
  2. Expectation step: compute the conditional expectation of the log-likelihood given the observations and the actual parameters and weights.
  3. Maximization step: update the parameters and weights by maximizing the actual expectation.
  4. Repeat expectation and maximization steps until convergence (of the likelihood).
How to estimate the parameters of a mixture model?

- Expectation-maximization with exponential families [Banerjee et al., 2005].
- Algorithm:
  1. Initialize the parameters and weights.
  2. Expectation step: compute the conditional expectation of the log-likelihood given the observations and the actual parameters and weights.
  3. Maximization step: update the parameters and weights by maximizing the actual expectation.
  4. Repeat expectation and maximization steps until convergence (of the likelihood).
- Distributions: this provides an estimate of the parameters $\theta_j$ and weights or $w_j$ of each component distribution of the mixture model.
How to estimate the parameters of a mixture model?

- Expectation-maximization with exponential families [Banerjee et al., 2005].
- Algorithm:
  1. Initialize the parameters and weights.
  2. Expectation step: compute the conditional expectation of the log-likelihood given the observations and the actual parameters and weights.
  3. Maximization step: update the parameters and weights by maximizing the actual expectation.
  4. Repeat expectation and maximization steps until convergence (of the likelihood).
- Distributions: this provides an estimate of the parameters $\theta_j$ and weights $\omega_j$ of each component distribution of the mixture model.
- Observations: it provides a probabilistic clustering of the observations as a by-product.
How to retrieve observations or distributions in a database?

- Nearest-neighbors and range search with Bregman divergences [Cayton, 2008, Cayton, 2009, Nielsen et al., 2009, Garcia et al., 2009].
How to retrieve observations or distributions in a database?

- Nearest-neighbors and range search with Bregman divergences [Cayton, 2008, Cayton, 2009, Nielsen et al., 2009, Garcia et al., 2009].
- Principle:
  1. Index the database with a Bregman ball tree.
  2. Choose the node whose centroid is the closest to the query.
  3. Prune the nodes by computing Bregman ball intersection or containment.
  4. Inspect the other nodes.
How to retrieve observations or distributions in a database?

- Nearest-neighbors and range search with Bregman divergences [Cayton, 2008, Cayton, 2009, Nielsen et al., 2009, Garcia et al., 2009].
- Principle:
  1. Index the database with a Bregman ball tree.
  2. Choose the node whose centroid is the closest to the query.
  3. Prune the nodes by computing Bregman ball intersection or containment.
  4. Inspect the other nodes.
- Intersection and containment can be efficiently solved by dichotomic walks on geodesics.

![Diagram of Bregman balls with different shapes and colors representing various types of geometry and probability distributions.]

**Figure:** Intersection and containment of Bregman balls.
How to detect a change in a series of observations or distributions?

- Previous approach based on threshold heuristic on the Bregman ball radius [Dessein & Cont, 2011].
- Purely geometric approach, robustness issues.

**Figure:** Change detection at time $t$. 
How to detect a change in a series of observations or distributions?

- Previous approach based on threshold heuristic on the Bregman ball radius [Dessein & Cont, 2011].
- Purely geometric approach, robustness issues.

**Figure**: Change detection at time $t$.

- CuSum or GLR-type algorithms with exponential families.
How to detect a change in a series of observations or distributions?

- Previous approach based on threshold heuristic on the Bregman ball radius [Dessein & Cont, 2011].
- Purely geometric approach, robustness issues.

![Figure: Change detection at time $t$.](image)

- CuSum or GLR-type algorithms with exponential families.
- Principle:
  1. Hypotheses formulation: $H_0$: $x_1, \ldots, x_n \sim p_{\theta_0}$
     $H_1^i$: $x_1, \ldots, x_i \sim p_{\theta_i}$, and $x_{i+1}, \ldots, x_n \sim p_{\theta_1}$.
  2. Estimation of the parameters after and before change: $\hat{\theta}_0$, $\hat{\theta}_0^i$, $\hat{\theta}_1^i$.
  4. Computationally efficient sequential updates of this test for the mles.
Outline

1. Information geometry
2. Machine learning aspects
3. Framework for audio processing
   - General system
   - Sound descriptors modeling
4. Some applications
How to design a system for audio processing?

**Scheme:**

1. Represent the incoming audio stream with short-time sound descriptors $d_j$.
2. Model these descriptors with probability distributions $p_{\theta_j}$ from a given exponential family.
3. Use the framework of computational information geometry on these distributions.

**Figure:** Schema of the general architecture of the system.
How to design a system for audio processing?

**Scheme:**
1. Represent the incoming audio stream with short-time sound descriptors $d_j$.
2. Model these descriptors with probability distributions $p_{\theta_j}$ from a given exponential family.
3. Use the framework of computational information geometry on these distributions.

- In particular, it allows to define the notion of similarity in an information setup through divergences.

**Figure:** Schema of the general architecture of the system.
How to design a system for audio processing?

**Scheme:**
1. Represent the incoming audio stream with short-time sound descriptors $d_j$.
2. Model these descriptors with probability distributions $p_{\theta_j}$ from a given exponential family.
3. Use the framework of computational information geometry on these distributions.

In particular, it allows to define the notion of similarity in an information setup through divergences.

**Potential applications:**
- Segmentation of audio streams.
- Audio content analysis.
- Sound processing and synthesis.
- Automatic structure discovery of audio signals.

**Figure:** Schema of the general architecture of the system.
How can we model sounds?

- Computation of a sound descriptor $d_j$:
  - Fourier or constant-Q transforms for information on the spectral content.
  - Mel-frequency cepstral coefficients for information on the timbre.
  - Many other possibilities.

**Figure**: Sound descriptors modeling.
How can we model sounds?

- Computation of a sound descriptor $d_j$:
  - Fourier or constant-Q transforms for information on the spectral content.
  - Mel-frequency cepstral coefficients for information on the timbre.
  - Many other possibilities.
- Modeling with a probability distribution $p_{\theta_j}$ from an exponential family:
  - Categorical distributions.
  - Multivariate Gaussian distributions.
  - Many other possibilities.

**Figure:** Sound descriptors modeling.
Outline

1. Information geometry
2. Machine learning aspects
3. Framework for audio processing
4. Some applications
   - Speaker segmentation
   - Music segmentation
   - Music similarity analysis
   - Musical structure discovery
   - Query by similarity
   - Audio recombination by concatenative synthesis
Speaker segmentation

Figure: Speaker change detection in a speech fragment.
Music segmentation

**Figure:** Segmentation of a polyphonic musical excerpt.
Music similarity analysis

**Figure:** Similarity analysis of the *1st Piano Sonate, 3rd Movement*, Beethoven.
Musical structure discovery

Figure: Structure discovery of the 1st Piano Sonate, 3rd Movement, Beethoven.
Figure: Query by similarity of the 1st Theme over the entire 1st Piano Sonata, 1st Movement, Beethoven.
Audio recombination by concatenative synthesis

Figure: Audio recombination of African drums by concatenative synthesis of congas.
Information geometry provides emerging tools for audio signal processing.
Toolbox in progress.
http://repmus.ircam.fr/music-information-geometry
Information geometry provides emerging tools for audio signal processing.
Toolbox in progress.
http://repmus.ircam.fr/music-information-geometry
Thank you for your attention! Questions?
This work was supported by a doctoral fellowship from the UPMC (EDITE) and by a grant from the JST-CNRS ICT (Improving the VR Experience).
Bibliography I


Clustering with Bregman divergences.

The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming.

Fast nearest neighbor retrieval for Bregman divergences.

Efficient Bregman range search.

Chentsov, N. N. (1982).


Tailored Bregman ball trees for effective nearest neighbors.

Sufficient statistics and intrinsic accuracy.

Information and accuracy attainable in the estimation of statistical parameters.

The representation levels of music information.