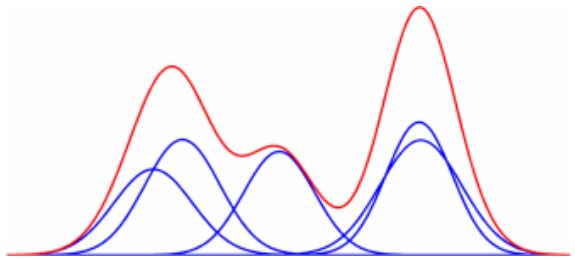
jMEF: A Java™ Library for Mixtures of Exponential Families



- A Java library to create, process and manage mixtures of exponential families (MEF):
  - Estimate a MEF using *Bregman soft clustering* (expectation-maximization using duality EFs$\leftrightarrow$ Bregman divergences).
  - Simplify a MEF using *Bregman hard clustering* (entropic vector quantization).
  - Hierarchical representation of a MEF using *Bregman hierarchical clustering*.
  - Retrieve the *optimal* number of components of a MEF using Bregman hierarchical clustering.
- Open-source:

  `http://www.lix.polytechnique.fr/~nielsen/MEF/`
- Cross platform (Java), with a Matlab® interface.

jMEF - Creating a MEF of 3 components

```
MixtureModel f = new MixtureModel(3);
// Choosen exponential family
f.EF        = new UnivariateGaussian();
// Set weights
f.weight[0] = 1.0/3.0;
f.weight[1] = 1.0/3.0;
f.weight[2] = 1.0/3.0;
// Parameters
PVector p1  = new PVector(2);
PVector p2  = new PVector(2);
PVector p3  = new PVector(2);
// Mu and sigma
p1.array[0] = 10; p1.array[1] = 9;
p2.array[0] = 20; p2.array[1] = 16;
p3.array[0] = 40; p3.array[1] = 25;
// Set the parameters
f.param[0]  = p1;
f.param[1]  = p2;
f.param[2]  = p3;
```

jMEF - Simplifying a MEF

- Let `f` be a MEF of n components.
- To simplify `f` into a MEF of m components ($m < n$), use the Bregman hard clustering:

```
MixtureModel g = BregmanHardClustering.simplify(f,m,type);
```

where `type` is equal to

  - `CLUSTERING_TYPE.RIGHT_SIDED`
  - `CLUSTERING_TYPE.LEFT_SIDED`
  - `CLUSTERING_TYPE.SYMMETRIC`

- For different values of m, we get image segmentation by GMMs:

$m = 1$ $m = 2$ $m = 4$ $m = 8$ $m = 16$ $m = 32$

jMEF - Hierarchical representation of a MEF

- Let `f` be a MEF of n components.
- The hierarchical representation of `f` is obtained using the Bregman hierarchical clustering:

```
HierarchicalMixtureModel h =
BregmanHierarchicalClustering.build(f, side, linkage);
```

where `linkage` is equal to

- `LINKAGE_CRITERION.MINIMUM_DISTANCE`
- `LINKAGE_CRITERION.MAXIMUM_DISTANCE`
- `LINKAGE_CRITERION.AVERAGE_DISTANCE`

jMEF - Hierarchical representation of a MEF

- Using the hierarchical representation `h`, we can:
- Simplify the initial MEF `f` into a MEF g1 of m components:

```
MixtureModel g1 = h.getResolution(m);
```

$m = 1$ $m = 2$ $m = 4$ $m = 8$ $m = 16$ $m = 32$

- Compute the optimal MEF g2 (most compact MEF satisfying a minimum quality t ($D_{KL}(f, g) < t$))

```
MixtureModel g2 = h.getOptimalMixtureModel(t);
```

jMEF - Bibliography

- Hierarchical Gaussian Mixture Model (ICASSP 2010)
- Levels of Details for Gaussian Mixture Models (ACCV 2009)
- Simplifying Gaussian Mixture Models Via Entropic Quantization (EUSIPCO 2009)
- Statistical exponential families: A digest with flash cards `arXiv 0911.4863` (2009)

Tutorials:
`http://www.lix.polytechnique.fr/~nielsen/MEF/`

Acknowledgments. Part of this research has been financially supported by ANR-07-BLAN-0328-01 (GAIA) and DIGITEO 2008-16D (GAS).