k-Maximum Likelihood Estimator for mixtures of generalized Gaussians

ICPR 2012, Tokyo, Japan

Olivier Schwander Aurélien Schutz Yannick Berthoumieu
Frank Nielsen

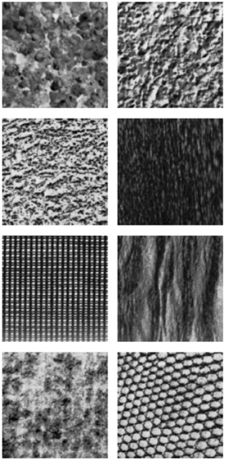
Laboratoire d'informatique, École Polytechnique, France
Laboratoire IMS, Université de Bordeaux, France
Sony Computer Science Laboratories Inc., Tokyo, Japan

November 14, 2012 (updated version)

Outline

Textures

Brodatz



Description

- Wavelet transform

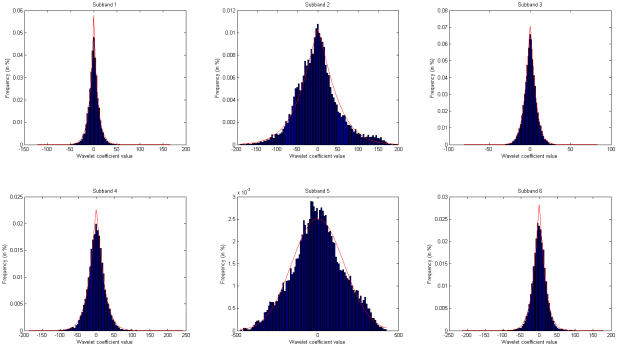
Tasks

- Classification
- Retrieval

Popular models

Modeling wavelet coefficient distribution

- generalized Gaussian distribution (Do 2002, Mallat 1996)
- mixture of generalized Gaussian distributions (Allili 2012)



Generalized Gaussian

Definition

$$f(x;\mu,\alpha,\beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} \exp\left(-\frac{|x-\mu|^\beta}{\alpha}\right)$$

- μ: mean (real number)
- α: scale (positive real number)
- β: shape (positive real number)

Multivariate version: a product of one dimensional laws

Properties and examples

Contains

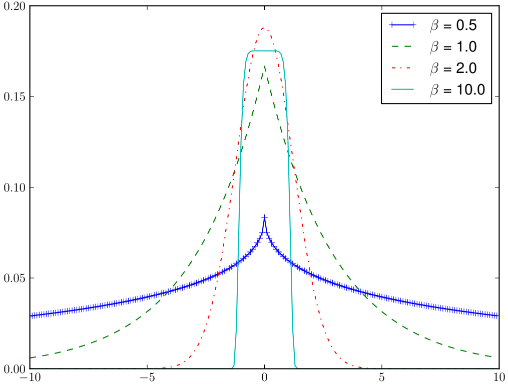
- Gaussian $\beta = 2$
- Laplace $\beta = 1$
- Uniform $\beta \to \infty$

Maximum likelihood estimator

- Iterative procedure (Newton-Raphson)

Exponential family

- For a fixed β



Exponential families

Definition

$$p(x;\lambda) = p_F(x;\theta) = \exp\left(\langle t(x)|\theta\rangle - F(\theta) + k(x)\right)$$

- λ source parameter
- $t(x)$ sufficient statistic
- θ natural parameter
- $F(\theta)$ log-normalizer
- $k(x)$ carrier measure

F is a stricly convex and differentiable function
$\langle\cdot|\cdot\rangle$ is a scalar product

Generalized Gaussian

Fixed μ and β

- $t(x) = -|x-\mu|^\beta$
- $\theta = \alpha^{-\beta}$
- $F(\theta) = -\beta\log(\theta) + \log\left(\frac{\beta}{2\Gamma(1/\beta)}\right)$
- $k(x) = 0$

A large class of distributions

Gaussian or normal (generic, isotropic Gaussian, diagonal Gaussian, rectified Gaussian or Wald distributions, log-normal), Poisson, Bernoulli, binomial, multinomial (trinomial, Hardy-Weinberg distribution), Laplacian, Gamma (including the chi-squared), Beta, exponential, Wishart, Dirichlet, Rayleigh, probability simplex, negative binomial distribution, Weibull, Fisher-von Mises, Pareto distributions, skew logistic, hyperbolic secant, negative binomial, etc.

With a large set of tools

- Bregman Soft Clustering (EM like algorithm)
- Bregman Hard Clustering (k-means like algorithm)
- Kullback-Leibler divergence (through Bregman divergence)

Strong links with the Bregman divergences (Banerjee 2005)

Bregman divergence

Definition and properties

- $B_F(p, q) = F(p) - F(q) + \langle p - q | \nabla F(q) \rangle$
- F is a stricly convex and differentiable function
- Centroids known in closed-form

Legendre duality

- $F^\star(\eta) = \sup_\theta \{\langle \theta, \eta \rangle - F(\theta)\}$
- $\eta = \nabla F(\theta)$, $\theta = \nabla F^\star(\eta)$

Bijection with exponential families

$$\log p_F(x|\theta) = -B_{F^*}(t(x) : \eta) + F^*(t(x)) + k(x)$$

Usual setup: expectation-maximization

Joint probability with missing component labels

- Observations from a finite mixture

$$p(x_1, z_1, \ldots, x_n, z_n) = \prod_i p(z_i|\omega)p(x_i|z_i, \theta)$$

- Marginalization

$$p(x_1, \ldots, x_n|\omega, \theta) = \prod_i \sum_j p(z_i = j|\omega)p(x_i|z_i = j, \theta)$$

EM maximizes

$$\bar{l} = \frac{1}{n} \log p(x_1, \ldots, z_n) = \frac{1}{n} \sum_i \log \sum_j p(z_i = j|\omega)p(x_i|z_i = j, \theta)$$

Complete log-likelihood

Complete average log-likelihood

$$\bar{l}' = \frac{1}{n} \log p(x_1, z_1, \ldots, x_n, z_n) = \frac{1}{n} \sum_i \log \prod_j \left((\omega_j p(x_i, \theta_j))^{\delta(z_i)} \right)$$

$$= \frac{1}{n} \sum_i \sum_j \delta(z_i) \left(\log p(x_i, \theta_j) + \log \omega_j \right)$$

But p is an exponential family

$$\log p(x_i, \theta_j) = \log p_F(x_i, \theta_j) = -B_{F^*}(t(x), \eta_j) + \underbrace{F^\star(t(x)) + k(x)}_{\text{does not depend on } \theta}$$

With fixed weights

Equivalent problem

- Minimizing

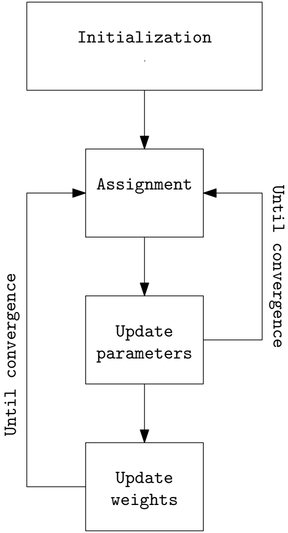
$$
\begin{aligned}
-\bar{l}' &= \frac{1}{n} \sum_i \sum_j \delta(z_i) \left(B_{F^*}(t(x), \eta_j) - \log \omega_j \right) \\
&= \frac{1}{n} \sum_i \min_j \left(B_{F^*}(t(x), \eta_j) - \log \omega_j \right)
\end{aligned}
$$

Bregman k-means with $B_{F^\star} - \log \omega_j$ for divergence

k-Maximum Likelihood estimator

Nielsen 2012

1. **Initialization** (random or k-MLE++)
2. **Assignment**
   $z_i = \arg\min B_{F^\star} - \log \omega_j$
   (gives a partition in cluster C_j)
3. **Update** of the η_j parameters
   $\eta_j = \frac{1}{|C_j|} \sum_{x \in C_i} t(x)$ (Bregman centroid)
4. **Goto** step 2 until local convergence
5. **Update** of the weights $\omega_j = \frac{|C_j|}{n}$
6. **Goto** step 2 until local convergence



Key points

k-MLE

- optimizes the *complete* log-likelihood
- is faster than EM
- converges finitely to a local maximum

Limitations

- All the components **must** belong to the same family
- $F^\star$ may be difficult to compute (without closed form)

What if each component belongs to a different EF ?

Direct applications of k-MLE

or of EM (Bregman Soft Clustering)

A mixture model

- with all components in same the mixture model
- generalized Gaussian sharing the same μ: same mean
- generalized Gaussian sharing the same β: same shape
- one degree of freedom: α (scale)

May be useful

- See mixtures of Laplace distributions ($\beta = 1$)

Not enough for texture description

Complete log-likelihood revisited

Complete average log-likelihood

$$\bar{l}' = \frac{1}{n} \log p(x_1, z_1, \ldots, x_n, z_n) = \frac{1}{n} \sum_i \sum_j \delta(z_i) \left(\log p(x_i, \theta_j) + \log \omega_j\right)$$

Each component is an exponential family

$$\bar{l}' = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \delta_j(z_i) \underbrace{\left(-B_{F_j{}^*}(t(x_i) : \eta_j) + F_j{}^*(t(x_i)) + k_j(x_i) + \log \omega_j\right)}_{-U_j\left(x_i, \eta_j\right)}$$

Optimizing the log-likelihood

Equivalent problem

- Minimizing

$$-\bar{l}' = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \delta_j(z_i) U_j\left(x_i, \eta_j\right)$$

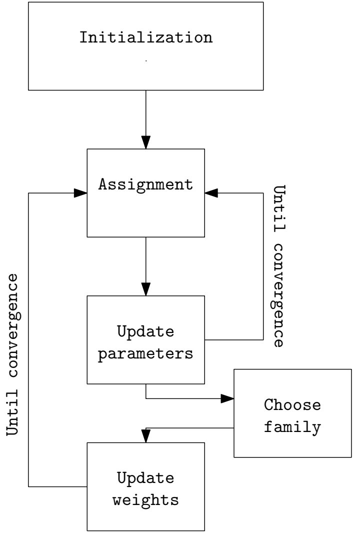
U_j

- Not a distance nor a divergence
- Can even be negative

k-means still works well (**Assignment** step with maximum likelihood)

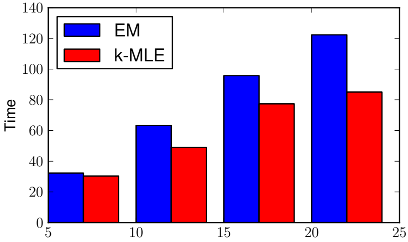
Full algorithm: k-MLE-GG

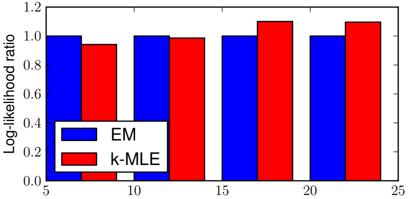
1. **Initialization**
2. **Assignment**
   $z_i = \arg\max_j \log(\omega_j p_{F_j}(x_i|\theta_j))$
3. **Update** of the η_j parameters
4. **Goto** step 2 until local convergence
5. **Choose** the exponential family (μ_j and β_j with MLE)
6. **Update** of the weights ω_j
7. **Goto** step 2 until local convergence



Comparaison with Gaussian EM

On simulated data



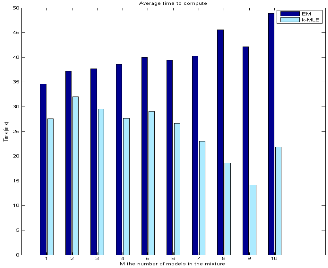


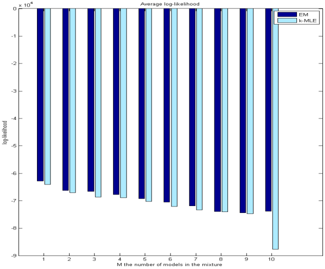
- A mixture of generalized Gaussian is faster to learn than a mixture of simple Gaussians !
- Performs similarly (log-likelihood)

Comparaison with generalized Gaussian EM

Allili 2010

On a texture of the Brodatz dataset





Performs similarly on a classification task

Conclusion

Contributions

- Extension of a powerful algorithm
- More general than k-MLE or EM
- Still faster than a classical EM
- Mixtures with components not belonging to the same exponential family

Perspectives

- Exponential law / Rayleigh $\rightarrow$ Weibull
- Any parametrized exponential family

Bibliography

- F. Nielsen *k-MLE: A fast algorithm for learning statistical mixture models* http://arxiv.org/abs/1203.5181
- M.S. Allili *Wavelet Modelling Using Finite Mixtures of Generalized Gaussian Distributions: Application to Texture Discrimination and Retrieval*. IEEE Trans. on Image Processing, , 2012.