Superresolution via Student-t Mixture Models

Johannes Hertrich | Technische Universität Berlin | 13. 07. 2020

Joint ANR-DFG Project with Gabriele Steidl, Jean-Francois Aujol, Dominique Bernard, Yannick Berthoumieu, Dang Phuong Lan Nguyen, Claudia Redenbach, Abdellatif Saadaldin

Contents

Student-t Mixture Models

Inertial Stochastic PALM (iSRPING)

Superresolution

Numerical Examples

Conclusions

Contents

Student-t Mixture Models

Inertial Stochastic PALM (iSRPING)

Superresolution

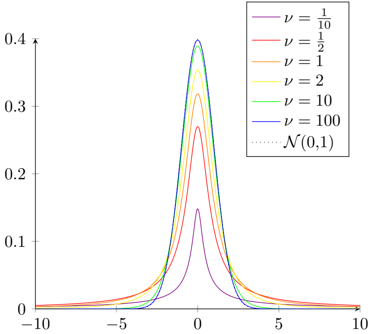
Numerical Examples

Conclusions

Student-t Distribution

For $\nu > 0, \mu \in \mathbb{R}^d, \Sigma \in \mathrm{SPD}(d)$, the multivariate Student-t distribution $T_\nu(\mu, \Sigma)$ is defined by the probability density function

$$f(x|\nu,\mu,\Sigma) = \frac{\Gamma\left(\frac{d+\nu}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)(\pi\nu)^{\frac{d}{2}}|\Sigma|} \frac{1}{\left(1+\frac{1}{\nu}\delta\right)^{\frac{d+\nu}{2}}}, \qquad \delta = (x-\mu)\Sigma^{-1}(x-\mu).$$



Technische Universität Berlin

Student-t mixture model with K components

Given parameters:

$\alpha \in \Delta_K$, $(\nu_k, \mu_k, \Sigma_k) \in \mathbb{R}_{>0} \times \mathbb{R}^d \times \mathrm{SPD}(d)$ for $k = 1, ..., K$

Generate a sample $x \in \mathbb{R}^d$ by the following procedure:

- Choose $k \in \{1, ..., K\}$ with $P(k) = \alpha_k$.
- Choose $x \sim T_{\nu_k}(\mu_k, \Sigma_k)$.

$\to$ Density function:

$$p(x) = \sum_{k=1}^K \alpha_k f(x|\nu_k, \mu_k, \Sigma_k), \quad f(x|\nu_k, \mu_k, \Sigma_k) = \frac{\Gamma\left(\frac{d+\nu_k}{2}\right)}{\Gamma\left(\frac{\nu_k}{2}\right)(\pi\nu_k)^{\frac{d}{2}}|\Sigma_k|} \frac{1}{\left(1+\frac{1}{\nu_k}\delta_k\right)^{\frac{d+\nu_k}{2}}},$$

where

$$\delta_k = (x - \mu_k)\Sigma_k^{-1}(x - \mu_k).$$

Parameter Estimation

- **Given:** samples $x_1, \ldots, x_n \in \mathbb{R}^d$.
- **Task:** estimate the parameters $(\alpha, \nu, \mu, \Sigma) \in \Delta_K \times \mathbb{R}^K_{>0} \times \mathbb{R}^{d \times K} \times \mathrm{SPD}(d)^K$ of a Student-t mixture model.

$\rightarrow$ Compute Maximum Likelihood estimator, i.e. minimize the negative log likelihood function

$$L(\alpha, \nu, \mu, \Sigma | x_1, \ldots, x_n) = -\sum_{i=1}^{n} \log\Big(\sum_{k=1}^{K} \alpha_k f(x_i | \nu_k, \mu_k, \Sigma_k)\Big).$$

Possible Algorithms:

- Expectation-Maximization (EM) Algorithm.
- Variants or Accelerations of the EM Algorithm[1].

$\rightarrow$ Drawback: Very slow for large data sets.

Solution: use **Stochastic PALM-based Algorithms**.

[1] Ref.: Hasannasab, H., Laus, Steidl, 2019

Contents

Student-t Mixture Models

Inertial Stochastic PALM (iSRPING)

Superresolution

Numerical Examples

Conclusions

Proximal Alternating Linearized Minimization (PALM)

Task: Minimize a function of the form

$$F(x_1, x_2) = H(x_1, x_2) + f(x_1) + g(x_2),$$

where

- $H\colon \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}$ is differentiable where the partial gradients $\nabla H(x_1, \cdot)$ and $\nabla H(\cdot, x_2)$ are globally Lipschitz continuous for all x_1, x_2.
- We assume that $H(x_1, x_2) = \sum_{i=1}^n h_i(x_1, x_2)$.
- $f\colon \mathbb{R}^{d_1} \to \mathbb{R} \cup \{\infty\}$ and $g\colon \mathbb{R}^{d_2} \to \mathbb{R} \cup \{\infty\}$ are proper and lower semicontinuous.

Now, the Proximal Alternating Linearized Minimization (PALM)[2] reads as

$$\begin{aligned} x_1^{k+1} &\in \operatorname{prox}^f_{\tau_1^k}\left(x_1^k - \tfrac{1}{\tau_1^k}\nabla_{x_1} H(x_1^k, x_2^k)\right) \\ x_2^{k+1} &\in \operatorname{prox}^g_{\tau_2^k}\left(x_2^k - \tfrac{1}{\tau_2^k}\nabla_{x_2} H(x_1^{k+1}, x_2^k)\right). \end{aligned}$$

[2] Ref.: Bolte, Sabach, Teboulle, 2014

Variants of PALM

Inertial PALM (Pock, Sabach, 2016):

$$y_1^k = x_1^k + \alpha_1^k(x_1^k - x_1^{k-1})$$
$$z_1^k = x_1^k + \beta_1^k(x_1^k - x_1^{k-1})$$
$$x_1^{k+1} \in \operatorname{prox}^f_{\tau_1^k}\left(y_1^k - \tfrac{1}{\tau_1^k}\nabla_{x_1} H(z_1^k, x_2^k)\right)$$

$$y_2^k = x_2^k + \alpha_2^k(x_2^k - x_2^{k-1})$$
$$z_2^k = x_2^k + \beta_2^k(x_2^k - x_2^{k-1})$$
$$x_2^{k+1} \in \operatorname{prox}^g_{\tau_2^k}\left(y_2^k - \tfrac{1}{\tau_2^k}\nabla_{x_2} H(x_1^{k+1}, z_2^k)\right)$$

Stochastic PALM (Xu, Yin, 2015) / SPRING (Driggs et. al. 2020):
$\rightarrow$ Replace the gradient by some stochastic gradient estimator $\tilde{\nabla}$.

$$x_1^{k+1} \in \operatorname{prox}^f_{\tau_1^k}\left(x_1^k - \tfrac{1}{\tau_1^k}\tilde{\nabla}_{x_1} H(x_1^k, x_2^k)\right)$$
$$x_2^{k+1} \in \operatorname{prox}^g_{\tau_2^k}\left(x_2^k - \tfrac{1}{\tau_2^k}\tilde{\nabla}_{x_2} H(x_1^{k+1}, x_2^k)\right).$$

Stochastic Gradient Estimators

- To show convergence of SPRING we need a so-called **variance reduced estimator** $\tilde{\nabla}$.
- This basically means that bias and variance of $\tilde{\nabla}$ applied on a sequence x^k become small, if the distance of x^k and x^{k-1} becomes small.
- Driggs et. al. have shown, that the SARAH estimator[3] has this property.

$$\tilde{\nabla}_{x_1} H(x_1^k, x_2^k) = \begin{cases} \nabla_{x_1} H(x_1^k, x_2^k), & \text{w.p. } \frac{1}{p}, \\ \frac{n}{b} \sum\limits_{i \in B_i^k} \nabla_{x_1} h_i(x_1^k, x_2^k) - \nabla_{x_1} h_i(x_1^{k-1}, x_2^{k-1}) + \tilde{\nabla}_{x_1} H(x_1^{k-1}, x_2^{k-1}), & \text{w.p. } 1 - \frac{1}{p}, \end{cases}$$

- Also other estimators as SAGA[4] fulfill this property.

[3] Ref.: Nguyen, Liu, Scheinberg, Takac, 2017
[4] Ref.: Defazio, Bach, Lacoste-Julien, 2014

Inertial Stochastic PALM (iSPRING)

Combine inertial and stochastic PALM to iSPRING[5]:

$$
\begin{aligned}
y_1^k &= x_1^k + \alpha_1^k(x_1^k - x_1^{k-1})\\
z_1^k &= x_1^k + \beta_1^k(x_1^k - x_1^{k-1})\\
x_1^{k+1} &\in \operatorname{prox}^f_{\tau_1^k}\Big(y_1^k - \tfrac{1}{\tau_1^k}\tilde\nabla_{x_1} H(z_1^k, x_2^k)\Big)\\
y_2^k &= x_2^k + \alpha_2^k(x_2^k - x_2^{k-1})\\
z_2^k &= x_2^k + \beta_2^k(x_2^k - x_2^{k-1})\\
x_2^{k+1} &\in \operatorname{prox}^g_{\tau_2^k}\Big(y_2^k - \tfrac{1}{\tau_2^k}\tilde\nabla_{x_2} H(x_1^{k+1}, z_2^k)\Big),
\end{aligned}
$$

– Similar assumptions on $\tilde\nabla$ as for SPRING.
– One can show, that the SARAH estimator fulfills these assumptions.
– We provide an implementation framework for iSPRING at `https://github.com/johertrich/iSPRING`.

[5] Ref.: H., Steidl, 2020

Convergence of iSPRING

Technische Universität Berlin

Theorem (H., Steidl, 2020)

Under certain assumptions, we have that

$$\mathbb{E}\left(\mathrm{dist}(0, \partial F(x_1^{k+1}, x_2^{k+1}))^2\right) \to 0 \quad \text{as } k \to \infty.$$

Further, for t drawn uniformly from $\{2, \ldots, T+1\}$, we get that

$$\mathbb{E}\left(\mathrm{dist}(0, \partial F(x_1^{t}, x_2^{t}))^2\right) \in \mathcal{O}(\tfrac{1}{T}).$$

If F additionally fulfills $F(x_1, x_2) - \underline{F} \leq \mu \mathrm{dist}(0, \partial F(x_1, x_2))^2$, where $\underline{F} = \inf_{x_1, x_2} F(x_1, x_2)$, then we have

$$\mathbb{E}\left(F(x_1^{T+1}, x_2^{T+1}) - \underline{F}\right) \in \mathcal{O}(\Theta^T)$$

for some $\Theta \in (0, 1)$.

iSPRING for Student-t Mixture Models

First idea: Rewrite the optimization problem as

$$F(\alpha,\nu,\mu,\Sigma) = H(\alpha,\nu,\mu,\Sigma) + f_1(\alpha) + f_2(\nu) + f_3(\mu) + f_4(\Sigma),$$

where $H := L$, $f_1 := \iota_{\Delta_K}$, $f_2 := \iota_{\mathbb{R}^K_{>0}}$, $f_3 := 0$, $f_4 := \iota_{\mathrm{SPD}(d)^K}$. BUT:

- f_2 and f_4 are not lower semi-continuous
- L is not defined on the whole Euclidean space.

$\rightarrow$ The convergence results are not applicable.

iSPRING for Student-t Mixture Models

Thus, we modify the model:

- Let $\mathrm{SPD}_\epsilon(d) := \{\Sigma \in \mathrm{SPD}(d) : \Sigma \succeq \epsilon I_d\}$.
- Consider the surjective mappings $\varphi_1 \colon \mathbb{R}^K \to \Delta_K$, $\varphi_2 \colon \mathbb{R}^K \to \mathbb{R}^K_{\geq \epsilon}$ and $\varphi_3 \colon \mathrm{Sym}(d)^K \to \mathrm{SPD}_\varepsilon(d)^K$ defined by

$$\varphi_1(\alpha) := \frac{\exp(\alpha)}{\sum_{j=1}^K \exp(\alpha_j)}, \quad \varphi_2(\nu) := \nu^2 + \epsilon, \quad \varphi_3(\Sigma) := \left(\Sigma_k^T \Sigma_k + \epsilon I_d\right)_{k=1}^K.$$

Now we solve

$$\operatorname*{arg\,min}_{\alpha\in\mathbb{R}^K, \nu\in\mathbb{R}^K, \mu\in\mathbb{R}^{d\times K}, \Sigma\in\mathrm{Sym}(d)^K} H(\alpha,\nu,\mu,\Sigma) := L(\varphi_1(\alpha), \varphi_2(\nu), \mu, \varphi_3(\Sigma)|\mathcal{X}).$$

Note that the functions f_i, $i = 1, \ldots, 4$ are just zero.

Contents

Student-t Mixture Models

Inertial Stochastic PALM (iSRPING)

Superresolution

Numerical Examples

Conclusions

Superresolution via Student-t Mixture Models

- We use two methods, which were originally proposed for Gaussian mixture models:
  - Expected Patch Log-Likelihood (EPLL, Zoran, Weiss 2011),
  - Joint Mixture Models (Sandeep, Jacob, 2016).
- We adapt this methods for Student-t mixture models.

Expected Patch Log-Likelihood (EPLL) for Student-t Mixture Models

We adapt the EPLL[6] for Student-t mixture models. Define:

- $x \in \mathbb{R}^N$ is the high resolution image.
- $y \in \mathbb{R}^M$ is the low resolution image.
- $A \in \mathbb{R}^{M \times N}$ is the superresolution operator.
- p is the density function of a $d = \tau^2$-dimensional Student-t mixture model.

Model: $y = Ax + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

Approximate

$$\operatorname*{arg\,min}_{x \in \mathbb{R}^N} \frac{d}{\sigma^2} \|Ax - y\|_2^2 - \sum_{i \in I} p(P_i(x)),$$

where P_i extracts the i-th patch.

$\rightarrow$ Maximum a posteriori (MAP) estimator.

[6] Ref.: Zoran, Weiss 2011

Technische Universität Berlin

Algorithm

Standard approximations lead to the following algorithm.

Algorithm 1 EPLL for Student-t mixture models

Input: Initialization $\hat{x} \in \mathbb{R}^N$, low resolution image $y \in \mathbb{R}^M$, superresolution operator $A \in \mathbb{R}^{M \times N}$, Student-$t$ mixture model $(\alpha, \nu, \mu, \Sigma)$.

Output: Reconstruction $\hat{x} \in \mathbb{R}^N$.

for $i \in I$ **do**

1. $\tilde{z}_i = P_i \hat{x}$.
2. $k_i^* = \arg\min_{1 \le k \le K} -\log(\alpha_k) - \log(f(\tilde{z}_i | \nu_k, \mu_k, \Sigma_k))$.
3. $\hat{z}_i = \arg\min_{z_i \in \mathbb{R}^d} \frac{\beta}{2} \|\tilde{z}_i - z_i\|_2^2 - \log(f(z_i | \nu_k, \mu_k, \Sigma_k))$.

end for

4. $\hat{x} = \left(A^T A + \frac{\beta \sigma^2}{d} \sum_{i \in I} P_i^T P_i\right)^{-1} \left(A^T y + \frac{\beta \sigma^2}{d} \sum_{i \in I} P_i^T \hat{z}_i\right)$.

Contents

Student-t Mixture Models

Inertial Stochastic PALM (iSRPING)

Superresolution

Numerical Examples

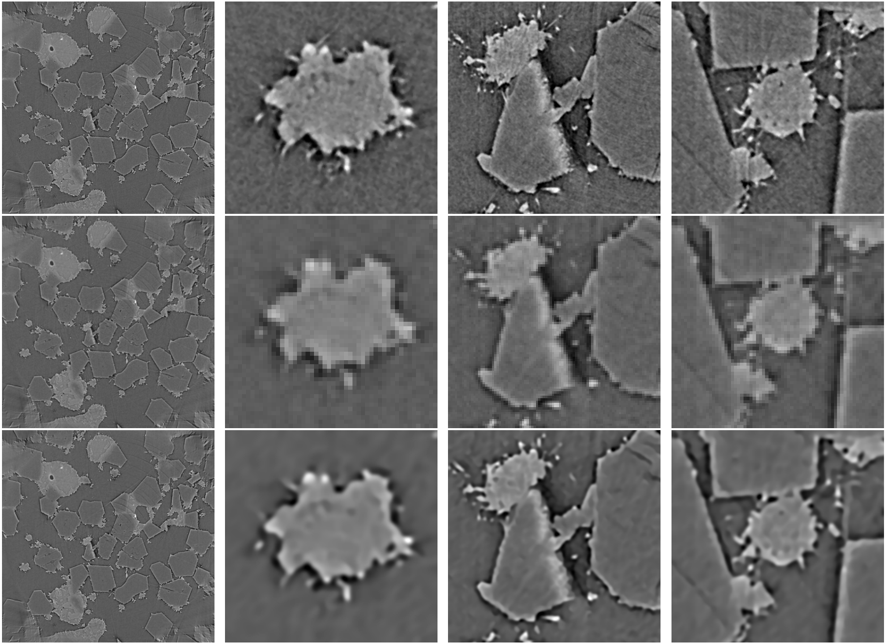
Conclusions

Technische Universität Berlin

Comparison to Gaussian mixture models

Reconstruct artificially downsampled test images with magnification $q = 2$.

Image	Algorithm	$K = 100$ PSNR	$K = 100$ SSIM	$K = 200$ PSNR	$K = 200$ SSIM	$K = 300$ PSNR	$K = 300$ SSIM
Cameraman	GMM-EPLL	**32.46**	**0.774**	**32.44**	0.770	**32.43**	0.769
	Student-t EPLL	32.19	0.770	32.21	0.771	32.17	0.769
	GMM-MMSE	32.00	0.771	32.08	**0.782**	32.16	**0.787**
	Student-t MMSE	32.08	**0.774**	32.07	0.777	31.84	0.770
	L_2-TV, $\lambda = 0.25$	31.25	0.767	–	–	–	–
Barbara	GMM-EPLL	25.15	**0.777**	25.15	0.778	25.14	0.777
	Student-t EPLL	**25.21**	0.773	**25.32**	**0.793**	**25.30**	**0.791**
	GMM-MMSE	25.01	0.776	25.01	0.778	25.01	0.778
	Student-t MMSE	24.86	0.760	24.92	0.768	24.89	0.769
	L_2-TV, $\lambda = 0.07$	25.17	0.754	–	–	–	–
Hill	GMM-EPLL	31.15	0.827	31.19	0.828	31.21	0.828
	Student-t EPLL	**31.48**	**0.844**	**31.39**	**0.843**	**31.52**	**0.845**
	GMM-MMSE	30.90	0.835	30.99	0.838	31.14	0.844
	Student-t MMSE	30.85	0.831	31.00	0.835	31.07	0.838
	L_2-TV, $\lambda = 0.35$	31.02	0.834	–	–	–	–



Top: original, middle: low resolution (artificially downsampled, $q = 4$), bottom: EPLL reconstruction with Student-t mixture models

Contents

Student-t Mixture Models

Inertial Stochastic PALM (iSRPING)

Superresolution

Numerical Examples

Conclusions

Technische Universität Berlin

Conclusions and Future Work

Conclusions:

- We proposed a new algorithm iSPRING based on PALM, which can be applied very efficient for various problems. In particular:
  - Estimating the parameters of Student-t mixture models,
  - Neural networks with constraints.
- We provide an implementation framework for iSPRING, which is very efficient and easy to use. `https://github.com/johertrich/iSPRING`.
- We adapted two superresolution methods for Student-t mixture models.
- On small test images, Student-t mixture models are slightly better than Gaussian mixture models.

Future work:

- Comparison of the algorithms for estimating the Student-t mixture models.
- The numerical part is still very experimental.

Joint Mixture Models[7]

Estimation of the mixture model:

- Let q be a magnification factor and let $(h_i, l_i) \in \mathbb{R}^{dq^2} \times \mathbb{R}^d$ for $i \in I$ be a given pairs of high resolution and low resolution patches.
- Estimate the parameters of a joint mixture model using the data points $(h_i^T, l_i^T)^T \in \mathbb{R}^{dq^2+d}$.

$\rightarrow$ resulting parameters: $(\alpha, \nu, \mu, \Sigma)$, where

$$\mu_k = \begin{pmatrix} \mu_{H_k} \\ \mu L_k \end{pmatrix}, \qquad \Sigma_k = \begin{pmatrix} \Sigma_{H_k} & \Sigma_{HL_k} \\ \Sigma_{HL_k}^T & \Sigma_{L_k} \end{pmatrix}.$$

[7] Ref.: Sandeep, Jacob 2016

Patch Estimation

Given a LR patch $y \in \mathbb{R}^d$ estimate a HR patch $x \in \mathbb{R}^{dq^2}$.

- Assume that $\begin{pmatrix} x \\ y \end{pmatrix}$ is a realization of a random variable

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim T_{\nu_k}(\mu_k, \Sigma_k).$$

- Select $k^* \in \{1, ..., K\}$ such that the likelihood that (x, y) belongs to component k^* is maximal, i.e.

$$k^* = \operatorname*{arg\,max}_{k=1,\dots,K} \alpha_k f(y|\nu_k, \mu_{L_k}, \Sigma_{L_k})$$

- Estimate HR patch x by

$$\hat{x} = \mathrm{MMSE}(X|Y = y) = \mathbb{E}(X|Y = y) = \mu_{H_{k^*}} + \Sigma_{HL_{k^*}} \Sigma_{L_{k^*}}^{-1} (y - \mu_{L_{k^*}}).$$

Technische Universität Berlin

We summarize the algorithm:

Algorithm 2 Joint Student-t mixture models

Input: Low resolution image $y \in \mathbb{R}^M$, Student-t mixture model $(\alpha, \nu, \mu, \Sigma)$.
Output: Reconstruction $\hat{x} \in \mathbb{R}^N$.
for $i \in I$ **do**

1. $y_i = P_i y$.
2. $k_i^* = \arg\min_{1 \le k \le K} -\log(\alpha_k) - \log(f(y_i | \nu_k, \mu_{L_k}, \Sigma_{L_k}))$.
3. $\hat{z}_i = \mu_{H_{k*}} + \Sigma_{HL_{k*}} \Sigma_{L_{k*}}^{-1} (y - \mu_{L_{k*}})$.

end for

4. Average the patches $\{\hat{z}_i\}_{i \in I}$ by $\hat{x}_{kl} = \frac{\sum_{i \in I} w_{ikl} \hat{z}_{ikl}}{\sum_{i \in I} w_{ikl}}$.