Detecting Mixed Density and Dimensionality: Applications to Brain Imaging

Guillermo Sapiro

University of Minnesota

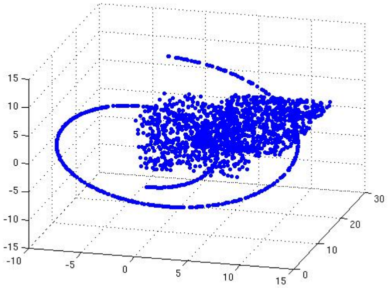
Gloria Haro, Gregory Randall, GS, NIPS 2006, IJCV 2008
Haro, Christophe Lenglet, Paul Thompson, GS, ISBI '08
Lenglet, Haro, Daniel Franc, Thompson, Kelvin Lim, GS, HBM '08

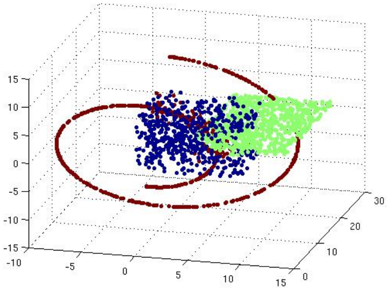
Detecting mixed dimensionality and density

Motivation

Goal

Detect and estimate different dimensions and densities in the same noisy point cloud data and cluster the points according to these characteristics.





Outline

1. Motivation
2. Local dimension estimation
3. Translated Poisson Model
4. Translated Poisson Mixture Model
5. Regularized TPMM
6. Experiments
7. Conclusions and future work

Local dimension estimation

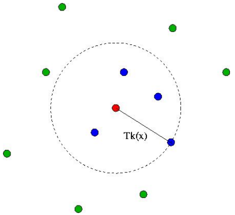
Levina and Bickel's approach

Basic idea: proportion of points falling into a ball.

$$\frac{k}{n} \approx f(x)V(m)R_k(x)^m$$

where:

- k: number of points inside ball.
- n: total number of points.
- $f(x)$: local density at point x.
- $V(m)$: volume of the unit sphere in $\mathbb{R}^m$.
- $R_k(x)$: Euclidean distance from x to its k-th nearest neighbor.



Local dimension estimation

Levina and Bickel's approach

Observable event: Number of points falling into a small sphere $B(R, x)$ (radius R, centered at x).

$$N(R, x) = \sum_{i=1}^{N} \mathbf{1}\{x_i \in B(R, x)\}$$

Making the **approximations**:

- Binomial process by a **Poisson process** ($n \to \infty$, k moderate, and $k/n \to 0$).
- $f(x) \approx const.$ in a small sphere.

then, the rate λ of the counting process N

$$\lambda(r, x) = f(x)V(m)mr^{m-1}$$

Local dimension estimation

Levina and Bickel's approach

Log-likelihood of the observed process $N(R, x)$

$$L(m(x), \theta(x)) = \int_0^R \log \lambda(r, x) dN(r, x) - \int_0^R \lambda(r, x) dr$$

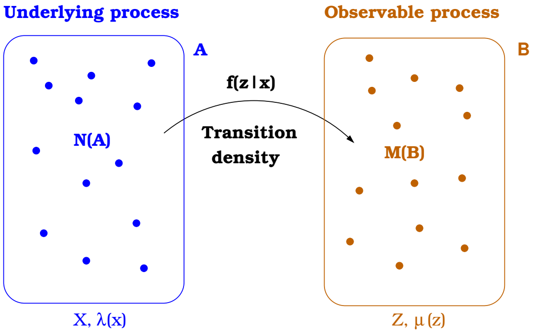
ML estimators satisfy $\partial L / \partial \theta = 0$ and $\partial L / \partial m = 0$ ($\theta = \log f(x)$).
Fixing the number of neighbors (kNN-graph) we obtain

$$\hat{m}(x) = \left[\frac{1}{k-1} \sum_{j=1}^{k-1} \log \frac{R_k(x)}{R_j(x)} \right]^{-1}$$

Same as Takens' estimator in dynamical systems.

Translated Poisson Model

Modeling a counting process under noise (Snyder & Miller)



- N(A) Poisson process with integrable intensity function $\lambda(x)$.
- Points translated **independently**.
- No insertions and deletions.

$\longrightarrow$ **M(B) Poisson process**

$$\mu(z) = \int_X f(z|x)\lambda(x)dx$$

Translated Poisson Model

In our case, $\lambda(r, x)$ is parametrized by the Euclidean distances r of the points. We consider a random translation $f(s|r)$ in the distances r. The intensity of the Poisson process in the output space, is given by

$$\mu(s, x_t) = \int_0^{R'} f(s|r)\lambda(r, x_t)dr.$$

Translated Poisson Model

In our case, $\lambda(r,x)$ is parametrized by the Euclidean distances r of the points. We consider a random translation $f(s|r)$ in the distances r. The intensity of the Poisson process in the output space, is given by

$$\mu(s,x_t) = \int_0^{R'} f(s|r)\lambda(r,x_t)dr.$$

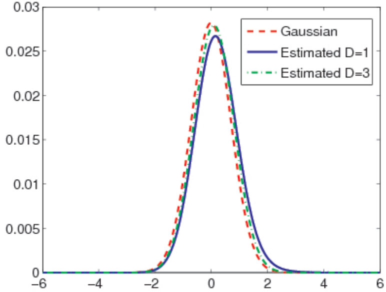
Particular case:

i.i.d. Gaussian noise in the points coord.

$$\hat{D}_{ij} \approx D_{ij} + W$$

where $W \sim N(0,2\sigma^2) * \hat{\chi}_p^2 * \check{\chi}_1^2$.

For sufficient SNR: $W \sim N(0,2\sigma^2)$.



Translated Poisson Model

Maximize the log-likelihood of the new Translated Poisson process.

$$m(x_t) = \left[\frac{1}{k-1} \sum_{i=1}^{k-1} \frac{\int_0^{R'} f(R_i(x_t)|r) r^{m-1} \log \frac{R_k(x_t)}{r} dr}{\int_0^{R'} f(R_i(x_t)|r) r^{m-1} dr} \right]^{-1}$$

Translated Poisson Model

Maximize the log-likelihood of the new Translated Poisson process.

$$m(x_t) = \left[\frac{1}{k-1} \sum_{i=1}^{k-1} \frac{\int_0^{R'} f(R_i(x_t)|r) r^{m-1} \log \frac{R_k(x_t)}{r} dr}{\int_0^{R'} f(R_i(x_t)|r) r^{m-1} dr} \right]^{-1}$$

We substitute r^{m-1} by its Taylor expansion around R_i (f has a small support concentrated around R_i).

Translated Poisson Model

Maximize the log-likelihood of the new Translated Poisson process.

$$m(x_t) = \left[\frac{1}{k-1} \sum_{i=1}^{k-1} \frac{\int_0^{R'} f(R_i(x_t)|r) r^{m-1} \log \frac{R_k(x_t)}{r} dr}{\int_0^{R'} f(R_i(x_t)|r) r^{m-1} dr} \right]^{-1}$$

We substitute r^{m-1} by its Taylor expansion around R_i
(f has a small support concentrated around R_i).

Local dimension estimator (noisy case):

$$m(x_t) \approx \left[\frac{1}{k-1} \sum_{i=1}^{k-1} \frac{\int_0^{R'} f(R_i|r) \log \frac{R_k}{r} dr}{\int_0^{R'} f(R_i|r) dr} \right]^{-1}$$

Translated Poisson Model

Maximize the log-likelihood of the new Translated Poisson process.

$$m(x_t) = \left[\frac{1}{k-1} \sum_{i=1}^{k-1} \frac{\int_0^{R'} f(R_i(x_t)|r) r^{m-1} \log \frac{R_k(x_t)}{r} dr}{\int_0^{R'} f(R_i(x_t)|r) r^{m-1} dr} \right]^{-1}$$

We substitute r^{m-1} by its Taylor expansion around R_i (f has a small support concentrated around R_i).

Local dimension estimator (noisy case):

$$m(x_t) \approx \left[\frac{1}{k-1} \sum_{i=1}^{k-1} \frac{\int_0^{R'} f(R_i|r) \log \frac{R_k}{r} dr}{\int_0^{R'} f(R_i|r) dr} \right]^{-1}$$

If $f(s|r) = \delta(s-r)$ (no noise) $\rightarrow$ Levina and Bickel estimator.

Translated Poisson Mixture Model (TPMM)

Detecting mixed dimensionality and density

Consider J mixture components (Translated Poisson distributions): vector of parameters $\psi = \{\pi^j, m^j, \theta^j; j = 1, \ldots, J\}$ where

- π^j is the mixture coefficient for class j,
- θ^j is the density parameter ($f^j = e^{\theta^j}$)
- m^j is the dimension.

Observable event: $y = N(R, x)$, # points inside ball $B(R, x)$.

Density function:

$$p(y_t|\psi) = \sum_{j=1}^{J} \pi^j p(y_t|\theta^j, m^j)$$

Translated Poisson Mixture Model (TPMM)

Observation sequence: $Y = \{y_t; t = 1, \ldots, T\}$, where $y_t = N(R, x_t)$.

Translated Poisson Mixture Model (TPMM)

Observation sequence: $Y = \{y_t; t = 1, \ldots, T\}$, where $y_t = N(R, x_t)$.

The complete-data density: $p(Z, Y|\psi) = \prod_{t=1}^{T} p(z_t, y_t|\psi)$.

Hidden-state information: $Z = \{z_t \in C; t = 1 \ldots T\}$, where $z_t = C^j$ means that the j-th mixture generates y_t.

Translated Poisson Mixture Model (TPMM)

Observation sequence: $Y = \{y_t; t = 1, \ldots, T\}$, where $y_t = N(R, x_t)$.

The complete-data density: $p(Z, Y|\psi) = \prod_{t=1}^{T} p(z_t, y_t|\psi)$.

Hidden-state information: $Z = \{z_t \in C; t = 1 \ldots T\}$, where $z_t = C^j$ means that the j-th mixture generates y_t.

$\longrightarrow$ Solved by the **EM algorithm**:

- **E-step:** Computation of the expectation of the membership functions, $h^j(y_t)$.
- **M-step:** Computation of the parameters π^j, m^j, θ^j of the J experts by maximizing the expectation of the log-likelihood w.r.t Z.

Regularized TPMM (R-TPMM)

Another interpretation of EM

EM is based on the following decomposition of the log-likelihood:

$$L(Y|\psi, H) = \sum_{t=1}^{T} \sum_{j=1}^{J} h^j(y_t) \log \left[p(y_t|\psi^j)\pi^j \right] - \sum_{t=1}^{T} \sum_{j=1}^{J} h^j(y_t) \log \left[h^j(y_t) \right],$$

where $H = \{h^j(y_t) \leq 1; t = 1, \ldots, T, j = 1, \ldots, J\}$.

First term: Expectation of $\sum_{t=1}^{T} \sum_{j=1}^{J} \delta_t^j \log \left[p(y_t|\psi^j)\pi^j \right]$ w.r.t. Z.

Second term: Entropy of the membership functions.

Regularized TPMM (R-TPMM)

Another interpretation of EM

EM can be seen as an **alternate optimization algorithm** of the previous log-likelihood.

E-step:

Maximization of $L(Y|\psi, H)$ w.r.t. H

with the additional constraint that $\sum_{j=1}^{J} h^j(y_t) = 1,\ t = 1, \ldots, T$.

M-step:

Maximization of $L(Y|\psi, H)$ w.r.t. ψ

with an additional constraint for the mixture probabilities: $\sum_{j=1}^{J} \pi^j = 1$.

Regularized TPMM (R-TPMM)

Extended functional

Inspired by the neighborhood EM (NEM) [Ambroise,Govaert].

$$F(\psi, H) = L(Y|\psi, H) + \alpha S(H)$$

where

- $S(H)$ is a regularization term.
- α is a regularization parameter.

Regularized TPMM (R-TPMM)

Regularization term

$$S(H) = -\sum_{t=1}^{T}\sum_{j=1}^{J} h^j(y_t)\mathcal{D}(t, j, X, H)$$

where $\mathcal{D}$ is a dissimilarity function.

Provides a **generic framework for introducing constraints** in the soft classification, besides the ones already present in the PMM model, dimensionality and density.

Regularized TPMM (R-TPMM)

Regularization term

$$S(H) = -\sum_{t=1}^{T}\sum_{j=1}^{J} h^j(y_t)\mathcal{D}(t, j, X, H)$$

where $\mathcal{D}$ is a dissimilarity function.

Provides a **generic framework for introducing constraints** in the soft classification, besides the ones already present in the PMM model, dimensionality and density.

Spatial/Temporal regularity

$$\mathcal{D}_R := \sum_{s\sim t}(1 - h^j(y_s))$$

Different neighborhoods $s \sim t$ result in different kinds of regularization.

Regularized TPMM (R-TPMM)

Algorithm R-TPMM

REQUIRE: The point cloud data, J, k, σ and α.

Regularized TPMM (R-TPMM)

Algorithm R-TPMM

REQUIRE: The point cloud data, J, k, σ and α.

1. **Initialization** of $\psi_0^j = \{\pi_0^j, m_0^j, \theta_0^j\}$ for all $j = 1, \ldots, J$.

Regularized TPMM (R-TPMM)

Algorithm R-TPMM

REQUIRE: The point cloud data, J, k, σ and α.

1. **Initialization** of $\psi_0^j = \{\pi_0^j, m_0^j, \theta_0^j\}$ for all $j = 1, \ldots, J$.
2. **Iterations** (until convergence of ψ_n^j):
   For each class $j = 1, \ldots, J$,

Regularized TPMM (R-TPMM)

Algorithm R-TPMM

REQUIRE: The point cloud data, J, k, σ and α.

1. **Initialization** of $\psi_0^j = \{\pi_0^j, m_0^j, \theta_0^j\}$ for all $j = 1, \ldots, J$.
2. **Iterations** (until convergence of ψ_n^j):
   For each class $j = 1, \ldots, J$,
   - **1st-step**: compute $h_{n+1}^j(y_t)$

$$h_{n+1}^j(y_t) = \frac{p(y_t|m_n^j, \theta_n^j)\pi_n^j e^{-\alpha \mathcal{D}'(t,j,X,H_n)}}{\sum_{l=1}^J p(y_t|m_n^l, \theta_n^l)\pi_n^l e^{-\alpha \mathcal{D}'(t,l,X,H_n)}},$$

Regularized TPMM (R-TPMM)

Algorithm R-TPMM

REQUIRE: The point cloud data, J, k, σ and α.

1. **Initialization** of $\psi_0^j = \{\pi_0^j, m_0^j, \theta_0^j\}$ for all $j = 1, \ldots, J$.
2. **Iterations** (until convergence of ψ_n^j):
   For each class $j = 1, \ldots, J$,
   - **1st-step**: compute $h_{n+1}^j(y_t)$

     $$h_{n+1}^j(y_t) = \frac{p(y_t|m_n^j, \theta_n^j)\pi_n^j e^{-\alpha\mathcal{D}'(t,j,X,H_n)}}{\sum_{l=1}^J p(y_t|m_n^l, \theta_n^l)\pi_n^l e^{-\alpha\mathcal{D}'(t,l,X,H_n)}},$$

   - **2nd-step**: compute $\psi_{n+1}^j = \{\pi_{n+1}^j, m_{n+1}^j, \theta_{n+1}^j\}$

     $$\psi_{n+1}^j = \arg\max_{\psi} F(\psi, H_{n+1}) + \lambda(\sum_{r=1}^{J} \pi^r - 1)$$

Regularized TPMM (R-TPMM)

Computation of parameters at step $n+1$:

$$
\begin{aligned}
\pi^j_{n+1} &= \frac{1}{T}\sum_{t=1}^{T} h^j_{n+1}(y_t) \\
m^j_{n+1} &= \left[\sum_t h^j_{n+1}(y_t)\hat{m}(x_t)^{-1} / \sum_t h^j_{n+1}(y_t)\right]^{-1} \\
f^j_{n+1} = e^{\theta^j_{n+1}} &= \left[\sum_t h^j_{n+1}(y_t)\hat{f}(x_t)^{-1} / \sum_t h^j_{n+1}(y_t)\right]^{-1}
\end{aligned}
$$

where $\hat{m}(x_t)$ and $\hat{f}(x_t)$ are the local estimators (Translated Poisson).
If $\sigma = 0$, PMM, they are the Levina and Bickel's estimators.

$\longrightarrow$ **Weighted harmonic means**

Regularized TPMM (R-TPMM)

Asymptotic behavior

Levina and Bickel's technique

$$E[\hat{m}(x)] = m_T, \qquad \mathrm{Var}[\hat{m}(x)] = \frac{m_T^2}{k-3}$$

(dividing by $k-2$ instead of $k-1$)

R-TPMM approach (hard clustering version)

$$E[\hat{m}^j] = m_T^j + \frac{m_T^j}{(k-1)N^j - 1}, \qquad \mathrm{Var}[\hat{m}^j] = (m_T^j)^2 O\left(\frac{1}{N^j(k-1)-4}\right)$$

where N^j is the number of points in class j.

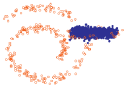
Experiments

Notation

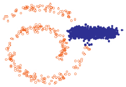
	$\sigma = 0$	$\sigma > 0$
$\alpha = 0$	PMM	TPMM
$\alpha > 0$	R-PMM	R-TPMM

Experiments

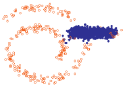
Synthetic data - two mixtures



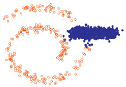
PMM



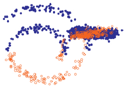
R-PMM



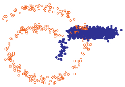
TPMM



R-TPMM



GPCA



Souvenir-Pless

Clustering of a spiral and a plane with noise.
Results with different algorithms.

Experiments

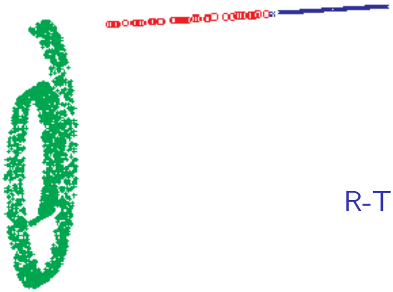
Synthetic data - two mixtures

	PMM		R-PMM		TPMM		R-TPMM	
	Estimated parameters							
m	**2.47**	**1.51**	**2.48**	**1.43**	**1.86**	**1.35**	**1.87**	**1.32**
θ	0.13	0.03	0.15	0.03	0.87	0.34	0.83	0.40
	Points in each class							
Pl.	764	36	800	0	784	16	800	0
Sp.	22	278	25	275	27	273	29	271

Estimated parameters of a spiral and a plane with noise ($k = 40$, $J = 2$).

Experiments

Clustering of a Swiss roll and a line with two different densities

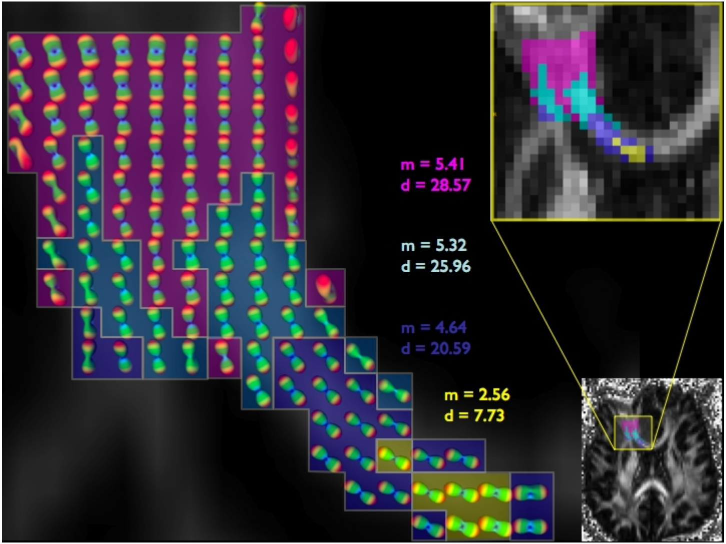


m: 1.98, 1.02, 0.99.
θ: 0.49, 0.53, 6.89.

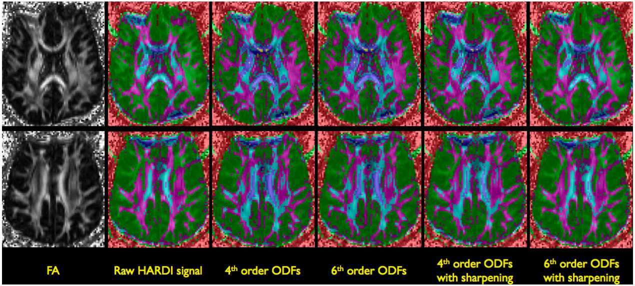
R-TPMM ($k = 20$, $J = 3$, $\alpha = 2$).

See Haro-Randall-Sapiro, NIPS 2006 and IJCV 2008 for numerous image/video examples.

Diffusion Imaging: Complexity in the Forceps Minor



Diffusion Imaging: Which Representation?



Color	*Red*	*Green*	*Blue*	*Yellow*	*L. blue*	*Purple*
HARDI						
Dim.	1.55	4.88	5.92	4.32	5.59	5.67
Dens.	9.27	16.01	10.69	2.42	13.18	15.85
Prob.	0.65	0.18	0.005	0.002	0.026	0.088
ODF 4						
Dim.	1.33	4.53	4.64	2.56	5.32	5.41
Dens.	12.53	26.70	20.59	7.73	25.96	28.57
Prob.	0.70	0.16	0.014	0.002	0.038	0.092
ODF 6						
Dim.	1.34	4.52	4.64	2.57	5.33	5.40
Dens.	12.54	26.64	20.57	7.74	25.94	28.49
Prob.	0.70	0.16	0.014	0.002	0.037	0.092

Conclusions and future work

Conclusions

- Algorithm to estimate and classify different dimensions and densities in noisy point cloud data.
- The noise is included in the statistical model.
- Natural way to introduce spatial/temporal regularization.
- Experiments in synthetic and real data.

Future work/ in progress

- Differentiate between manifolds of same dimension.
- Population analysis of HARDI/DTI data.

Thank you!