Stratification Learning:
Detecting Mixed Density and Dimensionality in Point Clouds

Gloria Haro

Universitat Politècnica de Catalunya (Spain)

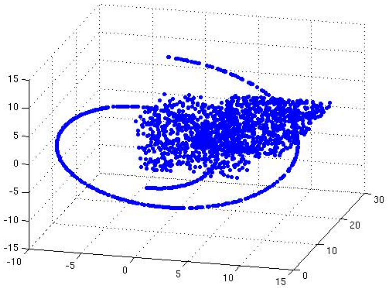
Joint work with:

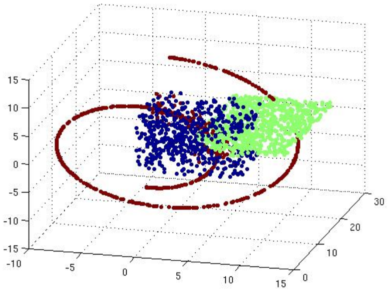
Gregory Randall: Universidad de la República (Uruguay)
Guillermo Sapiro: University of Minnesota (USA)

Motivation

Goal

Detect different dimensions – instead of a global dimension – in the same point cloud data.





Outline

1. Motivation
2. Previous work
3. Local dimension estimation
4. Detecting mixed dimensionality and density: PMM
5. Experiments PMM
6. Regularized PMM
7. Experiments R-PMM
8. Conclusions and future work

Dimension estimation

Previous work

1. **Projection methods**: global or local PCA, Isomap, MDS, ...
2. **Geometric methods**: based on fractal dimensions or nearest neighbor distances.
   - Correlation dimension.
   - Capacity dimension and packing numbers.
   - Geodesic entropic graphs.

Clustering by dimensionality: [Barbara and Chen], [Gionis et al.], [Souvenir and Pless], [Huang et al.], [Mordohai and Medioni].

Local dimension estimation

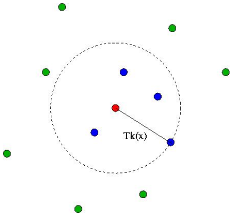
Levina and Bickel's approach

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

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 $k/n \to 0$).
- $f(x) \approx \textit{const}$. 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

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

Detecting mixed dimensionality and density

Our approach - Poisson Mixture Model (PMM)

Consider J mixture components:
vector of parameters $\psi = \{\pi^j, \theta^j, m^j; j = 1, \ldots, J\}$ 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:

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

Detecting mixed dimensionality and density

Our approach - Poisson Mixture Model (PMM)

Observation sequence: $Y = \{y_t; t = 1, \ldots, T\}$.

Detecting mixed dimensionality and density

Our approach - Poisson Mixture Model (PMM)

Observation sequence: $Y = \{y_t; t = 1, \ldots, T\}$.

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

Detecting mixed dimensionality and density

Our approach - Poisson Mixture Model (PMM)

Observation sequence: $Y = \{y_t; t = 1, \ldots, T\}$.

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. If we choose indicator variables

$$\delta_t^j \equiv \delta(z_t, C^j) = \begin{cases} 1 & \text{if } y_t \text{ generated by mixture} C^j, \\ 0 & \text{else.} \end{cases}$$

Detecting mixed dimensionality and density

Our approach - Poisson Mixture Model (PMM)

Observation sequence: $Y = \{y_t; t = 1, \ldots, T\}$.

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. If we choose indicator variables

$$\delta_t^j \equiv \delta(z_t, C^j) = \begin{cases} 1 & \text{if } y_t \text{ generated by mixture} C^j, \\ 0 & \text{else.} \end{cases}$$

Completed-data log-likelihood

$$\log p(Y, Z|\psi) = \sum_{t=1}^{T} \sum_{j=1}^{J} \delta_t^j \log \left[p(y_t|z_t = C^j, \psi^j) \pi^j \right]$$

Detecting mixed dimensionality and density

Our approach - Algorithm PMM

REQUIRE: The point cloud data, J, k.

Detecting mixed dimensionality and density

Our approach - Algorithm PMM

REQUIRE: The point cloud data, J, k.

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

Detecting mixed dimensionality and density

Our approach - Algorithm PMM

REQUIRE: The point cloud data, J, k.

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

Detecting mixed dimensionality and density

Our approach - Algorithm PMM

REQUIRE: The point cloud data, J, k.

1. **Initialization** of $\psi_0^j = \{\pi_0^j, m_0^j, \theta_0^j\}$ for all $j = 1, \ldots, J$.
2. **EM iterations** (until convergence of ψ_n^j):
   For each class $j = 1, \ldots, J$,
   - **E-step**: compute $h_n^j(y_t)$.

$$h_n^j(y_t) \equiv E[\delta_t^j | y_t, \psi_n] = P(\delta_t^j = 1 | y_t, \psi_n).$$

Detecting mixed dimensionality and density

Our approach - Algorithm PMM

REQUIRE: The point cloud data, J, k.

1. **Initialization** of $\psi_0^j = \{\pi_0^j, m_0^j, \theta_0^j\}$ for all $j = 1, \ldots, J$.
2. **EM iterations** (until convergence of ψ_n^j):
   For each class $j = 1, \ldots, J$,
   - **E-step**: compute $h_n^j(y_t)$.

     $$h_n^j(y_t) \equiv E[\delta_t^j | y_t, \psi_n] = P(\delta_t^j = 1 | y_t, \psi_n).$$

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

     $$\psi_{n+1}^j = \arg\max_{\psi} Q(\psi|\psi_n) + \lambda(\sum_{r=1}^{J} \pi^r - 1)$$

Detecting mixed dimensionality and density

Our approach - Algorithm PMM

REQUIRE: The point cloud data, J, k.

1. **Initialization** of $\psi_0^j = \{\pi_0^j, m_0^j, \theta_0^j\}$ for all $j = 1, \ldots, J$.
2. **EM iterations** (until convergence of ψ_n^j):
   For each class $j = 1, \ldots, J$,
   - **E-step**: compute $h_n^j(y_t)$.
   $$h_n^j(y_t) \equiv E[\delta_t^j | y_t, \psi_n] = P(\delta_t^j = 1 | y_t, \psi_n).$$
   - **M-step**: compute $\psi_{n+1}^j = \{\pi_{n+1}^j, m_{n+1}^j, \theta_{n+1}^j\}$
   $$\psi_{n+1}^j = \arg\max_{\psi} Q(\psi | \psi_n) + \lambda(\sum_{r=1}^{J} \pi^r - 1)$$

$\rightarrow$ **Soft clustering**

Detecting mixed dimensionality and density

Computation of parameters at step $n+1$:

$$\pi^j_{n+1} = \frac{1}{T}\sum_{t=1}^{T} h^j_{n+1}(y_t)$$

$$m^j_{n+1} = \left[\sum_t h^j_{n+1}(y_t)\hat{m}(x_t)^{-1} / \sum_t h^j_{n+1}(y_t)\right]^{-1}$$

$$f^j_{n+1} = e^{\theta^j_{n+1}} = \left[\sum_t h^j_{n+1}(y_t)\hat{f}(x_t)^{-1} / \sum_t h^j_{n+1}(y_t)\right]^{-1}$$

where $\hat{m}(x_t)$ and $\hat{f}(x_t)$ are the Levina and Bickel's estimators.

$\rightarrow$ **Weighted harmonic means**

Detecting mixed dimensionality and density

Asymptotic behaviour

Levina and Bickel's technique

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

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

PMM approach (hard clustering version)

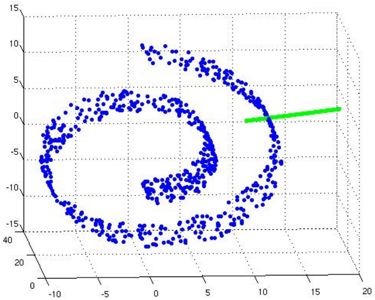
$$\mathrm{E}[\hat{m}^j] = m^j + \frac{m^j}{(k-1)N^j - 1}, \qquad \mathrm{Var}[\hat{m}^j] = (m^j)^2 O\left(\frac{1}{N^j(k-1)-4}\right)$$

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

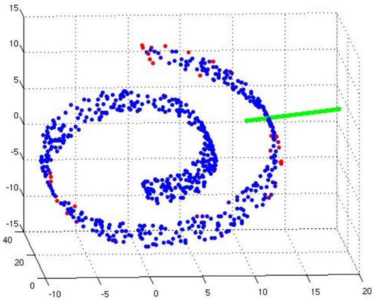
Experiments

Synthetic data - two mixtures

Swiss roll (700 points) and line (700 points) embedded in $\mathbb{R}^3$.
Poisson Mixture Model, $k = 10$ neighbors.



(a) 2 classes



(b) 3 classes

Experiments

Synthetic data - two manifolds

Swiss roll (700 points) and line (700 points) embedded in $\mathbb{R}^3$.
$k = 10$ neighbors.

Estimated parameters		
m	1.00	2.01
θ	5.70	2.48
π	0.50	0.50
% points in each class		
Line	100	0
SR	0	100

Table: Two Poisson distributions.

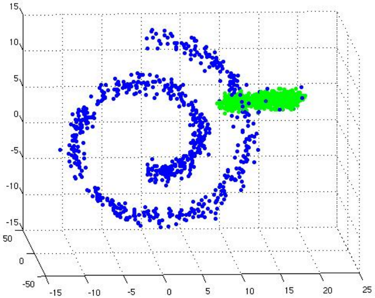
Estimated parameters			
m	1.00	2.01	2.16
θ	5.70	2.55	1.52
π	0.50	0.48	0.02
% points in each class			
Line	100	0	0
SR	0	96.57	3.43

Table: Three Poisson distributions.

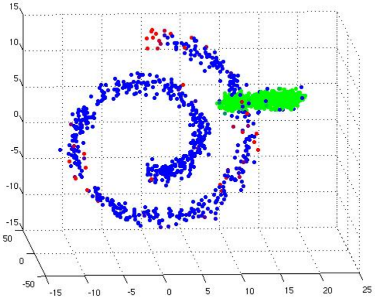
Synthetic noisy data - two manifolds

Swiss roll (700 points) and line (700 points) embedded in $\mathbb{R}^3$, noise $\sigma = 0.6$.

Poisson Mixture Model, $k = 20$ neighbors.



(c) 2 classes



(d) 3 classes

Experiments

Synthetic noisy data - two manifolds

Swiss roll (700 points) and line (700 points) embedded in $\mathbb{R}^3$, noise $\sigma = 0.6$.

$k = 20$ neighbors.

Estimated parameters		
m	3.02	2.38
θ	7.69	2.73
π	0.49	0.51
% points in each class		
Line	98.14	1.86
SR	0.86	99.14

Table: Two Poisson distributions.

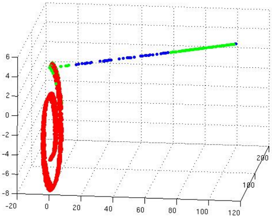
Estimated parameters			
m	3.01	2.40	2.26
θ	7.70	2.88	1.72
π	0.49	0.48	0.03
% points in each class			
Line	97.71	2.29	0
SR	0.71	93.00	6.29

Table: Three Poisson distributions.

Experiments

Synthetic data

Swiss roll (2500 points),
line 1 (100 points) and
line 2 (50 points),
embedded in $\mathbb{R}^3$.
$k = 20$ neighbors and 4 classes

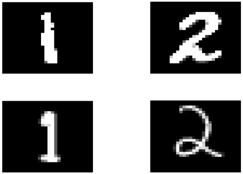


Estimated parameters				
m	1.94	1.04	0.98	1.93
θ	7.12	3.82	2.66	2.57
π	0.9330	0.0498	0.0167	0.0004
% points in each class				
Line	0.0	15.69	84.31	0.0
Line (dense)	0.0	99.00	1.00	0.0
Swiss Roll	98.92	1.08	0.0	0.0

Experiments

Real data - digits

MNIST database of handwritten digits: 784-dimensional image vectors, test set of 10.000 examples.
Mixture of digits one and two (1135 + 1032 points), $k = 10$ neighbors.



Some image examples

Estimated parameters		
m	8.50	12.82
θ	11.20	6.80
π	0.4901	0.5099
% points in each class		
Ones	93.48	6.52
Twos	0	100

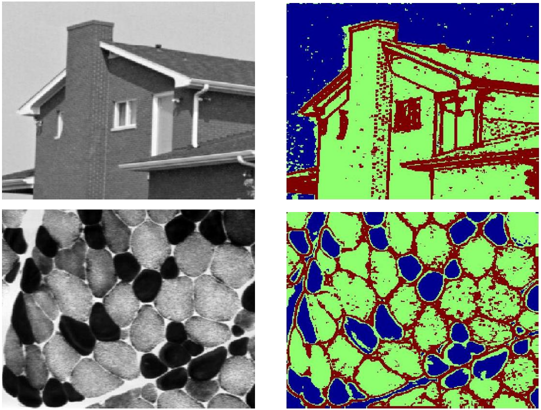
PMM 2 classes

Levina and Bickel: Ones: 9.13 Twos: 13.02 Mixture: 11.26
Costa and Hero: Ones: 8 Twos: 11 Mixture: 9

Experiments

Real data - image patches

$k = 18$ neighbors and 3 classes.



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 PMM (R-PMM)

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 PMM (R-PMM)

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 PMM (R-PMM)

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 PMM (R-PMM)

Dissimilarity functions

We propose two possibilities:

1. **Spatial/Temporal regularity**

$$\mathcal{D}_R := \sum_{s \sim t} (1 - h^j(y_s))^2$$

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

2. **Spatial intra-class compactness**

$$\mathcal{D}_C := \frac{\left|\left|x_t - X^j_{c,t}\right|\right|_2^2}{\frac{2}{J}\sum_{k=1}^{J}\left|\left|x_t - X^k_{c,t}\right|\right|_2^2},$$

where $X^j_{c,t}$ is the weighted centroid of class j without considering point x_t (the weights are $h^j(y_s)$).

Regularized PMM (R-PMM)

Algorithm R-PMM

REQUIRE: The point cloud data, J, k, 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 $h_{n+1}^j(y_t)$

     $$h_{n+1}^j(y_t) = \frac{p(y_t|m_n^j, \theta_n^j)\pi_n^j e^{-\alpha\mathcal{D}(t,j,X,H_n)}}{\sum_{l=1}^J p(y_t|m_n^l, \theta_n^l)\pi_n^l e^{-\alpha\mathcal{D}(t,l,X,H_n)}},$$

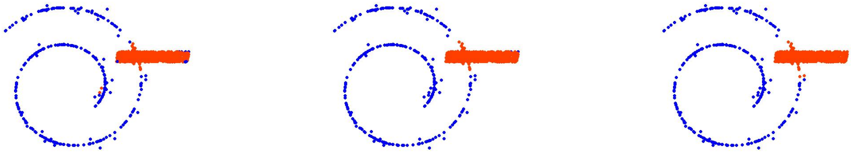
   - **2nd-step**: compute $\psi_{n+1}^j = \{\pi_{n+1}^j, m_{n+1}^j, \theta_{n+1}^j\}$

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

Experiments R-PMM

Synthetic Data

$k = 30,\ J = 2$

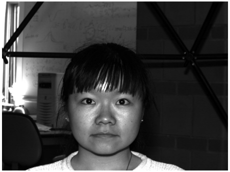


PMM | R-PMM ($\mathcal{D}_R$, $\alpha = 0.25$) | R-PMM ($\mathcal{D}_C$, $\alpha = 50$)

Gaussian noise of $\sigma = 0.66$ in 50 of the 300 points of the spiral.

Experiments R-PMM

Real Data - Yale Faces



Subject 5

Subject 6

Subject 7

	Estimated parameters			
Experiment	A: Sub. 5 and 6		B: Sub. 5, 6 and 7	
m	4.11	2.78	4.11	3.11
θ	5.16	2.73	4.77	2.60
π	0.89	0.11	0.81	0.19
	points in each class			
Subject 5	580	5	575	10
Subject 6	0	65	0	65
Subject 7	-	-	1	64

R-PMM with $\mathcal{D}_R$ (α=0.25, k=35, J=2).

Experiments R-PMM

Real Data - Activities in Video




	PMM				R-PMM ($\mathcal{D}_R$, $\alpha = 10$)			
	C1	C2	C3	C4	C1	C2	C3	C4
Wave	106	8	0	0	109	5	0	0
Jump in place	0	127	0	0	0	127	0	0
Walk	0	2	81	5	0	0	88	0
Jump	0	0	67	5	0	0	72	0

R-PMM $\mathcal{D}_R$ ($J = 4$, $k = 20$, 48×60 dimensional vectors).

Temporal regularization: 6 prev.and 6 post. frames as neighbors in $\mathcal{D}_R$.

Conclusions and future work

Conclusions

- Algorithm to estimate and classify different dimensions and densities in point cloud data.
- Natural way to introduce spatial/temporal regularization.
- Experiments in synthetic and real data.

Future work/ in progress

- Introduce the presence of noise in the model.
- Differentiate between manifolds of same dimension.
- Analysis of neuroscience data.

Thank you!