Regularized Mixed Dimensionality and Density Learning in Computer Vision

Gloria Haro
Dept. Teoria del Senyal i Comunicacions
Universitat Politècnica de Catalunya, Spain
gloria@gps.tsc.upc.edu

Gregory Randall
Instituto de Ingeniería Eléctrica
Universidad de la República, Uruguay
randall@fing.edu.uy

Guillermo Sapiro
Dept. of Electrical Engineering
University of Minnesota, USA
guille@umn.edu

Abstract

A framework for the regularized estimation of non-uniform dimensionality and density in high dimensional data is introduced in this work. This leads to learning stratifications, that is, mixture of manifolds representing different characteristics and complexities in the data set. The basic idea relies on modeling the high dimensional sample points as a process of Poisson mixtures, with regularizing restrictions and spatial continuity constraints. Theoretical asymptotic results for the model are presented as well. The presentation of the framework is complemented with artificial and real examples showing the importance of regularized stratification learning in computer vision applications.

1. Introduction

Recently, there has been significant interest in analyzing the intrinsic structure of high dimensional data, this is commonly known as *manifold learning*, e.g., [4, 5, 7, 13, 14, 17, 21]. Often, points that live in a high dimensional space can be parametrized by a number of parameters much smaller than the ambient dimension. A representation (embedding) of the data in a lower dimensional space is very helpful for analysis and computations on the dataset.

Most of the works on manifold learning rely on the hypothesis that all the points under analysis are samples of the same manifold and thus there is a unique intrinsic dimension. However, this is often not a good assumption. It is likely that, for example, a collection of image portraits of the same person under varying pose and illumination, lies on a manifold defined by a set of parameters related to the variations in pose and illumination. On the other hand, let us consider a set of images representing scanned digits. It might happen that the images representing the digit '1' can be described with a different number of parameters than the images for digit '3.' Videos of diverse human motions contain the same complexity variability. In these cases, it is important to detect that there are different complexities present in the same point cloud data.

This problem, clustering-by-dimensionality and *stratification learning*, has recently been addressed by some authors. Barbará and Chen, [3], defined a hard clustering technique based on the fractal dimension (box-counting) which also finds the number of clusters and the intrinsic dimension of each cluster. Gionis *et al.*, [9], propose a two-step algorithm: First, they estimate the local correlation dimension and density for each point; then, standard clustering techniques are used to cluster the two-dimensional representation (dimension + density) of the data. Souvenir and Pless, [19], compute a soft clustering based on Isomap [21]. After clustering, each cluster dimensionality is estimated following [14]. Huang *et al.*, [12], cluster linear subspaces with an algebraic geometric method based on polynomial differentiation and a Generalized PCA (GPCA), [22], which finds the number of linear subspaces and their intrinsic dimensions. The work of Mordohai and Medioni, [16], estimates the local dimension using tensor voting. A Poisson Mixture Model (PMM) was introduced by Haro *et al.*, [10], to simultaneously estimate non-uniform dimensionality and density and use them for soft clustering. Cao and Haralick, [6], propose a hard clustering by dimensionality: First, local dimensionality is computed via local PCA; and then, neighboring points are clustered together if they have the same dimension and if the error of representing the new cluster as a combination of basis functions in a kernel-based feature space is small.

Among these clustering-by-dimensionality techniques, only the one by Cao and Haralick includes spatial information in order to obtain a regularized classification. Recently, Lu and Vidal, [15], combined GPCA with an additional spatial constraint in a k-means fashion. They showed that, by adding this constraint, the classification is improved in the intersection of the linear subspaces.

In this paper, we first extend the framework introduced in [10] to include regularization and spatial constraints. We show that these new constraints can be easily incorporated within the PMM framework and come natural following a new interpretation of the model in [10]. These constraints can be adapted to either add spatial regularity in the classification or intra-class spatial compactness. Temporal regularization is also possible, within the same approach, by defining the proper neighborhood in the constraint. The interest in extending this particular model relies on its capability to deal with non-linear manifolds and simultaneously estimate the soft clustering and the intrinsic dimension and density of each cluster. This collection of attributes is not shared by any of the other above mentioned approaches. We complete the novel contributions by presenting asymptotic results on the proposed estimator and new examples for typical computer vision data.

In Section 2 we review the method proposed by Levina and Bickel, [14], which gives a local estimation of the intrinsic point cloud dimension. Inspired by this technique, a Poisson Mixture Model was proposed in [10], which simultaneously computes a soft clustering and estimates the intrinsic dimension and density of each cluster. This approach is reviewed and extended to include spatial terms in Section 3. Asymptotic results are presented here as well. We show experiments with synthetic and real data in Section 4, and finally, conclusions are presented in Section 5.

2. Local intrinsic dimension estimation

Levina and Bickel, [14], proposed a geometric and probabilistic method which estimates the local dimension and density of a point cloud data. This dimension estimator is equivalent to the one proposed in [20]. Their approach is based on the idea that if we sample an m-dimensional manifold with T points, the proportion of points that fall into a ball around a point x_t is $\frac{k}{T} \approx f(x_t)V(m)R_k(x_t)^m$. Here, the given point cloud, embedded in high dimensions D, is $X = \{x_t \in \mathbb{R}^D; t = 1, \ldots, T\}$, k is the number of points inside the ball, $f(x_t)$ is the local sampling density at point x_t, $V(m)$ is the volume of the unit sphere in $\mathbb{R}^m$, and $R_k(x_t)$ is the Euclidean distance from x_t to its k-th nearest neighbor (kNN). Then, they consider the inhomogeneous process $N(R, x_t)$, which counts the number of points falling into a small D-dimensional sphere $B(R, x_t)$ of radius R centered at x_t. This is a binomial process, and some assumptions need to be done to proceed. First, if $T \to \infty$, $k \to \infty$, and $k/T \to 0$, then we can approximate the binomial process by a Poisson process. Second, the density $f(x_t)$ is constant inside the sphere, a valid assumption for small R. With these assumptions, the rate λ of the counting process $N(R, x_t)$ can be written as $\lambda(R, x_t) = f(x_t)V(m)mR^{m-1}$. The log-likelihood of the process $N(R, x_t)$ is then given by

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

where $\theta(x_t) := \log f(x_t)$ is the density parameter and the first integral is a Riemann-Stieltjes integral [18]. The maximum likelihood estimators lead to a computation for the local dimension at point x_t, $m(x_t)$, depending on all the neighbors within a distance R from x_t [14]. In practice, it is more convenient to compute a fixed amount k of nearest neighbors. Thus, the local estimators at point x_t are

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

$$\theta(x_t) = \log \left((k-1) / \left(V(m(x_t)) R_k(x_t)^{m(x_t)} \right) \right) \quad (2)$$

where $V(m(x_t)) = (2\pi^{m(x_t)/2})/(m(x_t)\Gamma(\frac{m(x_t)}{2}))$, and $\Gamma(\frac{m(x_t)}{2}) = \int_0^\infty t^{m(x_t)/2-1} e^{-t} dt$. If the data points belong to the same manifold, the authors propose to average over all local estimators $m(x_t)$ in order to obtain a more robust estimator. However, if there are two or more manifolds with different dimensions, the average does not make sense, unless we first cluster according to dimensionality and then we estimate the dimensionality for each cluster. Another possibility is the simultaneous soft clustering and estimation technique described in Section 3.

3. Dimensionality and density estimation with simultaneous soft clustering

3.1. Poisson Mixture Model (PMM)

In [10], the authors proposed to study a stratification by extending the Levina and Bickel's technique. Instead of modeling each point and its local ball of radius R as a Poisson process and computing the maximum likelihood (ML) for each ball separately, all the possible balls are considered at the same time in the ML function. The probability density function for all the point cloud is a mixture of Poisson distributions with different parameters (dimension and density) in each class. This allows the presence of different intrinsic dimensions and densities in the dataset. These are automatically computed while being used for soft clustering.

Let us consider J different Poisson distributions in the mixture, each one with a (possibly) different dimension m and density parameter θ. Let us denote by ψ the vector set of parameters, $\psi = \{\psi^j = (\pi^j, \theta^j, m^j); j = 1, \ldots, J\}$, where π^j is the mixture coefficient for class j (the proportion of distribution j in the dataset), θ^j is its density parameter ($f^j = e^{\theta^j}$), and m^j is its dimension.

The observable event is, as in the Levina-Bickel approach, the number of points inside the ball $B(R, x_t)$ of radius R centered at point x_t, denoted by $y_t = N(R, x_t)$. The total number of observations is T' and $Y = \{y_t; t = 1, \ldots, T'\}$ is the observation sequence. Often, $T' \equiv T$, all points in the dataset are considered. Let us also denote by $p(\cdot)$ the probability density function and by $P(\cdot)$ the probability. The density function of the Poisson mixture model is given by

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

Since the observations follow a Poisson distribution,

$$p(y_t|\theta^j, m^j) = e^{\int_0^R \log \lambda^j(r)\, dN(r,x_t)} e^{-\int_0^R \lambda^j(r) dr},$$

where $\lambda^j(r) = e^{\theta^j} V(m^j) m^j r^{m^j-1}$. If Y contains T statistically independent variables (a standard assumption), then the probability density function of the observation sequence is the product of the individual probability densities, $p(y_t|\psi)$, and the log-likelihood is:

$$L(Y|\psi) = \log p(Y|\psi) = \sum_{t=1}^{T} \log p(y_t|\psi). \quad (3)$$

Let us consider the hidden-state information, that is, which mixture (or expert) generates each observation. We denote by $Z = \{z_t \in C; t = 1, \ldots, T\}$ the set of hidden variables and by $C = \{C^1, C^2, \ldots C^J\}$ the set of class labels. Then, $z_t = C^j$ means that the j-th mixture generates y_t. Using Z we can write the complete data log-likelihood as

$$\log p(Z, Y|\psi) = \sum_{t=1}^{T} \sum_{j=1}^{J} \delta_t^j \log \left[p(y_t|\psi^j) \pi^j \right], \quad (4)$$

where a set of indicator variables δ_t^j, called membership functions, is used in order to indicate the status of the hidden variables:

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

The unknown parameters in (4) are: The membership function of an expert (class), δ_t^j, the mixture probabilities, π^j, and the parameters of each expert, m^j and θ^j. Usually, problems involving a mixture of experts are solved by the Expectation Maximization (EM) algorithm [8]. The EM is based on the following decomposition of the log-likelihood (3):

$$\begin{aligned} L(Y|\psi, H) = &\sum_{t=1}^{T} \sum_{j=1}^{J} h^j(y_t) \log \left[p(y_t|\psi^j) \pi^j \right] \\ &- \sum_{t=1}^{T} \sum_{j=1}^{J} h^j(y_t) \log \left[h^j(y_t) \right], \end{aligned} \quad (5)$$

where $H = \{h^j(y_t) \leq 1; t = 1, \ldots, T, j = 1, \ldots, J\}$ and $h^j(y_t)$ is the probability that observation t belongs to mixture j. Thus, the set H is also unknown. Since the membership functions are indicator variables, the first term in (5) is the expectation of (4) with respect to Z. Also notice that the second term is the entropy of the membership functions.

An interesting interpretation of the EM algorithm is introduced in [11], where the EM is seen as an alternate optimization algorithm of the log-likelihood (5). Then, the E-step is nothing else than the maximization of $L(Y|\psi, H)$ with respect to H with the additional constraint that $\sum_{j=1}^{J} h^j(y_t) = 1$ for each observation $t = 1, \ldots, T$. Thus, the variables $h^j(y_t)$ at step $n+1$ of the optimization algorithm are

$$h_{n+1}^j(y_t) = \frac{p(y_t|m_n^j, \theta_n^j)\pi_n^j}{\sum_{l=1}^{J} p(y_t|m_n^l, \theta_n^l)\pi_n^l}. \quad (6)$$

In the same way, variables ψ are obtained by maximizing $L(Y|\psi, H)$ with respect to ψ with an additional constraint for the mixture probabilities: $\sum_{j=1}^{J} \pi^j{=}1$. This gives equations (10)-(12) for the variables at step $n+1$. In order to compute m_{n+1}^j we have used the same approach as in [14], by means of a k nearest neighbor graph. The PMM approach just described is summarized in **R-PMM Algorithm**, for the particular case of $\alpha = 0$ (no regularization).

3.2. Regularized and spatially constrained PMM

The PMM algorithm seeks a soft clustering according to dimensionality and density, but does not (explicitly) take into account spatial information. Adding regularization is the goal of this section. Regularization helps to improve the classification in noisy data and points lying close to manifold edges (see results in Figure 1). This is done inspired by the work in [1] for the neighborhood EM (NEM), where the authors extend the EM algorithm adding spatial constraints. This neighborhood spatial information is introduced as a penalization term in the log-likelihood, following Hathaway's EM interpretation [11]. In our context, we complete (5) with a spatial term $S(H)$,

$$F(\psi, H) = L(Y|\psi, H) + \alpha S(H), \quad (7)$$

where α is a parameter that controls the tradeoff between the spatial term and the likelihood. Its value is also related to the amount of noise in the data. Then, function F is maximized with an alternate optimization technique. Since the new term, S, only depends on H, the optimization procedure results in a EM-type algorithm with a modified membership probability that not only depends on the likelihood but also on the spatial criteria. The NEM algorithm uses (note the similitude with MRFs, see below)

$$S_{NEM}(H) = \sum_{t=1}^{T} \sum_{j=1}^{J} h^j(y_t) \sum_{s \sim t} h^j(y_s),$$

where $s \sim t$ indicates that there is a neighborhood relationship between observations s and t. By maximizing this term, we want, for each observation t, as many neighbors as possible with high probability of belonging to the same class as observation t, thus regularizing the classification. However, we will use a more general expression for $S(H)$ based on a dissimilarity measure, $\mathcal{D}$, between every observation and other observations in the sequence,

$$S(H) = -\sum_{t=1}^{T}\sum_{j=1}^{J} h^j(y_t)\mathcal{D}(t,j,X,H). \qquad (8)$$

The expression (8) provides a generic framework for introducing constraints in the soft classification, besides the ones already present in the PMM model, namely dimensionality and density. One possibility, as in the NEM algorithm, is to introduce spatial regularity. Then, as dissimilarity measure we use $\mathcal{D}_R$ defined as:

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

Different neighborhoods definitions in $\mathcal{D}_R$ result in different kinds of regularization. A natural choice is the manifold neighborhood, for that, we can define as neighbors the k nearest neighbors. However, for specific applications one might be interested in other neighborhoods, e.g., pixel neighborhoods or contiguous frames in video applications (see experiment in Figure 6).

We could also impose spatial intra-class compactness with the definition of a proper dissimilarity function,

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

where $X^j_{c,t}$ is the weighted centroid of class j without considering point x_t:

$$X^j_{c,t} = \frac{\sum_{s=1,s\neq t}^{T} h^j(y_s)x_s}{\sum_{s=1,s\neq t}^{T} h^j(y_s)}.$$

We study the effect of parameter α in Figures 2 and 3. Observe that, for a small value of the regularization parameter α, both $\mathcal{D}_R$ and $\mathcal{D}_C$ as $\mathcal{D}$ in (8) produce spatial regularization over the clustering with the original PMM. If we use a large value of α all the points will be classified in the same cluster. However, for intermediate values of α, the effect of (8) in the classification process is very different in both cases. If we use $\mathcal{D}_R$, the regularization is stronger (compared to lower values of α). On the other hand, the use of $\mathcal{D}_C$ will produce a k-means classification.

As noted in [1], the EM algorithm with additional constraints can be seen as finding the Gibbs distribution with energy $-F(\psi,H)$. In the particular case where the additional constraint is neighborhood dependent, $S_{NEM}(H)$ and $S(H)$ with $\mathcal{D}_R$, the Gibbs distribution defines a Markov Random Field.

The maximization of F (Equation (7)), is obtained as in [1], with an alternate optimization technique which results in an EM-type algorithm. Maximizing (7) with respect to H, with $S(H)$ defined in (8) – with the constraints $\sum_{j=1}^{J} h^j(y_t) = 1$ for each observation $t = 1,\ldots,T$, by means of Lagrange multipliers – results in the following expression for the membership probabilities:

$$h^j(y_t) = \frac{p(y_t|m^j,\theta^j)\pi^j_n e^{-\alpha\mathcal{D}(t,j,X,H)}}{\sum_{l=1}^{J} p(y_t|m^l,\theta^l)\pi^l e^{-\alpha\mathcal{D}(t,l,X,H)}}. \qquad (9)$$

Since the only term in (7) which depends on ψ is $L(Y|\psi,H)$, the optimal values of $\psi^j = \{(\pi^j,\theta^j,m^j)$ for $j = \{1,\ldots,J\}$ do not change with respect to the original PMM algorithm. The regularized version of the PMM algorithm is summarized in the **R-PMM Algorithm**.

R-PMM Algorithm *Regularized Poisson Mixture Model*

REQUIRE: The point cloud data, J (number of desired classes), k (scale of observation) and α (regularization parameter).
ENSURE: Regularized soft clustering according to dimensionality and density.

1. Initialization of $\psi_0 = \{\pi^j_0, m^j_0, \theta^j_0\}$ to any set of values which ensures that $\sum_{j=1}^{J}\pi^j_0 = 1$.
2. Iterations on n,
   For all $j = 1,\ldots J$, compute:
   - 1st step: Compute, for all $t = 1,\ldots,T$,

     $$h^j_{n+1}(y_t) = \frac{p(y_t|m^j_n,\theta^j_n)\pi^j_n e^{-\alpha\mathcal{D}(t,j,X,H_n)}}{\sum_{l=1}^{J} p(y_t|m^l_n,\theta^l_n)\pi^l_n e^{-\alpha\mathcal{D}(t,l,X,H_n)}},$$

     where $H_n = \{h^j_n(y_t);\ j = 1,\ldots,J,\ t = 1,\ldots,T\}$.
   - 2nd step: Compute

     $$\pi^j_{n+1} = \frac{1}{T}\sum_{t=1}^{T} h^j_n(y_t), \qquad (10)$$

     $$m^j_{n+1} = \left[\frac{\sum_{t=1}^{T} h^j_n(y_t)\sum_{j=1}^{k-1}\log\frac{R_k(y_t)}{R_j(y_t)}}{\sum_{t=1}^{T} h^j_n(y_t)(k-1)}\right]^{-1}, \qquad (11)$$

     $$\theta^j_{n+1} = \log\sum_{t=1}^{T} h^j_n(y_t)(k-1) - \log\left(V(m^j_n)\sum_{t=1}^{T} h^j_n(y_t)R_k(y_t)^{m^j_n}\right). \qquad (12)$$

Until convergence of ψ_n, that is, when $||\psi_{n+1}-\psi_n||_2 < \epsilon$, for a certain small value ϵ.

Remark. *If we write the estimators* (11) *and* (12) *as functions of the estimators* (1) *and* (2)*, we obtain*

$$m^j_{n+1} = \left[\sum_{t=1}^{T} h^j_n(y_t) m(x_t)^{-1} / \sum_{t=1}^{T} h^j_n(y_t)\right]^{-1},$$

$$f^j_{n+1} = e^{\theta^j_{n+1}} = \left[\sum_{t=1}^{T} h^j_n(y_t) f(x_t)^{-1} / \sum_{t=1}^{T} h^j_n(y_t)\right]^{-1},$$

where $f(x_t) = e^{\theta(x_t)}$. Notice that the estimators in the PMM (and R-PMM) approach are the inverse of the weighted average of the inverse estimators of Levina-Bickel. The weight at each point is the probability of the membership function. In the particular case of one unique class, $J = 1$, we obtain the global dimension estimator proposed by MacKay and Ghahramani (http://www.inference.phy.cam.ac.uk/mackay/dimension/).

As proved in [2], if α is small enough, (7) has a guaranteed global maximum for a fixed value of ψ, and the aditional term $S(H)$ does not affect the convergence of the EM-type algorithm. It can be shown that, for the case of $\mathcal{D}_R$, the corresponding bound on α is

$$\alpha_R < \frac{1}{2\max_{t,j}\sum_{s\sim t}(1 - h^j(x_s))}.$$

Notice that $\alpha_R < 1/(2k)$ in the worst case scenario. In the case of $\mathcal{D}_C$, the bound has a more complicated expression, and in the worst case scenario

$$\alpha_C < \left[4(J-1)(T-1)\max_s ||x_s||_1 \max_t \mathcal{B}(t)\right]^{-1},$$

where $\mathcal{B}(t) = \left(\max_j \left(\frac{||x_t - X^j_{c,t}||_1}{\sum_{s=1,s\neq t}^{T} h^j(x_s)}\right) \frac{\max_j \mathcal{D}'_C(t,j)}{\sum_{k=1}^{J} \mathcal{D}'_C(t,k)/J}\right)$, and $\mathcal{D}'_C(t,j) = ||x_t - X^j_{c,t}||_2^2$. The EM suffers from local maxima, this can be alleviated running the algorithm several times with different initializations. Different random subsets of points, from the original point cloud, may be used in each run. We have experimented with both approaches and the results are always similar if we initialize all the probabilities equally, that is, $\pi^j_0 = 1/J$ for $j = 1, \ldots, J$, which is the initialization we have used in the experiments here presented. We also normalize the distances, the maximum distance between a pair of points in the dataset is one.

3.3. Asymptotic analysis

Levina and Bickel show in [14] that under the assumptions $T \to \infty$, $k \to \infty$, and $k/T \to 0$, that is when the Poisson approximation is correct, the mean and variance of the dimension estimator (1) (with $k-2$ instead of $k-1$ in the denominator) are

$$E[m(x_t)] = m, \quad \text{Var}[m(x_t)] = \frac{m^2}{k-3}.$$

We can apply the same type of analysis to the PMM model in the particular case of hard clustering, that is

$$h^j(y_t) = \begin{cases} 1 & \text{if } j = \text{argmax}_i h^i(y_t), \\ 0 & \text{else.} \end{cases}$$

We assume, in addition, that all the points that belong to class j are well classified. Then, we obtain the following results

$$E[m^j] = \bar{m}_j + \frac{\bar{m}_j}{(k-1)N_j - 1},$$

$$\text{Var}[m^j] = \bar{m}_j^2 O\left(\frac{1}{(k-1)N_j - 4}\right),$$

where $\bar{m}_j$ is the correct intrinsic dimension of class j and N_j is the amount of points classified as class j. The basic lines of the analysis are as follows: We use the fact that R_k/R_j is distributed, under the Poisson assumption, as a Uniform(0,1) distribution, the log of such a distribution is an Exponential(1), and then, the sum of $(k-1)$ Exponential(1) distributed variables is a Gamma($k-1$,1). Now, for the case of PMM, we use the fact that the sum of N_j Gamma($k-1$,1) distributions is a Gamma($(k-1)N_j$,1) and then we use the properties of an inverse Gamma distribution. The analysis of the density estimator θ^j is the subject of current research.

4. Experimental results

We now present experimental results with synthetic and real data for the proposed Regularized PMM (R-PMM). We fixed α experimentally and found that the same value of α gives good results for different experiments of the same kind and with the same amount of noise. In some cases, the same value of α is good for different experiments, as it can be noticed in the experiments here presented.

First, we work with synthetic point cloud data formed by 300 samples of a 1D spiral and 800 of a 2D plane, both in 3D embedding space. We compare the results for the PMM algorithm, the R-PMM with $\mathcal{D}_R$, and the R-PMM with $\mathcal{D}_C$, Table 1 and Figure 1. All the cases where computed with two classes ($J = 2$) and 30 neighbors ($k = 30$). We also study the robustness to noise by adding Gaussian noise of standard deviation $\sigma = 0.66$ to 50 randomly picked samples of the spiral. Table 1 shows, for each algorithm, the obtained values for the mixture model parameters and also the quantitative result of the classification. Both the PMM and R-PMM are able to separate the manifolds. We also applied [14] and [7], and obtained dimensions of 1.68 and 1.46 respectively. These approaches consider that all the points are samples of the same manifold and give an estimated dimension which is an average of the actual dimensions. In Figure 1, we color each point according to the class it belongs to (the class whose membership probability is the largest).

	Estimated parameters					
	PMM		R-PMM $\mathcal{D}_R$		R-PMM $\mathcal{D}_C$	
m	1.91	1.10	1.91	1.09	1.91	1.08
θ	5.21	3.34	5.20	3.33	5.19	3.32
π	0.74	0.26	0.74	0.26	0.75	0.25
	points in each class					
Plane	790	10	798	2	800	0
Spiral	22	278	21	279	23	277

Table 1. *Estimated parameters and clustering results of a 1D spiral (with noise in 50 points) and a 2D plane ($k = 30$, $J = 2$).*

Notice how, for the two R-PMM versions and roughly for the PMM, the classification is robust to noise. As expected, the R-PMM algorithm (in its two versions) gives a more spatially regularized classification than the PMM algorithm: Observe that the points located at the edges of the plane and the missclassified points in the spiral in Fig. 1(a) are well classified in Fig. 1(b) and 1(c).

Figures 2 and 3 show the evolution, according to α, of the classification of the R-PMM with $\mathcal{D}_R$ and $\mathcal{D}_C$ respectively. As it can be observed, the classification with $\mathcal{D}_C$ is less sensible to the choice of parameter α since it is stable for a larger range of values. For intermediate values of α, $\mathcal{D}_C$ produces a k-means kind of classification while $\mathcal{D}_R$ increases the regularization by diffusing the membership valu

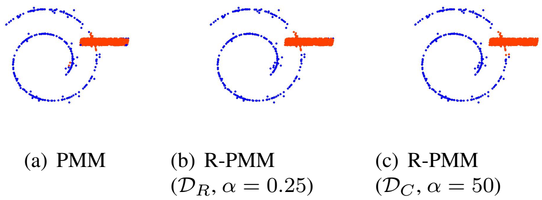
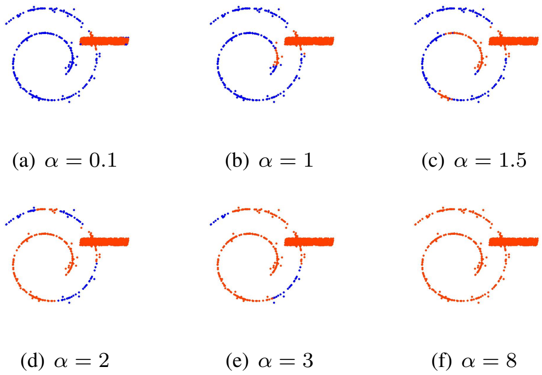
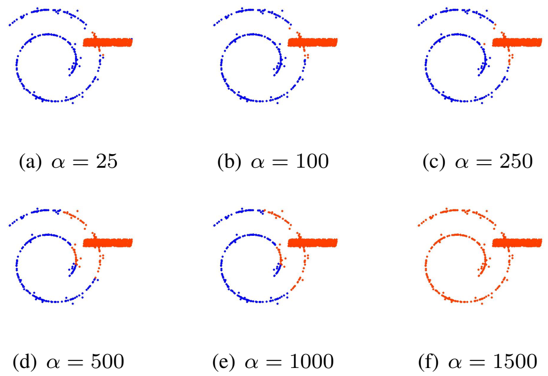

(a) PMM (b) R-PMM ($\mathcal{D}_R$, $\alpha = 0.25$) (c) R-PMM ($\mathcal{D}_C$, $\alpha = 50$)

Figure 1. *Clustering of a 1D spiral and a 2D plane ($k = 30$, $J = 2$). Gaussian noise of $\sigma = 0.66$ is added to 50 of the 300 points of the spiral. Points colored according to their classification.*


(a) $\alpha = 0.1$ (b) $\alpha = 1$ (c) $\alpha = 1.5$

(d) $\alpha = 2$ (e) $\alpha = 3$ (f) $\alpha = 8$

Figure 2. *Clustering of a 1D spiral and a 2D plane ($\mathcal{D}_R$, $k = 30$, $J = 2$). Evolution of the classification as parameter α increases.*


(a) $\alpha = 25$ (b) $\alpha = 100$ (c) $\alpha = 250$

(d) $\alpha = 500$ (e) $\alpha = 1000$ (f) $\alpha = 1500$

Figure 3. *Clustering of a 1D spiral and a 2D plane ($\mathcal{D}_C$, $k = 30$, $J = 2$). Evolution of the classification as parameter α increases. Observe how, for intermediate values of α, it produces a k-means classification.*

	Estimated parameters	
m	8.5	13.68
θ	10.11	6.20
π	0.50	0.50
	points in each class	
One	1061	74
Three	0	1010

Table 2. *Estimated parameters and clustering results of the mixture of digits '1' and '3' (R-PMM $\mathcal{D}_C$, α=50, J=2, k=10).*

As a test of the performance with real data, we first work with the MNIST database of handwritten digits,[1] which has a test set of 10.000 examples. Each digit is an image of 28 × 28 pixels and we treat the data as 784-dimensional vectors. We analyze the mixture of digits one and three, some examples of those scanned digits are in Figure 4. The obtained results are in Table 2. Levina-Bickel's technique gives a dimension value of 11.62 and Costa-Hero's 9.42. Like in the synthetic examples, these methods give a dimension in between the two different dimensions present in the point cloud. With the R-PMM algorithm, we are able to separate the points (images) corresponding to each digit, both sets have different dimensionality and density. We have observed that some other digits do have the same dimensionality, as expected.

Figure 4. *Examples of images of scanned digits (28×28 pixels). We use each image as a point in a 784 dimensional space where we cluster by dimensionality and density (see results in Table 2).*

We also analyze images from the Yale Face Database B,[2] which contains images of 10 subjects under 585 viewing conditions (9 poses and 65 illumination conditions), see Fig. 5. Each image has a size of 640×480 pixels. For computational reasons we subsampled the images by a factor of ten and use each 64×48 image as a vector in a high di-

[1] http://yann.lecun.com/exdb/mnist/

[2] http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html

	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

Table 3. *Estimated parameters and clustering results of the mixture of subject 5 (all poses, all illuminations) and subjects 6 and 7 (one pose, all illuminations) in the Yale Face Database B (R-PMM with $\mathcal{D}_R$, α=0.25, k=35, J=2). Experiment A (left): Subjects 5 and 6. Experiment B (right): Subjects 5, 6 and 7.*

mensional space. First, we analyze the point cloud formed by the 585 images of subject 5 (varying pose and illumination) together with the 65 images of subject 6 in the first pose only and under varying illuminations. The numerical results and confusion matrix using the R-PMM algorithm with $\mathcal{D}_R$ ($\alpha = 0.25$) are presented in Table 3 (left). Note how both subjects are well separated, and the set of images of subject 5 has a dimension one unity larger than the dimension for subject 6, since we do not consider the pose variation for this subject. The obtained dimension in this last case is close to three, this result is consistent with [15, 22]. As a second example, we add images of subject 7 (one pose under varying illumination) to the dataset used in the previous experiment. The results are presented in Table 3 (right). The set of images corresponding to subjects 6 and 7 are classified in the same class because they have lower complexity/dimensionality (only one pose) than the manifold of images corresponding to subject 5. Since the R-PMM clusters data according to dimensionality and density, we can not separate images of subjects 6 and 7 even if we set three classes in the algorithm. When $J = 3$, 56% of images of subject 5 are classified as one class, the other 44% are classified as a second class, and all the images of subject 6 and 7 (except for one) are classified together as a third class.

Figure 5. *Examples of images of subjects 5, 6, and 7 of the Yale Face Database B. See results in Table 3.*

Finally, we used the R-PMM framework to study different human activities in video, using public available data. [3] We created a point cloud with the frames of four videos corresponding to four different activities: Walking, jumping, waving, and jumping in place, all performed by the same person in a static background (see some frame examples in Figure 6). Each frame contains 144×180 pixels, and we subsampled each frame by a factor of 3 and used 48×60 dimensional vectors. This is mainly to speed up computations, actually the classification results are very similar without the subsampling process. The confusion matrix with the classification results using the PMM and the R-PMM algorithm (with $\mathcal{D}_R$ and $\alpha = 10$) are presented in Figure 6. In video applications, one may be interested in temporal regularization. For that, we consider a temporal neighborhood in $\mathcal{D}_R$, more concretely we take into account the 6 previous and 6 posterior frames in the regularization term. Although there are four different activities and we selected $J = 4$ classes in the algorithm, the classification is roughly done in three classes: waving, jumping in place, and walking and jumping while advancing (these two activities have the same dimensionality/complexity and density as detected by our proposed framework).

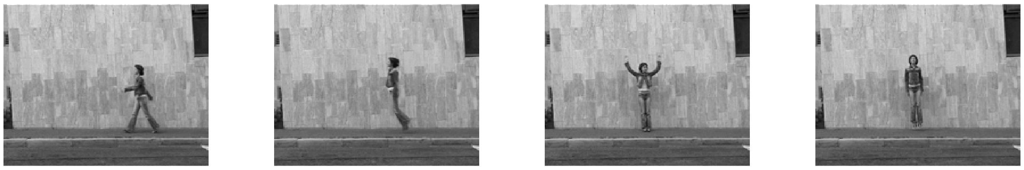
Regarding the computational time, the most expensive part is the kNN-graph. For the experiment B with Yale faces (Fig. 5, 715 points of dimension 3072) the execution takes 34.58s while 27.22s of the total time is spent in the computation of the kNN-graph. In the video experiment (Fig. 6, 401 points of dimension 2880) the total time and the kNN-graph time are, respectively, 10.67s and 8.04 (CPU: Pentium 4, 1.80 GHz).

5. Conclusions

In this work we introduced a framework for the simultaneous and regularized/constrained estimation of the intrinsic dimensionality and density of high dimensional point cloud data, as the basis for complexity/density based soft-clustering. We showed that regularization and spatial constraints can be naturally introduced in this approach. The experiments show the importance of adding regularization in the classification. With the proper dissimilarity function and neighborhood type, we are able to add spatial or temporal regularity in the classification or intra-class spatial compactness. Other type of constraints are possible under the same proposed framework. Asymptotic theoretical results were also presented.

We would like to follow this direction of work and study other constraints which can be useful for stratification learning. One possibility is to define a dissimilarity function which leads to separate manifolds that share the same dimensionality and density. This will define a new constraint that will also help in the classification process when there is an intersection of two manifolds (and where the algorithm fails at the present stage). Since the density depends on the dimension, we are intrinsically giving more importance to the dimension criterion in our framework. The control of the relative importance of these two criteria needs also to be addressed. Results in these directions will be reported

[3] http://www.wisdom.weizmann.ac.il/∼vision/SpaceTimeActions.html



	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	5	81	0	0	0	88
Jump	0	0	5	67	0	0	0	72

Figure 6. *Clustering of activities in video. Above: One frame of each different activity. Below: Confusion matrices for the PMM and the R-PMM algorithm (*$\mathcal{D}_R$*,* $J = 4$*,* $k = 20$*), taking the 6 previous and 6 posterior frames as neighbors in* $\mathcal{D}_R$*, which results in a temporal regularization. Note the importance of regularization. We work with* 48×60 *dimensional vectors.*

elsewhere.

Acknowledgments This work has been supported by ONR, DARPA, NSF, NGA, and the McKnight Foundation. GH was a postdoctoral associate at Institute of Mathematics and its Applications, University of Minnesota, USA, while performing this work.

References

[1] C. Ambroise and G. Govaert. Clustering of spatial data by the EM algorithm. In *geoENV I - Geostatistics for Environmental Applications*, 1996.

[2] C. Ambroise and G. Govaert. Convergence of an EM-type algorithm for spatial clustering. *Pattern Recognition Letters*, 19(10):919–927, 1998.

[3] D. Barbara and P. Chen. Using the fractal dimension to cluster datasets. In *Proceedings of the Sixth ACM SIGKDD*, pages 260–264, 2000.

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in NIPS 14*, 2002.

[5] M. Brand. Charting a manifold. In *Advances in NIPS 16*, 2002.

[6] W. Cao and R. Haralick. Nonlinear manifold clustering by dimensionality. In *Proceedings of the 18th International Conference on Pattern Recognition*, volume 1, pages 920–924, 2006.

[7] J. A. Costa and A. O. Hero. Geodesic entropic graphs for dimension and entropy estimation in manifold learning. *IEEE Trans. on Signal Processing*, 52(8):2210–2221, 2004.

[8] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data. *Journal of the Royal Statistical Society Ser. B*, 39:1–38, 1977.

[9] A. Gionis, A. Hinneburg, S. Papadimitriu, and P. Tsparas. Dimension induced clustering. In *Proceeding of the Eleventh ACM SIGKDD*, pages 51–60, 2005.

[10] G. Haro, G. Randall, and G. Sapiro. Stratification learning: Detecting mixed density and dimensionality in high dimensional point clouds. In *Advances in NIPS 19*, 2006.

[11] R. Hathaway. Another interpretation of the EM algorithm for mixture distributions. *Statistics and Probability Letters*, 4(2):53–56, 1986.

[12] K. Huang, Y. Ma, and R. Vidal. Minimum effective dimension for mixtures of subspaces: A robust GPCA algorithm and its applications. In *Proceedings of CVPR*, pages 631–638, 2004.

[13] B. Kegl. Intrinsic dimension estimation using packing numbers. In *Advances in NIPS 14*, 2002.

[14] E. Levina and P. Bickel. Maximum likelihood estimation of intrinsic dimension. In *Advances in NIPS 17*, 2005.

[15] L. Lu and R. Vidal. Combined central and subspace clustering for computer vision applications. In *Proceedings of the 23rd International Conference on Machine Learning*, volume 148, pages 593–600, 2006.

[16] P. Mordohai and G. Medioni. Unsupervised dimensionality estimation and manifold learning in high-dimensional spaces by tensor voting. In *IJCAI*, page 798, 2005.

[17] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[18] D. L. Snyder. *Random Point Processes*. Wiley, New York, 1975.

[19] R. Souvenir and R. Pless. Manifold clustering. In *ICCV*, pages 648–653, 2005.

[20] F. Takens. On the numerical determination of the dimension of an attractor. *Lecture notes in mathematics. Dynamical systems and bifurcations*, 1125:99–106, 1985.

[21] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[22] R. Vidal, Y. Ma, and J. Piazzi. A new GPCA algorithm for clustering subspaces by fitting, differentiating and dividing polynomials. In *IEEE CVPR*, pages 769–775, 2004.