Translated Poisson Mixture Model for Stratification Learning

Gloria Haro · Gregory Randall · Guillermo Sapiro

Abstract A framework for the regularized and robust estimation of non-uniform dimensionality and density in high dimensional noisy 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 translated Poisson mixtures, with regularizing restrictions, leading to a model which includes the presence of noise. The translated Poisson distribution is useful to model a noisy counting process, and it is derived from the noise-induced translation of a regular Poisson distribution. By maximizing the log-likelihood of the process counting the points falling into a local ball, we estimate the local dimension and density. We show that the sequence of all possible local countings in a point cloud formed by samples of a stratification can be modeled by a mixture of different translated Poisson distributions, thus allowing the presence of mixed dimensionality and densities in the same data set. With this statistical model, the parameters which best describe the data, estimated via expectation maximization, divide the points in different classes according to both dimensionality and density, together with an estimation of these quantities for each class. Theoretical asymptotic results for the model are presented as well. The presentation of the theoretical framework is complemented with artificial and real examples showing the importance of regularized stratification learning in high dimensional data analysis in general and computer vision and image analysis in particular.

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., (Belkin and Niyogi 2002; Brand 2002; Costa and Hero 2004; Kegl 2002; Levina and Bickel 2005; Roweis and Saul 2000; Tenenbaum et al. 2000). 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 both for analysis and computation on the dataset.

Most 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 correct 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. In contrast to this, let us consider a set of images representing scanned digits. It might happen that

G. Haro (✉)
Dept. Teoria del Senyal i Comunicacions, Universitat Politècnica de Catalunya, Barcelona, Spain
e-mail: gloria@gps.tsc.upc.edu

G. Randall
Instituto de Ingeniería Eléctrica, Universidad de la República, Montevideo, Uruguay
e-mail: randall@fing.edu.uy

G. Sapiro
Dept. of Electrical and Computer Engineering, University of Minnesota, Minneapolis, USA
e-mail: guille@umn.edu

the images representing the digit '1' can be described with a different number of parameters than the images for the digit '2.' 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 (noisy) point cloud data. This is the subject of this work.

This problem, clustering-by-dimensionality and *stratification learning*, has recently been explored in a handful of works. Barbara and Chen (2000), propose a hard clustering technique based on the fractal dimension (box-counting). Starting from an initial clustering, they incrementally add points into the cluster for which the change in the fractal dimension, after adding the point, is the lowest. They also find the number of clusters and the intrinsic dimension of the underlying manifolds. Gionis et al. (2005), 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 (2005), use an Expectation Maximization (EM) type of technique, combined with weighted geodesic multidimensional scaling (weighted ISOMAP (Tenenbaum et al. 2000)). The weights measure how well each point fits the underlying manifold defined by the current set of points in the cluster. After clustering, each cluster dimensionality is estimated following (Levina and Bickel 2005). Huang et al. (2004), Vidal et al. (2004), cluster linear subspaces with an algebraic geometric method based on polynomial differentiation, called Generalized PCA (GPCA), which also finds the number of linear subspaces and their intrinsic dimensions. An algorithm for clustering linear manifolds based on lossy coding has been proposed by Ma et al. (2007). Goh and Vidal (2007), extend (Polito and Perona 2002) to cluster a union of J non-intersecting k-connected nonlinear manifolds. It is done with the vectors spanning the null space of the LLE matrix (Roweis and Saul 2000), which are a linear combination of the membership vectors and the embedding vectors of the J connected components. The work of Mordohai and Medioni (2005), estimates the local dimension using tensor voting. Cao and Haralick (2006), 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 (2006), combine GPCA with an additional spatial constraint in a k-means fashion. They show that, by adding this constraint, the classification is improved in the intersection of the linear subspaces. From the computational geometry perspective, a Voronoi-based technique

to compute local dimensionality has been introduced in Dey et al. (2003), and demonstrated for 3D point cloud data. The diffusion distance framework (Coifman and Lafon 2006; Lafon et al. 2006), can work with stratifications, though no explicit estimation of the clusters is obtained, and single maps into Euclidean space are performed for the whole data set. Recently, and following in part the theory of persistent topology (Edelsbrunner et al. 2002), a framework for studying stratas based on local homology has been introduced in Bendich et al. (2007).

These recent works have clearly shown the necessity to go beyond manifold learning, into "stratification learning." In our work, we do not assume subspaces are linear, and we simultaneously estimate the soft clustering and the intrinsic dimension and density of the clusters while being robust to noise and outliers. This collection of attributes is not shared by any of the pioneering works just described. Our approach is an extension of the Levina and Bickel's local dimension estimator (Levina and Bickel 2005). They proposed to compute the intrinsic dimension at each point using a Maximum Likelihood (ML) estimator based on a Poisson distribution. We propose to compute an ML on the whole point cloud data at the same time (and not one for each point independently), based on a translated Poisson mixture model, which models the presence of noise and permits to have different classes (each one with their own dimension and sampling density). This technique automatically gives a soft clustering according to dimensionality and density, with an estimation of both quantities for each class. A preliminary version of this work was presented in Haro et al. (2006) and a regularized version together with asymptotic results in Haro et al. (2007). These techniques are particular cases of the more general translated Poisson model introduced in this paper in order to handle noise.[1]

The remainder of this paper is organized as follows: In Sect. 2 we review the method proposed by Levina and Bickel (2005), which gives a local estimation of the intrinsic dimension and has inspired our work. We reformulate and extend this approach in Sect. 3 in order to include the presence of noise in the statistical model. Section 4 explains our approach for robust stratification learning. We show experiments with synthetic and real data in Sect. 5, including comparisons with critical literature. Finally, conclusions are presented in Sect. 6.

---

[1] We should mention that in Haro et al. (2006) we compared the original framework (with no regularization or noise modelling as here developed), with a two step approach, where we first estimate the local dimensionality per point using the original Levina-Bickel approach, and then cluster following the information bottleneck approach (Tishby et al. 1999). This has been shown not only to be less elegant and mathematically funded than the approach here presented, but mush less robust, even when compared to the non-regularized and noise-transparent formulation.

2 Local Intrinsic Dimension Estimation

Levina and Bickel (2005), 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 Takens (1985) in the context of dynamical systems. 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 $\frac{k}{T} \approx \rho(x_t) V(m) R_k(x_t)^m$. 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, $\rho(x_t)$ 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 $\rho(x_t)$ is considered constant inside the sphere, a valid assumption for small R. Note that the latter assumption is only local, the global density does not need to be constant, only inside the local sphere. With these assumptions, the rate λ of the counting process $N(R, x_t)$ can be written as

$$\lambda(R, x_t) = \rho(x_t) V(m) m R^{m-1}. \tag{1}$$

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 $\theta(x_t) := \log \rho(x_t)$ is the density parameter and the first integral is a Riemann-Stieltjes integral (Snyder 1975). 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 (Levina and Bickel 2005). In practice, it is more convenient to compute a fixed amount k of nearest neighbors. Thus, the local estimators at point x_t are

$$m(x_t) = \left[\frac{1}{k-1} \sum_{j=1}^{k-1} \log \frac{R_k(x_t)}{R_j(x_t)} \right]^{-1}, \tag{2}$$

$$\theta(x_t) = \log \left((k-1) / \left(V(m(x_t)) R_k(x_t)^{m(x_t)} \right) \right), \tag{3}$$

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 estimate the dimensionality for each cluster. Another possibility is to include this in the process via the simultaneous soft clustering and estimation technique described in Sect. 4. Before this, let us present the proposed framework to naturally handle noise as part of the model.

3 Translated Poisson Model

Usually, point samples are contaminated with noise, thus the point process that we observe is not a simple sampling of a low dimensional manifold but a perturbation of this sample process. This can be modeled with a translated Poisson process (Snyder and Miller 1991), where an underlying (unobservable) point process is translated to an output (observable) point process. The input and output spaces of the points are not necessarily the same or even of the same dimension (clearly, noise brings points outside of the underlying manifold and into the higher dimensional embedding space). More concretely, an input point at location x in the input space X is randomly translated to a location z in the output space Z, according to a conditional probability density $f(z|x)$, called the *transition density*.

For our purposes, we are going to consider the particular case where each point is translated independently of the others and there are no deletions or insertions in the translation process (these more general cases are also studied in Snyder and Miller 1991). We have the following critical theorem which states that a translated Poisson process is also a Poisson process:

Theorem (Snyder and Miller 1991) *Let $\{N(A) : A \subseteq X\}$ be a Poisson process with an integrable intensity function $\{\lambda(x) : x \in X\}$. Points of this input point process are translated to the output space Z to form the output point process $\{M(B) : B \subseteq Z\}$, where each point is independently translated according to the transition density $f(z|x)$. Then, if there are no insertions and deletions, $\{M(B) : B \subseteq Z\}$ is a Poisson process with intensity*

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

Since the intensity of the Poisson process in our model is parametrized by the Euclidean distances of the points (and not by the points themselves, see previous section), we are going to consider a random translation in the distances. This means that we do not observe the original distances but noisy distances. Let $f(s|r)$ be the transition density

which defines the random process which translates a distance r in the input space to a distance s in the observable space. If $\lambda(r, x_t)$, defined in (1), is the local rate of the Poisson process which defines the counting process in the input space, then $\mu(s)$, the intensity of the Poisson process in the output space is given by

$$\mu(s, x_t) = \int_0^{R'} f(s|r) e^\theta V(m) m r^{m-1} dr. \qquad (4)$$

R' is different from the radius R considered in the counting process $N(R, x_t)$. We consider $R' > R$ in (4) since points originally at distance greater than R from x_t can be placed within a distance less than R after the translation process. In practice, the maximum translation is small (just a perturbation because of the noise), and we consider $R' = R + \sigma$ in the particular case of a Gaussian transition density (11). The log-likelihood of the translated Poisson process is

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

The parameters of the maximum log-likelihood are obtained by solving the system of equations $\partial L/\partial m = 0$ and $\partial L/\partial \theta = 0$. We then obtain the following expression for m, when we use the k nearest neighbors (k-NN) instead of the points within distance less to R,

$$m(x_t)$$
$$= \left[\frac{1}{k-1} \sum_{i=1}^{k-1} \frac{\int_0^{R'} f(R_i(x_t)|r) r^{m-1} \log \frac{R_k(x_t)}{r} dr}{\int_0^{R'} f(R_i(x_t)|r) r^{m-1} dr} \right]^{-1}, \qquad (5)$$

where, by an abuse of notation, we have identified $m = m(x_t)$ in the right hand side. Note that this expression reduces to the Levina and Bickel estimator (Levina and Bickel 2005) in the particular case that $f(s|r) = \delta(s-r)$, i.e., there is no translation of the original points. This corresponds to the ideal case with no noise.

Equation (5) is a nonlinear recursive expression in m which is difficult to solve. Thus, we are going to approximate it by an easier to compute closed expression. Since the translation density is modeling the effect of noise, the effective support of $f(s|r)$ is going to be concentrated around s. Then, we can substitute r^{m-1} in (5) by its Taylor expansion around R_i. Let us write (5) in the following way

$$m(x_t) = I^{-1} = \left[\frac{1}{k-1} \sum_{i=1}^{k-1} I_i \right]^{-1}, \qquad (6)$$

and expand r^{m-1} in the integral I_i via its Taylor series

$$I_i := \frac{\int_0^{R'} f(R_i|r) r^{m-1} \log \frac{R_k(x_t)}{r} dr}{\int_0^{R'} f(R_i|r) r^{m-1} dr}$$
$$= \frac{\int_0^{R'} f(R_i|r) \log \frac{R_k(x_t)}{r} dr + \Delta I_{N_i} + \dots}{\int_0^{R'} f(R_i|r) dr + \Delta I_{D_i} + \dots} = \frac{I_{N_i}}{I_{D_i}},$$

where

$$\Delta I_{N_i} := (m-1) R_i^{-1} \int_0^{R'} f(R_i|r)(r - R_i) \log \frac{R_k(x_t)}{r} dr, \qquad (7)$$

and

$$\Delta I_{D_i} := (m-1) R_i^{-1} \int_0^{R'} f(R_i|r)(r - R_i) dr. \qquad (8)$$

These integrals are small since the effective support of $f(R_i|r)$ has the same order than the level of noise (considered not very large), and the quantity $(r - R_i)$ is small in the vicinity of R_i. We can then approximate

$$I_i \approx \frac{\int_0^{R'} f(R_i|r) \log \frac{R_k(x_t)}{r} dr}{\int_0^{R'} f(R_i|r) dr}. \qquad (9)$$

Notice that with this approximation of I_i, the estimator (6) still reduces to the noise-free Levina-Bickel estimator (2), that is $I_i = \log \frac{R_k}{R_i}$, when $f(R_i|r) = \delta(R_i - r)$. In the more general case, (9) is the expected value of $\log \frac{R_k}{r}$ according to the transition density $f(R_i|r)$ and thus reducing the effect of noise. Using the approximation (9) in (6) we obtain

$$m(x_t) \approx \left[\frac{1}{k-1} \sum_{i=1}^{k-1} \frac{\int_0^{R'} f(R_i|r) \log \frac{R_k}{r} dr}{\int_0^{R'} f(R_i|r) dr} \right]^{-1}. \qquad (10)$$

We explicitly estimate, in the following section, the error produced in $m(x_t)$ when we use the approximation (10) instead of (5), for the particular important case of a Gaussian transition density,

$$f(s|r) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(s-r)^2}{2\sigma^2}\right). \qquad (11)$$

In this particular case that the coordinates are perturbed by Gaussian noise, the error in the Euclidean distance can be approximated by a Gaussian as well (see Appendix A for more details). Thus, the expression for the local dimension estimator becomes

$$m(x_t) \approx \left[\frac{1}{k-1} \sum_{i=1}^{k-1} \frac{\int_0^{R'} \exp\left(-\frac{(R_i-r)^2}{2\sigma^2}\right) \log \frac{R_k}{r} dr}{\int_0^{R'} \exp\left(-\frac{(R_i-r)^2}{2\sigma^2}\right) dr} \right]^{-1}. \qquad (12)$$

3.1 Approximation Error for a Gaussian Translation Density

In order to estimate the error of approximating (5) by (10), we compute the integrals (7) and (8), which are the largest order error terms of the numerator and denominator, respectively, in the approximation of $m(x_t)$. For the integral (8), notice that the Gaussian is even with respect to R_i and that $(r - R_i)$ is odd. Then, (8) is zero if the effective support of the Gaussian is within the interval $[0, R']$, that is essentially if $R_i \in [3\sigma, R' - 3\sigma]$. If $R_i \in [0, 3\sigma] \cup [R' - 3\sigma, R']$, (8) is bounded by $4.5\sigma^2(m - 1)/R_i$. We will use this bound for ΔI_{D_i} independently of the value of R_i. Regarding the integral (7), we use the Taylor expansion of $(r - R_i) \log \frac{R_k}{r}$ around R_i,

$$(r - R_i) \log \frac{R_k}{r} = (r - R_i) \log \frac{R_k}{R_i} - \frac{(r - R_i)^2}{R_i} + \cdots.$$

Again, we consider the worst case scenario, $R_i \in [0, 3\sigma] \cup [R' - 3\sigma, R']$, and we obtain

$$\Delta I_{N_i} \leq 4.5\sigma^2 \frac{m - 1}{R_i} \log \frac{R_k}{R_i}.$$

We use these bounds and error propagation theory to obtain the relative error on I_i,

$$\frac{\Delta I_i}{I_i} = \frac{\Delta I_{N_i}}{I_{N_i}} + \frac{\Delta I_{D_i}}{I_{D_i}}$$
$$= 4.5\sigma^2 \frac{m - 1}{R_i} \left(\frac{1}{I_{N_i}} \log \frac{R_k}{R_i} + \frac{1}{I_{D_i}} \right),$$

and the relative error on $m_k(x_t)$,

$$\frac{\Delta m(x_t)}{m(x_t)} = \frac{\Delta I}{I} = \frac{1}{I(k - 1)} \sum_i \Delta I_i,$$

which is bounded by

$$\frac{\Delta m(x_t)}{m(x_t)} \leq \frac{4.5\sigma^2(m(x_t) - 1)}{\min_i(R_i \tilde{R}_i^{m(x_t)-1})} \left(1 + \frac{m(x_t)}{m(x_t, \sigma = 0)} \right), \quad (13)$$

where $m(x_t, \sigma = 0)$ is (2), or equivalently, (5) with $\sigma = 0$, and $\tilde{R}_i^{m-1} = I_{D_i} = \int_0^{R'} f(R_i|r) r^{m-1} dr$. This provides a bound on the error of the approximation for the important case of Gaussian noise. Similar computations can be performed for other translation density (noise models). In the case of $\sigma = 0$ (no noise), the approximation error $\Delta m(x_t)$ is zero, as expected. If we consider $\tilde{R}_i \approx R_i$, the bound (13) is inversely proportional to the signal to noise ratio and proportional to $(m - 1)/R_i^{m-2}$, which is a decreasing function of the dimension m for $R_i > 1$. Note that the estimator $m(x_t)$, defined in (5), is invariant to distance rescalings so we can always ensure $R_i > 1$.

4 Dimensionality and Density Estimation with Simultaneous Soft Clustering

Having introduced the critical translational Poisson model, we are now ready to introduce the mixture of these models to address the problem of stratification learning for noisy point cloud data. We start with the basic model, and then introduce a regularization term. We conclude the presentation providing asymptotic results.

4.1 Translation Poisson Mixture Model (TPMM)

In Haro et al. (2006), we proposed to study stratifications 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 the whole point cloud becomes a mixture of Poisson distributions with different parameters (dimension and density) in each class. This allows for the presence of different intrinsic dimensions and densities in the dataset. These are automatically computed while being used for soft clustering. We extend this approach here to the more general case when we have mixtures of translated Poisson processes (thereby handling the noise).

Let us consider J different translated 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 ($\rho^j = e^{\theta^j}$), and m^j is its dimension. While in the Levina-Bickel approach the density is assumed locally constant (inside a ball), here the density is assumed constant inside a class (a single Poisson distribution defines each class). However, if there is a class with different densities the algorithm will cluster also according to density (not only dimension) and a single manifold will be represented by clusters of the same dimension but different densities. An example of that is shown in Fig. 5. If the number of classes is not sufficient to represent the dimension and density variability, the algorithm will give one or more classes with a dimension and/or density which are the (weighted) average of the actual features within the class. This is the standard result for under-clustering. On the other hand, we have experimentally observed that giving extra classes is reasonable robust, since the extra classes end-up being empty or identical to other classes in terms of parameters values.

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, and we use the translated Poisson model introduced in the previous section, we have

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

where $\lambda^j(r) = e^{\theta^j} V(m^j) m^j r^{m^j-1}$ and $\mu^j(s) = \int_0^{R'} f(s|r) e^{\theta^j} V(m^j) m^j r^{m^j-1} dr$. 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

$$L(Y|\psi) = \log p(Y|\psi) = \sum_{t=1}^{T} \log p(y_t|\psi). \qquad (14)$$

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

$$\log p(Z, Y|\psi) = \sum_{t=1}^{T} \sum_{j=1}^{J} \delta_t^j \log \left[p(y_t|\psi^j) \pi^j \right], \qquad (15)$$

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

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

The unknown parameters in (15) 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 (Dempster et al. 1977; Kung et al. 2004, Chap. 3). The EM is based on the following decomposition of the log-likelihood (14):

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

$$- \sum_{t=1}^{T} \sum_{j=1}^{J} h^j(y_t) \log \left[h^j(y_t) \right], \qquad (16)$$

where $H = \{h^j(y_t) \le 1; t = 1, \ldots, T, j = 1, \ldots, J\}$ and $h^j(y_t)$ is the probability that observation t belongs to mixture j: $h^j(y_t) = E_Z[\delta_t^j|y_t, \psi] = P(\delta_t^j = 1|y_t, \psi)$, where $E_Z(\cdot)$ is the expectation with respect to Z. Since the membership functions are indicator variables, the first term in (16) is the expectation of (15) 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 Hathaway (1986), where the EM is seen as an alternate optimization algorithm of the log-likelihood (16). 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

$$h_{n+1}^j(y_t) = \frac{p(y_t|m_n^j, \theta_n^j) \pi_n^j}{\sum_{l=1}^{J} p(y_t|m_n^l, \theta_n^l) \pi_n^l}. \qquad (17)$$

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 (21)–(23) for the variables at step $n + 1$. In order to compute m_{n+1}^j we have used the same approach as in Levina and Bickel (2005), by means of a k nearest neighbor graph. The TPMM approach just described is summarized in **R-TPMM Algorithm** below, for the particular case of $\alpha = 0$ (no regularization, see below).

4.2 Regularized TPMM

The TPMM algorithm seeks a soft clustering according to dimensionality and density, considering noise in the data, but does not (explicitly) take into account spatial information. Adding regularization is the goal of this section. Regularization further helps to improve the classification in noisy data and points lying close to manifold boundaries (see results in Figs. 1 and 2). This regularization is inspired in part by the work in Ambroise and Govaert (1996) 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 (Hathaway 1986). In our context, we complete (16) with a spatial term $S(H)$,

$$F(\psi, H) = L(Y|\psi, H) + \alpha S(H), \qquad (18)$$
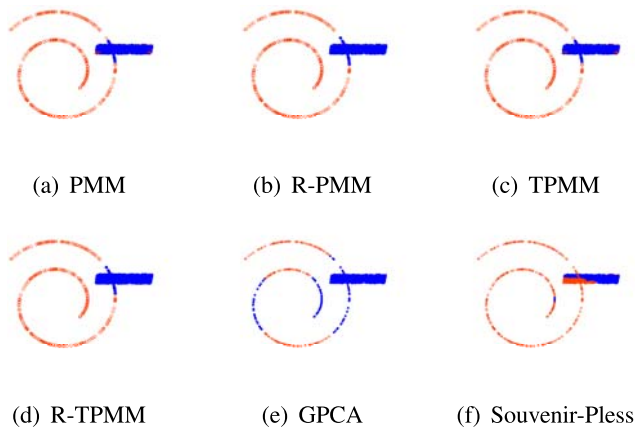
Fig. 1 Clustering of a spiral and a plane. Results with different algorithms (this is a color figure)
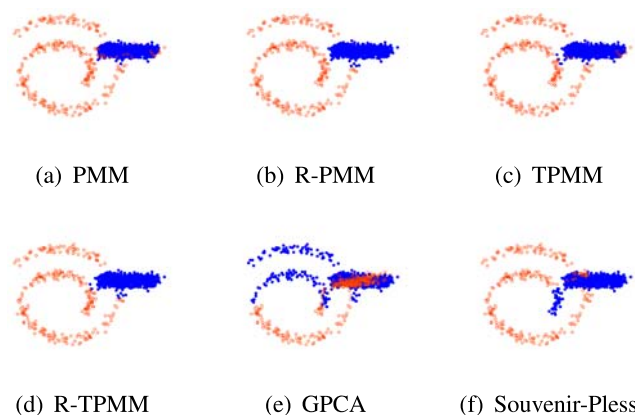

Fig. 2 Clustering of a spiral and a plane with noise. Results with different algorithms (this is a color figure)

where α is a parameter that controls the trade-off between the spatial term and the likelihood. Its value is also related to the amount of noise in the data.[2] Then, F is maximized with an alternate optimization technique. Since the new term, S, only depends on H, the optimization procedure results in an 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)

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

where $l\sim t$ indicates that there is a neighborhood relationship between observations l and t. By maximizing this term,

[2]The study of the possible connection between the regularization factor α and the level of noise and the translation density in the translated Poisson model is an interesting subject of future research. Note that this regularization is important beyond the noise, e.g., at the manifolds boundaries, see experimental results.

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,

$$S(H) = -\sum_{t=1}^{T}\sum_{j=1}^{J} h^j(y_t)\mathcal{D}(t,j,X,H). \quad (19)$$

The expression (19) provides a generic framework for introducing constraints in the soft classification, besides the ones already present in the TPMM model, namely dimensionality and density. One possibility, as in the NEM algorithm, is to introduce spatial regularity. Then, as dissimilarity measure we use $\mathcal{D} = \mathcal{D}_R$ defined as

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

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 Fig. 10 and Table 5).

As noted in Ambroise and Govaert (1996), the EM algorithm with additional constraints can be seen as finding the Gibbs distribution with energy $-F(\psi, H)$. In the particular case when the additional constraint is neighborhood dependent, $S_{\text{NEM}}(H)$ and $S(H)$ with $\mathcal{D}_R$, the Gibbs distribution defines a Markov Random Field.

The maximization of F (18), is obtained as in (Ambroise and Govaert 1996), with an alternate optimization technique which results in an EM-type algorithm. Maximizing (18) with respect to H, with $S(H)$ defined in (19)—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:

$$h^j(y_t) = \frac{p(y_t|m^j,\theta^j)\pi_n^j e^{-\alpha\mathcal{D}'(t,j,X,H)}}{\sum_{l=1}^{J} p(y_t|m^l,\theta^l)\pi^l e^{-\alpha\mathcal{D}'(t,l,X,H)}}, \quad (20)$$

where $\mathcal{D}'(t,l,X,H) = \sum_{l\sim t}(1 - 2h^j(y_l))$ in the particular case we are interested, $\mathcal{D} = \mathcal{D}_R$, and assuming that $l\sim t$ implies $t\sim l$. Since the only term in (18) 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 TPMM algorithm. The regularized version of the TPMM algorithm is summarized in the **R-TPMM Algorithm** below (Regularized Translated Poisson Mixture Model Algorithm).

The EM suffers from local maxima, this can be alleviated running the algorithm several times with different initializations. In particular, we add to the EM iterations an extra loop where the parameters m^j and θ^j of each class are reinitialized every odd iteration and π^j every even iteration.

R-TPMM Algorithm

REQUIRE: The point cloud data, J (number of desired classes), k (scale of observation), α (regularization parameter), and σ (noise level or full noise/translation function f).

ENSURE: Regularized soft clustering according to dimensionality and density.

1. Compute the local estimators

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

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

In particular, we use the definition of f given in (11).

2. Initialize $\psi_0 = \{\pi_0^j, m_0^j, \theta_0^j\}$ and $\bar\psi_0 = \{\bar\pi_0^j, \bar m_0^j, \bar\theta_0^j\}$ to any set of values which ensures that $\sum_j \pi_0^j = \sum_j \bar\pi_0^j = 1$ and $\bar H_0 = \{\bar h_0^j(y_t) = 1/J; \ j = 1, \ldots, J, \ t = 1, \ldots, T\}$.

3. Iterations on l,

   3A. If l is odd
   Set $\bar m_l^j = m_0^j$ and $\bar\theta_l^j = \theta_0^j$, for all $j = 1, \ldots, J$.
   Else
   Set $\bar\pi_l^j = 1/J$, for all $j = 1, \ldots, J$.

   3B. Iterations on n,
   For all $j = 1, \ldots, J$:
   3B.1: Compute, for all $t = 1, \ldots, T$,

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

   where $H_n = \{h_n^j(y_t); \ j = 1, \ldots, J, \ t = 1, \ldots, T\}$.
   3B.2: Compute

   $$\pi_{n+1}^j = \frac{1}{T} \sum_{t=1}^T h_n^j(y_t) \tag{21}$$

   $$m_{n+1}^j = \left[\sum_{t=1}^T h_n^j(y_t) m(x_t)^{-1} / \sum_{t=1}^T h_n^j(y_t) \right]^{-1} \tag{22}$$

$$\rho_{n+1}^j = e^{\theta_{n+1}^j}$$
$$= \left[\sum_{t=1}^T h_n^j(y_t) f(x_t)^{-1} / \sum_{t=1}^T h_n^j(y_t) \right]^{-1} \tag{23}$$

where $\rho(x_t) = e^{\theta(x_t)}$.

   Until convergence of ψ_n, that is, when $\|\psi_{n+1} - \psi_n\|_2 < \epsilon$, for a certain small value ϵ.
   Set $\bar\psi_{l+1} = \psi_n$ and $\bar H_{l+1} = H_n$.
   Until $\|\bar\psi_{l+1} - \bar\psi_l\|_2 < \epsilon$, $\|\bar H_{l+1} - \bar H_l\|_2 < \epsilon$ or $l = l_{\max}$.[3]

Remark 1 The PMM and R-PMM algorithms introduced respectively in Haro et al. (2006, 2007) are particular cases of the parameters α (regularization) and σ (noise) in the R-TPMM algorithm. Let us introduce the following notation for the particular cases of these parameters:

- PMM: $\alpha = 0$ and $\sigma = 0$.
- R-PMM: $\alpha > 0$ and $\sigma = 0$.
- TPMM: $\alpha = 0$ and $\sigma > 0$.
- R-TPMM: $\alpha > 0$ and $\sigma > 0$.

We will use the above notation in the experiments in Sect. 5.

Remark 2 Notice that the estimators (22)–(23) in the PMM and R-PMM approaches ($\sigma = 0$) are weighted harmonic means of the local estimators (2)–(3) of Levina-Bickel. The weight at each point is the probability of the membership function, h. In the particular case of a unique class, $J = 1$, we obtain the global dimension estimator proposed by MacKay and Ghahramani (http://www.inference.phy.cam.ac.uk/mackay/dimension/), a particular case of our proposed framework.

Remark 3 We are using the same level of noise σ for all the clusters. A better approach might be to use a σ suitable for each class. Although computationally speaking it will be more demanding, since we would have to recompute the local estimators $m(x_t)$ and $\theta(x_t)$ at each iteration with the σ of the assigned class. Moreover, the different σ would have to be estimated (this can be done for example as the value of σ which minimizes the estimated dimension in each class).

As proved in Ambroise and Govaert (1998), if α is small enough, (18) has a guaranteed global maximum for a fixed value of ψ, and the additional term $S(H)$ does not affect the convergence of the EM-type algorithm. It can be shown (see Appendix B) that, for the case of $\mathcal{D}_R$, the corresponding bound on α is $\alpha_R < 1/(2k)$.

---

[3]In the experiments we use $l_{\max} = 10$.

Using the same analysis as in Sect. 3.1 we find that the relative error produced in (22) by using the approximation (10) for $m(x_t)$ is

$$\frac{\Delta m^j}{m^j} \le \frac{4.5\sigma^2(m^j - 1)}{\min_{i,t}(R_i(y_t)\tilde{R}_i(y_t)^{m^j - 1})}\left(1 + \frac{m^j}{m^j(\sigma = 0)}\right),$$

where $m^j(\sigma = 0)$ is (22) with $\sigma = 0$, and $\tilde{R}_i(y_t)^{m^j - 1} = I_{D_i}(y_t)$.

4.3 Asymptotic Analysis

Levina and Bickel (2005) show 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 (2) (with $k - 2$ instead of $k - 1$ in the denominator) are

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

where m_T is the actual dimension. We can apply the same type of analysis to our model in the particular case of hard clustering, that is

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

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

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

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

where m_T^j is the correct intrinsic dimension of class j and N_j is the amount of points classified as class j. See Appendix C for the details of the proof. This result shows that the dimension estimator of each class is more biased when the intrinsic dimension increases. On the other hand, when there are more points in a class (N_j is larger), the bias is reduced. It is reduced also by considering more nearest neighbors, although there is a compromise for this value since increasing k affects the underlying hypothesis of constant density inside the ball. We have verified this result experimentally, and we found that the bias results in a underestimation of the intrinsic dimension (this behavior was also observed in the Levina-Bickel estimator (Levina and Bickel 2005)), and that it depends on the intrinsic dimension but not on the ambient dimension. We have also experimentally observed that a possible bias in the estimated dimension does not affect the clustering, unless the bias makes the estimated dimension of

one class to be close to any of the other clusters estimated dimensions.

The analysis of the density estimator θ^j is the subject of current research, as it is the study of the asymptotic behavior for the full soft clustering model.

5 Experimental Results

We now present experimental results with synthetic and real data for the proposed R-TPMM and its variants. We also compare some of the results with the ones obtained with GPCA (Vidal et al. 2003) and the Souvenir and Pless (2005) algorithms. We fixed α and σ experimentally. For α we usually use values in the interval $[0, 5]$. As for the case of σ we use a value in the order of the mean distance to the first neighbor: $\sigma = \nu \bar{R}_1$, where $\bar{R}_1 = \frac{1}{N}\sum_t R_1(x_t)$ and $0 \le \nu \le 1$. In the experiments with real data—digits, faces, video activities, and motion—we use the following values for ν: 0.4, 0.4, 0.25, and 1 respectively. In the first (artificial data) experiment, since we know the level of noise in the point coordinates, we use the estimated σ as computed in Appendix A. The only parameter in GPCA is the number of clusters. In the Souvenir-Pless algorithm the input parameters are the number of nearest neighbors and the dimension of each cluster. We also fixed these parameters experimentally in order to obtain the best classification results.

5.1 Synthetic Data

First, we work with a point cloud formed by 300 samples of a spiral and 800 of a plane, both in 3D embedding space. We compare the following algorithms: PMM, R-PMM, TPMM, R-TPMM, GPCA, and Souvenir-Pless. Figure 1 shows, for each algorithm, the point cloud with each point colored and marked differently according to its classification. In the different versions of our proposed algorithm we set $k = 30$, $J = 2$, $\alpha = 0.75$, and $\sigma = 0.1$. We test TPMM and R-TPMM with a small value of σ different than zero even if there is no noise just to show that a small error in the estimation of σ does not significantly affect the result. Notice that the regularized versions of our proposed algorithm improve the classification at the boundaries. In the Souvenir-Pless algorithm we use $k = 10$ and dimensions 2 and 2 (it gives a better result than using the actual dimensions, 2 and 1, as parameters). The GPCA algorithm does not give good results because it is designed for linear manifolds. Table 1 contains quantitative results of the different versions of our algorithm. Our approach gives some errors at the intersection of the two manifolds. This is due to the fact that a point in the intersection has points of the other manifold as some of its nearest neighbors. Thus, the extent of the classification

Table 1 Estimated parameters, dimension m and density $\rho = e^\theta$, in each class (C1 and C2), and clustering results of a plane and a spiral (denoted by Pl. and Sp. respectively). The four algorithms use $k = 30$ and $J = 2$

	PMM		R-PMM		TPMM		R-TPMM	
	C1	C2	C1	C2	C1	C2	C1	C2
	Estimated parameters for each class							
m	1.90	1.02	1.90	0.99	1.87	1.03	1.87	1.00
θ	1.01	1.10	0.99	1.14	1.05	1.09	1.02	1.12
	Number of points in each class							
Pl.	787	13	800	0	788	12	800	0
Sp.	21	279	22	278	21	279	23	277

Table 2 Estimated parameters, dimension m and density $\rho = e^\theta$, in each class (C1 and C2), and clustering results of a plane and a spiral with noise (denoted by Pl. and Sp. respectively). The four algorithms use $k = 40$ and $J = 2$

	PMM		R-PMM		TPMM		R-TPMM	
	C1	C2	C1	C2	C1	C2	C1	C2
	Estimated parameters for each class							
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
	Number of points in each class							
Pl.	764	36	800	0	784	16	800	0
Sp.	22	278	25	275	27	273	29	271

(a) R-TPMM   (b) GPCA   (c) Souvenir-Pless

Fig. 3 Clustering of a sphere and a curve with R-TPMM ($k = 20$, $J = 2$, $\alpha = 0.1$, and $\sigma = 0$), GPCA and Souvenir-Pless ($k = 20$, dimensions 2 and 1). R-TPMM works well in non-Euclidean manifolds. The estimated dimensions for each cluster in (a) are 0.98 and 2.05 for clusters in green and red respectively (this is a color figure)

(a) 2.5% outliers   (b) 2.5% outliers   (c) 25% outliers

(d) 50% outliers   (e) 75% outliers   (f) 100% outliers

Fig. 4 R-TPMM clustering of a spiral and a plane with different amount of added outliers, $k = 30$, $\alpha = 0.1$, and $\sigma = 0.1$. Example (a) is when considering two classes and the rest with three classes (this is a color figure)

errors in the intersection depends on the amount of k nearest neighbors considered. Note that the GPCA handles intersections well when working with linear manifolds (Huang et al. 2004). Addressing this problem is part of the ongoing efforts in the extensions of the proposed stratification learning framework.

Next, we added Gaussian noise, with standard deviation 0.66, to the point coordinates. Then, if we approximate the transition density with a Gaussian (see Appendix A), we use the estimated standard deviation $\sigma = 0.66\sqrt{2} = 0.93$. The rest of the parameters we use are $k = 40$, $J = 2$, $\alpha = 1$, and for Souvenir-Pless, $k = 20$ and dimensions 2 and 2. The qualitative comparison of the different algorithms can be seen in Fig. 2. Again, notice how the classification of the points at the manifold boundaries is better in the regularized versions. Table 2 contains the quantitative results for the different variants of the proposed algorithm. In particular, it can be seen that the translated versions give an estimation for the dimension m less sensible to noise.

Due to the statistical nature of the R-TPMM approach, it is not restricted to linear manifolds, such as GPCA (Vidal et al. 2004), nor to Euclidean manifolds, such as Isomap
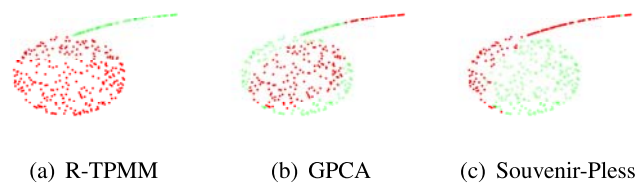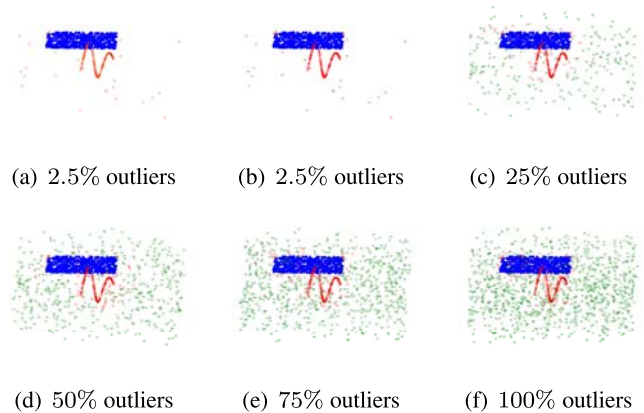
(Tenenbaum et al. 2000), on which the Souvenir-Pless technique (Souvenir and Pless 2005) is based on. This is manifested in Fig. 3, where we cluster a sphere and a curved line and compare the results of the R-TPMM, GPCA and Souvenir-Pless. The R-TPMM gives a 100% accurate clustering and the estimated dimensions are 0.98 for the line and 2.05 for the sphere.

Robustness to outliers has been studied for clustering linear manifolds. In Yang et al. (2006) a robust GPCA algorithm is proposed. A segmentation of linear subspaces based on information theory is proposed in Ma et al. (2007), and it has been shown to be robust to outliers. In order to see how the R-TPMM performs in the presence of outliers, we have added outliers to a set of points sampling a spiral and a plane. The original point coordinates are within the intervals $[-11, 21]$, $[5, 25]$, and $[-11, 14]$. The outliers follow a uniform distribution within the intervals $[-30, 30]$, $[-15, 35]$, and $[-30, 30]$. We use $\alpha = 0.1$ and $\sigma = 0.1$. Figure 4 shows the classification results for different amounts of outliers and number of classes: (a) $J = 2$, 2.5% outliers; (b) $J = 3$, 2.5% outliers; (c) $J = 3$, 50% outliers; (d) $J = 3$, 50% outliers; (e) $J = 3$, 75% outliers; (f) $J = 3$, 100% outliers. In the experiment (a) we set two classes and we obtain a class formed by the spiral and the outliers with an

Fig. 5 Clustering of a Swiss roll and a line with two different densities with R-TPMM, $k = 20$, $J = 3$, $\alpha = 2$, and $\sigma = 0$, no noise (this is a color figure)

(a) $\alpha = 0$ (b) $\alpha = 0.25$ (c) $\alpha = 1$
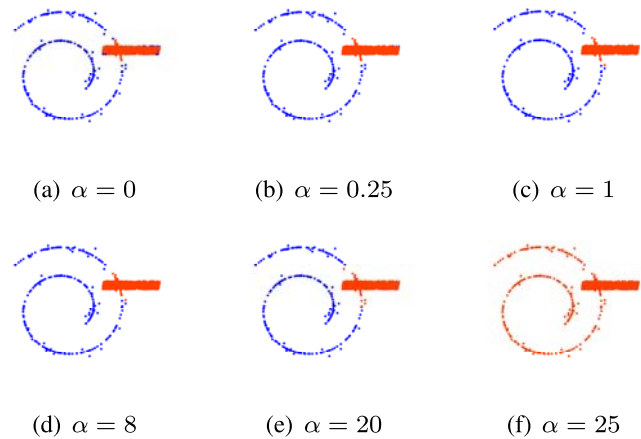
(d) $\alpha = 8$ (e) $\alpha = 20$ (f) $\alpha = 25$

Fig. 6 Clustering of a 1D spiral and a 2D plane (R-PMM, $k = 30$, $J = 2$). Evolution of the classification as the regularization parameter α increases

estimated dimension of 1.10, the second class is the plane with an estimated dimension of 1.87. Note that the estimation of the embedding dimensions are not affected by the outliers when its percentage is relatively low and we do not allow an extra class for the outliers. If we set three classes, experiments (b)–(f), the outliers are clustered as a separate class, and the only errors are at the intersections (due to points whose nearest neighbors actually belong to different classes). The dimensions obtained in each experiment are the following: (b) 1.06, 1.87, and 10.46; (c) 1.17, 1.88, and 3.32; (d) 1.23, 1.87, and 2.92; (e) 1.29, 1.83, and 2.97; (f) 1.28, 1.81, and 2.92. Independently of the amount of outliers, the algorithm is able to cluster apart the spiral and the plane, and the correct estimation of their embedding dimensions is not affected by the outliers. When the amount of outliers is small, 2.5% in (c), the estimated dimension for the class 'outliers' is very large due to the small amount of points belonging to this class, and we do not have enough samples of the class 'outlier' in each ball (there are mixed samples from the spiral and/or the plane). In these balls, the assumption of approximate constant density is not satisfied either. When the amount of outliers is larger, their estimated dimension is the same as the ambient dimension, since there is no intrinsic structure for these points.

The experiment in Fig. 5 illustrates how the soft clustering is done according to both dimensionality and density. The data consists of 2000 points on the Swiss roll, 400 on a line with high density and 50 on another less dense line. We have set $J = 3$ and the algorithm clusters the line in two different classes, according to the different densities. The estimated dimensions are: 1.98, 1.02 and 0.99. And the estimated densities: 0.49, 0.53 and 6.89 respectively.

Before presenting results on real data, we show how the regularization parameter α affects the classification. Figure 6 shows the evolution, according to α, of the classification of the spiral and the plane by the R-PMM with $\mathcal{D}_R$. We have perturbed 50 randomly picked samples of the spiral by a Gaussian noise of standard deviation $\sigma = 0.66$. It can be observed that a small amount of regularization helps in the classification, while when α increases, it produces a larger diffusion of the labelling, resulting in an inaccurate classification. This is due to the fact that the regularization component gains more importance than the log-likelihood term. In the limit, when α is quite large, the optimal solution is a single class. Of course, the "optimal" value of α, which marks these transitions, depends on each particular experiment (this is common in MRF-type approaches, and

the study of techniques from there to automatically compute this regularization parameter is an interesting open problem). For the experiments in this paper, we have worked with values of α within the interval $[0, 5]$. In Haro et al. (2007) we also tested the evolution of the classification with respect to α with another term $\mathcal{D}_C$ in the additional constraint $S(H)$, imposing spatial compactness within a class. Again, the dimension/density criterion for classification was more penalized against the extra term $S(H)$ for larger values of α and thus yielding rather a k-means kind of clustering (note that within the context of GPCA, Lu and Vidal (2006) also proposed a combination of k-means and dimensionality clustering).

5.2 Real Data

As a test of the performance with real data, we first work with the MNIST database of handwritten digits,[4] 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 two, some examples of those scanned digits as well as the clustering results are presented in Fig. 7. Observe how the classification improves adding regularization and including the noise in the model (translated Poisson). We have used R-PMM with $\alpha = 4$, TPMM with $\sigma = 1.5$, and R-TPMM with $\alpha = 1$ and $\sigma = 1.5$. Levina-Bickel's technique gives a dimension value of 11.26 and Costa-Hero's 9. These methods give a dimension in between the two different dimensions present in the point cloud. With the R-TPMM algorithm (and its variants), we are able to separate the points (images) corresponding to each digit and handle the noise and regularization. Both sets of digits have different dimensionality and density. We have

---

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

	PMM		R-PMM		TPMM		R-TPMM	
	C1	C2	C1	C2	C1	C2	C1	C2
	Estimated parameters for each class							
m	7.33	12.79	7.36	12.95	2.86	7.14	2.88	7.24
θ	−7.38	−23.99	−7.67	−23.26	−1.52	−12.70	−1.62	−12.90
	Number of points in each class							
'1'	1032	0	1032	0	1032	0	1029	3
'2'	70	1065	43	1092	36	1099	17	1118

Fig. 7 Clustering of scanned digits '1' and '2.' Some examples of digits and table with estimated parameters, dimension m and density $\rho = e^{\theta}$, for each class (C1 and C2), and clustering results for different variants of the R-TPMM algorithms with $J = 2$, $k = 30$ (since the density is $\rho = e^{\theta}$, $\rho \geq 0$ for $\theta \in \mathbb{R}$)

observed that some other digits do have the same dimensionality, as expected. Observe in the Table of Fig. 7 how the dimension is reduced with the (R-)TPMM, these values are much closer (than the ones with (R-)PMM) to the dimension obtained with Isomap, see graph in Fig. 8, separately applied to each one of the digits. The fact that the dimension is reduced when considering the translated process indicates that the high dimensions were originally due to the noise (this can be also inferred by observing the Isomap eigenvalues in Fig. 8).
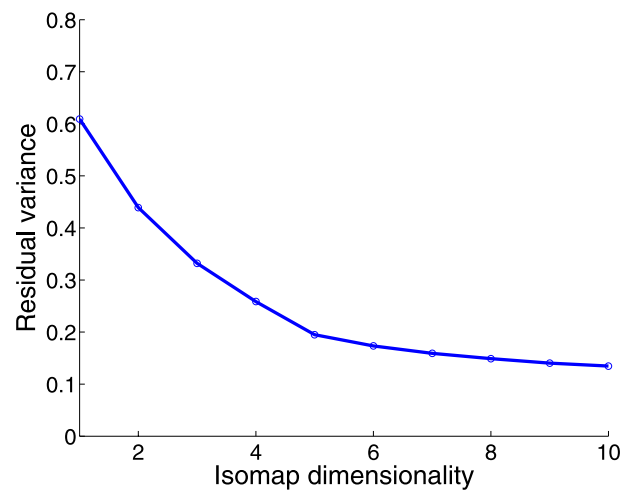
We also analyze images from the Yale Face Database B,[5] which contains images of 10 subjects under 585 viewing conditions (9 poses and 65 illumination conditions), see Fig. 9. Each image has a size of 640×480 pixels. For computational reasons we subsampled the images by a factor of ten and use each 64×48 image as a vector in a high dimensional space. 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 only in the first pose and under varying illuminations. The estimated dimensions and confusion matrices using the PMM and R-TPMM algorithms with $\alpha = 0.25$ and $\sigma = 1$ are presented in Table 3. 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 classification results are improved using regularization and the translated Poisson model. Observe also that changing the number of k nearest neighbors does not significantly change the results. Table 4 contains the confusion matrix obtained with the GPCA and the Souvenir-Pless algorithms. These algorithms are computed with a pre-projection of the data onto a 5-dimensional space.[6] This is necessary in the GPCA because, although not being an iterative algorithm, it consumes a lot of time in

(a) Digit '1'

(b) Digit '2'

Fig. 8 Isomap dimensionality of Digits one and two. The graph shows the residual variance of the first ten Isomap embedding dimensions

---

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

[6] We compute the SVD of the matrix data $I = U\Sigma V^{T}$ and consider the matrix formed by the first 5 columns of V^{T} as the embedded data. In GPCA we further use homogeneous coordinates (Vidal et al. 2004).

high dimensional spaces. For the Souvenir-Pless algorithm this point is not so critical but we obtained better classification results in the reduced dimensionality space. However,

Fig. 9 Examples of images of subjects 5 and 6 of the Yale Face Database B. See results in Table 3

Table 3 Dimension *m* for each class (C1 and C2) and clustering results of the mixture of subject 5 (all poses, all illuminations) and subject 6 (one pose, all illuminations) in the Yale Face Database B. PMM and R-TPMM ($\alpha = 0.25$, $\sigma = 1$) algorithms with two different values of *k*. The algorithms are applied in the 64×48 dimensional space

	PMM				R-TPMM			
	$k = 35$		$k = 50$		$k = 35$		$k = 50$	
	C1	C2	C1	C2	C1	C2	C1	C2
Estimated dimension for each class								
m	4.10	2.94	4.37	2.79	3.34	2.59	3.60	2.55
Number of points in each class								
S. 5	569	16	575	10	584	1	584	1
S. 6	0	65	0	65	0	65	0	65

Table 4 Clustering results of the mixture of subject 5 (all poses, all illuminations) and subject 6 (one pose, all illuminations) in the Yale Face Database B. We apply GPCA and Souvenir-Pless algorithms to the data pre-projected onto a 5 dimensional space

	GPCA		Souvenir-Pless	
	C1	C2	C1	C2
Number of points in each class				
Subject 5	325	260	476	109
Subject 6	0	65	20	45

with the proposed R-TPMM, we obtain better results in the original space.

It must be clarified that the R-TPMM is able to separate both subjects because their corresponding images lie in manifolds of different dimensions. However, if we consider, for example, a fixed pose under varying illuminations, in both subjects, all the points are classified in the same class since both manifolds have the same dimension (complexity). In this particular case, we tested the GPCA algorithm and it gives a 100% accurate classification.

The R-TPMM framework is also tested to study different human activities in video. We created a point cloud with the frames of a video of a person performing four different activities, with a static background: Standing, walking, jump-
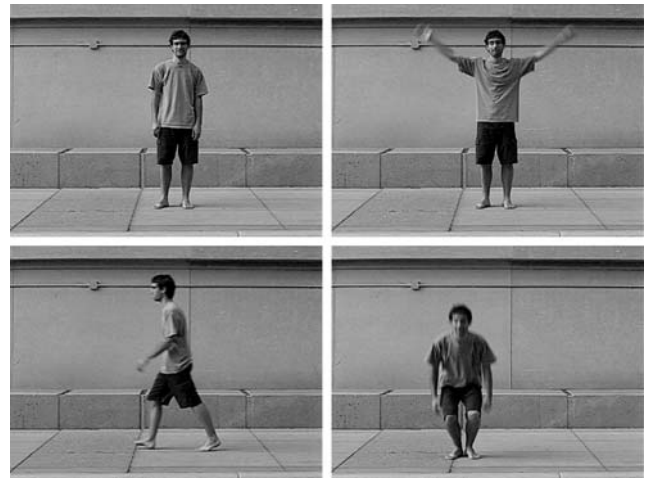

Fig. 10 Four sample frames of human activities in video

Table 5 Classifying human activities in video with the R-TPMM algorithm ($k = 10$, $\alpha = 40$, and $\sigma = 0.25$). We use the 6 previous and 6 posterior frames as neighbors in $\mathcal{D}_R$, which results in a temporal regularization. The global classification is 96% accurate

	Number of samples in each cluster			
	C1	C2	C3	C4
Standing	505	0	6	0
Walking	0	464	45	14
Waving	1	0	430	0
Jumping	0	0	0	207

ing, and arms waving. Each original frame is 480×640, sub-sampled to 48×64 pixels, with 1673 frames (see some frame examples in Fig. 10). This is mainly to speed up computations. 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. The confusion matrix with the classification results using the R-TPMM algorithm (with $k = 10$, $\alpha = 5$ and $\sigma = 0.25$) is presented in Table 5. The error in the classification affects only 4% of the frames.

Finally, we tested the R-TPMM algorithm in a motion segmentation application. We use a sequence of the Kanatani Lab,[7] see some examples of frames in Fig. 11. This sequence was originally used in Kanatani and Sugaya (2003) and then in Vidal et al. (2004). The data consists of the 2D projection coordinates of the trajectories along the sequence of some interest points. The sequence that we analyze corresponds to a car moving in a parking lot, and there are two different motions in the sequence. As in Vidal et al. (2004), we pre-project the data, originally in a 60-dimensional space
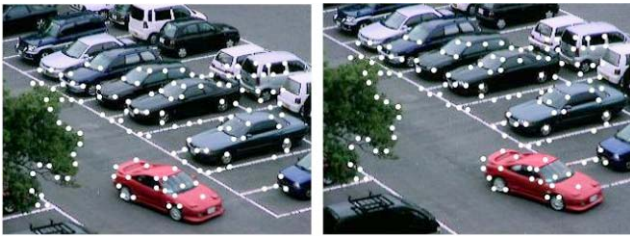
---

[7]http://www.suri.it.okayama-u.ac.jp/data.html

Fig. 11 Two frames of a sequence of the motion segmentation database of the Kanatani Laboratory

Table 6 Classification rates, using different methods, for the motion segmentation in the Kanatani Laboratory sequence (see example frames in Fig. 11)

Method	Effectiveness
Costeira-Kanade	60.3%
Ichimura	92.6%
Kanatani-Sugaya	100%
Souvenir-Press	93.38%
GPCA	100%
RTPMM	100%

(2 coordinates × 30 frames), onto a 5-dimensional space. In Table 6 we show the classification effectiveness for different methods: Costeira-Kanade, Ichimura, Kanatani-Sugaya (the three of them reported in Kanatani and Sugaya 2003), Souvenir-Press, GPCA and R-TPMM. For the R-TPMM we use $k = 10$, $\alpha = 2$ and $\sigma = 0.05$. We also tested our algorithm with the other two sequences used in Kanatani and Sugaya (2003), Vidal et al. (2004), and obtained a single class since the two different motions have the same dimension (complexity). Thus, it is necessary to introduce an additional constraint in the R-TPMM approach in order to deal with these cases.

Regarding the computational time of our proposed framework, the most expensive part is the kNN-graph. In the digits experiment (Fig. 7), 2167 points of dimension 784, the execution takes 18.37 s while 10.29 s of the total time is spent in the computation of the kNN-graph. For the experiment with the Yale faces (Fig. 9, 650 points of dimension 3072) the execution time is 7.64 s (3.70 s for the kNN-graph). In the video experiment (Fig. 10, 1673 points of dimension 3072), the total time and the kNN-graph time are, respectively, 29.78 s and 24.87 s (CPU: Pentium Core 2 Duo, 2.0 GHz, 2.0 GB memory).

6 Conclusions

In this paper we developed a framework for the simultaneous and regularized/constrained estimation of the intrinsic dimensionality and density of high dimensional noisy point cloud data sampled from a stratification, as the basis for complexity/density based soft-clustering. The algorithm is based on a statistical model which addresses the presence of noise in the measurements. Our previous related works (Haro et al. 2006, 2007) are particular cases of the R-TPMM algorithm introduced in this paper. We showed that regularization constraints can be naturally introduced in this approach. The experiments showed the importance of incorporating the noise in the model and also of adding regularization in the classification. We also showed that the algorithm is robust to outliers. With the proper dissimilarity function and neighborhood type, we are able to add spatial or temporal regularity in the classification and 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 different 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.

Appendix A: Estimation of the Distribution of Distance Errors

In this section we derive the distribution of the error in the distance between a pair of points when this distance is computed from noisy points. We are interested in the particular case when the noise follows an i.i.d. Gaussian distribution in each of the point coordinates.

Let $X = \{x_t \in \mathbb{R}^p; t = 1, \dots, T\}$ and $\hat{X} = \{\hat{x}_t \in \mathbb{R}^p; t = 1, \dots, T\}$ be two point clouds which are related in the following way: $\hat{x}_t = x_t + n_t$, for each index t, where $n_t \sim N(0, \sigma^2)$, i.e., $\hat{X}$ is a noisy version of X. Let D_{ij} (resp. $\hat{D}_{ij}$) be the Euclidean distance between points x_i and x_j (resp. $\hat{x}_i$ and $\hat{x}_j$). We can write $\hat{D}_{ij}$ as a function of the original points x_i and x_j:

$$\hat{D}_{ij} = \|\hat{x}_i - \hat{x}_j\|_2$$
$$= \left(D_{ij}^2 + \|n_i - n_j\|_2^2 + 2\langle (x_i - x_j), (n_i - n_j) \rangle \right)^{1/2}.$$

Expanding the previous expression in a Taylor series around D_{ij} (considering the rest of the terms sufficiently small),

we obtain,

$$\hat{D}_{ij} \approx D_{ij} + \frac{\langle (x_i - x_j), (n_i - n_j) \rangle}{D_{ij}} + \frac{\|n_i - n_j\|_2^2}{2D_{ij}}$$

$$- \frac{1}{8} \frac{(\langle (x_i - x_j), (n_i - n_j) \rangle)^2}{D_{ij}^3} + O(\sigma^3)$$

$$= D_{ij} + D_{n_1} + D_{n_2} + D_{n_3} + O(\sigma^3).$$

In order to estimate the probability density function of the three error terms D_{n_i}, $i = 1 \ldots 3$, in $\hat{D}_{ij}$ we make use of the following properties:

1. If $X \sim N(\mu, \sigma^2)$ and $a, b \in \mathbb{R}$, then $aX + b \sim N(a\mu + b, (a\sigma)^2)$.
2. If $X \sim N(\mu_X, \sigma_X^2)$ and $Y \sim N(\mu_Y, \sigma_Y^2)$ are independent variables, then:
   a) $X + Y \sim N(\mu_X + \mu_Y, \sigma_X^2 + \sigma_Y^2)$,
   b) $X - Y \sim N(\mu_X - \mu_Y, \sigma_X^2 + \sigma_Y^2)$.
3. If $X_1, \ldots, X_p$ are i.i.d. variables s.t. $X_i \sim N(\mu_i, \sigma_i^2)$, then $U = \sum_{i=1}^{p} (\frac{X_i - \mu_i}{\sigma_i})^2$ follows a Chi-square distribution with p degrees of freedom, $U \sim \chi_p^2$.
4. If X is a random variable with probability density function $f(x)$ and $Y = aX$, where $a \in \mathbb{R}$, then, the probability density function of Y is $\frac{1}{|a|} f(\frac{x}{a})$.
5. The probability density function of the sum of two independent random variables X and Y with probability density functions f and g is the convolution

$$(f * g)(x) = \int_{-\infty}^{\infty} f(y) g(x - y) \, dy.$$

The error term $D_{n_1} \sim N(0, 2\sigma^2)$, by using properties 1, 2(a) and 2(b) (notice that the denominator cancels out the weights in the numerator when adding the individual (constant) variances in each coordinate). The second term, $D_{n_2} \sim \hat{\chi}_p^2 = \frac{D_{ij}}{\sigma^2} \chi_p^2 (\frac{D_{ij}}{\sigma^2} x)$ (properties 2(b) and 3). And for the last term, using properties 1–4, $D_{n_3} \sim \check{\chi}_1^2 = \frac{32D_{ij}^2}{\sigma^2} \chi_1^2 (-\frac{32D_{ij}^2}{\sigma^2} x)$.

Finally, using the previous results and Property 5, we can write

$$\hat{D}_{ij} \approx D_{ij} + W; \quad \text{where } W \sim N(0, 2\sigma^2) * \hat{\chi}_p^2 * \check{\chi}_1^2.$$

In Fig. 12 we show the distribution $N(0, 2\sigma^2)$ compared with the estimated distribution W for $\sigma = 0.5$, $p = 3$ and two different values for D_{ij}: 1.0 and 3.0. As we can see in this figure, for a fixed σ, as D_{ij} gets larger, the distribution W is closer to a $N(0, 2\sigma^2)$ distribution. Then, for values $\frac{D_{ij}}{\sigma}$ not very small, that is, for sufficient SNR, we can approxi-
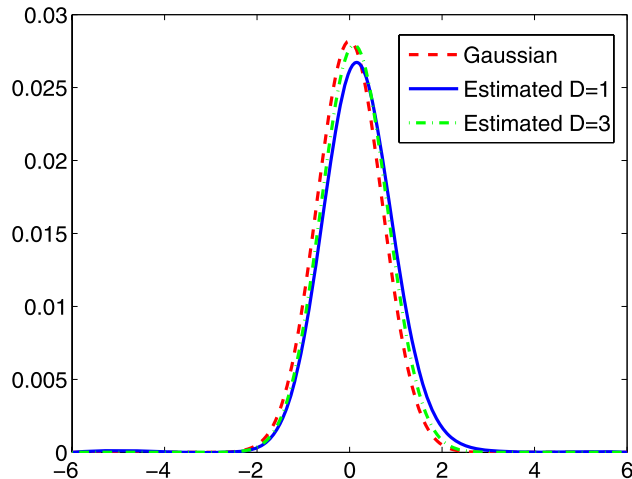

Fig. 12 Gaussian distribution, $N(0, 2\sigma^2)$ with $\sigma = 0.5$, compared to the estimated distribution W for two different values of D_{ij}: 1.0 and 3.0

mate the probability density function of the error in the distance as a Gaussian.

Appendix B: Bound on α for Convergence

We now show that, for a fixed ψ, $F(\psi, H)$ defined in (18) has a global maximum. For that, we follow the same lines as in Ambroise and Govaert (1998). Let us call $F_\psi(H)$ the functional (18) when ψ is fixed. $F_\psi(H)$ has a global maximum if it is strictly concave, i.e., if its Hessian matrix $\mathcal{H}$, with components

$$\mathcal{H}_{il, jt} = \frac{\delta^2 F}{\delta h_l^i \delta h_t^j} = \begin{cases} -1/h_t^j & \text{if } i = j \text{ and } l = t, \\ 2\alpha & \text{if } i = j \text{ and } l \sim t, \\ 0 & \text{otherwise}, \end{cases} \quad (24)$$

is strictly negative. The Gerschgorin-Hadamard Theorem tell us that the eigenvalues λ of this Hessian matrix belong to the union of discs indexed by (j, t) and defined by

$$|\lambda - \mathcal{H}_{jt, jt}| \le \sum_{(i,l) \ne (j,t)} |\mathcal{H}_{jt, il}|.$$

Substituting the last expression with values in (24) gives

$$\left| \lambda - \frac{1}{h_t^j} \right| \le \sum_{l \sim t} 2\alpha = 2\alpha k,$$

where k is the number of neighbors in the regularization term. Since $h_t^j \in [0, 1]$, $\mathcal{H}$ is strictly negative, i.e., every eigenvalue $\lambda < 0$, if $|\lambda - \frac{1}{h_t^j}| < 1$, and this is true for $\alpha < 1/(2k)$.

Appendix C: Proof of the Asymptotic Analysis

When we consider the particular case of hard clustering we have

$$h^j(y_t) = \delta_t^j = \begin{cases} 1 & \text{if } j = \arg\max_i h^i(y_t), \\ 0 & \text{otherwise.} \end{cases}$$

The estimator of the dimension in class j can be expressed as

$$m^j = \left[\frac{1}{N_j} \sum_{t=1}^{T} \delta_t^j \frac{1}{k-1} \sum_{i=1}^{k-1} \log \frac{R_k(y_t)}{\bar{R}_i(y_t)} \right]^{-1}, \qquad (25)$$

where N_j is the number of points clustered in class j and

$$\log \bar{R}_i(y_t) = \frac{\int_0^{R'} f(R_i(y_t)|r) \log r \, dr}{\int_0^{R'} f(R_i(y_t)|r) dr}. \qquad (26)$$

In the (R-)PMM approach we have $\bar{R}_i = R_i$. We can rewrite (25) as

$$m^j = N_j(k-1) \mathrm{m}_T^j Z^{-1}, \qquad (27)$$

where m_T^j is the actual dimension of class j and Z is

$$Z = \sum_{t=1}^{T} \delta_t^j Y_t; \qquad Y_t = \mathrm{m}_T^j \sum_{i=1}^{k-1} \log \frac{R_k(y_t)}{\bar{R}_i(y_t)}.$$

With the proper definition of the upper limit R' in the integral in (26) and the transition density $f(R_i|r)$ when R_i is close to R', we can guarantee that $\bar{R}_i \le R_k$ (always true in (R-)PMM). In this case, we use the fact that $(\bar{R}_i/R_k)^{\mathrm{m}_T^j}$ 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$(k-1, 1)$. Then, $Y_t \sim \mathrm{Gamma}(k-1, 1)$ and the sum of N_j Gamma$(k-1, 1)$ distributions gives $Z \sim \mathrm{Gamma}(N_j(k-1), 1)$ and $Z^{-1} \sim$ Inverse-Gamma$((k-1)N_j, 1)$. The expectation of Z^{-1} is $1/(N_j(k-1)-1)$, and substituting in (27), considering that $1 < N_j(k-1)$, yields

$$E[m^j] = \mathrm{m}_T^j + \frac{\mathrm{m}_T^j}{(k-1)N_j - 1}.$$

Regarding the variance,

$$\mathrm{Var}[m^j] = N_j^2(k-1)^2 \mathrm{Var}[Z^{-1}],$$

where

$$\mathrm{Var}[Z^{-1}] = \frac{1}{(N_j(k-1)-1)^2(N_j(k-1)-2)}.$$

We now define

$$a := \frac{2 - 5N_j(k-1)}{N_j^2(k-1)^2(N_j(k-1)-2)}.$$

After simple computations and under the hypothesis that $|a| < 1$, we obtain

$$\mathrm{Var}[m^j] = \frac{(\mathrm{m}_T^j)^2}{N_j(k-1)-2} \left[1 + \sum_{n=1}^{\infty} a^n \right],$$

and since the second term is smaller than the first one, we can write

$$\mathrm{Var}[m^j] = (\mathrm{m}_T^j)^2 O\left(\frac{1}{N_j(k-1)-2} \right).$$

References

Ambroise, C., & Govaert, G. (1996). Clustering of spatial data by the EM algorithm. In *geoENV I—Geostatistics for environmental applications*.

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

Barbara, D., & Chen, P. (2000). Using the fractal dimension to cluster datasets. In *Proceedings of the sixth ACM SIGKDD* (pp. 260–264).

Belkin, M., & Niyogi, P. (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in NIPS* (Vol. 14). Vancouver, Canada.

Bendich, P., Cohen-Steiner, D., Harer, J., Edelsbrunner, H., & Morozov, D. (2007). Inferring local homology from sampled stratified spaces. In *48th annual IEEE symposium on foundations of computer science* (pp. 536–546).

Brand, M. (2002). Charting a manifold. In *Advances in NIPS* (Vol. 16). Vancouver, Canada.

Cao, W., & Haralick, R. (2006). Nonlinear manifold clustering by dimensionality. In *Proceedings of the 18th international conference on pattern recognition* (Vol. 1, pp. 920–924).

Coifman, R. R., & Lafon, S. (2006). Diffusion maps. *Applied and Computational Harmonic Analysis*, *21*, 5–30. Special issue on Diffusion Maps and Wavelets.

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

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data. *Journal of the Royal Statistical Society Ser. B*, *39*, 1–38.

Dey, T. K., Giesen, J., Goswami, S., & Zhao, W. (2003). Shape dimension and approximation from samples. *Discrete and Computational Geometry*, *29*, 419–434.

Edelsbrunner, H., Letscher, D., & Zomorodian, A. (2002). Topological persistence and simplification. *Discrete Computational Geometry*, *28*, 511–533.

Gionis, A., Hinneburg, A., Papadimitriu, S., & Tsparas, P. (2005). Dimension induced clustering. In *Proceeding of the Eleventh ACM SIGKDD* (pp. 51–60).

Goh, A., & Vidal, R. (2007). Segmenting motions of different types by unsupervised manifold clustering. In *Proceedings of CVPR*.

Haro, G., Randall, G., & Sapiro, G. (2006). Stratification learning: Detecting mixed density and dimensionality in high dimensional point clouds. In *Advances in NIPS* (Vol. 19). Vancouver, Canada.

Haro, G., Randall, G., & Sapiro, G. (2007). Regularized mixed dimensionality and density learning in computer vision. In *Proceedings of 1st workshop on component analysis methods for classification, clustering, modeling and estimation problems in computer vision, in conjunction with CVPR*, Minneapolis, June 2007.

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

Huang, K., Ma, Y., & Vidal, R. (2004). Minimum effective dimension for mixtures of subspaces: A robust GPCA algorithm and its applications. In *Proceedings of CVPR* (pp. 631–638).

Kanatani, K., & Sugaya, Y. (2003). Multi-stage optimization for multi-body motion segmentation. In *Proceedings of the Australia-Japan advanced workshop on computer vision* (pp. 25–31), September 2003.

Kegl, B. (2002). Intrinsic dimension estimation using packing numbers. In *Advances in NIPS* (Vol. 14). Vancouver, Canada.

Kung, S. Y., Mak, M. W., & Lin, S. H. (2004). *Biometric authentication: a machine learning approach*. New York: Prentice Hall.

Lafon, S., Keller, Y., & Coifman, R. R. (2006). Data fusion and multi-cue data matching by diffusion maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(11), 1784–1797.

Levina, E., & Bickel, P. J. (2005). Maximum likelihood estimation of intrinsic dimension. In *Advances in NIPS* (Vol. 17). Vancouver, Canada.

Lu, L., & Vidal, R. (2006). Combined central and subspace clustering for computer vision applications. In *Proceedings of the 23rd international conference on machine learning* (Vol. 148, pp. 593–600).

Ma, Y., Derksen, H., Hong, W., & Wright, J. (2007). Segmentation of multivariate mixed data via lossy coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(9), 1546–1562.

Mordohai, P., & Medioni, G. (2005). Unsupervised dimensionality estimation and manifold learning in high-dimensional spaces by tensor voting. In *IJCAI* (p. 798).

Polito, M., & Perona, P. (2002). Grouping and dimensionality reduction by locally linear embedding. In *Advances in NIPS* (Vol. 14). Vancouver, Canada.

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

Snyder, D. L. (1975). *Random point processes*. New York: Wiley.

Snyder, D. L., & Miller, M. I. (1991). *Random point processes in time and space*. Berlin: Springer.

Souvenir, R., & Pless, R. (2005). Manifold clustering. In *ICCV* (pp. 648–653).

Takens, F. (1985). On the numerical determination of the dimension of an attractor. *Lecture notes in mathematics: Vol. 1125. Dynamical systems and bifurcations* (pp. 99–106).

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

Tishby, N., Pereira, F., & Bialek, W. (1999). The information bottleneck method. In *Proceedings of the 37-th annual allerton conference on communication, control and computing* (pp. 368–377).

Vidal, R., Ma, Y., & Piazzi, J. (2003). Generalized principal component analysis (GPCA). In *Proceedings of CVPR* (pp. 621–628).

Vidal, R., Ma, Y., & Sastry, S. (2004). Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(12).

Yang, A. Y., Rao, S. R., & Ma, Y. (2006). Robust statistical estimation and segmentation of multiple subspaces. In *CVPR workshop on 25 years of RANSAC*.