Information Geometric Density Estimation

Ke Sun and Stéphane Marchand-Maillet

{Ke.Sun, Stephane.Marchand-Maillet}@unige.ch
Viper Group, Computer Vision and Multimedia Laboratory, University of Geneva

Abstract. We investigate kernel density estimation where the kernel function varies from point to point. Density estimation in the input space means to find a set of coordinates on a statistical manifold. This novel perspective helps to combine efforts from information geometry and machine learning to spawn a family of density estimators. We present example models with simulations. We discuss the principle and theory of such density estimation.

Keywords: Information Geometry, Density Estimation, Statistical Learning
PACS: 07.05.Mh

INTRODUCTION

Density estimation in $\Re^m$ aims to approximate an underlying true distribution $\mathscr{T}(\boldsymbol{x})$ which, by assumption, generates a given set of samples $\{\boldsymbol{x}_i\}_{i=1}^n$. Kernel density estimation (KDE) as a non-parametric approach, without assuming any parametric model, approximates $\mathscr{T}(\boldsymbol{x})$ by

$$p(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} p_i(\boldsymbol{x}), \tag{1}$$

where $p_i(\boldsymbol{x})$, or simply p_i, is a local density function (kernel). We focus on the case where $p_i(\boldsymbol{x}) = G(\boldsymbol{x}\,|\,\boldsymbol{x}_i, \Sigma_i)$ is a multivariate Gaussian distribution with mean $\boldsymbol{x}_i$ and covariance matrix Σ_i. The proposed methodology can be extended to other kernels.

This type of estimators appeared more than half a century ago [13]. A large volume of past efforts concentrated on the case $\Sigma_i = h^2 I$, where I is the identity matrix, and how to choose a proper band-width h. This classical case is commonly referred to as Parzen-Rosenblatt window method and is abbreviated as "Parzen" in this paper. The modern computing power presents the opportunity to multiply the number of parameters to better describe the data. Following the developments of mixture models [10], manifold Parzen windows (MParzen) [21, 4] finds $\{\Sigma_i\}$ that are different for different samples, so that $p(\boldsymbol{x})$ follows the local principal directions of the data manifold. In a supervised setting, where $\{\boldsymbol{x}_i\}$ are labelled, neighbourhood component analysis (NCA) [8] learns a global Σ to maximize the leave-one-out nearest-neighbour (NN) classification accuracy. As the unsupervised counterpart of NCA, local component analysis (LCA) [16] learns either a global Σ or a set $\{\Sigma_i\}$ to maximize the leave-one-out likelihood. The idea to align local Gaussian distributions also appears in non-linear dimensionality reduction (NLDR) methods [5]. From a unified geometric perspective, these efforts can be viewed as learning a data geometry of the input space $\Re^m$, where the metric near $\boldsymbol{x}_i$ is approximately Σ_i^{-1}, and the geodesics are likely passing through the data [11].

This paper introduces a meta-method called information geometric density estimation (IGDE) [1]. It implements eq. (1) by embedding $\{\boldsymbol{x}_i\}$ into a statistical manifold, then using the embedding images $\{p_i\}$ as the local densities. Its design involves constructing a low-dimensional sub-manifold of an ambient statistical manifold as the embedding target space, and choosing a measurement scheme of the embedding $\{p_i\}$ on this sub-manifold. We present in more detail a sub-family of IGDE which utilize neighbour-based learning methods. They try to make nearby densities of the same class to be similar so as to *form a data manifold*, and to prevent the densities grow into the class gaps.

In the following, we first review the space of Gaussian distributions. Then we introduce neighbour-based IGDE with demonstrative experiments. In the end, we discuss the design principles of IGDE methods and their theoretical background.

THE GAUSSIAN MANIFOLD

All m-dimensional multi-variate Gaussian distributions form a statistical manifold $\mathscr{M}^m = \{(\boldsymbol{\mu}, \Sigma)\}$ [2], where any point $(\boldsymbol{\mu}, \Sigma)$ is a Gaussian distribution $G(\cdot \,|\, \boldsymbol{\mu}, \Sigma)$. The Riemannian geometry of $\mathscr{M}$ is defined by the Fisher information metric (FIM) [14], the unique Riemannian metric under some conditions including invariance.

Distance is a basic measure that can be used in our problem. With respect to FIM, the geodesic distance $\texttt{dist}_{ij}$, i.e. the length of a shortest path, between two points $p_i = (\boldsymbol{\mu}_i, \Sigma_i)$ and $p_j = (\boldsymbol{\mu}_j, \Sigma_j)$ on $\mathscr{M}$ is, in general, complex. We point the reader to [7] for a comprehensive study of different cases. We use the (asymmetric) Kullback-Leibler (KL) divergence $\delta_{ij} = \int d\boldsymbol{x} p_i(\boldsymbol{x}) \left(\log p_i(\boldsymbol{x}) - \log p_j(\boldsymbol{x})\right)$ as an approximation of $\texttt{dist}_{ij}^2/2$. This approximation is accurate when p_i and p_j are close enough [1].

$\mathscr{M}$ is equipped with a pair of dually affine connections [1]. As $\mathscr{M}$ is in the exponential family, we can write any distribution $(\boldsymbol{\mu}, \Sigma)$ in the canonical form $G(\boldsymbol{x} \,|\, \boldsymbol{\mu}, \Sigma) = \exp\left(tr(\boldsymbol{\theta}_1 \boldsymbol{x}\boldsymbol{x}^T) + \boldsymbol{\theta}_2^T \boldsymbol{x} - \psi(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)\right)$, where ψ is a convex potential function, and $tr(\cdot)$ is the trace. With respect to the canonical parameters $\boldsymbol{\theta}_1 = -\Sigma^{-1}/2$, $\boldsymbol{\theta}_2 = \Sigma^{-1}\boldsymbol{\mu}$, the coefficients of an e-connection vanish. A sub-manifold of $\mathscr{M}$ is called *e-flat* if it is linear in these θ-coordinates. Correspondingly, the expectation parameters $\boldsymbol{\eta}_1 = E(\boldsymbol{x}\boldsymbol{x}^T) = \Sigma + \boldsymbol{\mu}\boldsymbol{\mu}^T$, $\boldsymbol{\eta}_2 = E(\boldsymbol{x}) = \boldsymbol{\mu}$ make an m-connection, which is dual to the e-connection, vanish. A sub-manifold that is linear in the η-coordinates is called *m-flat*. This dually-flat structure has a deep connection with machine learning dynamics [1].

NON-PARAMETRIC NEIGHBOUR-BASED IGDE

From an information geometric view, the density estimator in eq. (1) works by finding an embedding $\mathscr{E} : \Re^m \to \mathscr{M}$ of the input samples $\{\boldsymbol{x}_i\}$ to the Gaussian manifold $\mathscr{M}$, then

---

[1] This paper concentrates on KDE-like density estimation. The discussed principle, however, is not limited to the non-parametric case. We use the general term "IGDE" for future parametric extensions.

[2] The upper-script "m" in $\mathscr{M}^m$ does not denote the dimensionality of $\mathscr{M}$, which is $m(m+3)/2$, but denotes the dimensionality of the associated random variable. $\mathscr{M}^m$ can be simply denoted as $\mathscr{M}$.

giving a statistical mixture of $\{\mathscr{E}(\boldsymbol{x}_i)\}$ with uniform weights. We do not impose a parametric structure $\mathscr{E}(\boldsymbol{x}\,|\,\boldsymbol{\Theta})$ of the embedding $\mathscr{E}$. Instead, we assume $\forall i$, $\mathscr{E}(\boldsymbol{x}_i) = (\boldsymbol{x}_i, \Sigma_i)$, and each $(\boldsymbol{x}_i, \Sigma_i)$ lies on a specific sub-manifold of $\mathscr{M}$. Then, we optimize the embedding with respect to the free parameters $\{\Sigma_i\}$ by utilizing neighbour-based learning methods [3, 9, 8] to preserve pair-wise local information. In the target embedding, this local information is encoded into a probability matrix $P_{n\times n}$. Each row $\boldsymbol{p}_i = (p_{i1}, \ldots, p_{in})$ is normalized, representing the probability for $\mathscr{E}(\boldsymbol{x}_i)$ selecting each other $\mathscr{E}(\boldsymbol{x}_j)$ as its neighbour with respect to the information geometry of $\mathscr{M}$. It can be defined as

$$p_{ij} = \frac{\exp(-\delta_{ij})}{\sum_{j:i\neq j}\exp(-\delta_{ij})} \text{ or } p^t_{ij} = \frac{1/(1+\delta_{ij})}{\sum_{j:i\neq j} 1/(1+\delta_{ij})} \quad (\forall j \neq i). \tag{2}$$

In the following, the upper-script "t" denotes symbols that are associated with p^t_{ij} in eq. (2). The quality of the embedding is measured by a cost function f, usually in the form $f = -tr(Q^T P)$ or $f = -tr(Q^T \log P)$, where $Q = (q_{ij})_{n\times n} \geq 0$ is a fixed target weight matrix based on the input samples. Minimizing f means to align P to Q in the best possible way, and to inject the input information to the output embedding. Usually, f is smooth so that we can write its total differential in the form $df = tr(W^T dD)$, where $D = (\delta_{ij})_{n\times n}$, and $W = (w_{ij})_{n\times n}$ means the pair-wise forces applied on the embedding points during learning. Table 1 lists several possible implementations based on NCA [8], stochastic neighbour embeddings (SNE) [9, 20] and Laplacian eigenmaps (LE) [3]. Note, these methods were mostly used for embedding the input data into a low-dimensional Euclidean space.

TABLE 1. Different neighbour-based learning methods that can be applied to IGDE. "Same class" means, if i and j are in the same class and $i \neq j$, then $q_{ij} = 1$; otherwise $q_{ij} = 0$. "$\circ$" and "$\oslash$" are the element-wise product and division of two matrices, respectively. $\boldsymbol{e} = (1, 1, \ldots, 1)^T$. NCA can be based on L_1 norm (the "NCA" row) or KL-divergence (the "NCA-KL" row).

	f	Q	W	W^t
NCA	$-tr(Q^T P)$	same class	$(Q-(Q\circ P)\boldsymbol{e}\boldsymbol{e}^T)\circ P$	$(Q-(Q\circ P)\boldsymbol{e}\boldsymbol{e}^T)\circ P \oslash (1+D)$
NCA-KL	$-tr(Q^T \log P)$	same class	$Q-(Q\boldsymbol{e}\boldsymbol{e}^T)\circ P$	$(Q-(Q\boldsymbol{e}\boldsymbol{e}^T)\circ P) \oslash (1+D)$
SNE	$-tr(Q^T \log P)$	heat kernel+normalize	$Q-P$	$(Q-P) \oslash (1+D)$
LE	$tr(Q^T D)$	heat kernel	Q	—

As an example, we look into how to adapt L_1-norm-based NCA (the first case in the "NCA" row in table 1) to supervised density estimation based on a set of labeled samples $\{(\boldsymbol{x}_i, y_i) : \boldsymbol{x} \in \Re^m; y_i \in \{1, \ldots, L\}\}$. In a cross-validation scenario, each $\mathscr{E}(\boldsymbol{x}_i)$ $(i = 1, \ldots, n)$ selects a random neighbour $\mathscr{E}(\boldsymbol{x}_j)$ according to $\boldsymbol{p}_i$ and gets classified to the class y_j, which is correct if and only if $y_i = y_j$. The classification accuracy can be maximized with respect to the embedding points $\{\mathscr{E}(\boldsymbol{x}_i)\}$. According to table 1, if $y_i = y_j$, then $w_{ij} = p_{ij}(1 - \sum_{j:y_i=y_j} p_{ij}) \geq 0$, decaying with increasing δ_{ij}. This means, nearby densities within the same class are attracting each other, which helps to better describe the data manifold [21]. If $y_i \neq y_j$, then $w_{ij} = -p_{ij}\sum_{j:y_i=y_j} p_{ij} \leq 0$, decaying with increasing δ_{ij}. Nearby densities of different classes are repelling each other, which helps to clear the gap between two classes.

If we perform the above embedding on $\mathscr{M}^m$ without any constraints, the problem of over-fitting arises due a large number of free parameters in the order of $O(nm^2)$. We

must impose certain regularity conditions and construct a sub-manifold as the target space. First, we assume that the local densities are deteriorated and focused on a l-dimensional hyperplane in $\Re^m$. Equivalently, we apply a linear projection $U_{m\times l}$ ($l \leq m$) on $\{\boldsymbol{x}_i\}$ and estimate the density of $\{U^T\boldsymbol{x}_i\}$ instead. U can be either precomputed using dimensionality reduction techniques, e.g. NCA [8], and fixed during learning, or learned by *integrating dimensionality reduction with density estimation*. In the latter case, the scale of U must be constrained, e.g. by $U^TU = I_{l\times l}$, to avoid trivial solutions. Moreover, we assume that $\forall i$, $\Sigma_i = S_iS_i^T + h^2I$, where $h > 0$ is a pre-fixed minimum bandwidth, and S_i is a $l \times r$ ($r \leq l$) matrix which satisfies $tr(S_iS_i^T) = (\tau - 1)h^2$. $\tau \geq 1$ is also a pre-fixed parameter, meaning the highest possible ratio of Σ_i's largest eigenvalue to its smallest eigenvalue. The above two assumptions constrain the embedding to a singular region on $\mathscr{M}$, where the Gaussian distributions deteriorate. The number of free parameters in $\{\mathscr{E}(\boldsymbol{x}_i)\}$ is reduced to nlr.

The pair-wise KL divergence δ_{ij} with respect to the above assumptions is

$$\delta_{ij}(U,S_i,S_j) = \frac{1}{2}tr\left(\left(U^T(\boldsymbol{x}_i-\boldsymbol{x}_j)(\boldsymbol{x}_i-\boldsymbol{x}_j)^TU+\Sigma_i\right)\Sigma_j^{-1}\right) - \frac{1}{2}\log|\Sigma_i\Sigma_j^{-1}| - \frac{l}{2}. \quad (3)$$

In this paper, "$|\cdot|$" denotes either the determinant or the volume. Recall that $df = tr(W^TdD)$. This together with eq. (3) and $\Sigma_i = S_iS_i^T + h^2I$ gives

$$\frac{\partial f}{\partial U} = \sum_{i=1}^n\sum_{j=1}^n w_{ij}\frac{\partial\delta_{ij}}{\partial U} = \sum_{i=1}^n\sum_{j=1}^n w_{ji}(\boldsymbol{x}_i-\boldsymbol{x}_j)(\boldsymbol{x}_i-\boldsymbol{x}_j)^TU\Sigma_i^{-1}, \quad (4)$$

$$\begin{aligned}\frac{\partial f}{\partial S_i} =& 2\sum_{j=1}^n\left(w_{ij}\frac{\partial\delta_{ij}}{\partial\Sigma_i} + w_{ji}\frac{\partial\delta_{ji}}{\partial\Sigma_i}\right)S_i = \left[\sum_{j=1}^n\left(w_{ji}-w_{ij}\right)\Sigma_i^{-1}\right.\\ &\left.+\sum_{j=1}^n w_{ij}\Sigma_j^{-1} - \Sigma_i^{-1}\sum_{j=1}^n w_{ji}\left(\Sigma_j + U^T(\boldsymbol{x}_i-\boldsymbol{x}_j)(\boldsymbol{x}_i-\boldsymbol{x}_j)^TU\right)\Sigma_i^{-1}\right]S_i. \quad (5)\end{aligned}$$

The projection of the gradient in eq. (5) on the constraint $tr(S_iS_i^T) = (\tau-1)h^2$ is $\partial f/\partial S_i - tr(S_i^T\partial f/\partial S_i)S_i/((\tau-1)h^2)$. Based on this projected gradient, as well as eq. (4) if U has to be learned, learning can be implemented by any gradient-based optimizer, which has to carefully avoid local optima.

The hyper-parameters l, r, h and τ have to be tuned. l is the reduced dimensionality after the global projection U. For data visualization, one can set $l = 2$ or 3. When the data is (assumed to be) pre-processed by dimensionality reduction methods, one can leave $U = I$ and $l = m$. This l appears in any other density estimator which integrates dimensionality reduction. Therefore, it is fair to say that, neighbour-based IGDE has 3 hyper-parameters and an optional module to perform dimensionality reduction. r, usually in the range $1 \sim 10$, is the rank of the local S_i's. It corresponds to the intrinsic dimensionality of the data and can be set accordingly [19]. h and τ determine the shape and total energy of each p_i. An empirical range of τ is $2 \sim 5$. Large values of l, r, τ and small values of h are likely to cause over-fitting. One can choose an optimal set of hyper-parameters by cross-validation for high likelihood on the validation sets.

EXAMPLES

We present IGDE examples, not for a systematical experimental study, but to discuss the advantages of IGDE. In particular, we compare the two variations of IGDE-NCA in the "NCA" row of table 1, denoted, in order, by IGDE-NCA^g and IGDE-NCA^t, with Parzen and MParzen in supervised density estimation.

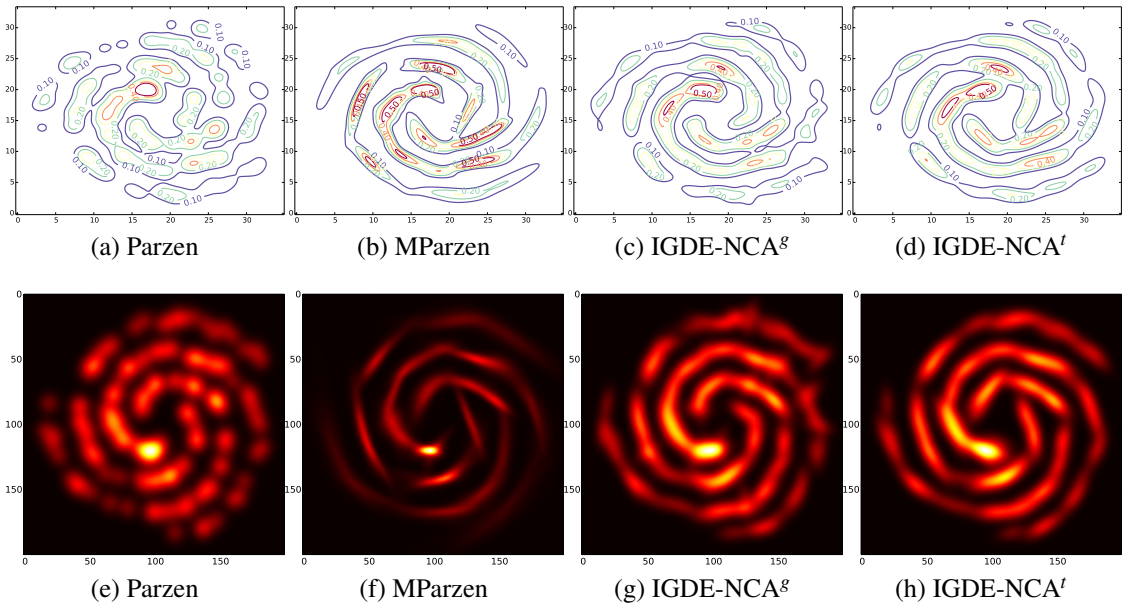
The `spiral` and `pathbased` datasets[3] consist of 2D point clouds with 3 classes each. `Spiral` resembles 3 entangled spirals (see figure 1). `Pathbased` resembles two blobs inside a circle (see figure 2(c)). In each run, the associated dataset is added Gaussian noise $G(\cdot\,|\,\mathbf{0},\sigma^2_{noise}I)$, and then, half of the dataset is randomly sampled for training, where 20% is used for validation. In this supervised case, MParzen is based on k-NN *within the same class*. IGDE-NCA is implemented by simple gradient descent with momentum. All hyper-parameters, including Parzen's $h \in \{0.01, 0.02, \ldots, 2.00\}$, MParzen's regularization parameter $\sigma \in \{0.01, 0.02, \ldots, 0.10, 0.2, \ldots, 1.0\}$, neighbourhood size $k \in \{1, 2, \ldots, 20\}$, number of principal components $d \in \{1, 2\}$, and IGDE-NCA's $h \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$, $\tau = 4$, $r = 1$, $l = 2$, are tuned to minimize the same-class average negative log-likelihood (SANLL) $-\sum_{i=1}^{n} \log(\frac{1}{n_i}\sum_{j:y_i=y_j} p_j(\boldsymbol{x}_i))/n$ on the validation set, where n_i is the number of training samples in the same class as the test sample i. Note, we choose a coarse configuration grid for IGDE-NCA, because it requires a time-consuming training process. In theory, on a fine grid, IGDE-NCA can achieve even better performance.

Figure 1 shows the density contours and color-maps in one trial, where σ_{noise} is 0.4. Parzen is not able to capture well the data manifold. Its density is discontinuous and looks blurred. MParzen is likely to over-fit, producing many zigzags and looking skinny and angular. This is because MParzen is based on the k-NN graph, which is not robust to noise. IGDE-NCA^g often presents some local bumps. This is because its learner often ends in a sub-optimal region, where f is almost flat with tiny gradient. In essence, it is easy for $\mathscr{E}(\boldsymbol{x}_i)$ and $\mathscr{E}(\boldsymbol{x}_j)$ from different classes to have a small p_{ij} in eq. (2) due to the locality of the Gaussian kernel. IGDE-NCA^g needs more advanced optimization than simple gradient descent to give better results. The visualization by IGDE-NCA^t is, in general, more appealing. The similarity p^t_{ij} between two nearby $\mathscr{E}(\boldsymbol{x}_i)$ and $\mathscr{E}(\boldsymbol{x}_j)$ is *exaggerated by a non-local kernel*, which helps to enhance the forces (W or W^t in table 1) during learning. Such density maps are only intuitive measurements and vary slightly across different runs. Figure 2(a,b) shows the testing SANLL on 3 different noise levels. Remarkably, IGDE-NCA^t achieves much better performance with smaller variance as compared to the other methods. MParzen has a low SANLL (but large variations) on `spiral` added with small-noise, because the data has a smooth manifold structure. It performs poor on `pathbased`, which has two blobs (clustered structure). At a large noise level, both IGDE-NCA^g and IGDE-NCA^t are preferred over MParzen.

To conclude, the good performance of MParzen on manifold-structured datasets relies on the validation process. It is limited by the noisy k-NN graph and the lack of a learning process to exchange information between neighbourhoods. IGDE-NCA, as

---

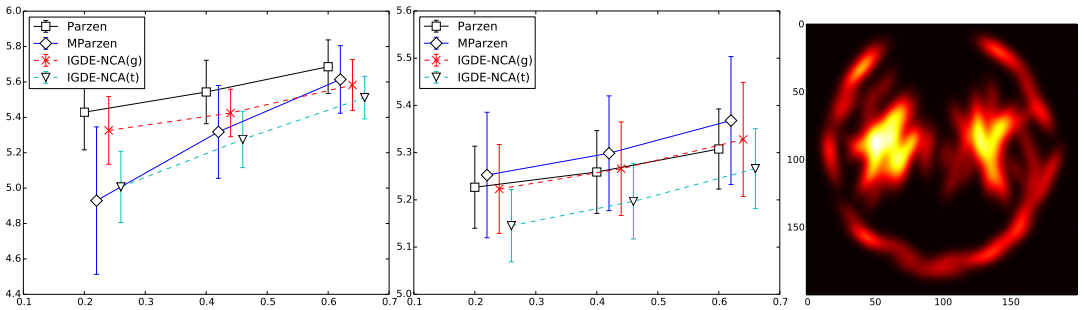
[3] `http://cs.joensuu.fi/sipu/datasets/`



(a) Parzen (b) MParzen (c) IGDE-NCA^g (d) IGDE-NCA^t

(e) Parzen (f) MParzen (g) IGDE-NCA^g (h) IGDE-NCA^t

FIGURE 1. Contours (top) and color-maps (bottom) of the estimated density on the `spiral` dataset.



(a) `Spiral` (b) `Pathbased` (c) IGDE-NCA^t

FIGURE 2. (a-b) Testing SANLL (avg.±std.) against 3 noise levels (0.2, 0.4, 0.6) after 300 runs on two datasets. (c) The density map of `pathbased` by IGDE-NCA^t in one trial.

a demonstration of the IGDE concept, is designed to overcome both difficulties. It implements an information flow between nearby structures. As preliminary experiments, this concept is verified by its good performance on two different toy datasets. IGDE-NCA inherits the $O(n^2)$ computational complexity of NCA and thus does not scale up well. It needs further developments to be applied on real data.

DISCUSSION

The key proposal of this manuscript is to implement the density estimator in eq. (1) by optimizing an embedding $\mathscr{E} : \Re^m \to \mathscr{M}$. A family of IGDE methods can be spawned along the following two axes.

① There is a wide array of embedding target spaces. The ambient manifold $\mathscr{M}$ can be non-Gaussian depending on the type of data. For example, in graph-based density estimation, e.g. social network analysis, one could use the statistical simplex as $\mathscr{M}$, where any point $p_i \in \mathscr{M}$ is a local distribution on the graph nodes. Usually, a sub-manifold $\mathscr{M}_\theta \subset \mathscr{M}$ is constructed to reduce the model flexibility. Its definition involves a *decomposition of global information and local information*. In MParzen [21] and neighbour-based IGDE, $\boldsymbol{\eta}_1^i = h^2 I + (\boldsymbol{x}_i \boldsymbol{x}_i^T + S_i S_i^T)$ is a linear combination of the global $h^2 I$ and the local low-rank $(\boldsymbol{x}_i \boldsymbol{x}_i^T + S_i S_i^T)$, and $\boldsymbol{\eta}_2^i = \boldsymbol{x}_i$ is fixed. This m-flat structure (see section 2) makes it easy to constrain the total energy and effective support of p_i in $\Re^m$, and to avoid singularities on $\mathscr{M}$. Alternatively, in LCA [16], they assume that p_i is a product of a global Gaussian $(\boldsymbol{\mu}, \Sigma)$ and a local Gaussian $(\boldsymbol{x}_i, \Sigma_i)$. This can be written as $\boldsymbol{\theta}_1^i = -\Sigma^{-1}/2 - \Sigma_i^{-1}/2$, $\boldsymbol{\theta}_2^i = \Sigma^{-1}\boldsymbol{\mu} + \Sigma_i^{-1}\boldsymbol{x}_i$. This e-flat structure helps decompose global information (e.g. high dimensional noise or global metric) and local information which are independent, because a sum in the θ-coordinates corresponds to a product of probabilities. ② Another direction to develop IGDE is on how to measure the embedding on $\mathscr{M}_\theta$. Locally, we usually have to approximate the geometric quantities, e.g. distance, on $\mathscr{M}_\theta$ defined by FIM. Besides KL-divergence used in this paper, there is a pool of information divergences with diverse properties [12]. At a global scale, the overall cost of the embedding can refer to efforts on NLDR [3, 9, 5, 20], which contributed diverse objective functions and heuristics.

The goal of IGDE can be understood by the minimum description length (MDL) principle [15] formulated as

$$\min \left[-\sum_{i=1}^{n} \log_2 p(\boldsymbol{x}_i) + |enc(\mathscr{O})| + n \log_2 \frac{|\mathscr{O}|}{\varepsilon} \right]. \tag{6}$$

The first term is the encoding length of the data given a fixed model p in eq. (1). Minimizing this term pulls $\{p_i\}$ towards the boundary $\Sigma = \mathbf{0}$ of $\mathscr{M}$, making p like the empirical distribution. In neighbour-based IGDE, this strength is weakly conducted by constraining the energy of each p_i. The second and third terms in eq. (6) express the length of p in a hierarchical coding scheme. We first find a sub-manifold $\mathscr{O} \subset \mathscr{M}$ enclosing $\{p_i\}$. Its description length is $|enc(\mathscr{O})|$ ("enc" is for "encoding"). This $\mathscr{O}$ corresponds to the data manifold after the embedding $\mathscr{E}$. Then, we cover $\mathscr{O}$ with tiny patches $\{\mathscr{P}_1, \mathscr{P}_2, \dots\}$ with equal volume ε (similar to figure 3.3 in [2]) and zero overlap. Within each patch $\mathscr{P}_i$, all distributions are regarded the same. The code length to record each p_i is $\log_2(|\mathscr{O}|/\varepsilon)$. Minimizing the last two terms in eq. (6) pulls $\{p_i\}$ to the high entropy region on $\mathscr{M}$, and gives $\{p_i\}$ a *a low dimensional and compact enclosure* $\mathscr{O}$. In neighbour-based IGDE, this strength is conducted through NLDR to form local low-rank structures on $\mathscr{M}$. In addition to eq. (6), in a supervised scenario, where each class is represented by an $\mathscr{O}_i$, maximizing the margin between these sub-manifolds reduces the description length of supervised information (see [17] pp. 194).

By the third term in eq. (6), the encoding length of p scales with n. This means high storage complexity and the risk of over-flexible models. This is the price for a nearly assumption-free estimator, as the manifold assumption is only a weak assumption. To tackle these difficulties, one way is to build a parametric $\mathscr{E}(\boldsymbol{x}|\boldsymbol{\Theta})$ [4]. Alternatively, a two-step procedure is to "simplify" the resulting density [18].

IGDE is similar to information geometric dimensionality reduction (IGDR) [6] in that they both investigate the low-dimensional structures of a set of points on a statistical manifold. They have very different objectives, though. IGDR performs dimensionality reduction on a given set of probability density functions based on their pair-wise information geometric measurements. IGDE learns a set of probability density functions, corresponding to the input samples, to estimate a density function in the input space.

ACKNOWLEDGMENTS

This work is partly supported by the European COST Action on Multilingual and Multifaceted Interactive Information Access (MUMIA) via the Swiss State Secretariat for Education and Research (SER grant C11.0043).

REFERENCES

1. S. Amari and H. Nagaoka. *Methods of Information Geometry*, volume 191 of *Translations of Mathematical Monographs*. AMS and OUP, 2000. (Published in Japanese in 1993).
2. V. Balasubramanian. MDL, Bayesian inference, and the geometry of the space of probability distributions. In *Advances in Minimum Description Length: Theory and Applications*, pages 81–99. MIT Press, 2005.
3. M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, 2003.
4. Y. Bengio, H. Larochelle, and P. Vincent. Non-local manifold Parzen windows. In *NIPS 18*, pages 115–122. MIT Press, 2006.
5. M. Brand. Charting a manifold. In *NIPS 15*, pages 961–968. MIT Press, 2003.
6. K. M. Carter, R. Raich, W. G. Finn, and A. O. Hero. Information-geometric dimensionality reduction. *IEEE Signal Process. Mag.*, 28(2):89–99, 2011.
7. S. I. R. Costa, S. A. Santos, and J. E. Strapasson. Fisher information distance: a geometrical reading. *arXiv*, 1210.2354v3 [stat.ME], 2014.
8. J. Goldberger, G. E. Hinton, S. T. Roweis, and R. Salakhutdinov. Neighbourhood components analysis. In *NIPS 17*, pages 513–520. MIT Press, 2005.
9. G. E. Hinton and S. T. Roweis. Stochastic neighbor embedding. In *NIPS 15*, pages 833–840. MIT Press, 2003.
10. G. E. Hinton, M. Revow, and P. Dayan. Recognizing handwritten digits using mixtures of linear models. In *NIPS 7*, pages 1015–1022. MIT Press, 1995.
11. G. Lebanon. Learning Riemannian metrics. In *UAI*, pages 362–369, 2003.
12. F. Nielsen and R. Nock. Sided and symmetrized Bregman centroids. *IEEE Trans. Inf. Theory*, 55(6): 2882–2904, 2009.
13. E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Statist.*, 33(3): 1065–1076, 1962.
14. C. R. Rao. Information and accuracy attainable in the estimation of statistical parameters. *Bull. Cal. Math. Soc.*, 37(3):81–91, 1945.
15. J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.
16. N. Le Roux and F. Bach. Local component analysis. *arXiv*, 1109.0093 [cs.LG], 2011.
17. B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.
18. O. Schwander and F. Nielsen. Learning mixtures by simplifying kernel density estimators. In *Matrix Information Geometry*, pages 403–426. Springer, 2012.
19. K. Sun and S. Marchand-Maillet. An information geometry of statistical manifold learning. In *ICML, JMLR: W&CP 32(1)*, pages 1–9, 2014.
20. L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *JMLR*, 9(Nov):2579–2605, 2008.
21. P. Vincent and Y. Bengio. Manifold Parzen windows. In *NIPS 15*, pages 825–832. MIT Press, 2003.