ISDE : Independence Structure Density Estimation

Louis Pujol

ISDE: Independence Structure Density Estimation

Louis Pujol*

Abstract

Density estimation appears as a subroutine in many learning procedures, so it is of interest to have efficient methods for it to perform in practical situations. Multidimensional density estimation suffers from the curse of dimensionality. A solution to this problem is to add a structural hypothesis through an undirected graphical model on the underlying distribution. We propose ISDE (Independence Structure Density Estimation), an algorithm designed to estimate a density and an undirected graphical model from a particular family of graphs corresponding to Independence Structure (IS), a situation where we can separate features into independent groups. ISDE works for moderately high-dimensional data (up to a few dozen features), and it is useable in parametric and nonparametric situations. Existing methods on nonparametric graphical model estimation focus on multidimensional dependencies only through pairwise ones: ISDE does not suffer from this restriction and can address structures not yet covered by available algorithms. In this paper, we present the existing theory about IS, explain the construction of our algorithm and prove its effectiveness. This is done on synthetic data both quantitatively, through measures of density estimation performance under Kullback-Leibler loss, and qualitatively, in terms of capability to recover IS. By applying ISDE on mass cytometry datasets, we also show how it performs both quantitatively and qualitatively on real-world datasets. Then we provide information about running time.

Keywords— Multivariate Density Estimation, Independence Structure

*Université Paris-Saclay, CNRS, Inria, Laboratoire de Mathématiques d'Orsay, 91405, Orsay, France. louis.pujol@universite-paris-saclay.fr

1 Introduction

Unsupervised Learning and Density Estimation Unsupervised learning is an important field of data analysis. It aims to design methods to extract meaningful information from a dataset with little prior knowledge. A central task in unsupervised learning is density estimation. Given a sample $X_1, \dots, X_N$ drawn independently from a random variable X on $\mathbb{R}^d$ with a density f, the goal is to build an estimator $\hat{f}$ of f. This question finds many applications, and density estimation is a building block for many learning tasks such as clustering ([Chazal et al., 2013], [Campello et al., 2013]) or anomaly detection ([Chandola et al., 2009]) among others.

Nonparametric and Parametric Density Estimation The easiest way to do density estimation is to consider parametric models. Here data is supposed to be drawn from a probability distribution known up to a finite-dimensional parameter θ. Estimating the density is then equivalent to estimating θ. One example is the centered multivariate Gaussian framework, where the parameter θ is the covariance matrix Σ. An introduction to parametric statistics can be found in [Wasserman, 2004], chapter 9. This approach suffers from a lack of flexibility as it strongly constrains the model.

At the other end of the spectrum lies nonparametric density estimation. In this framework, densities are no longer considered members of some finite-dimensional family but are supposed to belong to a set of functions with a given regularity (Lipschitz or Hölder, for example). An introduction to the subject can be found in [Tsybakov, 2008].

Curse of Dimensionality When dealing with multidimensional data, one must be aware of the issues that the number of features can imply. The complexity of a statistical problem can be evaluated through minimax risk, quantifying the statistical error in a worst-case scenario. In the covariance estimation problem, without further assumption on the covariance matrix, the minimax risk under the Frobenius norm is proportional to $\frac{d}{N}$ (see [Cai et al., 2010]). In the nonparametric framework, the minimax rate is influenced by two parameters: a regularity parameter β and dimension d , the rate of convergence for the squared L_2 loss

is typically proportional to $N^{\frac{-2\beta}{2\beta+d}}$ (see [Goldenshluger and Lepski, 2014] for an exhaustive coverage of the topic).

We remark that the dependence on the dimension is adversarial in both situations. The higher d, the more complex the density estimation problem is. This phenomenon is a manifestation of the so-called curse of dimensionality. For practitioners, it means that it should be adventurous to use a multivariate density estimator if the sample size is limited and the dimension becomes large, especially in the case of nonparametric estimation. A solution is to assume that unknown densities belong to a structured class of functions.

Structural Density Estimation with Undirected Graphical Models A way to consider a structure for a multivariate random variable is to study its undirected graphical model (introduction to the field can be found in [Giraud, 2014] and more in-depth cover in [Wainwright and Jordan, 2008]). As we will not consider directed graphical models, we always consider that graphs are undirected in the sequel. Given a graph $G = (V, E)$ whose vertices correspond to the features $(X^1, \ldots, X^d)$ we say that G is a graphical model for X if the following condition is satisfied:

$$(i, j) \notin E \Rightarrow X^i \perp\!\!\!\perp X^j | (X^k)_{k \notin (i,j)}. \tag{1}$$

Constraints on the graph associated with a distribution impose a structure on the density, and such a structure can help overcome the curse of dimensionality. However, learning a graphical model is a complex task in many situations. An exception is the multivariate Gaussian framework, where data distribution is a multivariate normal $\mathcal{N}(0, \Sigma)$. Here the estimation of the graphical model is possible through estimation of the inverse of Σ. This setting is known as the Gaussian Graphical Model (GGM). Different methods are available: graphical lasso [Friedman et al., 2008] which imposes a sparse structure for the graph, is probably the most famous example.

In a fully nonparametric setting, up to our knowledge, one method is available: Forest Density Estimation (FDE) [Liu et al., 2011] which is limited to graphs without cycles.

Independence Structure In the present work, we focus on the model of Independence Structure (IS) for multivariate density developed by [Lepski, 2013] and studied by [Rebelles, 2015]. It contains d-dimensional densities, which can be decomposed as a product of low-dimensional marginals, forming a partition of the original features. For the graphical model, it corresponds to graphs that are composed of disjoint fully-connected cliques.

These authors have shown that if the density enjoys the property that the size of the biggest block of the partition is equal to $k < d$, then the complexity of density estimation, measured through minimax rate, is related to k instead of the ambient dimension d. However, these works rely on the analysis of hardly implementable estimators.

Moderately High Dimension Setting In recent years, attention was put on high-dimensional problems, where the number of features can vary from hundreds to thousands. We are interested here in situations of moderately high dimension, where the number of features can vary from a few ones to a few dozens. It is of particular interest to distinguish both paradigms as we will develop algorithmic solutions that allow exhaustive search over admissible structures in moderately high dimension but become too time-consuming in high dimension.

Our Contribution We have developed Independence Structure Density Estimation (ISDE), a method designed to simultaneously compute a partition of the features and a density estimation relying on this partition. Our method enjoys reasonable running time for moderately high-dimensional problems and can be combined with any density estimation technique, so it covers parametric as well as nonparametric settings.

For GGM, an algorithm already exists [Devijver and Gallopin, 2018], but up to our knowledge, we are the first to design an algorithm to deal with nonparametric density estimation under the model of IS.

Organization of the Paper In section 2 we briefly review existing work about IS. In section 3 we present ISDE. In section 4 we compare our method with some

existing ones for the task of density estimation under Kullback-Leibler loss for synthetic and real-world datasets from mass cytometry experiments before analyzing its running time in section 5.

Notations Let f be a density function (a nonnegative real function whose integral is equal to 1) over $\mathbb{R}^d$. If we think of f from a statistical viewpoint, it is natural to refer to the indices $\{1, \dots, d\}$ as the features.

Let $S \subset \{1, \dots, d\}$, we denote by f_S the marginal density of f over S. For all $x = (x_1, \dots, x_d) \in \mathbb{R}^d$

$$f_S(x) = \int f(x) \prod_{i \notin S} dx_i. \tag{2}$$

With a slight abuse of notation, to highlight the fact that $f_S(x)$ does not depend on $(x_i)_{i \notin S}$, we write $f_S(x_S)$ instead of $f_S(x)$.

Let k be an positive integer not greater than d. We denote by Set_d^k the set of all subsets of $\{1, \dots, d\}$ with cardinal not greater than k and by Part_d^k the collection of all partitions of $\{1, \dots, d\}$ constructed with blocks in Set_d^k. We also introduce the shortcuts $\mathrm{Set}_d = \mathrm{Set}_d^d$ and $\mathrm{Part}_d = \mathrm{Part}_d^d$.

2 Independence Structure

In this section we review some theory about nonparametric density estimation and IS model.

Minimax Risk Let $X_1, \dots, X_N$ be *iid* realizations of a random variable in $\mathbb{R}^d$ admitting a density f. The goal of density estimation is to construct an estimator $\hat{f}$ of the density. We can measure the hardness of such an estimation task using the minimax framework. Assume that the true density belongs to some known model $\mathcal{F}$ and let D be a (pseudo)distance on $\mathcal{F}$, the minimax risk is defined as follows:

$$\mathcal{R}(D, \mathcal{F}) := \inf_{\hat{f}} \sup_{f \in \mathcal{F}} \mathbb{E}\left[D(f, \hat{f})\right] \tag{3}$$

where the inf is taken over all measurable functions from the data to $\mathcal{F}$. More specifically, a great part of the literature on the topic deals with the asymptotic regime of $\mathcal{R}(D, \mathcal{F})$ with respect to N.

Hölder Balls Let g be a function from $\mathbb{R}^d$ to $\mathbb{R}$. Let $\gamma = (\gamma_1, \ldots, \gamma_d) \in \mathbb{N}^d$ be a multiindex and let $|\gamma| = \sum_{i=1}^d \gamma_i$ be its order. The partial differentiate operator D^γ is defined as follows

$$D^\gamma g = \frac{\partial^{|\gamma|} g}{\partial_{x_1}^{\gamma_i} \ldots \partial_{x_d}^{\gamma_d}}. \tag{4}$$

For $\beta, H > 0$ let us consider $\mathcal{F} = \mathcal{H}^\beta(d, H)$ the Hölder ball over $\mathbb{R}^d$ defined as follows. If we denote by s the larger integer strictly lower than β and let $\delta = \beta - s \in (0, 1]$, $g : \mathbb{R}^d \to \mathbb{R}$ belongs to $\mathcal{H}^\beta(d, H)$ if both following conditions are fulfilled.

$$\begin{cases} \max\limits_{|\gamma| \leq s} \sup\limits_{x \in \mathbb{R}^d} |D^\gamma g(x)| \leq H \\ \max\limits_{|\gamma| = s} \sup\limits_{x, y \in \mathbb{R}^d} |D^\gamma g(x) - D^\gamma g(y)| \leq H \|x - y\|^\delta. \end{cases} \tag{5}$$

Minimax Risk over Hölder Balls In [Hasminskii et al., 1990], the minimax rate of this family of functions was studied considering L_p distances. In particular, the result with the squared L_2 distance is the following

$$\mathcal{R}\left(\|.\|_2^2, \mathcal{H}^\beta(d, H)\right) \sim N^{-\frac{2\beta}{2\beta + d}}. \tag{6}$$

We can interpret this bound as a manifestation of the curse of dimensionality because of its dependence on d. A solution is to consider the IS model introduced in [Lepski, 2013].

Independence Structure For $k \leq d$, we define a family of functions:

$$\mathcal{D}_d^k = \left\{ f : \mathbb{R}^d \to \mathbb{R} |\ \exists \mathcal{P} \in \mathrm{Part}_d^k : f(x) = \prod_{S \in \mathcal{P}} f_S(x_S) \right\}. \tag{7}$$

In probabilistic terms, a density f over $\mathbb{R}^d$ belongs to $\mathcal{D}_d^k$ if we can group these features into independent blocks. Another viewpoint is that the random variable characterized by f admits a graphical model, a collection of disjoint fully connected cliques of size not greater than k. It was showed in [Rebelles, 2015] that

$$\mathrm{R}\left(\|.\|_2^2, \mathcal{H}^\beta(d,H) \cap \mathcal{D}_d^k\right) \sim N^{-\frac{2\beta}{2\beta+k}}. \tag{8}$$

The striking fact here is that the hardness of the estimation problem is no longer related to the ambient dimension but instead to the size of the biggest block of the partition on which the density function is decomposable.

Practical Consideration In order to compute minimax rates, an appropriate estimator has been defined in [Rebelles, 2015], but it is not practically computable. However, we believe that the IS model could be of practical interest as it leads to qualitative information about data through the IS and tackles the curse of dimensionality.

3 ISDE

In this section, we present ISDE, our algorithm designed to perform simultaneously density estimation and independence partition selection in a moderately high-dimensional setting. Let k be an input parameter. We aim to provide a method taking point cloud as input and outputting an IS (a partition of the features in Part_d^k) and a density estimator as a product of marginal estimators.

Hyperparameters Optimization Let Θ denote a hyperparameter space adapted to our problem. A parameter $\theta \in \Theta$ corresponds to a collection of partition-indexed parameters $(\theta_\mathcal{P})_{\mathcal{P}\in\mathrm{Part}_d^k}$, each of them being a list of parameter for marginal density estimates: $\theta_\mathcal{P} = (\theta_\mathcal{P}(S))_{S\in\mathcal{P}}$. Then to each $\theta \in \Theta$ is associated a family of density estimators satisfying IS condition:

$$\left(\hat{f}_\mathcal{P}^\theta\right)_{\mathcal{P}\in\mathrm{Part}_d^k} = \left(\prod_{S\in\mathcal{P}} \hat{f}_S^{\theta_\mathcal{P}(S)}\right)_{\mathcal{P}\in\mathrm{Part}_d^k}. \tag{9}$$

We do not specify which set of hyperparameters Θ we take. This choice will be an input of ISDE as we want our method to be usable indifferently with any local density estimator.

Number of Partitions and Complexity Bottleneck Apparently, we need to compute a density estimation of the form $\prod_{S\in\mathcal{P}} \hat{f}_S^{\theta_{\mathcal{P}}(S)}$ for all $\mathcal{P} \in \text{Part}_d^k$. However, as we will see, the number of partitions is rapidly very high, even for moderately high-dimensional settings. Then we need to avoid this complexity bottleneck by carefully designing our algorithm.

Let us start by comparing S_d and B_d, the respective cardinals of Set_d and Part_d. We have $S_d = 2^d - 1$ and B_d is known as the Bell number of order d. table 1 shows how these quantities compare for dimension lying between 10 and 15.

d	10	11	12	13	14	15
S_d	$1,023$	$2,047$	$4,095$	$8,191$	$16,383$	$32,767$
B_d	$115,975$	$678,570$	$4,213,597$	$27,644,437$	$190,899,322$	$1,382,958,545$

Table 1: Number of partitions vs number of subsets

Even if we restrict ourselves to small values of k, the difference remains important. We denote S_d^k and B_d^k the cardinals of Set_d^k and Part_d^k. It is simple to see that

$$S_d^k = \sum_{i=1}^{k} \binom{d}{k}. \tag{10}$$

For B_d^k exact computation is harder but we can prove that (see appendix A.1)

$$B_d^k \geq B_d^2 = 1 + \binom{d}{2} + \frac{\binom{d}{2}\binom{d-2}{2}}{2!} + \frac{\binom{d}{2}\binom{d-2}{2}\binom{d-4}{2}}{3!} \cdots + \frac{\binom{d}{2}\cdots\binom{d-2(\lfloor d/2\rfloor-1)}{2}}{(\lfloor d/2\rfloor)!} \tag{11}$$

and notice that $B_d^2 \underset{d\to\infty}{\sim} d^{\frac{d}{2}}$ while $S_d^k \underset{d\to\infty}{\sim} d^k$. For values of d corresponding to moderately high-dimensional settings, some computations are gathered in table 2 (the values of B_d^2 are approximations).

d	20	30	40	50
S_d^3	$1,350$	$4,525$	$10,700$	$20,875$
B_d^2	2.4×10^{10}	6.1×10^{17}	7.3×10^{25}	2.8×10^{34}

Table 2: Number of partitions vs number of subsets

These computations indicate that it would be beneficial to find a way to avoid the computation of B_d^k estimators. Intuitively, as estimators are combinations of marginals estimators, it seems reasonable to decouple marginal estimations from partition selection. We will now see that we can implement this idea through an appropriate choice of the loss function.

Choice of Loss Function Though theory about IS focuses on L_p losses. We found it more convenient to rephrase the estimation problem using Kullback-Leibler (KL) divergence as a discrepancy measure. The reason is that KL involves log-densities, making it is well suited for densities in $\mathcal{D}_d^k$ as the logarithm of a product of marginal densities becomes the sum of the marginal log-densities.

For an estimator $\hat{f}$, the Kullback-Leibler divergence is defined as follows:

$$\mathrm{KL}\left(f \| \hat{f}\right) = P\left[\log\left(\frac{f}{\hat{f}}\right)\right] = P\left[\log\left(f\right)\right] - P\left[\log(\hat{f})\right] \tag{12}$$

where for any function g, $P[g] = \int g(x)f(x)dx$. We see that minimizing $\mathrm{KL}\left(f \| \hat{f}\right)$ is equivalent to maximizing $P\left[\log(\hat{f})\right]$.

Optimization Problem Then, the optimization problem we want to solve rewrites as follows:

$$\max_{\mathcal{P}\in\mathrm{Part}_d^k,\theta\in\Theta} P\left[\log(\hat{f}_{\mathcal{P}}^{\theta})\right] \tag{13}$$

$$= \max_{\mathcal{P}\in\mathrm{Part}_d^k,\theta\in\Theta} \sum_{S\in\mathcal{P}} P\left[\log(\hat{f}_S^{\theta_{\mathcal{P}}(S)})\right] \tag{14}$$

$$= \max_{\mathcal{P}\in\mathrm{Part}_d^k} \sum_{S\in\mathcal{P}} \max_{\theta\in\Theta} P\left[\log(\hat{f}_S^{\theta_{\mathcal{P}}(S)})\right] \tag{15}$$

With this formulation, it appears that the optimization of $\theta_{\mathcal{P}}$ can be done through independent optimizations of the parameters $(\theta_{\mathcal{P}}(S))_{S\in\mathcal{P}}$. What is more, if the same subset S is shared by two partitions $\mathcal{P}$ and $\mathcal{P}'$ we have:

$$\arg\max_{\theta_{\mathcal{P}}(S)} P\left[\hat{f}_S^{\theta_{\mathcal{P}}(S)}\right] = \arg\max_{\theta_{\mathcal{P}'}(S)} P\left[\hat{f}_S^{\theta_{\mathcal{P}'}(S)}\right]. \tag{16}$$

Then it is only necessary to consider a hyperparameter space indexed by Set_d^k: $\Theta = (\theta(S))_{S\in\mathrm{Set}_d}$. We can rewrite the optimization task as follows:

$$\max_{\mathcal{P}\in\mathrm{Part}_d^k} \sum_{S\in\mathcal{P}} \left\{\max_{\theta\in\Theta} P\left[\log(\hat{f}_S^{\theta(S)})\right]\right\}. \tag{17}$$

Then under KL loss, hyperparameters optimization and partitions selection can be decoupled, leading to the necessity of computing S_d^k density estimators instead of B_d^k. As highlighted previously, it leads to an appreciable gain in terms of algorithmic complexity.

Empirical Formulation of the Optimization Problem The rephrasing above indicates that under KL loss, hyperparameters optimization and partition selection become two separated tasks. This decoupling incites us to design an algorithm consisting of two steps: first, compute a marginal estimation for all subsets of features and then find the best combination. Unfortunately, the optimization problem equation (17) cannot be solved as it requires the knowledge of P. Here we explain how we construct an empirical version of it.

Let n and m be two positive integers such that $m + n = N$. The dataset $X_1, \ldots, X_N$ is split into two disjoint subsamples:

- $W_1, \ldots, W_m$ used to compute marginal estimators $(\hat{f}_S)_{S \in \mathrm{Set}_d}$
- $Z_1, \ldots, Z_n$ used to compute empirical log-likelihoods $(\ell_n(S))_{S \in \mathrm{Set}_d}$ for these estimators. We have $\ell_n(S) = \frac{1}{n} \sum_{i=1}^{n} \log\left(\hat{f}_S(Z_i)\right)$

Let $\ell_n(\mathcal{P}) = \sum_{S \in \mathcal{P}} \ell_n(S)$, the empirical optimization task can be written as:

$$\max_{\mathcal{P} \in \mathrm{Part}_d^k} \ell_n(\mathcal{P}) = \max_{\mathcal{P} \in \mathrm{Part}_d^k} \sum_{S \in \mathcal{P}} \ell_n(S). \tag{18}$$

Partition Selection A naive approach to solve equation (18) is to compute $\ell_n(\mathcal{P})$ for every partition of Part_d^k and then find the optimal one. However, this approach becomes time-consuming as d grows. Therefore, it will be appreciable to reformulate this optimization to speed up computation. It is possible through linear programming under constraints reformulation of equation (18):

Solve:

$$\max_{x \in \mathbb{R}^{\mathrm{Set}_d^k}} \sum_{S \in \mathrm{Set}_d^k} \ell_n(S) x(S) \tag{19}$$

Under constraints:

$$Ax = (1, \ldots, 1)^{\mathrm{T}} \tag{20}$$

$$x \in \{0, 1\}^{S_d^k} \tag{21}$$

Where x is a binary vector representing which elements of Set_d^k are selected, and A is a $d \times S_d^k$ matrix where each column is a binary vector representing the composition of one of the sets of Set_d^k. The condition $Ax = (1, \ldots, 1)^{\mathrm{T}}$ then ensures that each feature is chosen once, implying that the sets selected through x form a partition.

We validate this approach through a running time comparison (see table below) between the implementation of a naive approach and a linear program

solver. In this experiment, we fix the quantities $(\ell_n(S))_{S\in \mathrm{Set}_d^k}$, the naive approach consists in a for loop (implemented here in Python), computing $\ell_n(\mathcal{P})$ for all $\mathcal{P} \in \mathrm{Part}_d^k$ and returning the maximum. For the LP formulation, computations are made with the Python package PulP [Mitchell et al., 2011]. With the naive formulation and choice $k = d$, partition selection takes approximately 3 hours in dimension 15 but less than 10 seconds with LP formulation.

d	9	10	11	12	13	14	15
Naive Formulation	0.2	0.9	5.2	32.5	219.9	1304.4	10437.5
LP Formulation	0.1	0.2	0.4	0.8	1.9	4.1	9.1

Table 3: Running Times (seconds): linear programming vs naive approach for partition selection

Conclusion The resulting algorithm is algorithm 1. It enjoys the following properties:

- It exploits the decoupling of marginal density estimation and partition selection offered by choice of KL as discrepancy measure: it optimizes over partitions in Part_d^k even if it only requires the computation of Set_d^k marginal estimators
- It is versatile: it is useable with any multivariate density estimation algorithm as an input

4 Experiments

4.1 Synthetic Data

In this section, we show the performance of ISDE on simulated data satisfying IS. To illustrate the versatility of our method, we apply it in two scenarios: Gaussian framework and nonparametric framework.

input : $X_1, \dots, X_N \in \mathbb{R}^d$, k integer with $k \leq d$, integers m and n and a subroutine to perform multidimensional density estimation
output: Partition $\hat{\mathcal{P}} \in \mathrm{Part}_d^k$, marginal estimates $(\hat{f}_S)_{S \in \hat{\mathcal{P}}}$
begin
  for $S \in \mathrm{Set}_d^k$ **do**
    Compute $\hat{f}_S(W_1, \dots, W_m)$ thanks to the density estimation subroutine
    Compute $\ell_n(S)$
  end
  Compute $\hat{\mathcal{P}} \in \underset{\mathcal{P} \in \mathrm{Part}_d^k}{\arg\max} \sum_{S \in \mathcal{P}} \ell_n(S)$ using linear programming formulation
end

Algorithm 1: ISDE

4.1.1 Gaussian Data with IS

Data Generating Process The Gaussian Graphical Models (GGM) theory indicates that edges of the undirected graphical model associated with a Gaussian distribution $\mathcal{N}(0, \Sigma)$ are the non-zero entries of the precision matrix Σ^{-1}. As the inverse operator preserves the block-diagonal structure, we can easily simulate data from a multivariate Gaussian with an IS.

For a positive integer s and a real number $\sigma \in (0, 1)$ we denotes by Σ_σ^s the $s \times s$ matrix whose diagonal entries are 1 and nondiagonal entries are σ. Then for a list of positive integers $S = [s_1, \dots s_K]$ we define the block diagonal matrix:

$$\Sigma_\sigma^S = \begin{pmatrix} \Sigma_\sigma^{s_1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \Sigma_\sigma^{s_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \Sigma_\sigma^{s_K} \end{pmatrix} \tag{22}$$

The distribution $\mathcal{N}\left(0, \Sigma_\sigma^S\right)$ satisfies the IS condition with partition $\left(\left\{\sum_{i=1}^{j-1} s_i + 1, \dots, \sum_{i=1}^{j} s_i\right\}\right)_{j=1,\dots,K}$.

Evaluation Scheme If $\hat{\Sigma}$ and Σ are respectively the estimated and the true covariance, the Kullback-Leibler risk can be explicitly computed (see appendix A.2):

$$\mathrm{KL}\left(\mathcal{N}(0,\Sigma)\|\mathcal{N}(0,\hat{\Sigma})\right) = \sum_{v\in \mathrm{Sp}(A)} \frac{v - \log(1+v)}{2} \tag{23}$$

where $A = (\hat{\Sigma}^{-1} - \Sigma^{-1})\Sigma$.

Benchmarked Methods Two methods will be compared to ISDE for the task of covariance estimation.

The first estimator is the simple **Empirical Covariance**, which is the maximum likelihood estimator if the covariance does not enjoy any particular structure.

The second estimator is **Block-Diagonal Covariance Selection** (BDCS) developed in [Devijver and Gallopin, 2018]. It aims to estimate an IS in the context of GGM. This algorithm works in two steps:

- Compute a family of nested partitions candidates to be the IS
- Choose a partition in this family using a slope heuristic approach

More details can be found in the original paper. Up to our knowledge, this is the only work dealing specifically with IS in the GGM framework.

ISDE Inputs We run algorithm 1 with $k = d$, $m = n = 0.5 \times N$ and simple empirical covariance as multivariate density estimator.

Performance We compare the three methods described above for fixed σ, N, and different structures S. We have gathered results in terms of KL loss are in table 4. We have repeated each experiment 5 times, and the scores displayed are the mean KL losses and standard deviation over these repetitions.

S	[2, 2]	[4, 4, 1]	[4, 3, 2, 3]	[4, 4, 3, 3 ,2]
ISDE	$\mathbf{0.60 \pm 0.21}$	1.88 ± 0.52	2.85 ± 0.60	5.30 ± 0.96
BDCS	$\mathbf{0.60 \pm 0.21}$	$\mathbf{1.72 \pm 0.46}$	$\mathbf{2.63 \pm 1.01}$	$\mathbf{4.42 \pm 1.80}$
Empirical	0.80 ± 0.20	3.62 ± 0.53	6.88 ± 0.84	12.63 ± 0.83

Table 4: Gaussian: KL Losses ($.10^3$) - $\sigma = 0.7$, $N = 6000$

Recovery We are interested not only in performance, but we also want to find the correct partition in order to get qualitative information about datasets. In table 5 we collect, for the same experiment as above, the rate of recovery of the proper partition. In parentheses is displayed the rate of admissible output partition: a partition is admissible if all the blocks of the original partition are subsets of blocks of this one.

S	[2, 2]	[4, 4, 1]	[4, 3, 2, 3]	[4, 4, 3, 3 ,2]
ISDE	100%(100%)	80%(100%)	40%(100%)	0%(100%)
BDCS	100%(100%)	100%(100%)	80%(100%)	60%(100%)

Table 5: Gaussian: Recovery - $\sigma = 0.7$, $N = 6000$

Conclusion We remark that BDCS is the most efficient method for the task of density estimation in GGM under IS. We can explain it as ISDE tends to select admissible partition but fails to select the exact IS when the dimension grows. BDCS inherently penalizes more useless blocks merging, making it more accurate in this setting.

However, ISDE performs significantly better than a naive empirical covariance, proving that it benefits from the IS.

We want to highlight the difference between ISDE and BDCS. BDCS starts by selecting a family of up to d nested partitions and then selects among them. This approach uses a preliminary covariance estimator to design this family of nested partitions. This approach is reasonable as for Gaussian data, pairwise dependencies entirely determine multidimensional dependencies between features. Outside the scope of GGM, this approach does not remain valid as features of a random variable can be pairwise independent but mutually dependent. ISDE can

handle more general settings as it selects among a set of partitions with blocks of cardinal potentially more significant than 2.

4.1.2 Nonparametric Data with IS

Data Generating Process For a given list of positive integer (structure) $S = [s_1, \ldots, s_K]$, the data generating process is defined as follows. For each $s_i \in S$, we define a s_i dimensional dataset drawn from P_i:

- If $s_i = 1$, P_i is the uniform distribution over $[0, 1]$
- If $s_i = 2$, P_i is a distribution corresponding to data sample near two concentric circles with different radii
- If $s_i = 3$, a sample X from P_i is obtained as follows: let Y_1 and Y_2 be two independent Bernoulli variables with probability of success 0.5 and $Y_3 = |Y_1 - Y_2|$. X is then drawn from the multivariate Gaussian distribution $\mathcal{N}((Y_1, Y_2, Y_3), 0.08 \times I_3)$. This is a situation where features of P_i are pairwise independent but not mutually independent
- If $s_i \geq 4$, P_i is a mixture of two multivariate Gaussian distributions, one centered in $(0, \ldots, 0)$, the other in $(1, \ldots, 1)$

The final dataset results from their concatenation, plus featurewise rescaling so that each value lies between 0 and 1. This rescaling step does not affect the IS as it is done featurewise.

Evaluation Scheme Here the Kullback-Leibler loss between true density f and an estimator $\hat{f}$ is not computable. In order to evaluate the performance of an estimator, we compute the empirical log-likelihood on a validation set $X^{\text{valid}} = X_1^{\text{valid}}, \ldots, X_M^{\text{valid}}$ drawn independently from the same distribution as $X_1, \ldots, X_N$:

$$\text{Score}(\hat{f}) = \frac{1}{M} \sum_{i=1}^{M} \log\left(\hat{f}\left(X_i^{\text{valid}}\right)\right). \tag{24}$$

The set $X^{\text{valid}} = X_1^{\text{valid}}, \ldots, X_M^{\text{valid}}$ is never used to estimate $\hat{f}$. In the experiments below, we set $M = 5000$.

Benchmarked Methods Two estimators will be compared to ISDE for the task of nonparametric multivariate density estimation.

The first one is **Cross-Validated Kernel Density Estimator** (CVKDE) with Gaussian kernel. For a given bandwidth $h > 0$ we define the Gaussian kernel density estimator associated to h as

$$\hat{f}_h(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{\exp\left(-\frac{(X_i - x)^{\mathrm{T}}(X_i - x)}{2h^2}\right)}{(2\pi)^{d/2} h^d}. \tag{25}$$

The final estimator is $\hat{f}_{\hat{h}}$ where $\hat{h}$ is selected through a cross-validation scheme in order to maximize empirical log-likelihood. We ran our experiments with a 5-fold cross-validation and tested bandwidths belonging to a regular grid on a log-scale from 0.01 to 1 with 30 values.

The second one is **Forest Density Estimation** (FDE) , designed to estimate a graphical model for nonparametric densities ([Liu et al., 2011]). The estimated graph is a forest, a graph without a cycle. In this case, the density can be expressed only through 1 and 2-dimensional marginals. If $G = (V, E)$ is a forest, a random variable with density f admitting G as a graphical model enjoys the following formulation

$$f(x) = \prod_{(i,j) \in E} \frac{f_{\{i,j\}}(x_i, x_j)}{f_{\{i\}}(x_i) f_{\{j\}}(x_j)} \prod_{k=1}^{d} f_{\{k\}}(x_k). \tag{26}$$

Estimating such a density only requires the estimation of marginals up to dimension 2. We consider estimators of the form equation (25). Then for each couple of features, a score is computed quantifying the loss of information induced by assuming they are independent. After that, a preliminary tree (connected graph without cycle) is built using Kruskal's algorithm, and then the tree is pruned using held-out data. More details can be found in the original paper. As the authors do not provide an empirical bandwidth selection method, we plug the bandwidths parameters learned by a cross-validation scheme, as described above.

ISDE Inputs We run algorithm 1 with $k = d$, $m = n = 0.5 \times N$. For density estimation subroutine, we tested two options: CVKDE as presented above (ISDE_CVDE) and kernel density estimator with Gaussian kernel with a fixed h equals to 0.05 (ISDE_Fixed_h).

Performance table 6 shows empirical log-likelihood on validation data for methods listed above and for different ISs.

	[2, 2, 1]	[3, 3, 3]	[4, 4, 2, 2]
ISDE_CVKDE	$\mathbf{1.83 \pm 0.08}$	$\mathbf{4.05 \pm 0.15}$	$\mathbf{6.30 \pm 0.25}$
ISDE_Fixed_h	1.01 ± 0.02	4.04 ± 0.14	5.55 ± 0.25
FDE	$\mathbf{1.83 \pm 0.08}$	2.88 ± 0.14	5.89 ± 0.33
CVKDE	0.56 ± 0.03	3.49 ± 0.11	3.96 ± 0.16

Table 6: Nonparametric: Empirical log-likelihood on validation data - $N = 5000$

Recovery table 7 shows the recovery rates of the IS for ISDE_CVDE and ISDE_Fixed_h.

	[2, 2, 1]	[3, 3, 3]	[4, 4, 2, 2]
ISDE_CVDE	100%(100%)	100%(100%)	100%(100%)
ISDE_Fixed_h	100%(100%)	100%(100%)	100%(100%)

Table 7: Nonparametric: Recovery - $N = 5000$

Conclusion For $[2, 2, 1]$, ISDE_CVKDE and FDE give similar results as they output the same graph and the same bandwidths. ISDE_CVKDE has better results than ISDE_Fixed_h even if they recover the proper IS because of bandwidths optimization in ISDE_CVKDE.

For $[3, 3, 3]$, as features are pairwise independent, FDE outputs at every try a graph without any edge and computes the density as a product of one-dimensional marginals, leading to poor results in comparison to ISDE_CVKDE and ISDE_Fixed_h. Here no difference is observed between ISDE_Fixed_h and ISDE_CVKDE. An explanation is that the parameter $h = 0.05$ is not far from optimized bandwidths.

For $[4, 4, 2, 2]$, FDE outputs a subgraph of the actual graphical model at every try. It leads to better estimation than CVKDE but worse than ISDE_CVDE and ISDE_Fixed_h, which learn the proper IS at every try.

Thus, ISDE_Fixed_h and ISDE_CVKDE lead to better results than FDE for the task of structured density estimation under KL loss under IS. ISDE_CVKDE outperforms ISDE_Fixed_h as it optimizes over a set of bandwidths for every marginal estimator. The recovery study indicates that we can learn IS using ISDE_Fixed_h or ISDE_CVKDE indifferently. However, if the running time is not critical, we recommend using a cross-validation scheme to select bandwidths.

Here we remark that we recover exactly the IS for the considered settings. One can wonder why we do not observe, as it was the case for GGM, that admissible partitions are outputted but not precisely the IS. We believe that this is because, in a nonparametric scenario, a useless merging of blocks in the partition is strongly penalized by ISDE as the dimension has a more substantial negative impact on our ability to estimate a density than in a Gaussian setting. Then the hold-out scheme implemented in ISDE (by splitting X into W and Z in algorithm 1) penalizes efficiently too big blocks in partitions and leads to accurate recovery of IS.

4.2 Real-world data

This section is devoted to the presentation of some outputs on real-world datasets. In addition to studying the performance of ISDE in terms of log-likelihood, it is the occasion to illustrate how we can interpret the outputted partition.

Datasets The datasets presented here are the output of mass cytometry experiments. Cytometry allows high-throughput measurements at a single-cell level over a cell sample. Two types of information about cells are collected. Some are about the cell's geometry, and others about the abundance of some targeted proteins at their surface. The number of events for cytometry experiments on blood samples usually lies between $10,000$ and $1,000,000$, and the number of features can vary from a few ones to approximately 50.

We present here results on two public cytometry datasets, used in a benchmark of clustering methods paper [Weber and Robinson, 2016], Levine13 and Levine32. Both are experiments on bone marrow cells extracted from healthy human donors with 13 and 32 features.

4.2.1 Quantitative evaluation

Benchmarked Algorithms For these experiments, the assumption that the data follow a multivariate Gaussian distribution is irrelevant, and the associate methods for estimating density lead to poor results, then we did not include them in this benchmark.

We have decided to compare FDE, CVKDE, and ISDE_CVKDE (the value of k depends on the dimension, we selected $k = 3$ for Levine32 and $k = 5$ for Levine13 to keep computations fast).

In addition, we have added a more realistic parametric approach in our configuration, namely a Gaussian Mixture (GM) model with a selection of the number of components. This model has seemed particularly adapted to cytometry as we naturally expect in this context that the data forms clusters representing cell populations.

Let n_C be a positive integer corresponding to the number of components in the mixture. Let $p = (p_1, \ldots, p_{n_C})$ be a collection of nonnegative real number such that $\sum_{i=1}^{n_C} p_i = 1$, $\mu = (\mu_1, \ldots, \mu_{n_C})$ a collection of vector in $\mathbb{R}^d$ and $\Sigma = (\Sigma_1, \ldots, \Sigma_{n_C})$ a collection of $d \times d$ definite positive matrices. The density $f_{(n_C,p,\mu,\Sigma)}$ of the Gaussian mixture model associated with the parameters (n_C, p, μ, Σ) is

$$f_{(n_C,p,\mu,\Sigma)} = \sum_{i=1}^{n_C} p_i f_{\mu_i,\Sigma_i} \tag{27}$$

where f_{μ_i,Σ_i} is the density of the multivariate Gaussian random variable with mean μ_i and covariance matrix Σ_i.

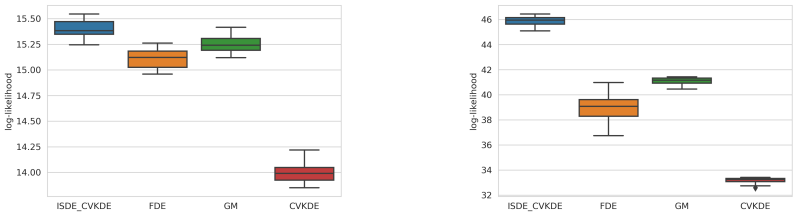
Given n_C and a dataset, it is possible to compute estimate parameters $(\hat{p}, \hat{\mu}, \hat{\Sigma})$ with the EM algorithm [Dempster et al., 1977]. As we do not know the

optimal number of components in advance, a strategy is to fit a Gaussian mixture model for different values of n_C (from 1 to 30 in our experiments) and select the number of components in the mixture with a cross-validation scheme. We rely on the implementation of these methods provided by scikit-learn [Buitinck et al., 2013].

Testing ISDE against other density estimation methods is a way to evaluate how this model can explain the data well, but we have to be careful and keep in mind that we do not have any way to ensure that data enjoys an IS.

Experimental Setup From each dataset we have extracted a train sample with $N = 5000$ events, this train sample is exclusively used to compute estimators $\hat{f}_{\text{CVKDE}}$, $\hat{f}_{\text{FDE}}$, $\hat{f}_{\text{ISDE\_CVKDE}}$ and $\hat{f}_{\text{GM}}$. For ISDE_CVKDE we fixed $m = 3000$ and $n = 2000$. Then to compare between these density estimators, we sampled 20 datasets with 2000 events from the data that were not used to compute estimators.

Results Outputs of our experiments can be visualized in figure 1.



(a) Levine13 (b) Levine32

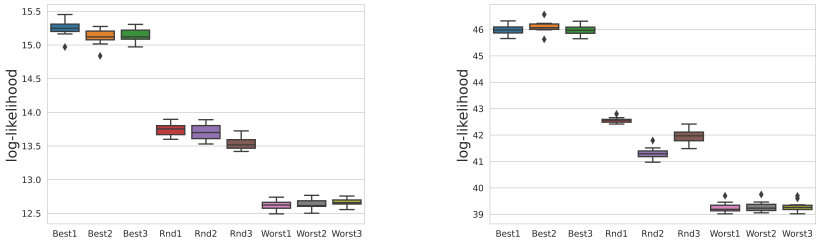
Figure 1: Comparison of empirical log-likelihood on validation data for different density estimation methods

We remark that using ISDE_CVKDE leads to better empirical log-likelihood on validation data. CVKDE in the ambient dimension is always the worst estimator. GM is slightly better than FDE for both datasets, and the gap between performances of FDE/GM and ISDE_CVKDE is higher in dimension 32 than in dimension 13. We conclude that the approach of ISDE with a limited size of blocks seems to be a relevant model for these datasets as it could outperform other model-based approaches in terms of log-likelihood.

4.2.2 Qualitative Interpretation

We believe that the added value of our method is that ISs are easy to understand and useable as a tool to interpret data. After validating the pertinence of ISDE in comparison with other methods through quantitative analysis, we now provide some insight into the capacity of ISDE to deliver meaningful qualitative information.

Nontriviality of Outputted Partition The first question to ask is if the gain in terms of empirical log-likelihood is due to the specific outputted partition $\hat{\mathcal{P}}$ or if any other partition of features in Part_d^k with a fixed value of k could achieve it. To answer this question, we have computed empirical log-likelihood on 10 validation sets of size $2,000$ for the three best partitions outputted by ISDE, the three worst ones regarding the optimization task, and three random partitions in Part_d^k. To compute not the optimal, but the second one, the third one, and so on, it suffices to add constraints on the partition selection problem that artificially exclude some partitions from the optimization. To compute the worst partitions, switching the optimization from maximization to minimization suffices. Random partitions are computed by generating a random permutation σ of $\{1, \ldots, d\}$ and then gather consecutive features in $\{\sigma(1), \ldots, \sigma(d)\}$ in groups with sizes drawn uniformly between 1 and k.



(a) Levine13 (b) Levine32

Figure 2: Comparison of empirical log-likelihood on validation data for best, worst and random partitions

These experiments seem to indicate that

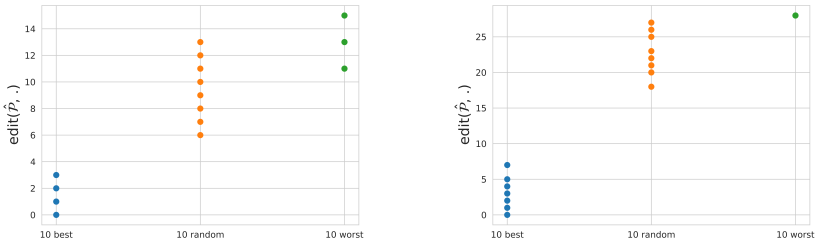
- ISDE outputted specific partitions that lead to better estimators in terms of log-likelihood on empirical data than the random partitions. In that sense, the information provided by ISDE on these datasets is not trivial.
- not only the optimal one $\hat{\mathcal{P}}$ but a collection of partitions seem to lead to optimal scores.

With that in mind, it could be interesting to determine if the collection of partitions leading to optimal results are close in some sense. To this end, it is necessary to introduce a notion of distance between partitions.

Edit Distance given two partition $\mathcal{P}$ and $\mathcal{P}'$ in Part_d^k it is possible to define a distance between $\mathcal{P}$ and $\mathcal{P}'$ called edit distance ([Brown et al., 2007]) and denoted by $\text{edit}(\mathcal{P}, \mathcal{P}')$. This distance corresponds to the minimal number of operations required to go from $\mathcal{P}$ to $\mathcal{P}'$ where an operation can split a block into two ones or merge two blocks. The edit distance defines a distance on Part_d^k in the mathematical sense as it is nonnegative, symmetric, equal to zero only if we compute the distance from one partition to itself and enjoys triangular inequality.

Correlation between Edit Distance and Density Estimation We will now see how the edit distance from $\hat{\mathcal{P}}$ to $\mathcal{P}$ correlates with the empirical log-likelihood on validation data for $\hat{f}_{\mathcal{P}}$.

Firstly, we can visualize the edit distance from $\hat{\mathcal{P}}$ to the 10 best partitions (excluding $\hat{\mathcal{P}}$) in the sense of the problem of partition selection, 10 random partitions, and the 10 worst partitions.



(a) Levine13 (b) Levine32

Figure 3: Edit distance from $\hat{\mathcal{P}}$ for 10 best, 10 random and 10 worst partitions

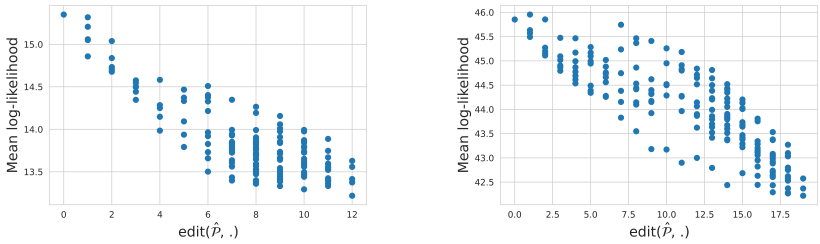
These observations seem to correlate well with what we have observed previously in terms of log-likelihood.

Secondly, we explore the space Part_d^k by defining a random walk considering the topology induced by edit. We define a random walk $(\mathcal{P}_0, \mathcal{P}_1, \dots)$ as follows: at each step we go from $\mathcal{P}_i$ to $\mathcal{P}_{i+1}$ with $\text{edit}(\mathcal{P}_i, \mathcal{P}_{i+1}) = 1$. To do so, it suffices to randomly choose an operation (edit or merge) and apply it to randomly selected block(s) of $\mathcal{P}_i$ while controlling that we stay in Part_d^k.

To observe a possible correlation between $\text{edit}(\hat{\mathcal{P}}, .)$ and log-likelihood on validation data, we have implemented the following protocol

- do 5 walks of length 40 with $\hat{\mathcal{P}}$ as starting point and store all visited partitions.
- for the 200 selected partitions, compute empirical log-likelihood on ten resampling of validation data and store the mean value.

Then we plot these scores against $\text{edit}(\hat{\mathcal{P}}, .)$.



(a) Levine13 (b) Levine32

Figure 4: Mean log-likelihood on validation data with respect to edit distance from $\hat{\mathcal{P}}$ for the partitions visited by the random walk

For both datasets, we observe a clear negative correlation between edit$(\hat{\mathcal{P}}, .)$ and empirical log-likelihood on validation data. These observations indicate that the topology induced by the distance edit on Part_d^k is meaningful in the sense that the more a partition $\mathcal{P}$ far from $\hat{\mathcal{P}}$, the less optimal is the estimator $\hat{f}_{\mathcal{P}}$

Exhaustive Analysis For the dataset Levine13, as the cardinal of Part_{13}^5 is $25,719,630$, it is possible to store the entire family of empirical log-likelihood computed thanks to the data $Z_1, \ldots, Z_n$ on ISDE: $(\ell_n(\mathcal{P}))_{\mathcal{P}\in\text{Part}_{13}^5}$. Such an exhaustive analysis is impossible for Levine32 as the number of partitions in Part_{32}^3 exceed 10^{19}. The distribution of $(\ell_n(\mathcal{P}))_{\mathcal{P}\in\text{Part}_{13}^5}$ can be visualized thanks to an histogram.

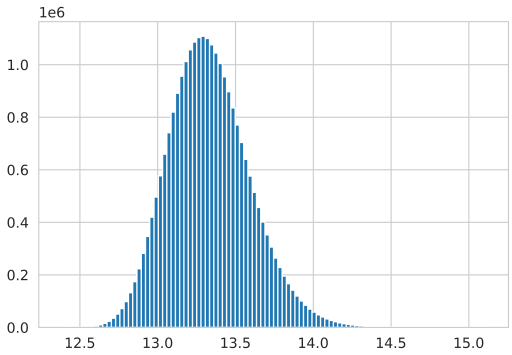


Figure 5: Distribution of $(\ell_n(\mathcal{P}))_{\mathcal{P}\in\text{Part}_{13}^5}$

If we select the partition with a score higher than 14.6, it remains $1,941$ elements. Then for these, we compute empirical log-likelihood again on validation data and represent it against edit$(\hat{\mathcal{P}}, .)$. This is a way to ask the uniqueness of the optimal partition $\hat{\mathcal{P}}$. If another partition $\mathcal{P}$ a significantly positive value of edit$(\hat{\mathcal{P}}, \mathcal{P})$ gives as good results as $\hat{\mathcal{P}}$, it will indicate that there are other local maximums than $\hat{\mathcal{P}}$.

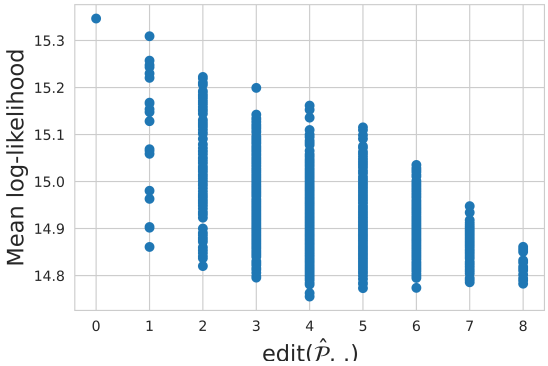


Figure 6: Mean log-likelihood on validation data with respect to edit distance from $\hat{\mathcal{P}}$ for $1,941$ best partitions

Conclusion This analysis of the space Part_d^k equipped with edit distance in terms of empirical log-likelihood for $\hat{f}_{\mathcal{P}}$ has led us to the conclusion that the qualitative information provided by ISDE through $\hat{\mathcal{P}}$ is nontrivial for these datasets as random partitions in Part_d^k does not lead to optimal scores. We also prove that the density estimation score deteriorates as the edit distance from $\hat{\mathcal{P}}$ increases, indicating that edit distance is a relevant metric to explore Part_d^k in the context of density estimation under IS. Then an exhaustive analysis on Levine13 indicates that we can consider the optimal partition as unique.

These conclusions depend on the specific datasets presented here and could become invalid for other ones. We did not test our method on other datasets than ones from mass cytometry, but as we provide the code to reproduce our experiments, our aim is that anyone interested in the method can replicate these analyses for other data.

5 Running time

In this section, we analyze ISDE in terms of running time for the settings presented in section 4. We have run All experiments on a laptop with the following hardware: CPU Intel(R) Xeon(R) W-10885M CPU @ 2.40GHz and GPU: Nvidia Quadro RTX 3000 Mobile. All computations involving Gaussian kernels have been performed on GPU using the KeOps package [Charlier et al., 2021]. Note that the same code can be run without a GPU as KeOps automatically parallelize on CPU if there is no available GPU.

Our goal in this section is not to compare the running time of ISDE with other presented methods but rather to highlight the fact that its running time is acceptable for reasonable data size.

The running time of algorithm 1 is influenced by the subroutine for marginal density estimations and by the parameters d, k, m and n. Here we give running times for experimental settings described in section 4.1.

5.1 Synthetic Data

For the GGM scenario, we considered empirical covariance as the subroutine for multivariate density estimation. table 8 shows the running time of ISDE for the parameters described in section 4.1.

d	4	9	12	16
Time (seconds)	0.020	0.346	3.005	60.099

Table 8: Mean running times (seconds): ISDE with Empirical Covariance

For the nonparametric scenario, we used both CVKDE and a Gaussian KDE with fixed bandwidth as subroutines for multivariate density estimation. table 9 shows the running time of ISDE_CVKDE and ISDE_Fixed_h for the parameters described in section 4.1:

d	5	9	12
ISDE_CVKDE	15.961	297.235	2611.502
ISDE_Fixed_h	0.143	2.392	21.217

Table 9: Mean running times (seconds): ISDE_CVKDE and ISDE_Fixed_h

For $m = n = 2500$ and in dimension 15, ISDE_Fixed_h runs in approximately 30 minutes. ISDE_CVKDE (5-fold cross validation over 30 values for bandwidths) runs in approximately 43 minutes in dimension 12. To cover the moderately high-dimensional setting keeping running times reasonable, it is possible to use ISDE_Fixed_h instead of ISDE_CVKDE if data has more than 11 or 12 features.

5.2 Real Data

The running time for ISDE_CVKDE in the conditions presented above was 29min 2s for Levine13 and 1hr 1min 5s for Levine32.

6 Conclusion

ISDE is an algorithm that outputs an estimate of a density function of a point cloud, taking into account an IS for data in moderately high dimension. To design it, we reduced the number of hyperparameters with an appropriate choice of the loss function and, through linear programming reformulation, made the partition selection step faster than was previously possible. This leads to reasonable running time even on a laptop for the considered datasets. The code is available and ready to be used by anyone interested in this method.

As mentioned before, ISDE is versatile: it takes any basic multidimensional density estimator as input so that it can be used in parametric as well as in nonparametric frameworks. It is also exhaustive as it searches over all partitions of features with given maximal block size. To our knowledge, we are the first to

propose a method that takes into consideration IS in the context of nonparametric density estimation with kernel density estimators.

We validated its performance on synthetic data on GGM and on a non-parametric framework under IS. This performance was measured in terms of KL loss, comparatively with other methods, and IS recovery. Applying ISDE to mass cytometry data has indicated that it could accurately estimate density over real-world datasets and extract qualitative information about their features through the outputted partition.

Code availability The code to reproduce the experiments presented here are available at https://github.com/Louis-Pujol/ISDE-Paper.

Data availability Original datasets were downloaded from the repository presented in [Weber and Robinson, 2016] and available at the address https://flowrepository.org/id/FR-FCM-ZZPH.

Declarations of interest None.

Acknowledgement The author is thankful to Marc Glisse[1] and Pascal Massart[2] for their constructive remarks on this work.

Funding This work was supported by the program Paris Region Ph.D. of DIM Mathinnov and was partly supported by the French ANR Chair in Artificial Intelligence TopAI - ANR-19-CHIA-0001.

[1]Inria Saclay
[2]Université Paris-Saclay

References

[Brown et al., 2007] Brown, D. P., Krishnamurthy, N., and Sjölander, K. (2007). Automated protein subfamily identification and classification. *PLoS computational biology*, 3(8):e160.

[Buitinck et al., 2013] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

[Cai et al., 2010] Cai, T. T., Zhang, C.-H., and Zhou, H. H. (2010). Optimal rates of convergence for covariance matrix estimation. *The Annals of Statistics*, 38(4):2118–2144.

[Campello et al., 2013] Campello, R. J., Moulavi, D., and Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer.

[Chandola et al., 2009] Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58.

[Charlier et al., 2021] Charlier, B., Feydy, J., Glaunès, J. A., Collin, F.-D., and Durif, G. (2021). Kernel operations on the gpu, with autodiff, without memory overflows. *Journal of Machine Learning Research*, 22(74):1–6.

[Chazal et al., 2013] Chazal, F., Guibas, L. J., Oudot, S. Y., and Skraba, P. (2013). Persistence-based clustering in riemannian manifolds. *Journal of the ACM (JACM)*, 60(6):1–38.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

[Devijver and Gallopin, 2018] Devijver, E. and Gallopin, M. (2018). Block-diagonal covariance selection for high-dimensional gaussian graphical models. *Journal of the American Statistical Association*, 113(521):306–314.

[Friedman et al., 2008] Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

[Giraud, 2014] Giraud, C. (2014). *Introduction to high-dimensional statistics*. Chapman and Hall/CRC.

[Goldenshluger and Lepski, 2014] Goldenshluger, A. and Lepski, O. (2014). On adaptive minimax density estimation on r d. *Probability Theory and Related Fields*, 159(3):479–543.

[Hasminskii et al., 1990] Hasminskii, R., Ibragimov, I., et al. (1990). On density estimation in the view of kolmogorov's ideas in approximation theory. *The Annals of Statistics*, 18(3):999–1010.

[Lepski, 2013] Lepski, O. (2013). Multivariate density estimation under sup-norm loss: oracle approach, adaptation and independence structure. *Annals of Statistics*, 41(2):1005–1034.

[Liu et al., 2011] Liu, H., Xu, M., Gu, H., Gupta, A., Lafferty, J., and Wasserman, L. (2011). Forest density estimation. *The Journal of Machine Learning Research*, 12:907–951.

[Mitchell et al., 2011] Mitchell, S., Consulting, S. M., and Dunning, I. (2011). Pulp: A linear programming toolkit for python.

[Rebelles, 2015] Rebelles, G. (2015). Lp adaptive estimation of an anisotropic density under independence hypothesis. *Electronic journal of statistics*, 9(1):106–134.

[Tsybakov, 2008] Tsybakov, A. B. (2008). *Introduction to Nonparametric Estimation*. Springer Publishing Company, Incorporated, 1st edition.

[Wainwright and Jordan, 2008] Wainwright, M. J. and Jordan, M. I. (2008). *Graphical models, exponential families, and variational inference*. Now Publishers Inc.

[Wasserman, 2004] Wasserman, L. (2004). *All of statistics: a concise course in statistical inference*, volume 26. Springer.

[Weber and Robinson, 2016] Weber, L. M. and Robinson, M. D. (2016). Comparison of clustering methods for high-dimensional single-cell flow and mass cytometry data. *Cytometry Part A*, 89(12):1084–1096.

A Appendix

A.1 Computation of B_d^2

Let us prove the following formula :

$$B_d^2 = \sum_{i=1}^{\lfloor d/2 \rfloor} \frac{\prod_{j=0}^{i-1} \binom{d-2j}{2}}{i!} \tag{28}$$

$$= 1 + \binom{d}{2} + \frac{\binom{d}{2}\binom{d-2}{2}}{2!} + \frac{\binom{d}{2}\binom{d-2}{2}\binom{d-4}{2}}{3!} \cdots + \frac{\binom{d}{2} \cdots \binom{d-2(\lfloor d/2 \rfloor -1)}{2}}{(\lfloor d/2 \rfloor)!} \tag{29}$$

For a nonnegative integer i, let us denote by $B_d^2[i]$ the number of partitions of Part_d^k with exactly i blocks of size 2. A first remark is that $B_d^2[i] = 0$ as soon as $i > \lfloor d/2 \rfloor$, then

$$B_d^2 = \sum_{i=0}^{\lfloor d/2 \rfloor} B_d^2[i]. \tag{30}$$

Now, we evaluate $B_d^2[i]$. It is not hard to count the number of possibilities to select i pairs of distinct elements of $\{1, \ldots, d\}$ taking into account in which order there were selected. For the first pair there are $\binom{d}{2}$ choices, then $\binom{d-2}{2}$ choices for selecting another pair among the other variables and so on. Then there are $\prod_{j=0}^{i-1} \binom{d-2j}{2}$ ordered pairs of variables of $\{1, \ldots, d\}$.

As selecting a partition in Part_d^k is equivalent to an unordered choice of pairs of variables, it remains to divide by the number of permutation of i elements, $i!$. Then

$$B_d^2[i] = \frac{\prod_{j=0}^{i-1} \binom{d-2j}{2}}{i!}. \tag{31}$$

A.2 Computation of $\mathrm{KL}\left(\mathcal{N}(0,\Sigma_1)\|\mathcal{N}(0,\Sigma_2)\right)$

Let us prove that if Σ_1 and Σ_2 are two covariance matrix, then

$$\mathrm{KL}\left(\mathcal{N}(0,\Sigma_1)\|\mathcal{N}(0,\Sigma_2)\right) = \sum_{v\in \mathrm{Sp}(A)} \frac{v-\log(1+v)}{2} \tag{32}$$

where $A = (\Sigma_2^{-1} - {\Sigma_1}^{-1})\Sigma_1$.

First of all, for a covariance matrix Σ, the density f_Σ of $\mathcal{N}(0,\Sigma)$ is given by

$$\forall x \in \mathbb{R}^d f_\Sigma(x) = \frac{1}{(2\pi)^{d/2}(\det\Sigma)^{1/2}} \exp\left(-\frac{1}{2}x^{\mathrm{T}}\Sigma^{-1}x\right). \tag{33}$$

We compute the KL divergence between f_{Σ_1} and f_{Σ_2}

$$\begin{aligned}
\mathrm{KL}\left(f_{\Sigma_1}\|f_{\Sigma_2}\right) &= \int \log\left(\frac{f_{\Sigma_1}(x)}{f_{\Sigma_2}(x)}\right) f_{\Sigma_1}(x)dx && (34)\\
&= \frac{1}{2}\log\frac{\det\Sigma_2}{\det\Sigma_1}\underbrace{\int f_{\Sigma_1}(x)dx}_{=1} && (35)\\
&\quad + \frac{1}{2}\underbrace{\int x^{\mathrm{T}}\Sigma_2^{-1}x f_{\Sigma_1}(x)dx}_{=\mathrm{Tr}(\Sigma_2^{-1}\Sigma_1)} && (36)\\
&\quad + \frac{1}{2}\underbrace{\int x^{\mathrm{T}}{\Sigma_1}^{-1}x f_{\Sigma_1}(x)dx}_{=\mathrm{Tr}({\Sigma_1}^{-1}\Sigma_1)=d} && (37)\\
&= \frac{1}{2}\left(\log\det\Sigma_2 - \log\det\Sigma_1 + \mathrm{Tr}\left(\Sigma_2^{-1}\Sigma_1\right) - d\right) && (38)
\end{aligned}$$

We remark that

$$\mathrm{Tr}\left(\Sigma_2^{-1}\Sigma_1\right) - d = \mathrm{Tr}(A) = \sum_{v\in\mathrm{Sp}(A)} v \tag{39}$$

We also remark that $\log\left(\frac{\det\Sigma_1}{\det\Sigma_2}\right) = \log\left(\det\Sigma_2^{-1}\Sigma_1\right)$ and as if v is an eigenvalue of A, $1+v$ is an eigenvalue of $\Sigma_2^{-1}\Sigma_1$ we have

$$\log\left(\frac{\det\Sigma_1}{\det\Sigma_2}\right) = \sum_{v\in\mathrm{Sp}(A)} \log(1+v) \tag{40}$$

Combining these results with equation (38) leads to the desired formula.