LIDL: LOCAL INTRINSIC DIMENSION ESTIMATION USING APPROXIMATE LIKELIHOOD

Piotr Tempczyk[*]
University of Warsaw
& opium.sh, Poland

Adam Goliński
University of Oxford
United Kingdom

Przemysław Spurek
Jagiellonian University
Krakow, Poland

Jacek Tabor
Jagiellonian University
Krakow, Poland

ABSTRACT

Most of the existing methods for estimating the local intrinsic dimension of a data distribution do not scale well to high dimensions because they rely on investigating the nearest neighbors structure, which may cause problems due to the curse of dimensionality. We attempt to address that challenge by proposing a new method for Local Intrinsic Dimension estimation using approximate Likelihood (LIDL) which makes use of the recent progress in likelihood estimation in high dimensions—the normalizing flow methods. We empirically show that on standard benchmarks for this problem, our method yields more accurate estimates than previous algorithms and that, unlike other methods, it scales well to problems with thousands of dimensions. What is more, we anticipate this new approach to improve further with continuing advances in the density estimation literature.

1 INTRODUCTION

One of the important problems in representation learning is estimating the intrinsic dimensionality (ID) of lower-dimensional data embedded in a higher-dimensional observed space (Ansuini et al., 2019; Li et al., 2018; Rubenstein et al., 2018), which we will refer to as IDE. It is a well-studied problem in the context of dimensionality reduction, clustering, and classification problems (Camastra & Staiano, 2016; Kleindessner & Luxburg, 2015; Vapnik, 2013). ID is also relevant for some prototype-based clustering algorithms (Claussen & Villmann, 2005; Struski et al., 2018). ID estimation can be used as a powerful analytical tool to study the process of training and representation learning in deep neural networks (Li et al., 2018; Ansuini et al., 2019). Also in the context of representation learning, Rubenstein et al. (2018) show how the mismatch between the latent space dimensionality and the dataset ID may hurt the performance of auto-encoder-based generative models like VAE (Kingma & Welling, 2014), WAE (Tolstikhin et al., 2017), or CWAE (Knop et al., 2020).

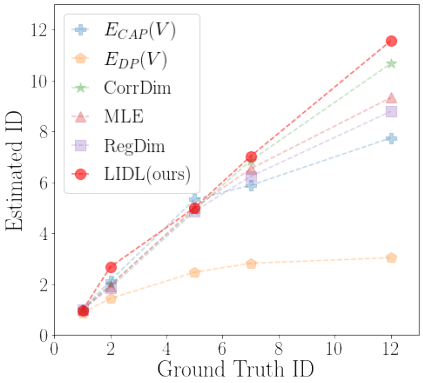

Figure 1: Comparison of LIDL and methods from Kleindessner & Luxburg (2015). Based on Table 1.

IDE methods can be divided into two broad categories: global and local (Camastra & Staiano, 2016). Global methods aim to give a single estimate of the dimensionality of the entire dataset. However, reducing the description of a dataset's ID structure to just a single number might discard the nuanced structure, such as when the data lies on a union of manifolds with different numbers of dimensions (which is the case for real-world datasets). On the contrary, the local methods (Carter et al., 2009) try to estimate the local intrinsic dimensionality (LID) of data manifold in the neighborhood of an arbitrary point on that manifold. This approach gives more insight into the nature of the dataset and provides more options to summarize the dimensionality of the manifold from the global perspective – similarly to how the estimate of the density of a random variable provides richer information than just the estimates of its summary statistics. Studying the local dimensionality allows to reason on

[*]Correspondence to: tempczyk@mimuw.edu.pl

the level of single samples, which can lead to new insights e.g. by looking at single datapoints and how their properties are related to their LID estimates. A good review and comparison of a wide range of methods for global and local ID estimation can be found in (Camastra & Staiano, 2016).

Most of the existing methods for LID estimation (Kleindessner & Luxburg, 2015; Levina & Bickel, 2004; Hino et al., 2017; Camastra & Staiano, 2016; Rozza et al., 2012; Ceruti et al., 2014; Camastra & Vinciarelli, 2002) tend to analyze the local structure of the data manifold by investigating data point's nearest neighbours structure or pairwise distances. Unfortunately, such approaches generally do not scale well to high dimensions. To address this problem, we propose a new method for Local Intrinsic Dimension estimation using approximate Likelihood (LIDL), which circumvents that challenge by making use of the recent progress in normalizing flows (NF) (Dinh et al., 2014; Rezende & Mohamed, 2015; Kingma & Dhariwal, 2018). Our method makes use of the observation that the local dimensionality of the manifold can be estimated using approximate likelihood of the data distribution perturbed with a small Gaussian noise of varying magnitude.

Our contributions are: we introduce a new way of local intrinsic dimension estimation, that uses approximate likelihood instead of a nearest neighbour approach; we show that our method outperforms other approaches on standard benchmark problems for this task (see Fig. 1); we demonstrate that our method scales to high-dimensional settings (see Table 2) and works on image datasets (Fig. 4).

2 Local Intrinsic Dimensionality

We often know that a particular dataset comes from some T-dimensional manifold, but we observe it after it has been embedded into a higher-dimensional space of $\mathbb{R}^M$, $T < M$. While such setting is common for structured problems, it is also hypothesised to hold to some extent in less structured settings, e.g., for natural images. This is termed the manifold hypothesis (Fefferman et al., 2016) and it motivates applying the intrinsic dimension estimation methods to less structured datasets. As pointed out in the introduction, the data does not necessarily come from a single T-dimensional manifold, but instead it might lie on a union of manifolds of different dimensions, which might or might not intersect. Paying particular attention to such settings is important, because they allow to showcase some advantages of the local over the global ID estimation methods. To give the problem a more formal framing we adapt a definition from (Kleindessner & Luxburg, 2015) in Appendix A.

3 Method

Let us start with building an intuition about the core observation that underpins our method. First, in the context of the definition of the problem setting from the previous section, note that in the absence of noise $_i$, the probability density of the data in the observed space $p_D(\mathbf{x})$ has to be described using a Dirac delta function. Unfortunately, this makes it impossible for computational methods to express the probability density on M because the density becomes numerically singular. This is illustrated in the top row, middle column of Fig. 2 as an example of a 1D set being embedded in a 2D space.

To circumvent this problem, consider perturbing the data points from the original distribution $p_D(\mathbf{x})$ with a small Gaussian noise $N_M(0;\ I_M)$, such that $2\ \mathbb{R}^M$, I_M denotes an M-dimensional identity matrix, is a parameter determining the noise magnitude, and N_K denotes a K-dimensional normal distribution. The summation of random variables, the original data and the noise, corresponds to a convolution of the probability density functions such that the resulting density is $p\ (\mathbf{x}) = p_D(\mathbf{x})\quad N_M(\mathbf{x}j0;\ I_M)$, which no longer causes numerical issues and can be approximated using density estimation methods.

Adding the noise not only allows us to approximate the density of the perturbed data, but is also the key to estimating the local intrinsic dimension. Locally, around every point $\mathbf{x}$, space M can be divided into two linear subspaces: subspace that is locally tangent to the manifold $M^{\mathbf{x}}_T$, and the subspace orthogonal to the manifold $M^{\mathbf{x}}_O$. The the fact the convolution with a normal distribution displaces some of the probability mass from the tangent subspace into the orthogonal subspace decreases the amount of probability mass in the tangent subspace in proportion to the dimensionality of the orthogonal subspace. See Fig. 2 to build an intuition about this process. We assume that the density along the locally tangent subspace is locally constant such that the convolution along the tangent subspace has no effect on the density in that subspace, more discussion in Appendix B.

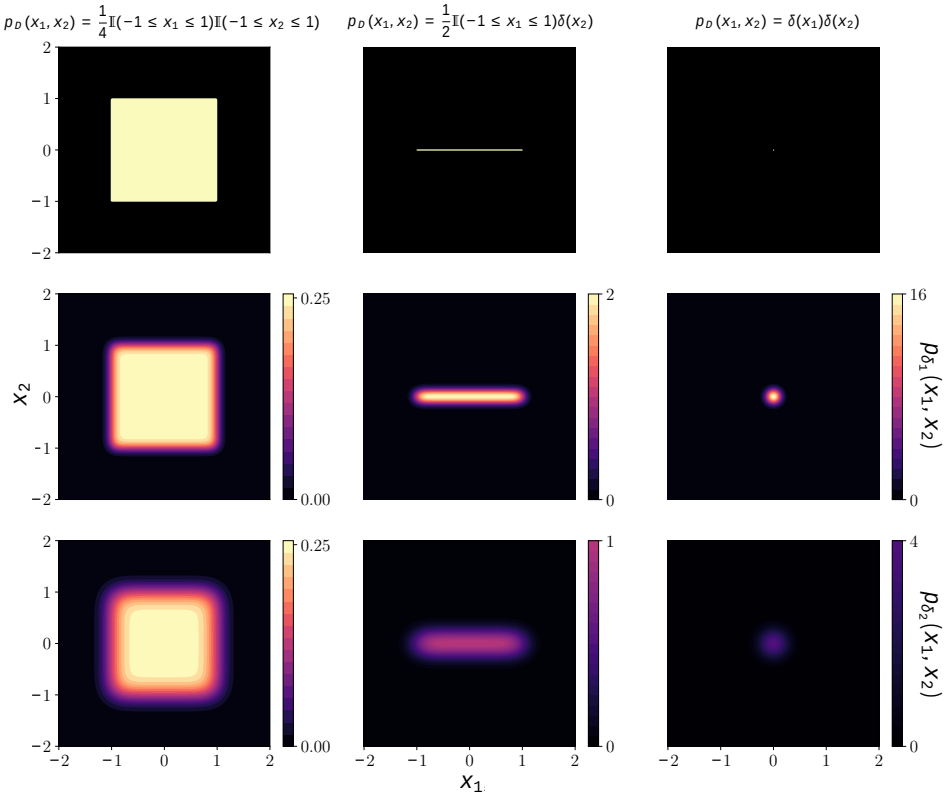

Figure 2: Illustrating the intuition of the core mechanism behind our LIDL method. [Top row] Three example densities $p_D(x_1; x_2)$ on space M of different underlying dimensionality of X: $T = 2; 1; 0$, respectively. The form of the densities are provided above the figures. [Middle and bottom rows] The probability density functions after addition of noise: $p(\mathbf{x}) = p_D(\mathbf{x}) \quad N_M(\mathbf{x}j0; \ I_M)$. Middle and bottom rows correspond to two different noise magnitudes, $_1$ and $_2 = 2\ _1$, respectively. Note that the symbol is overloaded, the Dirac delta term ( ) should not be confused with specifying the magnitude of the noise. The density colors are consistent per-column (for the middle and bottom rows), such that the colors in each of the columns are directly comparable, and the maximum value of the colorbar gives the maximum value of the density on a given plot. Let's say that our goal is to determine the local intrinsic dimensionality for the neighbourhood of point $(0; 0)$. The key observation that will allow us to answer that question is that for different dimensionalities of X (i.e., different columns) the difference between densities $p_{\ 1}(0; 0)$ and $p_{\ 2}(0; 0)$ (i.e., middle and bottom rows) differs. The precise relationship between those quantities is dictated by Eq. 1 what allows us to determine the local intrinsic dimension at a particular point on the distribution by evaluating densities for different values of . In this particular case we observe that: for the left column the density remains constant as the noise increases; for the middle column it is halved when the noise amplitude is doubled; for the right column the density is quartered. Having access only to the evaluations $p_{\ 1}(0; 0)$ and $p_{\ 2}(0; 0)$, by following Eq. 1 we would conclude that $T = 2; 1; 0$ for the neighbourhood of point (0,0) for columns left to right, respectively.

Consider how the probability density is displaced by the act of convolution at $\mathbf{x}^{(n)}$, which is the n^{th} point in the dataset. Three distinct effects take place: 1. The probability density from point $\mathbf{x}^{(n)}$ is displaced into neighbouring points in the tangent subspace, 2. The probability density from point $\mathbf{x}^{(n)}$ is displaced into neighbouring points in the orthogonal subspace, 3. The probability density from the neighbouring points in the tangent subspace is displaced onto the point $\mathbf{x}^{(n)}$. By the assumption above and the symmetry of the isotropic normal distribution as the convolution kernel, the effects 1 and 3 cancel out. This implies that the amount of mass displaced from any point $\mathbf{x}^{(n)}$ in the tangent subspace will depend on the dimensionality of the local tangent and orthogonal subspaces. This implies that the density $p(\mathbf{x}^{(n)})$ for a point $\mathbf{x}^{(n)}$ can be expressed as a product of two independent probability distributions, one for each subspace we are considering. The first

term is the original density with support on the tangent space which is an integral of p_D over the orthogonal subspace $p_T(\mathbf{x}^{(n)}) = \int_{O} p_D(\mathbf{x}^{(n)}) d M_{O}^{\mathbf{x}^{(n)}}$. To define the second term first let $O^{(n)} = M \quad T^{(n)}$ be the dimensionality of the orthogonal subspace at point $\mathbf{x}^{(n)}$. Now, the second term is the probability density of the component of the noise in the $O^{(n)}$–dimensional orthogonal subspace, i.e., $N_{O^{(n)}}(0; \ I_{O^{(n)}})$. Then, $p\ (\mathbf{x}^{(n)}) = p_T(\mathbf{x}^{(n)}) \ \ N_{O^{(n)}}(0j0; \ I_{O^{(n)}}) = p_T(\mathbf{x}^{(n)})$ $(2 \quad ^2)^{-O^{(n)}=2}$, leading to

$$\log p\ (\mathbf{x}^{(n)}) = \log p_T(\mathbf{x}^{(n)}) \quad \frac{O^{(n)}}{2} \log 2 \quad ^2 = \quad O^{(n)} \log \quad + (\text{const w.r.t.} \quad): \tag{1}$$

If we are able to evaluate $p\ (\mathbf{x}^{(n)})$ for at least two different values of , then we can use this relationship to estimate the value of $O^{(n)}$ for a datapoint $\mathbf{x}^{(n)}$. Since we know M we obtain an estimate of the local intrinsic dimension $T^{(n)}$.

LIDL Algorithm
Now, let us consider how to use the method proposed above in practice. To evaluate the probabilities $p\ (\mathbf{x})$ we use NF, which these days can scale even to high-dimensional data like images. We learn $q\ (\mathbf{x})$: a model that approximates $p\ (\mathbf{x})$. NF are trained on the data points from the dataset with a Gaussian noise of appropriate magnitude added to them. For each datapoint in every new batch, we sample a new noise perturbation . When we learn separate models for different values $_i$ we can use linear regression to obtain an estimate of $O^{(n)}$ at some particular point $\mathbf{x}^{(n)}$. This estimate can then be used to calculate $T^{(n)}$, the LID.

To estimate LID for a set of N points $S = f\mathbf{x}^{(1)}; :::; \mathbf{x}^{(n)}; :::; \mathbf{x}^{(N)}g$ in the dataset D we have to fit $d > 1$ models F_i $(i = 1; :::; d)$ to d datasets D_i. Each D_i is perturbed version of the original D with different noise $N_M(0; \ _i I_M)$ added to the whole dataset. The probability density for the same coordinates is different when estimated by different F_i and it decreases monotonically as the value of $_i$ increases. If we then calculate the values $\log q_i(\mathbf{x}^{(n)})$ for the same point $\mathbf{x}^{(n)}$, we can use the linear regression to fit Eq. 1, where the slope is equal to $O^{(n)} = M \quad T^{(n)}$. The full algorithm is presented in detail in Algorithm 1 in Appendix C.

4 Experiments

We ran a series of experiments to verify LIDL, to compare it with other state-of-the-art algorithms for LID estimation and to analyze how it behaves on real-world datasets.

4.1 Lollipop dataset

For the first experiment, we used an synthetic 1D/2D dataset shown in Fig. 3. We trained the MAF (Papamakarios et al., 2017) model to fit 8 models and estimated the average density per model for the points from "head" and "stick" parts of the dataset. Than we fitted regression to those averages. Calculated LID estimate for points in the "head" of the lollipop was 1.96, and for points from the "stick" part estimated LID was 1.00.

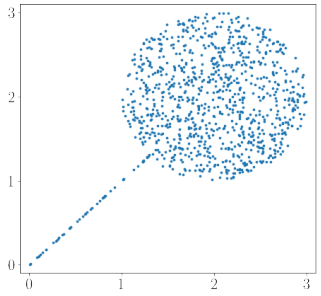

Figure 3: Lollipop dataset used in our experiments.

4.2 Comparison with other algorithms on synthetic datasets

We collated LIDL with other LID estimation algorithms (Levina & Bickel, 2004; Kleindessner & Luxburg, 2015) for intrinsic dimension estimation by comparing it with estimates in Table 1 from Kleindessner & Luxburg (2015). Each algorithm in Kleindessner & Luxburg (2015) was tested on a dataset of size 1000, so we used the data set of the same size for LIDL training (we used 750 examples from this set for training and 250 for validation). We used MAF and RQ-NSF (Durkan et al., 2019) for this experiments. The results of this comparison (each one with a standard deviation calculated using 10 experiments) are presented in Table 1 and in Fig. 1. From this experiments we can clearly see, that LIDL yields result close to the original dimensionality for low-dimensional datasets and gives better estimates in higher dimensions.

Table 1: LIDL comparison with algorithms from Table 1 in (Kleindessner & Luxburg, 2015).

Distribution	ID	E_{CAP} (V)	E_{DP} (V)	MLE	CorrDim	RegDim	LIDL
uniform on a helix in R^3	1	1.00 :05	0.88 :01	1.00 :01	1.00 :11	0.99 :01	0.97 :15
Swiss roll in R^3	2	2.14 :05	1.44 :01	1.94 :02	1.99 :23	1.87 :04	2.68 :35
$N_5(0; I)$ R^5	5	5.33 :07	2.47 :01	5.00 :04	4.91 :56	4.86 :05	5.00 :02
uniform on sphere S^7 R^8	7	5.88 :06	2.82 :01	6.53 :07	6.85 :66	6.23 :09	7.02 :18
uniform on $[0; 1]^{12}$ in R^{12}	12	7.74 :08	3.04 :01	9.32 :10	10.66 1:18	8.78 :10	11.55 :33

Table 2: LIDL estimated ID compared with CorrDim estimate in higher dimensions.

Distribution	ID	CorrDim	LIDL
$N_1(0; I)$ R^2	1	1.00 :001	1.02 :04
$N_{10}(0; I)$ R^{20}	10	7.45 :02	10.14 :08
$N_{100}(0; I)$ R^{200}	100	30.18 :13	100.92 :62
$N_{1000}(0; I)$ R^{2000}	1000	102.64 :85	1048.42 21:52

4.3 High-dimensional synthetic datasets

We compared LIDL with CorrDim on four datasets of size 10K with Gaussian distribution embedded in higher dimensional space with ID equals 1, 10, 100, and 1000 dimensions. Results are presented in Table 2 and show, that LIDL scales to higher dimensions unlike CorrDim algorithm.

4.4 Image datasets

We ran LIDL on MNIST (image size 32x32x1) (LeCun & Cortes, 2010), FMNIST (32x32x1) (Xiao et al., 2017) and Celeb-A (64x64x3) (Liu et al., 2015) datasets using Glow (Kingma & Dhariwal, 2018) as a density model. Before training all images were normalized to have pixel values between $(-0.5; 0.5)$. Estimated dimensionalities for MNIST images span roughly from 40 to 250, for FMNIST those numbers are 100 and 900, and for Celeb-A we estimated dimensionalities between 2500 and 6500. After sorting the dataset according to the local intrinsic dimension of individual data points, we observed that visually more complicated examples have higher estimated dimensionalities. Some small, medium and high dimensional images from those datasets are shown in Fig. 4 and 7. More details about experiments can be found in Appendix D and E.

Figure 4: At the first 3 rows we show MNIST images with low, medium and high LID estimates from LIDL. 3 middle rows show the same for FMNIST and 3 bottom show the same for Celeb-A.

5 Conclusions

We introduced the algorithm for LID estimation based on NF as density estimators, provided the theoretical justification for it, and showed that it can scale to datasets of thousands of dimensions. Our approach, however, is limited by the ability of NF models to scale to even higher dimensions. For now, we are not able to cope with datasets of images consisting of millions of pixels. We hope that current intensive research on NF models will make them able to scale to those datasets eventually and automatically make LIDL to be able to estimate LID for them as well.

REFERENCES

Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 6111–6122, 2019.

Francesco Camastra and Antonino Staiano. Intrinsic dimension estimation: Advances and open problems. *Information Sciences*, 328:26–41, 2016.

Francesco Camastra and Alessandro Vinciarelli. Vinciarelli, a.: Estimating the intrinsic dimension of data with a fractal-based method. ieee trans. pami 24, 1404-1407. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, 05 2002. doi: 10.1109/TPAMI.2002.1039212.

Kevin M Carter, Raviv Raich, and Alfred O Hero III. On local intrinsic dimension estimation and its applications. *IEEE Transactions on Signal Processing*, 58(2):650–663, 2009.

Gabriel B Cavallari, Leonardo SF Ribeiro, and Moacir A Ponti. Unsupervised representation learning using convolutional and stacked auto-encoders: a domain and cross-domain feature space analysis. In *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pp. 440–446. IEEE, 2018.

Claudio Ceruti, Simone Bassis, Alessandro Rozza, Gabriele Lombardi, Elena Casiraghi, and Paola Campadelli. Danco: An intrinsic dimensionality estimator exploiting angle and norm concentration. *Pattern Recognition*, 47(8):2569–2581, 2014. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2014.02.013. URL https://www.sciencedirect.com/science/article/pii/S003132031400065X.

Jens Christian Claussen and Thomas Villmann. Magni cation control in winner relaxing neural gas. *Neurocomputing*, 63:125–137, 2005.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline ows. *arXiv preprint arXiv:1906.04032*, 2019.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. n ows: normalizing ows in pytorch, November 2020. URL https://doi.org/10.5281/zenodo.4296287.

Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

Hideitsu Hino, Jun Fujiki, Shotaro Akaho, and Noboru Murata. Local intrinsic dimension estimation by generalized linear modeling. *Neural Computation*, 29(7):1838–1878, 2017.

Diederik P Kingma and Prafulla Dhariwal. Glow: Generative ow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018.

D.P. Kingma and M. Welling. Auto-encoding variational Bayes. *arXiv:1312.6114*, 2014.

Matthäus Kleindessner and Ulrike Luxburg. Dimensionality estimation without distances. In *Artificial Intelligence and Statistics*, pp. 471–479, 2015.

Szymon Knop, Przemys aw Spurek, Jacek Tabor, Igor Podolak, Marcin Mazur, and Stanis aw Jastrzebski. Cramer-wold auto-encoder. *Journal of Machine Learning Research*, 21, 2020.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. http://yann.lecun.com/exdb/mnist/.

Elizaveta Levina and Peter Bickel. Maximum likelihood estimation of intrinsic dimension. *Advances in neural information processing systems*, 17:777–784, 2004.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In Proceedings of International Conference on Computer Vision (ICCV), December 2015.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. arXiv preprint arXiv:1705.07057, 2017.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In International Conference on Machine Learning, pp. 1530–1538. PMLR, 2015.

A. Rozza, G. Lombardi, C. Ceruti, E. Casiraghi, and P. Campadelli. Novel high intrinsic dimensionality estimators. Machine Learning, 2012.

Paul K Rubenstein, Bernhard Schoelkopf, and Ilya Tolstikhin. On the latent space of wasserstein auto-encoders. arXiv preprint arXiv:1802.03761, 2018.

ukasz Struski, Jacek Tabor, and Przemys aw Spurek. Lossy compression approach to subspace clustering. Information Sciences, 435:161–183, 2018.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. arXiv preprint arXiv:1711.01558, 2017.

Vladimir Vapnik. The nature of statistical learning theory. Springer science & business media, 2013.

Yasi Wang, Hongxun Yao, and Sicheng Zhao. Auto-encoder based dimensionality reduction. Neurocomputing, 184:232–242, 2016.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747, 2017.

Appendices

A Problem definition

Let us assume the data manifold is a union of J separate manifolds, and we will use a discrete indexing variable e, $1 \leq e \leq J$. Let $X_e \subset \mathbb{R}^{T_e}$ be a set of low-dimensional sets of dimensionality T_e, $\varphi_e : X_e \to \mathcal{M} \subset \mathbb{R}^M$ a set of smooth embeddings of X_e into a high-dimensional space $\mathcal{M}$, p_{X_e} a probability density function supported on X_e, and p_e a probability mass function over the indexing variable e. Assume that a datapoint $\mathbf{x}^{(n)}$ is drawn by first sampling a random variable $E^{(n)} \sim p_e$ which decides which of the submanifolds the datapoint will drawn from, and then drawing the sample from an appropriate distribution $\mathbf{x}^{(n)} \sim p_{X_{E^{(n)}}}$. Next, the samples $\mathbf{x}^{(n)}$ are embedded into the observation space $\mathbb{R}^M$ via a corresponding $\varphi_{E^{(n)}}$, possibly disturbed by noise $\boldsymbol{\varepsilon}^{(n)} \in \mathbb{R}^M$, resulting in the samples $\mathbf{s}^{(n)} = \varphi_{E^{(n)}}(\mathbf{x}^{(n)}) + \boldsymbol{\varepsilon}^{(n)}$. The resulting probability density function supported on $\mathcal{M}$ is denoted as $p_D(\mathbf{x})$. Given a dataset of samples $\mathbf{s}^{(n)}$, the task of LID estimation is to infer the corresponding values of $T_{E^{(n)}}$.

B The assumption of constant density along the tangent subspace

Let us revisit the assumption we made in the previous section about the density along the tangent subspace p_T being constant, e.g., as per Fig. 5a.

Naturally, in general the density p_T will not be constant and so we need to question ourselves how much of an error will that introduce to our LID estimates. To answer that question, consider that the degree of violation of our assumption is a product of two factors: the magnitudes $\varepsilon_1, \varepsilon_2$ of ε used to make the LID estimates, and the rate of change of p_T of which worst case could be analyzed by

considering by Lipschitz constants of the distributions p_{X_e} and the associated functions ψ_e. The smaller the values, the larger the rate of change of p_{X_e} can be while keeping the degree of violation of our assumption fixed. This means that we can alleviate this violation by using small values of (as much as numerical precision allows), but it does introduce limitations on the smoothness of p_{X_e} and ψ_e—we will be able to perform better LID estimation for smoother densities.

(a) Constant density (b) Varying density

Figure 5: Comparison of the two possible scenarios of density along the tangent subspace: (a) constant density, which corresponds to the top middle panel in Fig. 2, and (b) varying density. Note that even though the densities $p_{\mathcal{S}}$ are singular, as described in the caption of Fig. 2, the values of $p_{\mathcal{S}}$ are finite since defined over the tangent subspace.

C ALGORITHM

LIDL algorithm is written out in details in Algorithm 1.

Algorithm 1: LIDL: Local Intrinsic Dimension estimation using approximate Likelihood.

Input: Dataset $\mathcal{D}$;
d – number of models to estimate;
M – dimensionality of the data space;
$\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_d)$ – list of values of σ_i;
Set S of N points from $\mathbb{R}^M$, at which we want to estimate LID;
Result: List of T_j's – LID estimates for points in S;
for σ_i in $\boldsymbol{\sigma}$ **do**
  initialize model F_i;
  while F_i not converged **do**
    Sample batch b from $\mathcal{D}$;
    Sample noise n_i from $\mathcal{N}(0, \sigma_i I)$;
    $b_i \leftarrow b + n_i$;
    Make training step on F_i using b_i;
  end
end
for x_j in S **do**
  for F_i in $(F_1, \ldots, F_d)$ **do**
    Estimate likelihood q_{ij} at point x_j using model F_i;
  end
  Calculate regression coefficient Ω_j for a list of d pairs:
  $((\log q_{1j}, \log \sigma_1), \ldots, (\log q_{dj}, \log \sigma_d))$;
  $T_j \leftarrow M - \Omega_j$;
end

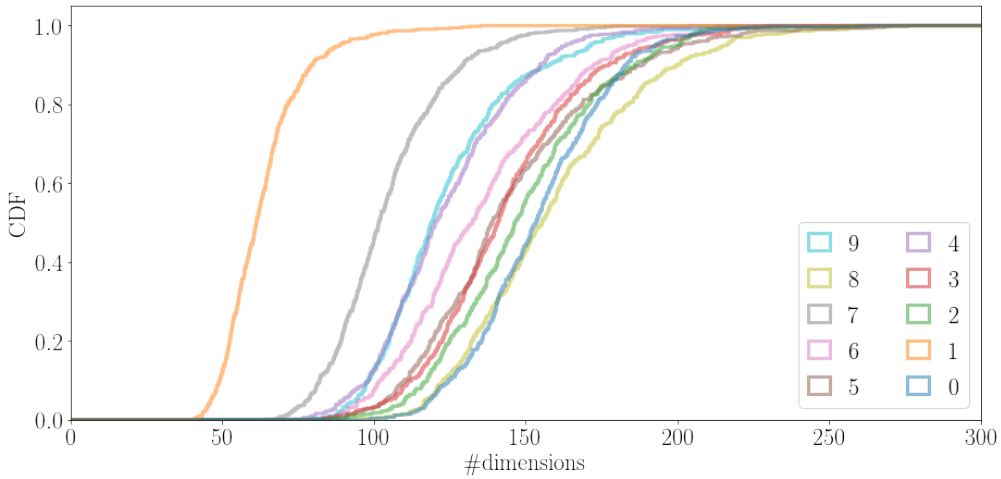

Figure 6: Empirical cumulative distribution function (CDF) of 5000 examples from MNIST dataset. Each line represents CDF for separate class in the dataset. Class number (which also is a represented digit in this case) can be found in the legend.

D MORE ON LIDL ESTIMATES ON MNIST

We also used LIDL estimates for MNIST dataset to analyse how the LID distribution looks for different classes. Empirical CDFs of those values for each digit separately are presented in Fig. 6. Classes 1s and 7s have the lowest LIDs and 8s and 0s have the highest dimensionalities, which is consistent with the visual complexity of those classes.

We can ask a question: how LIDL estimates are close to underlying LID? The estimated IDs reported for some MNIST digits in Table 1 in (Kleindessner & Luxburg, 2015) are between 3.07 and 18. On the other hand Cavallari et al. (Cavallari et al., 2018) and Wang et al. (Wang et al., 2016) used auto-encoder representation of MNIST as an input to SVN digit classifier and they achieved the best classification results for an auto-encoder with latent space size greater than 100. This means that we need more than 100 dimensions to encode an average MNIST digit, which is more consistent with our result than with (Kleindessner & Luxburg, 2015).

E NF IMPLEMENTATIONS

For non-image datasets we used MAF and RQ-NSF implementations from nflows library (Durkan et al., 2020). For images we used PyTorch implementation of glow from `https://github.com/rosinality/glow-pytorch`.

F ACKNOWLEDGEMENTS

I wish to thank various people for their contribution to this project: Marek Cygan for overall support during this project and valuable and constructive suggestions on the manuscript; Maciej Dziubiński, Dominik Filipak, Piotr Kozakowski and Maciej Śliwowski for their valuable and constructive suggestions on the manuscript and Wacława Tempczyk for her help with method derivation.

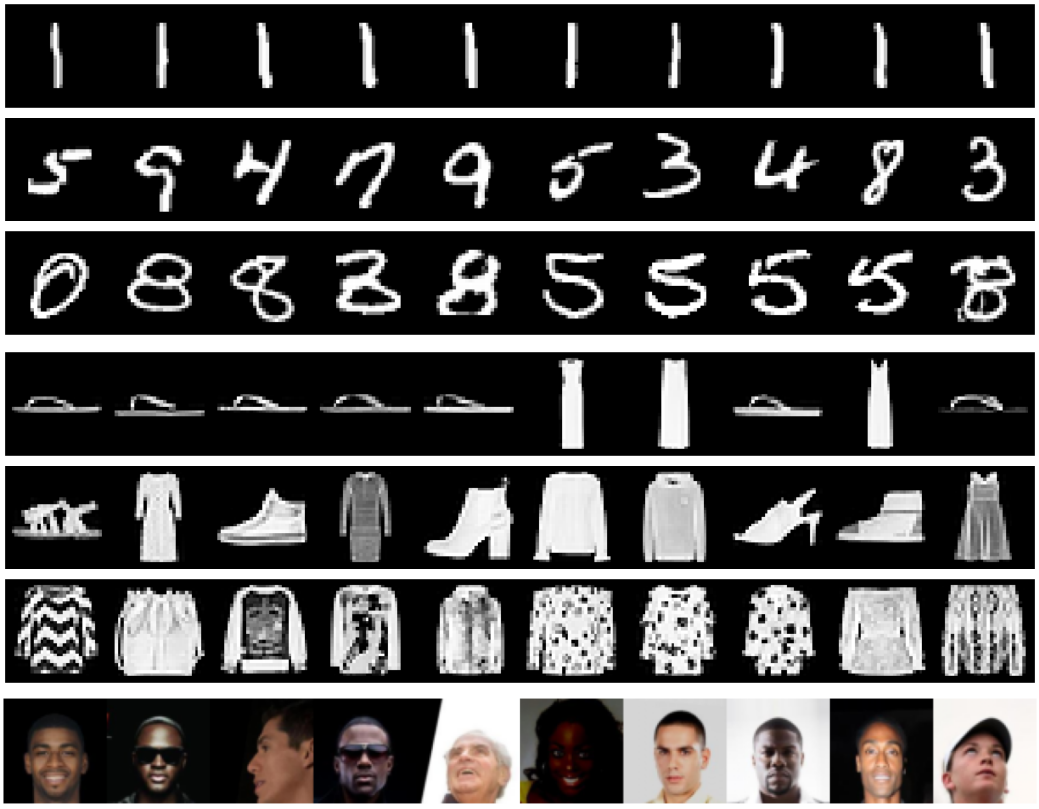

Figure 7: Closer look at Fig. 4