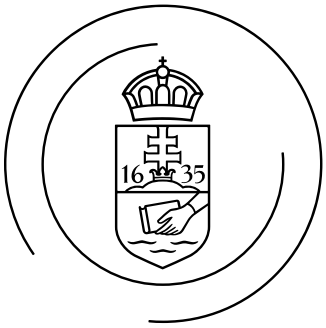
Math project

Entropy estimation
The Kozachenko-Leonenko method

Emma Lukács

Supervisor: Adrián Csiszárik



EÖTVÖS LORÁND
UNIVERSITY | BUDAPEST

Applied Mathematics MSc
2023/1

1. Introduction

The information theory paradigm, rooted in Shannon's foundational work from the 1940s [5], has gained significant traction in Machine Learning and Neural Networks. Self-supervised learning involves models predicting one input part from another, reflecting principles akin to entropy maximization. Despite its historical significance, several fundamental questions persist in the field. The usage of information entropy encounters challenges in real-world scenarios due to the absence of underlying Probability Density Functions (PDFs), leaving only observed data. This obstacle necessitates accurate entropy estimation solely from observed data, driving the need for flexible, non-parametric methods like the k-nearest neighbor (kNN) approach pioneered by Kozachenko and Leonenko [3]. However, the classical kNN estimator exhibits bias, particularly in higher dimensions [4]. My research focus centered on creating specific high-dimensional scenarios challenging the Kozachenko-Leonenko estimator, exploring methods to project data into lower-dimensional spaces to preserve vital information, aiming to improve results and alleviate biases.

2. The Kozachenko-Leonenko estimate

2.1. Definition (Entropy)**.** Let X be a discrete random variable with probability mass function $P_X(x)$, $x \in \mathcal{X}$. The *entropy* (or *Shannon entropy*) of X is

$$H(X) = \mathbb{E}\left[\log \frac{1}{P_X(X)}\right] = \sum_{x\in\mathcal{X}} P_X(x) \log \frac{1}{P_X(x)} \tag{1}$$

$$= \int_{\mathcal{X}} P_X(x) \log \frac{1}{P_X(x)}\, dx. \tag{2}$$

2.2. Definition (k-Nearest Neighbour Kozachenko-Leonenko estimator)**.** According to definition introduced in the paper written by Ao and Li [1]. Let $x_1, x_2, \ldots, x_n$ $(n \geq 3)$ be i.i.d. random variables with density f on $\mathbb{R}^d$. Let us indentify the k-nearest neighbors (in terms of the p-norm distance) for each x_i and define the smallest closed ball covering them as:

$$B(x_i, \frac{\varepsilon_i}{2}) = \{x \in \mathbb{R}^d | \|x - x_i\|_p \leq \frac{\varepsilon}{2}\},$$

where ε is twice the distance of x_i and its k-th nearest neighbour, and the mass of $B(x_i, \frac{\varepsilon_i}{2})$ is:

$$q_i(\varepsilon_i) = \int_{x\in B(x_i, \frac{\varepsilon_i}{2})} P_X(x)dx \Rightarrow \mathbb{E}(\log(q_i)) = \psi(k) - \psi(N),$$

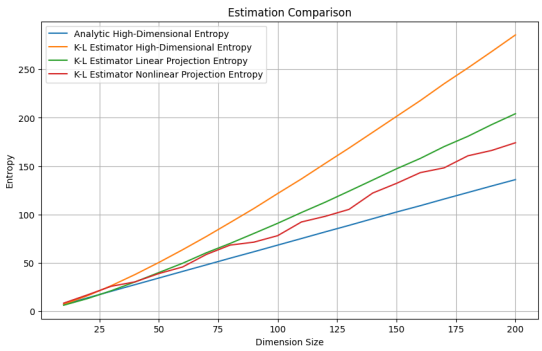
where $\psi(N)$ is equal to $\frac{\Gamma'(x)}{\Gamma(x)}$ with $\Gamma(x)$ being the Gamma function. The main assumption of the KL estimation is that the density is constant within the unit ball approximated by $q_i(\varepsilon_i) \approx c_d \varepsilon_i^d P_X(x_i)$, where d is the dimension of X and c_d is given by $\frac{\Gamma(1+\frac{1}{p})^d}{\Gamma(1+\frac{d}{p})}$, which is the volume of the d-dimensional unit ball according to the given p-norm. This yields the final KL-estimator formula:

$$\hat{H}_{KL} = \psi(k) + \psi(N) + \log c_d + \frac{d}{N}\sum_{i=1}^{N} \log(\varepsilon_i). \tag{3}$$

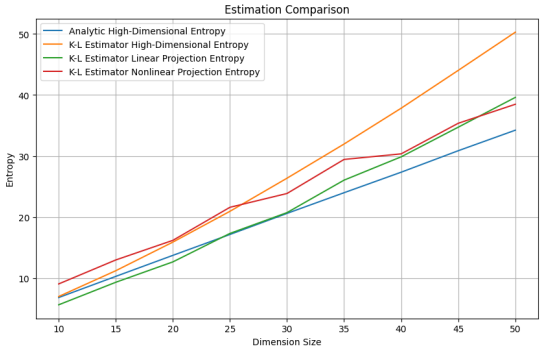
3. Dimensionality Reduction and Entropy Estimation Analysis

I utilized autoencoders, structures comprising an encoder function f_{enc} and a decoder function f_{dec}, to retain crucial information when projecting into a lower-dimensional space. The encoder, f_{enc}, mapped high-dimensional input data $\mathbf{x}$ from $\mathbb{R}^{\text{input dim}}$ to a lower-dimensional latent representation $\mathbf{z}$ in $\mathbb{R}^{\text{latent dim}}$, denoted as $\mathbf{z} = f_{\text{enc}}(\mathbf{x})$. Simultaneously, the decoder function f_{dec} aimed to reconstruct the input from the encoded representation, achieving $\mathbf{x}' = f_{\text{dec}}(\mathbf{z})$ [2]. Two specific model designs were explored: the

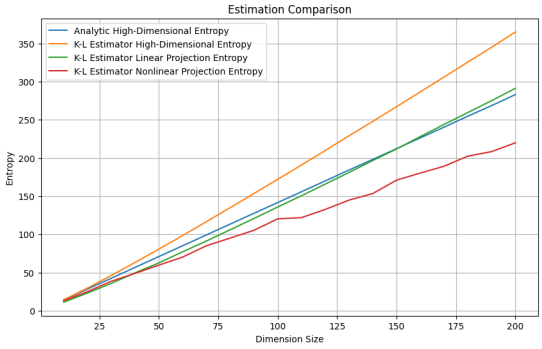
Nonlinear Autoencoder, featuring multiple hidden layers (300-200-100-hidden size) utilizing linear and Rectified Linear Unit (ReLU) activation layers for both encoder and decoder, and the **Linear Autoencoder**, a simpler architecture using a single linear layer for both encoding and decoding. Training involved Mean Squared Error (MSE) loss function and Adam optimizer. For data generation, I simulated scenarios where data predominantly clustered around a hyperplane using custom methods. Initially, I generated multidimensional data with a normal distribution, emphasizing data concentration around the hyperplane by scaling down specific lower dimensions by a factor of 0.4. Another strategy involved creating sets of decorrelated Gaussian data by employing Cholesky decomposition on a specified covariance matrix, transforming randomly generated data to achieve decorrelation based on the matrix.



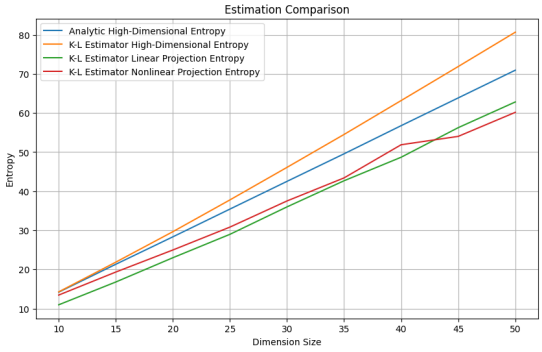
(a) First data - full dimension range



(b) First data - lower dimension range



(c) Second data - full dimension range



(d) Second data - lower dimension range

Figure 1. Comparison of entropy estimation across dimension sizes

The plot color scheme is coherent: blue indicates the analytic entropy, orange represents the original dimensional KL-estimate, green stands for the linear projection KL-estimate, and red signifies the nonlinear projection. The 'first data' pertains to the initial data generation process, while the 'second data' corresponds to the decorrelated Gaussian data.

I computed entropy for the original high-dimensional dataset using both the analytical method and the KL-estimate, alongside the lower-dimensional representations obtained from two autoencoder models. The resulting visualization compares entropy estimates across various dimension sizes while maintaining a fixed $\frac{4}{5}$ ratio between lower and higher dimensions. In Figure 1, the top row illustrates outcomes from the initial data generation process. Overall, the nonlinear model projection yielded the most favorable results. However, focusing solely on the lower-dimensional outcomes shown on the right side, the linear model's low dimensional estimate exhibited closer proximity to the analytical result. Particularly notable is the significantly improved entropy estimation within the linear projection space, especially beyond dimension 60 in the case of decorrelated Gaussian data. It's apparent that using both autoencoder models I could outperform the direct KL estimation of entropy in the original space. This prompts an exploration into the underlying reasons for this phenomenon, urging a deeper analysis. To delve further into this observation, I aim to investigate additional data generation processes that pose challenges for the estimation of entropy, providing a more comprehensive understanding of the models' efficacy in capturing essential information for accurate entropy estimation.

Bibliography

[1] Ziqiao Ao and Jinglai Li. Entropy estimation via uniformization. *Artificial Intelligence*, 322:103954, 2023.

[2] Dor Bank, Noam Koenigstein, and Raja Giryes. Autoencoders, 2021.

[3] L. F. Kozachenko and N. N. Leonenko. Sample estimate of the entropy of a random vector. *Probl. Peredachi Inf.*, 23(2):9–16, 1987. Translated from: Problems Inform. Transmission, 23(2):95–101, 1987.

[4] Chien Lu and Jaakko Peltonen. Enhancing nearest neighbor based entropy estimator for high dimensional distributions via bootstrapping local ellipsoid. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:5013–5020, 04 2020.

[5] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.