Advanced Methods of Data Analysis: Normalizing Flows

Leonhard Moske
(Dated: August 10, 2022)

In this paper, a recent method named normalizing flows is utilized to construct density estimators for the purpose of classification. Normalizing flows are a class of transformation, which can be trained with gradient descend to estimate the distribution of data or to generate data in a toy model.

In this paper, the expressiveness of fully connected neural networks is used in a class of transformation known as normalizing flows in order to construct density estimator. Further these estimators are utilized to classify stars.

more explizit

I. INTRODUCTION

Normalizing flows is a powerful method that utilizes the transformation of random variables for either density estimation or for generative sampling.

To demonstrate the functionality of normalizing flows we investigate the *AstroML RR-Lyrae variable stars data*-dataset which contains a record of *RR-Lyrae variable stars*, which are periodic variable stars, found in globular cluster, that serve as standard candles to measure extra galactic distances. The features provided are four intensities at different wavelength filters corresponding to the photometric system. The four features are: $\{u-g, g-r, r-i, i-z\}$, where u is at 365 nm, g at 464 nm, r at 658 nm, i at 806 nm and z at 900 nm. The dataset contains background stars further named background and the *RR-Lyrae variable stars*, which we are going to call signal.

statistic of signal and background

II. THEORY

To classify data into different categories (background and signal) using the density of these categories we have to evaluate the probability density distributions at the data and compare the result, i.e. choose the category with the highest probability. In this case we have only two categories (background named 0, signal named 1), thus we can use a test statistic like:

$$t(\text{data}) = \ln\left(\frac{p(0|\text{data})}{p(1|\text{data})}\right)$$

We should classify the data as background if $t > 0$ and as signal if $t < 0$. In practice this cut has to be chosen with respect to the validation of the classifiers.

So we have to construct an algorithm that allows us to evaluate the probability density of both categories.

A. functions of random variables

To estimate the density distribution we make use of the formula of transformations of random variables, which lets us connect a simple distribution that we can evaluate and the distribution of the data. This allows us to approximate the evaluation of the data density at points, i.e. new data.

Let z be a random variable distributed as $r(z)$ then a random variable $x = f(z|\theta)$, where f is a invertible and differentiable function with parameters θ, is distributed as $q(x)$ with:

$$q(x) = r(z)\left|\frac{\mathrm{d}z}{\mathrm{d}x}\right| = r(z)\left|\det J_f(z)\right|^{-1}$$

We call $r(z)$ the base distribution and $q(x)$ the target distribution.

To get a density estimation of new data x' with some parameters we would vary θ until $q(x)$ is close to the target distribution of the data $p(x)$, then we can compute $r(f^{-1}(x'|\theta))\left|\det J_f(x'|\theta)\right|$ which is the estimate for the probability of the data. To do this we have to be able to compute the inverse transformation, its Jacobian determinant and evaluate the base distribution. As the base distribution we choose a multidimensional normal distribution. The dimension of this distribution is the number of features since the flow f has to be injective.

citations

B. parameterize the transformation

To achieve the needed expressiveness of the transformation we construct it by composing smaller so called coupling layers:

$$\begin{aligned} f &= f_0 \circ f_1 \circ ... \circ f_K \qquad k = 0, 1...K \\ z_k &= f_k(z_{k-1}) \\ z_K &= x \end{aligned}$$

We achieve invertibility and differentiability of f if all coupling layers f_k are differentiable and invertible. Also the Jacobian determinant is calculated as:

$$\begin{aligned} \ln\left|\det J_{f^{-1}}(x)\right| &= \ln\left|\prod_{k=0}^{K} \det J_{f_k^{-1}}(x_{k-1})\right| \\ &= \sum_{k=0}^{K} \ln\left|\det J_{f_k^{-1}}(x_{k-1})\right| \end{aligned}$$

Also because the the coupling layers have to be invertible they have to be injective, having the same number of input as output dimensions.

citations

picutre normalizing flows

diagram spectrum google: u g r i z wavelength filter astrology

citation

hist of features

classifier by density estimation: if we have den-

C. training the transformation

In order to receive a distribution $q(x)$ that represents the target distribution $p(x)$ closely we have to vary the parameters θ of the transformation f. One can use gradient descent with a divergence as a loss function. In this implementation the Kullback-Leibler (KL) divergence is used as one of the most popular.

In this case where we have samples of the target distribution it is suitable to work with the forward KL divergence between $p(x)$ and $q(x|\theta)$:

$$\begin{aligned}\mathcal{L}(\theta) &= D_{KL}\left[p(x)\|q(x|\theta)\right] \\ &= -\mathbb{E}_{p(x)}\left[\ln(q(x|\theta))\right] + \text{const} \\ &\approx -\frac{1}{N}\sum_{n=1}^{N}\ln(r(f^{-1}(x_n|\theta))) + \ln\left|\det J_{f^{-1}}(x_n|\theta)\right|\end{aligned}$$

where the x_n are the sampled target data. Thus we have to be able to compute f^{-1}, its jacobian determinant and evaluate $r(z)$ and since we want do use gradient descend we need to differentiate through them.

III. NORMALIZING FLOW CATEGORIES

list of different parametrizations -¿ all requirements in subsection

IV. METHODS

training testing val split
fixed base dist
tendorflow bijectors
how training (train utils)
adam optimizer
implementation mit tensorflow. citation LUKAS...
how checking (roc und hist der t und fraction of richtig class)

V. RESULTS

loss function over time
hist of test stat der versch nf.
plot der fraction of richtig class
vllt plot prob density

VI. SUMMARY

VII. CONCLUSION