Soft Brownian Offset

Release 1.1

Felix Möller

Aug 25, 2021

CONTENTS:

CHAPTER

ONE

INTRODUCTION

Soft Brownian Offset (SBO) defines an iterative approach to translate points by a most likely distance from a given dataset. It can be used for generating out-of-distribution (OOD) samples. It is based on Gaussian Hyperspheric Offset (GHO), which is also included in this package (see *below*).

CHAPTER

TWO

INSTALLATION

This project is hosted on PyPI and can therefore be installed easily through `pip`:

```
pip install sbo
```

Dependending on your setup you may need to add `--user` after the install.

CHAPTER

THREE

USAGE

For brevity's sake here's a short introduction to the library's usage:

```
from sklearn.datasets import make_moons
from sbo import soft_brownian_offset

X, _ = make_moons(n_samples=60, noise=.08)
X_ood = soft_brownian_offset(X, d_min=.35, d_off=.24, n_samples=120, softness=0)
```

CHAPTER

FOUR

BACKGROUND

The technique allows for trivial OOD generation – as shown above – or more complex schemes that apply the transformation of learned representations. For an in-depth look at the latter please refer to the paper that is available as open access from the CVF. For citations please see *cite*.

CHAPTER

FIVE

PARAMETER OVERVIEW

The following plot gives an overview of possible choices for `d_min` (d^-), `d_off` (d^+) and `softness` (σ):

It was created using the following Python code:

```
#!/usr/bin/env python3
# Creates a plot for Soft Brownian Offset (SBO)

import numpy as np
import pylab as plt
import itertools
import sys

from matplotlib import cm
from sklearn.datasets import make_moons

from sbo import soft_brownian_offset

plt.rc('text', usetex=True)

c = cm.tab10.colors

def plot_data(X, y, ax=plt):
    ax.scatter(X[:, 0], X[:, 1], marker='x', s=20, label='ID', alpha=alpha, c=[c[-1]])
    ax.scatter(y[:, 0], y[:, 1], marker='+', label='SBO', alpha=alpha, c=[c[-6]])

def plot_mindist(X, y, ax=plt):
    if len(X.shape) == 1:
        X = X[:, None]
    if len(y.shape) == 1:
        y = y[:, None]
    ax.hist(pairwise_distances(y, X).min(axis=1), bins=len(y) // 10)
    ax.set_xlabel("Minimum distance from ood to id")
    ax.set_ylabel("Count")

def plot_data_mindist(X, y):
    fig, ax = plt.subplots(1, 2)
    plot_data(X, y, ax=ax[0])
    plot_mindist(X, y, ax=ax[1])
    plt.show()


n_samples_id = 60
n_samples_ood = 150
```

(continues on next page)

(continued from previous page)

```
noise = .08
show_progress = False
alpha = .6

n_colrow = 3
d_min = np.linspace(.25, .45, n_colrow)
softness = np.linspace(0, 1, n_colrow)
fig, ax = plt.subplots(n_colrow, n_colrow, sharex=True, sharey=True, figsize=(8.5, 9))

X, _ = make_moons(n_samples=n_samples_id, noise=noise)
for i, (d_min_, softness_) in enumerate(itertools.product(d_min, softness)):
    xy = i // n_colrow, i % n_colrow
    d_off_ = d_min_ * .7
    ax[xy].set_title(f"$d^- = {d_min_:.2f}\ d^+ = {d_off_:.2f}\ \sigma = {softness_}$
↪")
    if softness_ == 0:
        softness_ = False
    y = soft_brownian_offset(X, d_min_, d_off_, n_samples=n_samples_ood,␣
↪softness=softness_, show_progress=show_progress)
    plot_data(X, y, ax=ax[xy])
    if i // n_colrow == len(d_min) - 1:
        ax[xy].set_xlabel("$x_1$")
    if i % n_colrow == 0:
        ax[xy].set_ylabel("$x_2$")
ax[0, n_colrow - 1].legend(loc='upper right')

plt.tight_layout()
plt.show()
```

CHAPTER

SIX

GAUSSIAN HYPERSPHERIC OFFSET

GHO is the basis for SBO and assumes $\boldsymbol{X} \sim \mathcal{N}$. The following code's result displays the shortcomings if the assumption does not hold:

```
from sklearn.datasets import make_moons
from sbo import soft_brownian_offset, gaussian_hyperspheric_offset

X, _ = make_moons(n_samples=60, noise=.08)
X_ood = (gaussian_hyperspheric_offset(n_samples=220, mu=2, std=.3, n_dim=X.ndim) + X.
↪mean()) * X.std()
```

CHAPTER

SEVEN

CITE

Please cite SBO in your paper if it helps your research:

```
@inproceedings{MBH21,
  author    = {Möller, Felix and Botache, Diego and Huseljic, Denis and Heidecker,␣
↪Florian and Bieshaar, Maarten and Sick, Bernhard},
  booktitle = {{Proc. of CVPR SAIAD Workshop}},
  title     = {{Out-of-distribution Detection and Generation using Soft Brownian␣
↪Offset Sampling and Autoencoders}},
  year      = 2021
}
```