
tensor-fm

Gabriele Modena

Jan 05, 2022

CONTENTS:

1	tensor-fm	3
1.1	Install	3
1.2	Usage examples	3
1.2.1	Tensorflow	3
1.2.2	Scikit-learn estimator	3
1.3	Performance	4
1.4	Limitations and known issues	4
2	API	5
2.1	sklearn	5
2.2	Tensorflow	5
3	MIT License	7

Second Order Factorization Machine implemented with [Tensorflow](#) and [scikit-learn](#).

[Documentation Status](#)

TENSOR-FM

A scikit-learn compatible order 2 Factorization Machine, implemented atop TensorFlow 2. The algorithm is described in <http://www.csie.ntu.edu.tw/~b97053/paper/Rendle2010FM.pdf>. For an higher level overview of the method see <http://nowave.it/factorization-machines-with-tensorflow.html>.

This package is a port to Tensorflow 2 of the code presented in that blog post. The goal of this project is to experiment with different optimization strategies for classical ML models, and scalability of TF2 backends.

1.1 Install

The latest development version of `tensorfm` can be installed from its [github repo](#) with:

```
pip install git+https://github.com/gmodena/tensor-fm
```

1.2 Usage examples

Tensorflow and scikit-learn APIs are provided.

1.2.1 Tensorflow

The tensorflow implementation of Factorization Machines lives under `tensor-fm/tensorfm/base.py`. An example of how to work with this API can be found in `tensor-fm/tests/test_base.py`.

1.2.2 Scikit-learn estimator

`tensorfm.sklearn` exposes two sklearn compatible estimators: `FactorizationMachineRegressor` and `FactorizationMachineClassifier`.

Example

```
from tensorfm.sklearn import FactorizationMachineRegressor
...
fm = FactorizationMachineRegressor()
fm.fit(X, y)
fm.predict(X)
```

See also `examples/movielens.py`

1.3 Performance

All parameters and settings being equal, I noticed a considerable performance degradation of `FactorizationMachineRegressor` (MSE on train/test) on movielens compared to the tensorflow 1 implementation from <http://nowave.it/factorization-machines-with-tensorflow.html>. Possibly related, a test in the `check_regressors_train` suite (sklearn) fails due to a low R^2 . As a workaround `FactorizationMachineRegressor` sets the `poor_score` tag to `True`.

1.4 Limitations and known issues

Operations on sparse matrices are currently not supported. Training continues till `max_iter` is reached, we should stop if performance does not improve for a certain number of iterations.

2.1 sklearn

2.2 Tensorflow

MIT LICENSE

Copyright (c) 2020, 2021 Gabriele Modena

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.