## Source code for cornac.data.dataset

<https://www.ctolib.com/PreferredAI-cornac.html>

<https://cornac.readthedocs.io/en/latest/_modules/cornac/data/dataset.html#Dataset>

目录

[Source code for cornac.data.dataset 1](#_Toc59629200)

[Cornac：推荐系统实验框架库 1](#_Toc59629201)

[Cornac.data 4](#_Toc59629202)

[Text modality 5](#_Toc59629203)

### Cornac：推荐系统实验框架库

Cornac是一个用于多模式推荐系统的比较框架，主要针对于利用辅助性数据（比如：项目描述文本和图片、社交网络等等）。同现有的机器学习库（比如TensorFlow, Pytorch）高度兼容。

**Cornac experiment**

**import** cornac

#导入并分割数据集

ml\_100k = cornac.datasets.movielens.load\_feedback(variant="100K")

rs = cornac.eval\_methods.RatioSplit(data=ml\_100k, test\_size=0.2, rating\_threshold=4.0, seed=123)

#调用BRP模块，参数分别为：

bpr = cornac.models.BPR(k=10, max\_iter=200, learning\_rate=0.001, lambda\_reg=0.01, seed=123)

#选择评估工具

#将上述部分组合起来，运行

cornac.Experiment(

eval\_method=rs,

models=[mf, pmf, bpr],

metrics=[mae, rmse, recall, ndcg, auc, mAP],

user\_based=**True**

).run()

下面将根据cornac.data.dataset源码进行说明分析：

**def** \_\_init\_\_(

self,

num\_users,

num\_items,

uid\_map,

iid\_map,

uir\_tuple,

timestamps=**None**,

seed=**None**,

):

参数：

num\_users: int, required

用户的数量

num\_items: int, required

项目的数量

uid\_map: :obj:`OrderDict`, required

字典：从用户原始id到索引

iid\_map: :obj:`OrderDict`, required

从项目原始id到索引

uir\_tuple: tuple, required

Tuple of 3 numpy arrays (user\_indices, item\_indices, rating\_values).

3元组（用户索引，项目索引，评估分数）

timestamps: numpy.array, optional, default: None

对应与3元组的时间戳数组

seed: int, optional, default: None

Random seed for reproducing data sampling.

属性：

num\_ratings: int

Number of rating observations in the dataset.

uir\_tuple: tuple

Tuple three numpy arrays (user\_indices, item\_indices, rating\_values).

timestamps: numpy.array

Numpy array of timestamps corresponding to feedback in `uir\_tuple`.

This is only available when input data is in `UIRT` format.

**@property**

**def** user\_indices(self):

*"""An iterator over the user indices"""*

**return** self.uid\_map.values()

**@property**

**def** item\_indices(self):

*"""An iterator over the item indices"""*

**return** self.iid\_map.values()

**@property**

**def** user\_data(self):

*"""Data organized by user. A dictionary where keys are users,*

*values are tuples of two lists (items, ratings) interacted by the corresponding users.*

*"""*

**@property**

**def** item\_data(self):

*"""Data organized by item. A dictionary where keys are items,*

*values are tuples of two lists (users, ratings) interacted with the corresponding items.*

*"""*

**@classmethod**

**def** build(

cls,

data,

fmt="UIR",

global\_uid\_map=**None**,

global\_iid\_map=**None**,

seed=**None**,

exclude\_unknowns=**False**,

):

*"""Constructing Dataset from given data of specific format.*

方法：

**@classmethod**

**def** build(

cls,

data,

fmt="UIR",

global\_uid\_map=**None**,

global\_iid\_map=**None**,

seed=**None**,

exclude\_unknowns=**False**,

):

*"""Constructing Dataset from given data of specific format.*

*Returns*

*-------*

*res: :obj:`<cornac.data.Dataset>`*

*Dataset object.*

*"""*

**@classmethod**

**def** from\_uir(cls, data, seed=**None**):

*"""Constructing Dataset from UIR (User, Item, Rating) triplet data.*

**@classmethod**

**def** from\_uirt(cls, data, seed=**None**):

*"""Constructing Dataset from UIRT (User, Item, Rating, Timestamp)*

*quadruplet data.*

**return** cls.build(data, fmt="UIRT", seed=seed)

**def** uir\_iter(self, batch\_size=1, shuffle=**False**, binary=**False**, num\_zeros=0):

*"""Create an iterator over data yielding batch of users, items, and rating values*

**def** uij\_iter(self, batch\_size=1, shuffle=**False**, neg\_sampling="uniform"):

*"""Create an iterator over data yielding batch of users, positive items, and negative items*

**def** is\_unk\_user(self, user\_idx):

*"""Return whether or not a user is unknown given the user index"""*

**return** user\_idx >= self.num\_users

**def** is\_unk\_item(self, item\_idx):

*"""Return whether or not an item is unknown given the item index"""*

**return** item\_idx >= self.num\_items

**def** add\_modalities(self, \*\*kwargs):

self.user\_feature = kwargs.get("user\_feature", **None**)

self.item\_feature = kwargs.get("item\_feature", **None**)

self.user\_text = kwargs.get("user\_text", **None**)

self.item\_text = kwargs.get("item\_text", **None**)

self.user\_image = kwargs.get("user\_image", **None**)

self.item\_image = kwargs.get("item\_image", **None**)

self.user\_graph = kwargs.get("user\_graph", **None**)

self.item\_graph = kwargs.get("item\_graph", **None**)

self.sentiment = kwargs.get("sentiment", **None**)

self.review\_text = kwargs.get("review\_text", **None**)

## Cornac.data

Cornac提供多种内容模式，总共有如下几种：

PACKAGE CONTENTS

graph

image

modality

reader

testset

text

trainset

CLASSES

builtins.object

cornac.data.reader.Reader

cornac.data.testset.TestSet

cornac.data.testset.MultimodalTestSet

cornac.data.trainset.TrainSet

cornac.data.trainset.MatrixTrainSet

cornac.data.trainset.MultimodalTrainSet

cornac.data.modality.Modality(builtins.object)

cornac.data.modality.FeatureModality

cornac.data.graph.GraphModality

cornac.data.image.ImageModality

cornac.data.text.TextModality

class FeatureModality(Modality)

| FeatureModality(features=None, ids=None, copy=False, normalized=False, \*\*kwargs)

***class*cornac.data.text.Tokenizer**

**tokenize(*t: str*) → List[str]**

Splitting text into tokens.

***class*cornac.data.text.BaseTokenizer(*sep: str = ' '*, *pre\_rules: List[Callable[str*, *str]] = None*, *stop\_words: Union[List*, *str] = None*)**

***class*cornac.data.text.TextModality(*corpus: List[str] = None*, *ids: List = None*, *tokenizer: cornac.data.text.Tokenizer = None*, *vocab: cornac.data.text.Vocabulary = None*, *max\_vocab: int = None*, *max\_doc\_freq: Union[float*, *int] = 1.0*, *min\_doc\_freq: int = 1*, *tfidf\_params: Dict = None*, *\*\*kwargs*)**

### Text modality

|  |  |
| --- | --- |
| **Par**  **Ame**  **ters:** | * **corpus** (*List[*[*str*](https://docs.python.org/3/library/stdtypes.html#str)*], default = None*) – List of user/item texts that the indices are aligned with *ids*. * **ids** (*List, default = None*) – List of user/item ids that the indices are aligned with *corpus*. If None, the indices of provided *corpus* will be used as *ids*. * **tokenizer** ([*Tokenizer*](https://cornac.readthedocs.io/en/latest/data.html#cornac.data.text.Tokenizer)*, optional, default = None*) – Tokenizer for text splitting. If None, the BaseTokenizer will be used. * **vocab** ([*Vocabulary*](https://cornac.readthedocs.io/en/latest/data.html#cornac.data.text.Vocabulary)*, optional, default = None*) – Vocabulary of tokens. It contains mapping between tokens to their integer ids and vice versa. * **max\_vocab** ([*int*](https://docs.python.org/3/library/functions.html#int)*, optional, default = None*) – The maximum size of the vocabulary. If vocab is provided, this will be ignored. * **max\_doc\_freq** (*float in range [0.0, 1.0] or*[*int*](https://docs.python.org/3/library/functions.html#int)*, default=1.0*) – When building the vocabulary ignore terms that have a document frequency strictly higher than the given threshold (corpus-specific stop words). If float, the value represents a proportion of documents, int for absolute counts. If *vocab* is not None, this will be ignored. * **min\_doc\_freq** (*float in range [0.0, 1.0] or*[*int*](https://docs.python.org/3/library/functions.html#int)*, default=1*) – When building the vocabulary ignore terms that have a document frequency strictly lower than the given threshold. This value is also called cut-off in the literature. If float, the value represents a proportion of documents, int absolute counts. If *vocab* is not None, this will be ignored. * **tfidf\_params** ([*dict*](https://docs.python.org/3/library/stdtypes.html#dict)*or*[*None*](https://docs.python.org/3/library/constants.html#None)*, optional, default=None*) –   If *None*, a default arguments of **<cornac.data.text.IfidfVectorizer>** will be used. List of parameters:  **’binary’ :boolean, default=False**  If True, all non zero counts are set to 1.  **’norm’ :’l1’, ‘l2’ or None, optional, default=’l2’**  Each output row will have unit norm, either: \* ‘l2’: Sum of squares of vector elements is 1. The cosine similarity between two vectors is their dot product when l2 norm has been applied. \* ‘l1’: Sum of absolute values of vector elements is 1. See **utils.common.normalize()**  **’use\_idf’ :boolean, default=True**  Enable inverse-document-frequency reweighting.  **’smooth\_idf’ :boolean, default=True**  Smooth idf weights by adding one to document frequencies, as if an extra document was seen containing every term in the collection exactly once. Prevents zero divisions.  **’sublinear\_tf’ :boolean (default=False)**  Apply sublinear tf scaling, i.e. replace tf with 1 + log(tf). |