

pu

The user factors (only exists if `fit()` has been called)

Type

numpy array of size (n_users, n_factors)

qi

The item factors (only exists if `fit()` has been called)

Type

numpy array of size (n_items, n_factors)

bu

The user biases (only exists if `fit()` has been called)

Type

numpy array of size (n_users)

bi

The item biases (only exists if `fit()` has been called)

Type

numpy array of size (n_items)

```
class surprise.prediction_algorithms.matrix_factorization.SVDpp(n_factors=20, n_epochs=20,
init_mean=0, init_std_dev=0.1, lr_all=0.007, reg_all=0.02, lr_bu=None, lr_bi=None, lr_pu=None,
lr_qi=None, lr_yj=None, reg_bu=None, reg_bi=None, reg_pu=None, reg_qi=None, reg_yj=None,
random_state=None, verbose=False, cache_ratings=False)
```

Bases: `AlgoBase`

The SVD++ algorithm, an extension of `SVD` taking into account implicit ratings.

The prediction \hat{r}_{ui} is set as:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j \right)$$

Where the y_j terms are a new set of item factors that capture implicit ratings. Here, an implicit rating describes the fact that a user u rated an item j , regardless of the rating value.

If user u is unknown, then the bias b_u and the factors p_u are assumed to be zero. The same applies for item i with b_i , q_i and y_i .

For details, see section 4 of [Kor08]. See also [RRSK10], section 5.3.1.

Just as for `svd`, the parameters are learned using a SGD on the regularized squared error objective.

Baselines are initialized to `0`. User and item factors are randomly initialized according to a normal distribution, which can be tuned using the `init_mean` and `init_std_dev` parameters.

You have control over the learning rate γ and the regularization term λ . Both can be different for each kind of parameter (see below). By default, learning rates are set to `0.005` and regularization terms are set to `0.02`.

Parameters

n_factors – The number of factors. Default is `20`.

n_epochs – The number of iteration of the SGD procedure. Default is `20`.

cache_ratings – Whether or not to cache ratings during *fit()*. This should speed-up the training, and has a higher memory footprint. Default is `False`.

init_mean – The mean of the normal distribution for factor vectors initialization. Default is `0`.

init_std_dev – The standard deviation of the normal distribution for factor vectors initialization. Default is `0.1`.

lr_all – The learning rate for all parameters. Default is `0.007`.

reg_all – The regularization term for all parameters. Default is `0.02`.

lr_bu – The learning rate for b_u . Takes precedence over `lr_all` if set. Default is `None`.

lr_bi – The learning rate for b_i . Takes precedence over `lr_all` if set. Default is `None`.

lr_pu – The learning rate for p_u . Takes precedence over `lr_all` if set. Default is `None`.

lr_qi – The learning rate for q_i . Takes precedence over `lr_all` if set. Default is `None`.

lr_yj – The learning rate for y_j . Takes precedence over `lr_all` if set. Default is `None`.

reg_bu – The regularization term for b_u . Takes precedence over `reg_all` if set. Default is `None`.

reg_bi – The regularization term for b_i . Takes precedence over `reg_all` if set. Default is `None`.

reg_pu – The regularization term for p_u . Takes precedence over `reg_all` if set. Default is `None`.

reg_qi – The regularization term for q_i . Takes precedence over `reg_all` if set. Default is `None`.

reg_yj – The regularization term for y_j . Takes precedence over `reg_all` if set. Default is `None`.

random_state (int, RandomState instance from numpy, or `None`) – Determines the RNG that will be used for initialization. If int, `random_state` will be used as a seed for a new RNG. This is useful to get the same initialization over multiple calls to `fit()`. If RandomState instance, this same instance is used as RNG. If `None`, the current RNG from numpy is used. Default is `None`.

verbose – If `True`, prints the current epoch. Default is `False`.

pu

The user factors (only exists if `fit()` has been called)

Type

numpy array of size (n_users, n_factors)

qi

The item factors (only exists if `fit()` has been called)

Type

numpy array of size (n_items, n_factors)

yj

The (implicit) item factors (only exists if `fit()` has been called)

Type

numpy array of size (n_items, n_factors)

bu

The user biases (only exists if `fit()` has been called)

Type

numpy array of size (n_users)

bi

The item biases (only exists if `fit()` has been called)

Type

numpy array of size (n_items)

```
class surprise.prediction_algorithms.matrix_factorization.NMF(n_factors=15, n_epochs=50,
biased=False, reg_pu=0.06, reg_qi=0.06, reg_bu=0.02, reg_bi=0.02, lr_bu=0.005, lr_bi=0.005,
init_low=0, init_high=1, random_state=None, verbose=False)
```

Bases: `AlgoBase`