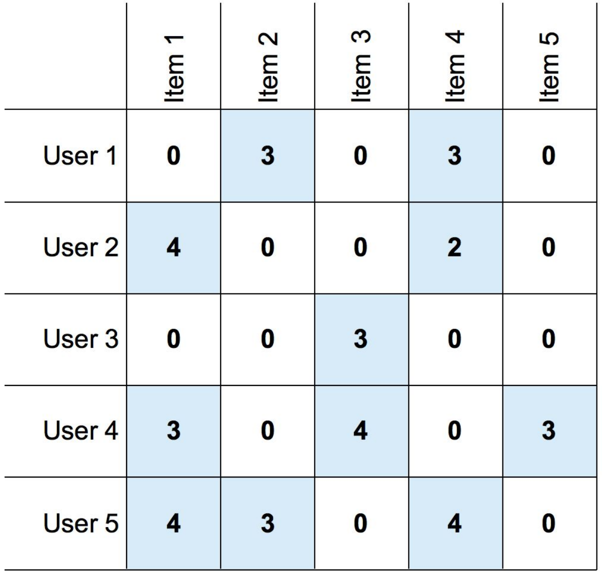
Collaborative Filtering

Learning Portfolio 7

universität
innsbruck

Collaborative Filtering

Collaborative filtering is a technique used in Recommender Systems, so that past similar preferences of users inform future preferences. It works by displaying the preferences of each user in a vector. The similarity between users is measured as cosine similarity. Computed cosine similarity in turn can be used as weights for the ratings of other users to predict a rating for a certain user.



	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	0	3	0	3	0
User 2	4	0	0	2	0
User 3	0	0	3	0	0
User 4	3	0	4	0	3
User 5	4	3	0	4	0

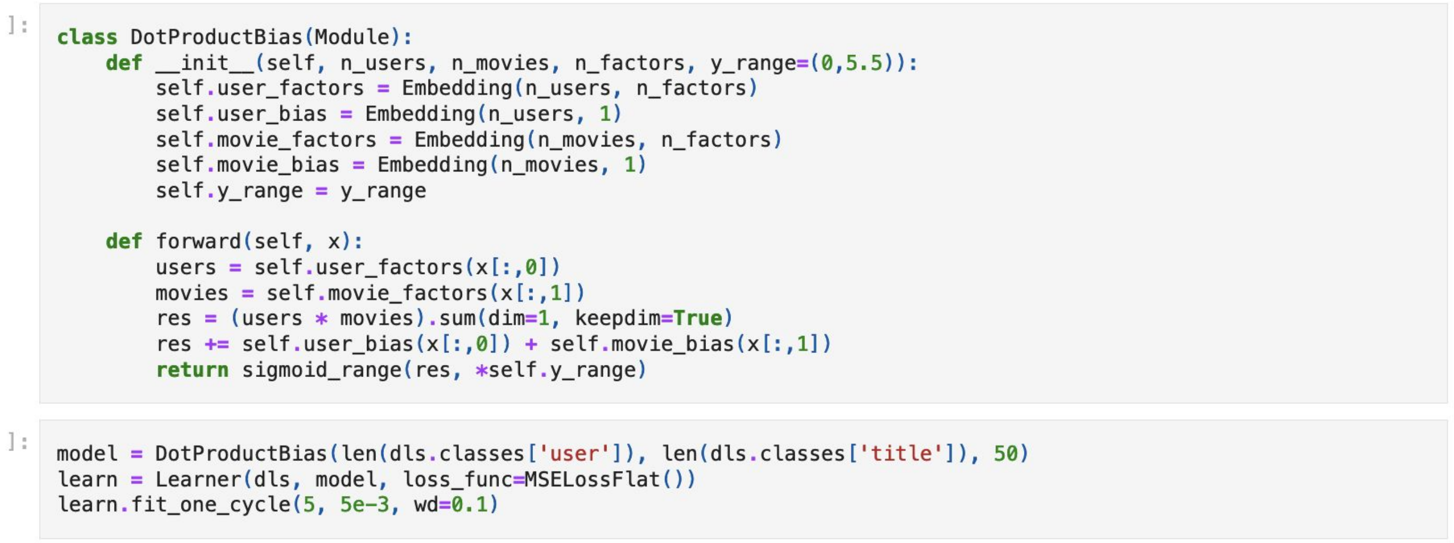
Collaborative Filtering: Limitations and Mitigations

Limitation	**Possible Mitigation**
Grey Sheep Problem	Prediction of ratings using metadata
Black Sheep Problem	Prediction of ratings using metadata
Matrix sparsity	Predict ratings based on actions taken (e.g. views)
Matrix size	Batch training, Gradient accumulation
Subgroup overrepresentation	Monitoring
Bootstrapping	Use average rating for new user/item

universität innsbruck

Building a collaborative filtering model

Building a collaborative filtering model
from scratch



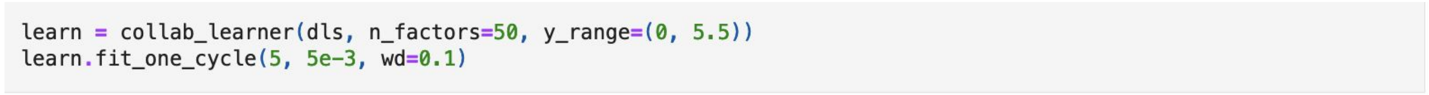
```
]:
class DotProductBias(Module):
    def __init__(self, n_users, n_movies, n_factors, y_range=(0,5.5)):
        self.user_factors = Embedding(n_users, n_factors)
        self.user_bias = Embedding(n_users, 1)
        self.movie_factors = Embedding(n_movies, n_factors)
        self.movie_bias = Embedding(n_movies, 1)
        self.y_range = y_range

    def forward(self, x):
        users = self.user_factors(x[:,0])
        movies = self.movie_factors(x[:,1])
        res = (users * movies).sum(dim=1, keepdim=True)
        res += self.user_bias(x[:,0]) + self.movie_bias(x[:,1])
        return sigmoid_range(res, *self.y_range)
```

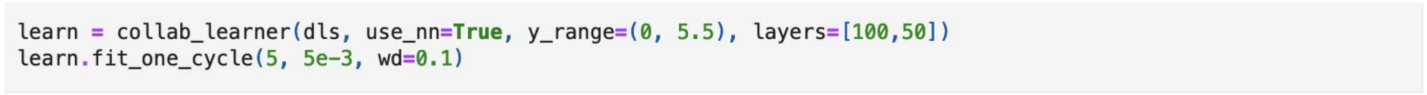
```
]:
model = DotProductBias(len(dls.classes['user']), len(dls.classes['title']), 50)
learn = Learner(dls, model, loss_func=MSELossFlat())
learn.fit_one_cycle(5, 5e-3, wd=0.1)
```

Building a collaborative filtering model

Using fastAI classes



```
learn = collab_learner(dls, n_factors=50, y_range=(0, 5.5))
learn.fit_one_cycle(5, 5e-3, wd=0.1)
```



```
learn = collab_learner(dls, use_nn=True, y_range=(0, 5.5), layers=[100,50])
learn.fit_one_cycle(5, 5e-3, wd=0.1)
```

Kontakt

Fabian Leuk
12215478
fabian.leuk@student.uibk.ac.at