Deep Models of Interactions Across Sets

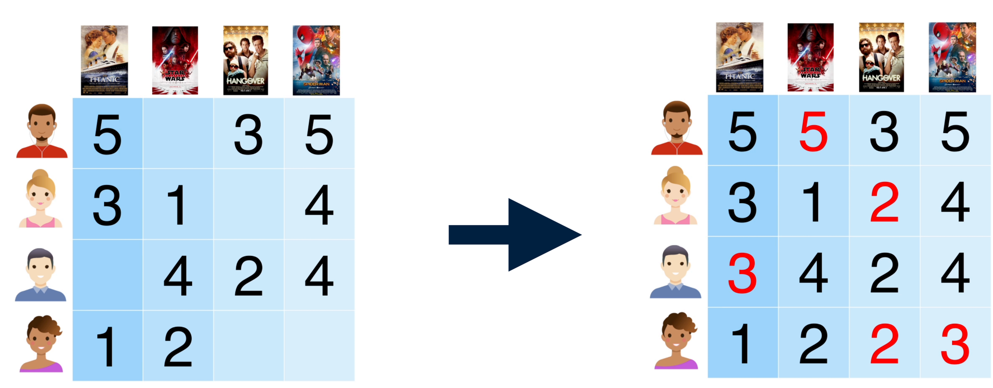
Jason Hartford, Devon R. Graham, Kevin Leyton-Brown, Siamak Ravanbakhsh
University of British Columbia

ABSTRACT

- We present the most expressive parameter tying scheme possible for operating on exchangeable data.
- We perform a number of experiments using this parameter tying scheme in two different deep models.
- State-of-the-art results on benchmark matrix completion tasks.
- Very strong results applying trained models to previously unseen data, including from disjoint datasets.

EXCHANGEABLE DATA

- Example: Movie recommender system.



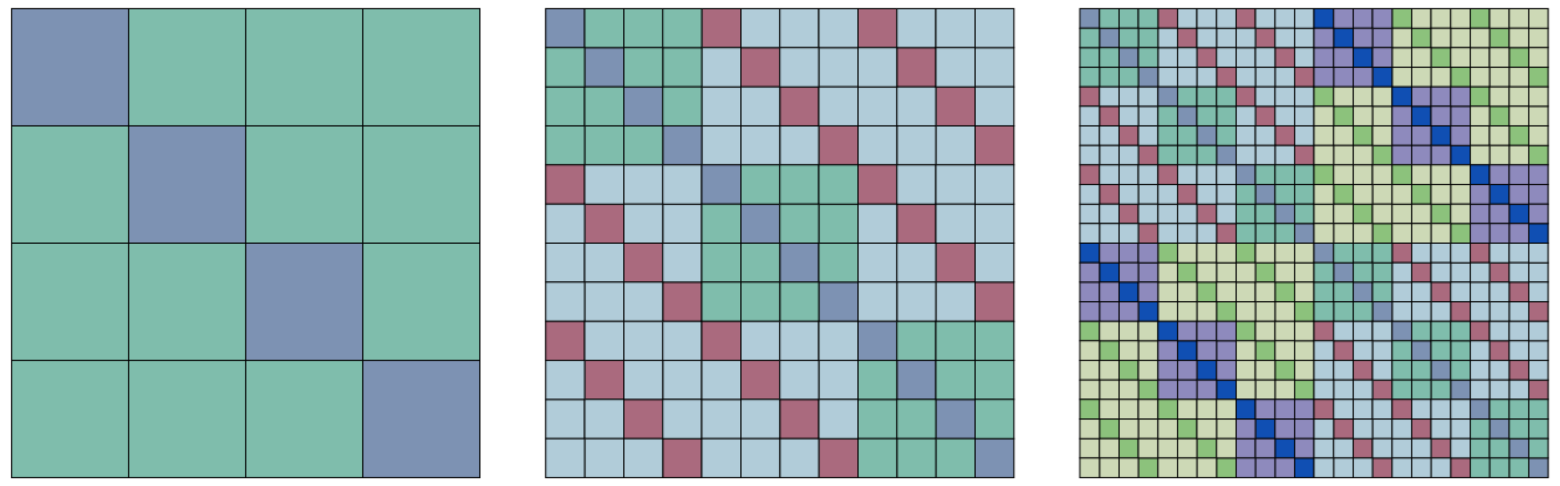
- Permuting rows/columns does not change input data.

EXCHANGEABLE MATRIX LAYER

- Define each output element of our layer to be:

$$Y_{i,j} = \sigma\left(w_1 X_{i,j} + w_2 \sum_{i'=1}^{n} X_{i',j} + w_3 \sum_{j'=1}^{m} X_{i,j'} + w_4 \sum_{i'=1}^{n} \sum_{j'=1}^{m} X_{i',j'} + w_5 \right)$$

- Equivalent to a simple parameter tying scheme. Shown here for 1, 2 and 3D cases. Each colour represents a tied parameter.



Exchangeable vector | Exchangeable matrix | Exchangeable tensor

- Layer extends to higher dimensions and to sparse inputs.
- Constant number of parameters, but each parameter can be a vector/matrix to achieve better expressivity, similar to convolutions.
- Can incorporate per-row, per-column, and global input features.

EQUIVARIANT MODEL

- **We want to constrain our model to make the same predictions over all row/column permutations of the input.**
- **Equivalent to seeing all possible permutations of rows/ columns of the input data.**
- **Amounts to cheap data augmentation.**

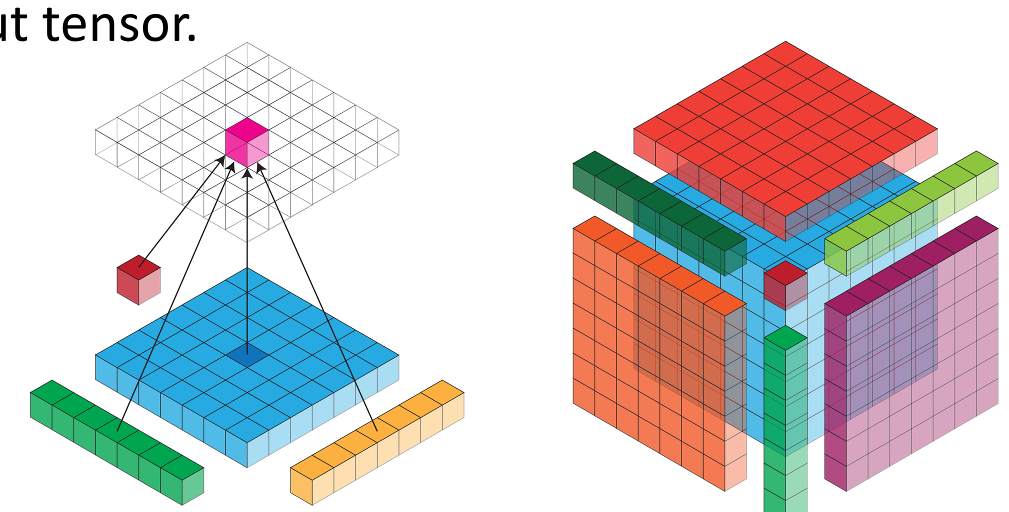
THEORETICAL RESULTS

Exchangeable Matrix Layer ⟷ **Equivariant Model**

- Will make the same predictions on any row/column permutation of input.
- Permutations *across* rows/columns will result in different predictions.
- Increasing number of parameters violates equivariance.

BUILDING DEEP MODELS

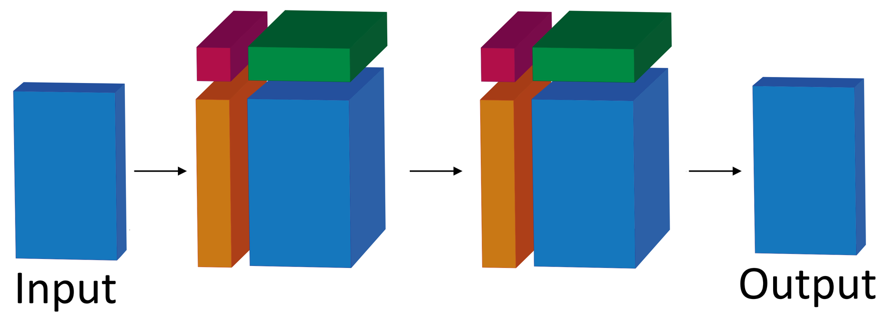
- Our exchangeable matrix layer pools over the dimensions of the input tensor.



Matrix Input | 3D tensor input

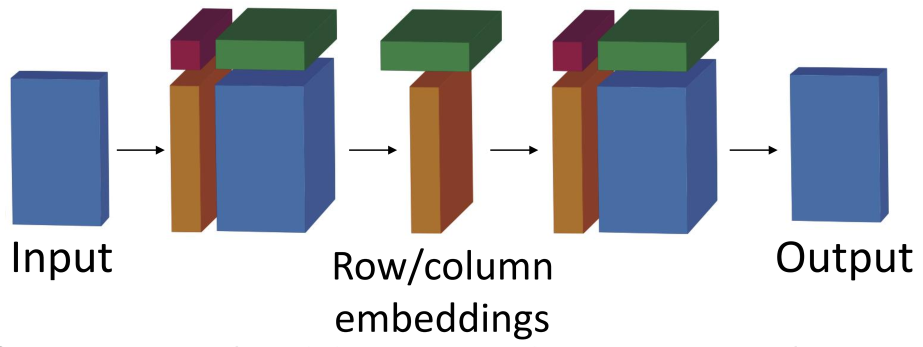
- We use this layer to build deep models.

Self-Supervised Model



- Randomly removes entries, then is trained to predict their value.

Factorized Autoencoder



- Produces row/column feature embeddings, and recreates the input from these embeddings.

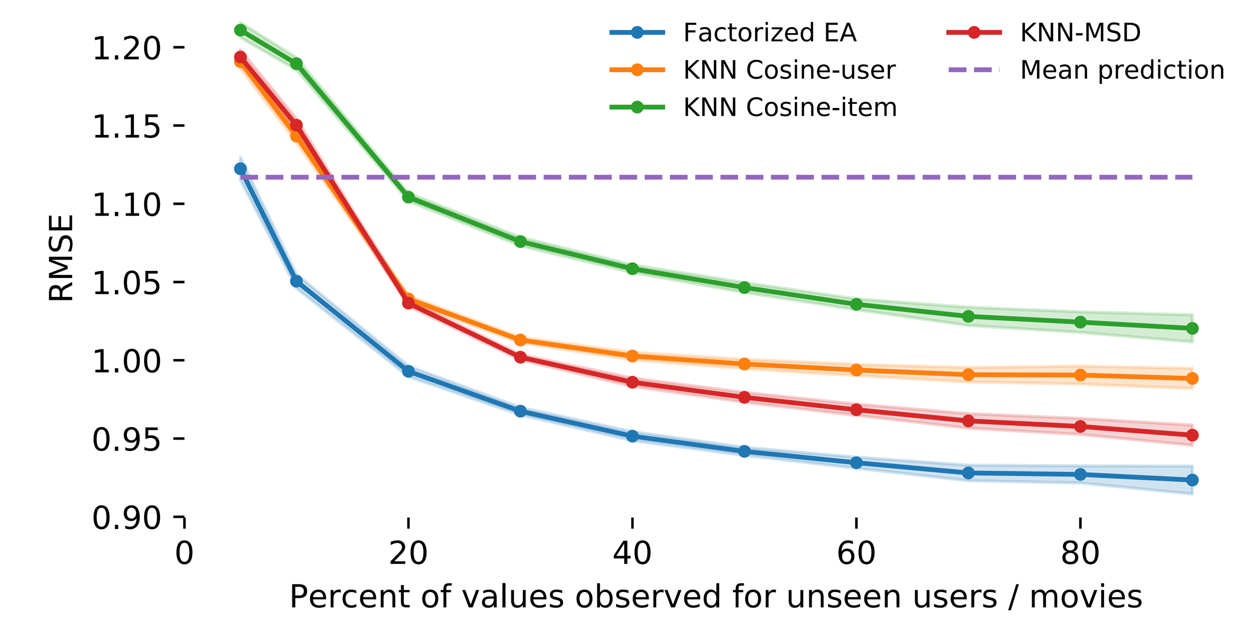
EXPERIMENTAL RESULTS

- State-of-the-art for matrix completion (RMSE):

Model	ML-100k	Flixter	Douban	YahooMusic
Rao et al., 2015	0.945	1.313	0.833	38.0
Monti et al., 2017	0.929	1.179	0.801	22.4
Berg et al., 2017	**0.910**	0.941	**0.734**	20.5
Ours	**0.910**	**0.908**	0.738	**20.0**
Ours (trained on ML-100k)	0.910	0.987	0.766	23.3

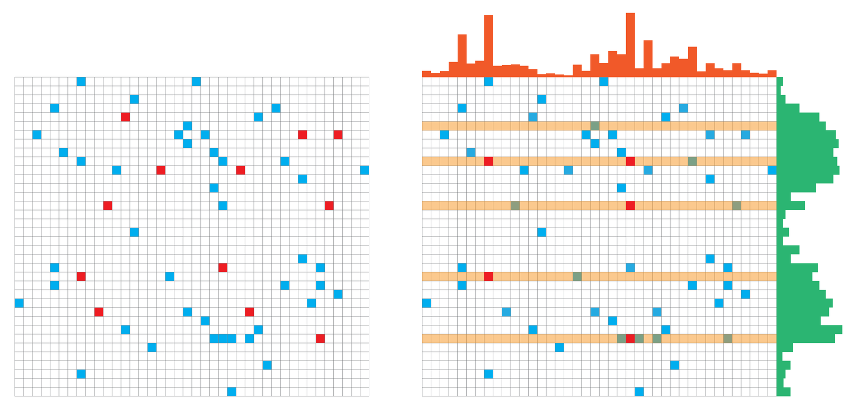
APPLICATION TO NEW DATA

- Trained on MovieLens-100k.
- Evaluated on subsets of MovieLens-1M.
- Use p% of subset as "observed", predict remaining (1-p)%.



SUBSAMPLING IN LARGE MATRICES

- Our models take the whole data matrix as input.
- With large datasets, deep models do not fit into GPU memory.



Uniform Sampling | Conditional Sampling

- Subsampling sparsifies interactions, limiting model performance.

Effects of Subsampling