Representation Learning: Serial-Autoencoder for Personalized Recommendation

Yi ZHU, Yishuai GENG, Yun LI, Jipeng QIANG, Xindong WU

Frontiers of Computer Science, DOI: 10.1007/s11704-023-2441-1

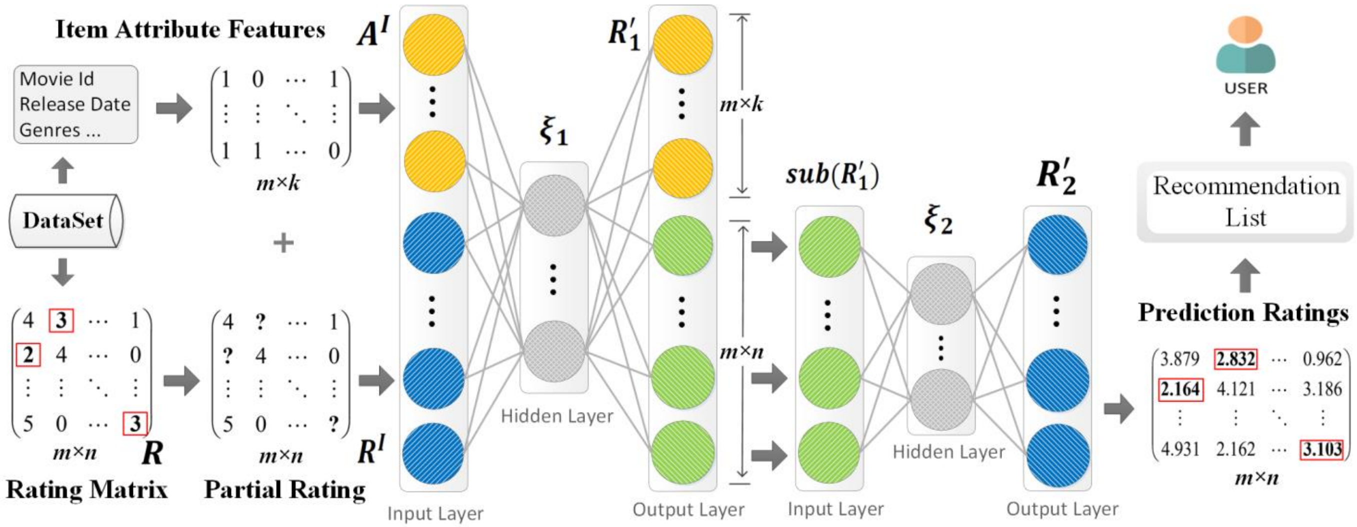
Problems & Ideas

Problems:

- Most models introduce auxiliary information by expanding the dimensionality of the input layer, and the unequal dimensionality of the input and output layers of the model leads to the loss of some key feature information during reconstruction.
- Most existing autoencoder-based recommendation methods rely on the single autoencoder model, which poses challenges for learning different features of data.

Ideas:

- The reconstructed rating information can be further enhanced efficiently by retaining the reconstructed auxiliary information of the decoding layer.
- A serially connected autoencoder can learn a higher-level and robust feature representation of the predicted rating information.



The whole framework of the proposed serial autoencoder method for personalized recommendation.

Experimental results

Table 1: The performance of our method in MovieTweeting 10K.

Metrics	Methods	Proportion of training data				
		50%	60%	70%	80%	90%
MAE	NMF	1.102	1.059	1.044	1.014	1.003
	SVD++	1.082	1.032	1.021	1.008	0.989
	Wide&Deep	1.065	0.983	0.952	0.927	0.909
	NCF	0.996	0.962	0.933	0.912	0.903
	NFM	1.008	0.979	0.952	0.927	0.914
	HCRSA	1.080	1.029	0.989	0.964	0.945
	GraphRec	0.997	0.956	0.929	0.909	0.893
	LightGCN	0.996	0.948	0.922	**0.902**	**0.891**
	PRKG	1.018	0.976	0.941	0.911	0.896
	SAPR	**0.994**	**0.942**	**0.915**	0.903	0.893
RMSE	NMF	1.303	1.261	1.231	1.153	1.125
	SVD++	1.268	1.224	1.188	1.152	1.132
	Wide&Deep	1.201	1.158	1.126	1.093	1.062
	NCF	1.142	1.109	1.082	1.051	1.031
	NFM	1.153	1.122	1.102	1.068	1.045
	HCRSA	1.201	1.169	1.128	1.096	1.087
	GraphRec	1.136	1.108	1.067	1.036	1.010
	LightGCN	1.134	1.089	1.064	**1.023**	**1.006**
	PRKG	1.167	1.126	1.072	1.034	1.012
	SAPR	**1.132**	**1.084**	**1.061**	1.024	1.008

Tip: The bolder ones mean better.

Table 2: The performance of our method in MovieLens 100K.

Metrics	Methods	Proportion of training data				
		50%	60%	70%	80%	90%
MAE	NMF	0.769	0.765	0.761	0.758	0.755
	SVD++	0.752	0.747	0.741	0.726	0.722
	Wide&Deep	0.721	0.718	0.715	0.712	0.708
	NCF	0.717	0.711	0.704	0.699	0.693
	NFM	0.718	0.709	0.705	0.701	0.697
	HCRSA	0.727	0.724	0.713	0.711	0.703
	GraphRec	0.721	0.714	0.709	0.703	0.701
	LightGCN	0.719	0.711	0.705	0.696	**0.686**
	MetaHIN	0.792	0.786	0.781	0.774	0.768
	PRKG	0.729	0.723	0.714	0.704	0.698
	SAPR	**0.715**	**0.704**	**0.701**	**0.692**	0.687
RMSE	NMF	0.991	0.976	0.965	0.960	0.963
	SVD++	0.979	0.965	0.949	0.932	0.924
	Wide&Deep	0.922	0.917	0.913	0.910	0.908
	NCF	0.914	0.911	0.909	0.907	0.903
	NFM	0.917	0.914	0.910	0.909	0.904
	HCRSA	0.927	0.921	0.907	0.8905	0.897
	GraphRec	0.919	0.908	0.899	0.891	0.887
	LightGCN	0.916	0.903	0.897	0.885	**0.873**
	MetaHIN	1.047	1.032	1.017	1.004	0.989
	PRKG	0.928	0.917	0.913	0.899	0.895
	SAPR	**0.909**	**0.898**	**0.890**	**0.882**	0.874

Tip: The bolder ones mean better.

Table 3: The performance of our method in MovieLens 1M.

Metrics	Methods	Proportion of training data				
		50%	60%	70%	80%	90%
MAE	NMF	0.735	0.727	0.718	0.711	0.708
	SVD++	0.683	0.678	0.674	0.668	0.666
	Wide&Deep	0.702	0.697	0.693	0.694	0.689
	NCF	0.696	0.691	0.686	0.683	0.677
	NFM	0.693	0.688	0.686	0.682	0.679
	HCRSA	0.692	0.687	0.681	0.675	0.668
	GraphRec	0.683	0.679	0.673	0.668	0.664
	LightGCN	0.682	0.676	0.671	0.666	0.659
	MetaHIN	0.756	0.748	0.741	0.736	0.729
	PRKG	0.705	0.696	0.690	0.684	0.679
	SAPR	**0.680**	**0.672**	**0.666**	**0.662**	**0.656**
RMSE	NMF	0.928	0.923	0.918	0.914	0.911
	SVD++	0.879	0.866	0.859	0.851	0.848
	Wide&Deep	0.887	0.881	0.875	0.872	0.868
	NCF	0.882	0.875	0.869	0.863	0.858
	NFM	0.879	0.876	0.871	0.867	0.862
	HCRSA	0.892	0.885	0.879	0.871	0.863
	GraphRec	0.875	0.866	0.857	0.851	0.847
	LightGCN	0.871	0.863	0.853	0.849	0.843
	MetaHIN	0.983	0.968	0.957	0.945	0.932
	PRKG	0.898	0.888	0.881	0.874	0.868
	SAPR	**0.867**	**0.858**	**0.851**	**0.847**	**0.839**

Tip: The bolder ones mean better.

Conclusions:

1. Our method performs better than most models and can effectively alleviate the data sparsity problem.
2. Our method is able to achieve better performance by introducing relatively less side information, which also shows the advantage of the serial autoencoder structure in learning.
3. Our method has a simple structure and universality.