Scalable Approximate NonSymmetric Autoencoder for Collaborative Filtering

Authors: Martin Spišák[1,2], Radek Bartyzal[1], Antonín Hoskovec[1,3], Ladislav Peška[2], Miroslav Tůma[2]

[1]GLAMI, [2]Faculty of Mathematics and Physics, Charles University, [3]Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague

GitHub

Abstract

In the field of recommender systems, **shallow autoencoders** have recently gained significant attention. One of the most highly acclaimed shallow autoencoders is EASE^R, favored for its competitive recommendation accuracy and simultaneous simplicity. However, the **poor scalability of EASE^R** (both in time and **especially in memory**) severely restricts its use in production environments with vast item sets. In this paper, we propose a **hyperefficient factorization technique for sparse approximate inversion** of the data-Gram matrix used in EASE^R. The resulting autoencoder, **SANSA**, is an **end-to-end sparse** solution with **prescribable density** and almost **arbitrarily low memory requirements — even for training**. As such, SANSA allows us to **effortlessly scale the concept of EASE^R to millions of items** and beyond.

Architecture

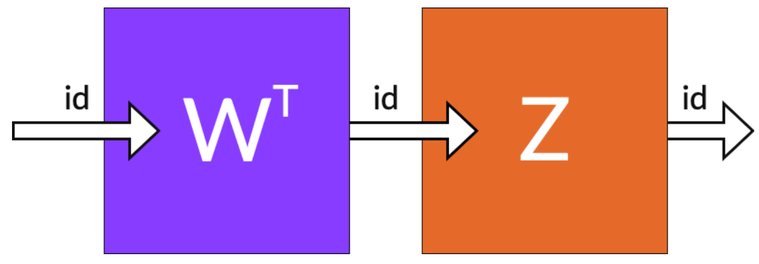

Figure 1. SANSA is a sparse nonsymmetric encoder–decoder model. To disallow recommending input items, we mask the prediction vector, or add an input–output residual connection.

Motivation

Shallow neural networks are simple yet often outperform deep learning approaches in collaborative filtering tasks [1].

Embarrassingly Shallow Autoencoder (EASE^R) [2] is a single-layer neural network with no bias term and no activation. To learn its weight matrix B, EASE^R solves the following convex optimization problem (X is the user-item interaction matrix):

$$\min_B \|X - XB\|_F^2 + \lambda\|B\|_F^2 \text{ s.t. } \operatorname{diag}(B) = \vec{0} \quad (1)$$

Benefits of EASE^R

- Full-rank weights provide higher model capacity compared to low-rank models.
- Scarce feedback from individual users is not a problem when we have enough users in the training data.
- **EASE^R uses long chains of user–item feedback to model item similarity.**

The training procedure uses *closed-form solution* of (1) instead of gradient descent, yielding lowered training complexity[a]. However, this process relies on the calculation of $A^{-1} = (X^TX + \lambda I)^{-1}$, introducing **two challenges for practical application:**

1. Computing A^{-1} is **expensive**: $O(n^{2.3755})$ using Coopersmith-Winograd.
2. Despite the sparsity of input data X, A^{-1} (and also the weights) will be **dense**.

Even if training complexity isn't a problem, the model must fit in RAM for inference. For catalogs with **1 million items, the model size is 4 TB** (using `float32`).

Previous approaches

- *low-rank* factorization of A^{-1} — ELSA [3]
- *full-rank* approximation via inverses of leading item clusters (*local*) — MRF [4]

[a]The complexity depends only on the number of items n.

How to scale EASE^R to *millions* of items?

Approximate EASE^R using a sparse model

- preserve properties of A^{-1} — full rank, symmetric positive definite (SPD)
- enable **arbitrary model compression** — allow users to specify weight density of the resulting model

Method: **factorized sparse approximate inversion**

- sophisticated approaches developed for numerical solvers [5]
- **extract global dominant information** from user–item interaction graph
- A is SPD $\Rightarrow$ increased algorithm efficiency, higher compression

The approximate inverse is computed in three steps:

1. approximate (or incomplete) *sparse* Cholesky factorization
2. *free* initial approximation of the inverse factor
3. (optional) refinement based on Frobenius norm minimization

Scalable Approximate NonSymmetric Autoencoder (SANSA)

input user–item interaction matrix X, L2 regularization λ
compute sparse $LDL^T \approx P(X^TX + \lambda I)P^T$ (for a permutation P)
compute sparse $K \approx L^{-1}$
$W \leftarrow KP$
$Z_0 \leftarrow D^{-1}W$
$\vec{r} \leftarrow \operatorname{diag}(W^TZ_0)$
$Z \leftarrow$ scale the columns of Z_0 by $-1/\vec{r}$
return W^T, Z

Algorithm 1. The training procedure of SANSA is based on factorized sparse approximate inversion. Proper choice of fill-in reducing permutation P is critical for accuracy and efficiency. The final scaling is applied to the decoder only.

Experiments on large, sparse dataset

Dataset properties

#Users: 52 643
#Items: 91 599
#Interactions: 2 984 108
(2 380 730 train, 603 378 test)
Density: 0.062%

Item-item density: 3.937%

Results

- 3x faster training with 10x less memory compared to MRF
- orders of magnitude faster and cheaper than other models
- new state-of-the-art accuracy on the dataset

Amazon Books

results reprinted from [6]:

	SANSA (ICF)	MRF ($r=0$)	MRF ($r=0.5$)	EASE^R	SLIM	ITEMCF	ULTRAGCN
recall@20	**0.077**	0.071	0.069	0.071	0.075	0.074	0.068
nDCG@20	**0.064**	0.058	0.055	0.057	0.060	0.061	0.056
Training resources							
vCPU	**2**	16	16	28	28	28	20*
memory usage (GB):							
peak	**9.18**	96.45	96.58	—— not measured; costly ——			
average	**3.87**	49.12	49.75	—— not measured; costly ——			
time	**49 s**	172 s	167 s	222 m	316 m	57 m	45 m

*and a GPU (RTX 2080)

Table 1. Thanks to end-to-end sparsity, training SANSA (ICF) on 2 vCPUs takes about 3 times less than MRF on 16 vCPUs and orders of magnitude less than non-sparse model training. The training of SANSA requires minuscule memory — unparalleled even with MRF. As a bonus, it also achieves new state-of-the-art accuracy. The standard error in accuracy measurements is about 0.0005.

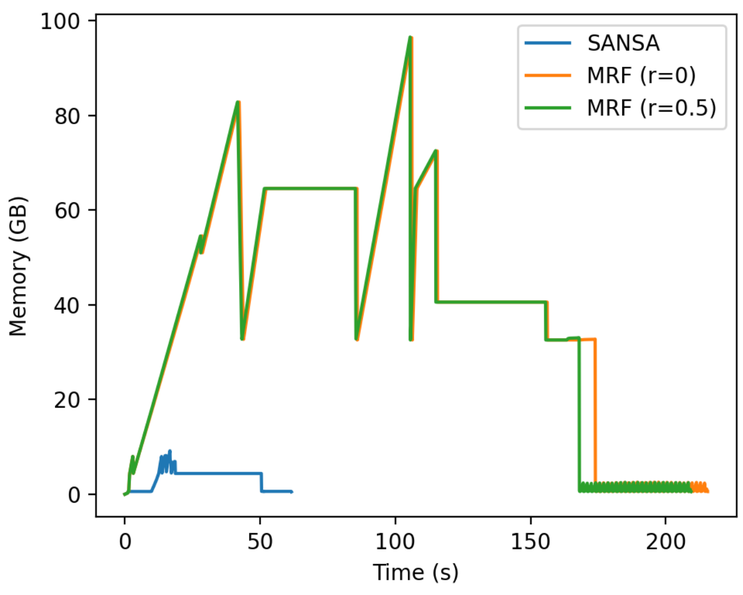

Figure 2. Comparison of time and memory usage of SANSA versus MRF on Amazon Books. The final flatline on each graph corresponds to the evaluation.

Conclusion

Popular shallow autoencoder EASE^R leverages long-distance user–item interaction chains. This ability positively affects the quality of recommendations but also prohibitively increases training and inference costs on large item catalogs.

We introduce a solution to significantly decrease training costs and model size using modern numerical methods for sparse approximate inversion. These techniques are scalable and robust enough to find critical (even long-distance) information. By exploiting the inherent sparsity of user–item interaction data, our end-to-end sparse method achieves substantial efficiency gains over previous approaches that attempt to overpower the problem using dense block operations.

The resulting model SANSA provides a robust yet attainable baseline model for researchers with limited resources and large-scale industry environments with millions of items.

References

[1] Yushun Dong, Jundong Li, and Tobias Schnabel. When newer is not better: Does deep learning really benefit recommendation from implicit feedback? In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 942–952, New York, NY, USA, 2023. Association for Computing Machinery.

[2] Harald Steck. Embarrassingly shallow autoencoders for sparse data. In *The World Wide Web Conference*, WWW '19, page 3251–3257, New York, NY, USA, 2019. Association for Computing Machinery.

[3] Vojtěch Vančura, Rodrigo Alves, Petr Kasalický, and Pavel Kordík. Scalable linear shallow autoencoder for collaborative filtering. In *Proceedings of the 16th ACM Conference on Recommender Systems*, RecSys '22, page 604–609, New York, NY, USA, 2022. Association for Computing Machinery.

[4] Harald Steck. Markov random fields for collaborative filtering. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc.

[5] M. Benzi and M. Tůma. A comparative study of sparse approximate inverse preconditioners. *ANM*, 30(2-3):305–340, 1999.

[6] The BARS Community. Barsmatch: A benchmark for candidate item matching, 2023.

martin.spisak@glami.cz

17th ACM Conference on Recommender Systems (RecSys '23), September 18–22, 2023, Singapore, Singapore.

GLAMI

FACULTY OF MATHEMATICS AND PHYSICS Charles University