A Ridge Regression Ensemble Framework for Collaborative Filtering with Explicit and Implicit Feedback

Kevin Barbieri [*1] **Lena Jankoschek** [*1] **Loris Keist** [*1] **Elias Pinter** [*1]

Abstract

Collaborative filtering is widely used for personalized recommendations, yet individual algorithms often struggle to generalize across diverse data types. This presents a challenge for creating effective recommendation systems that can handle both explicit ratings and implicit interactions. In this work, we propose an ensemble model that combines SVD++, Alternating least squares (ALS), neural collaborative filtering (NCF), and Bayesian Factorization Machines (BFM) using Ridge Regression. Our method integrates explicit ratings and implicit signals, and further enhances SVD++ with contrastive learning to improve representation quality. Our ensemble shows strong predictive accuracy and highlights the potential of hybrid models for future recommender systems.

1. Introduction

Recommender systems provide personalized suggestions using collaborative filtering, which predicts unseen preferences from user-item interactions. Interactions can be explicit (e.g., ratings) or implicit (e.g., views or clicks). We focus on recommending scientific papers using two datasets: one with 1.1M explicit ratings (1–5 scale) from 10,000 scientists on 1,000 papers, and another with 328,839 implicit "intent to read" interactions. This recommendation problem can be framed as completing a sparse rating matrix $A \in \mathbb{R}^{10,000 \times 1000}$, where the goal is to predict ratings for papers a scientist has not yet evaluated.

We extend the two provided baselines by implementing two additional ones for comparison. Separately, we construct an ensemble that combines ALS, NCF, SVD++, and BFMs using Ridge Regression. Notably, we enhance SVD++ with contrastive learning to better align user and item embeddings, improving prediction accuracy. The design of our ensemble allows for seamless extension with additional models, enabling continuous improvement.

[*]Equal contribution [1]Department of Computer Science, ETH Zurich, Switzerland. Correspondence to: Elias Pinter <epinter@ethz.ch>.

Computational Intelligence Lab FS2025, ETH Zuerich

2. Models and Methods

All our models used in the ensemble were trained for 300 epochs. For the PyTorch-based models (SVD++ and NCF), we employed early stopping, halting training when no further improvement was observed on the validation set after a specified patience period. Additionally, hyperparameters were tuned through systematic experimentation on the validation set to optimize performance.

2.1. Baseline Models

To evaluate the performance of our model, we compare it with four different baseline methods, which are described below.

2.1.1. Embedding Dot Product Model

The Embedding Dot Product Model (Salakhutdinov & Mnih, 2007; Koren et al., 2009) represents scientists and papers as fixed-dimensional embedding vectors and predicts ratings using their dot product.

Let $\boldsymbol{p}_s \in \mathbb{R}^k$ denote the embedding of scientist s, and $\boldsymbol{q}_p \in \mathbb{R}^k$ the embedding of paper p. The predicted rating $\hat{r}_{sp}$ is:

$$\hat{r}_{sp} = \boldsymbol{p}_s^\top \boldsymbol{q}_p$$

The embeddings are learned by minimizing the squared error over the observed ratings Ω, with ℓ_2-regularization to prevent overfitting.

2.1.2. Singular Value Decomposition

Singular Value Decomposition (SVD) (Funk, 2006; Koren et al., 2009) is a matrix factorization method for collaborative filtering that models user-item interactions using low-dimensional latent factors.

The predicted rating $\hat{r}_{sp}$ of scientist s for paper p is given by:

$$\hat{r}_{sp} = \mu + b_s + b_p + \boldsymbol{q}_p^\top \boldsymbol{p}_s$$

where:

$\mu \in \mathbb{R}$: global average rating

$b_s \in \mathbb{R}$: bias of scientist s

$b_p \in \mathbb{R}$: bias of paper p

$\boldsymbol{p}_s \in \mathbb{R}^k$: k-dimensional latent vector of scientist s

$\boldsymbol{q}_p \in \mathbb{R}^k$: k-dimensional latent vector of paper p

The bias terms account for systematic tendencies of scientists and papers. The dot product $\boldsymbol{q}_p^\top \boldsymbol{p}_s$ captures the *interaction* between the scientist and the paper, modeling how well the paper matches the scientist's preferences.

2.1.3. Singular Value Projection

Singular Value Projection (SVP) (Jain et al., 2010) is a matrix completion method that reconstructs a low-rank approximation of the rating matrix through iterative optimization. Unlike SVD, which learns latent vectors, SVP directly optimizes the full matrix under a rank constraint.
Given a partially observed rating matrix $R \in \mathbb{R}^{s \times p}$, the goal is to find a low-rank matrix $\hat{R}$ that minimizes the squared error over the observed entries Ω:

$$\hat{R} = \arg \min_{\text{rank}(X) \leq k} \|P_\Omega(X - R)\|_F^2$$

Here, $P_\Omega(\cdot)$ is the projection onto observed entries and k is the target rank. The optimization alternates between gradient descent on the observed error and projection onto the rank-k space via truncated SVD, which ensures the solution remains low-rank throughout optimization.

2.1.4. Singular Value Thresholding

Singular Value Thresholding (SVT) (Cai et al., 2010) is a matrix completion algorithm that recovers a low-rank approximation of a partially observed matrix by minimizing the nuclear norm. Unlike SVP, which enforces a hard rank constraint, SVT promotes low-rank solutions via a convex relaxation.
Given a partially observed rating matrix $R \in \mathbb{R}^{s \times p}$, where each entry corresponds to a rating by scientist s for paper p, SVT solves:

$$\hat{R} = \arg\min_X \|X\|_* \quad \text{subject to} \quad P_\Omega(X) = P_\Omega(R)$$

where, in addition to the variables defined in SVP, $\|X\|_*$ denotes the *nuclear norm* of X (i.e., the sum of its singular values), which provides a convex relaxation of the rank minimization problem—making it computationally feasible while still encouraging low-rank solutions.

2.2. SVD++

SVD++ (Koren, 2008) is an extension of the standard SVD algorithm that incorporates implicit feedback into the modeling process. While standard SVD relies solely on explicit ratings, SVD++ leverages user behavior signals to improve predictions. The key idea in SVD++ is to enrich the scientist representation by incorporating implicit feedback. The vector $\boldsymbol{p}_s$ is augmented with a normalized sum of the embeddings $\boldsymbol{y}_j$ of papers in $N(s)$, enabling the model to capture user preferences even without explicit ratings. We further extended this model in our implementation by incorporating a second implicit feedback component: papers a scientist wants to read. This second implicit feedback allows the model to more accurately infer scientist interests by taking into account future intent. Including this information leads to the following updated formulation:

$$\hat{r}_{sp} = \mu + b_s + b_p + \boldsymbol{q}_p^\top \left(\boldsymbol{p}_s + |N(s)|^{-\frac{1}{2}} \sum_{j \in N(s)} \boldsymbol{y}_j + |W(s)|^{-\frac{1}{2}} \sum_{j \in W(s)} \boldsymbol{y}_j^{(w)} \right)$$

where, in addition to the variables defined in the standard SVD formulation, we introduce the following:

$N(s)$: All papers a scientist s has rated

$W(s)$: All papers a scientist s wants to read

$\boldsymbol{y}_j \in \mathbb{R}^k$: k-dim. latent vector of paper j rated by a scientist

$\boldsymbol{y}_j^{(w)} \in \mathbb{R}^k$: k-dim. latent vector of paper j to be read by a scientist

2.2.1. Contrastive Learning

We extended the SVD++ model with a contrastive loss that encourages scientist embeddings to be closer to positively interacted (rated) papers and farther from randomly sampled negatives. For a scientist s, positive item i^+, and negative item i^-, we compute cosine similarities and apply a hinge loss:

$$\mathcal{L}_{\text{contrast}} = \max(0, \gamma - \cos(\mathbf{p}_s, \mathbf{q}_{i^+}) + \cos(\mathbf{p}_s, \mathbf{q}_{i^-}))$$

This term is added to the standard MSE loss during training, weighted by a scaling factor λ. This approach is inspired by contrastive learning methods commonly used in recommendation systems (Hsieh et al., 2017).

2.3. Bayesian Factorization Machines

Bayesian Factorization Machines (BFMs) are an extension of Standard Factorization Machines as first proposed by Rendle. (Rendle, 2010) The model of such machines for an input $\mathbf{x} \in \mathbb{R}^n$ is represented by the following equation:

$$\hat{y}(\boldsymbol{x}) = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle x_i x_j$$

where:

$w_0 \in \mathbb{R}$: global bias

$w_i \in \mathbb{R}$: weight of the i-th variable

$\mathbf{v}_i \in \mathbb{R}^k$: k-dimensional latent representaion of the i-th variable

$\langle \mathbf{v}_i, \mathbf{v}_j \rangle$: interaction between the i-th and j-th variable

In our problem, where both scientists and papers are represented as one-hot encoded vectors, the model becomes:

$$\hat{\boldsymbol{r}}_{sp} = w_0 + w_s + w_p + \langle \mathbf{v}_s, \mathbf{v}_p \rangle$$

2.3.1. Standard Model

Bayesian Factorization Machines as proposed by (Freudenthaler et al., 2011) extend this model by introducing a probabilistic framework using Bayesian Inference by placing priors on w_0, w_i and $\mathbf{v}_i$ and using a likelihood for y_i. The parameters can then be optimized via MAP-estimation with approximate inference methods like Markov Chain Monte Carlo or Gibbs Sampling.
For implementing them, we made use of the myFM library (Ohtsuki, 2025), oriented ourselves on their tutorials, and extended them to be able to make use of our second dataset.

2.3.2. Implicit Features

BFMs typically model explicit user–item interactions but can be extended with implicit feedback to improve performance in sparse settings (Rendle, 2012). Prior work highlights the importance of such signals in recommender systems (Rendle et al., 2019). Following the SVD++ approach, we treat both rated and wishlisted papers as implicit feedback to better capture scientist preferences. We construct relation-aware feature vectors for scientists and papers based on their co-occurrence in the training and implicit datasets. Each scientist is represented by a normalized vector over all papers they have rated or wishlisted, and each paper by the scientists who interacted with it. These are stored using `RelationBlocks` from the `myFM` library.

2.3.3. Ordinal Regression

There also exists a variant of Bayesian Factorization Machines that treats the problem as an ordinal regression task. In this approach, the target variable is considered ordinal, meaning that its values have a clear order, but the distances between those values are not directly interpretable (e.g., *Strongly Disagree* to *Strongly Agree*).(McCullagh, 1980)

2.4. Alternating Least Squares

Alternating Least Squares (ALS) is another matrix factorization technique that tries to approximate a sparse rating matrix $A \in \mathbb{R}^{n\times m}$ as a low-rank product $A \approx UV^\top$, with $U \in \mathbb{R}^{n\times k}$ and $V \in \mathbb{R}^{m\times k}$. The goal is to find these matrices by minimizing the regularized squared error over observed entries:

$$\min_{U,V} \quad \frac{1}{2}\sum_{(i,j)\in\Omega}\left(A_{ij} - \boldsymbol{u}_i^\top \boldsymbol{v}_j\right)^2 + \frac{\lambda}{2}\left(\|U\|_F^2 + \|V\|_F^2\right)$$

where Ω is the set of observed ratings and λ is a regularization term.
This objective is non-convex in the joint parameters (U, V), but it is convex in one set of variables when the other is fixed. For example, fixing V, the objective simplifies to:

$$\min_{U} \quad \frac{1}{2}\sum_{(i,j)\in\Omega}\left(A_{ij} - \boldsymbol{u}_i^\top \boldsymbol{v}_j\right)^2 + \frac{\lambda}{2}\|U\|_F^2$$

which is a convex quadratic function in U. The same holds symmetrically for V when U is fixed.
We implemented ALS using Apache Spark's `pyspark.ml.recommendation` library (Meng et al., 2016). The model was trained using explicit ratings only, without incorporating wishlist data.

2.5. Neural Collaborative Filtering

Neural Collaborative Filtering (NCF) generalizes the fixed inner-product of classic matrix factorization (MF) by replacing it with a learnable non-linear interaction function between user (scientist) and item (paper) embeddings using a neural architecture. Specifically, classical Neural Matrix Factorization (NeuMF) (He et al., 2017) combines two branches:

Generalized Matrix Factorization (GMF) branch:

$p_s, q_p \in \mathbb{R}^d$: the embeddings
$z_{GMF} = p_s \odot q_p$: element-wise product interaction
$\hat{r}_{sp}^{GMF} = w_{GMF}^T z_{GMF} + b$

Mulit-Layer Perceptron (MLP) branch:

$p_s, q_p \in \mathbb{R}^d$: the embeddings
$h^{(0)} = [p_s;\ q_p] \in \mathbb{R}^{2d}$,
$h^{(l)} = \phi(W^{(l)}h^{(l-1)} + b^{(l)}$ for $l = 1, ..., L$

where

$W^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$: weights of layer l
$b^{(l)} \in \mathbb{R}^{d_l}$: bias of layer l
$d_0 = 2d$ and d_l: size of the last hidden layer
$\phi(\cdot)$: activation function

The two models are trained individually and then stitched together using a fusion model, which learns a scalar fusion weight $\alpha \in (0, 1)$:

$$\hat{r}_{sp} = \alpha\hat{r}_{sp}^{GMF} + (1-\alpha)\hat{r}_{sp}^{MLP}$$

We present a modified version of this model, which combines both pathways into a single module capturing both linear (GMF) and non-linear (MLP) interactions.

2.5.1. Single-Node NCF

The Single-Node NCF model directly implements both GMF and MLP in one model, following the Recommenders library (Team, 2023). For each scientist s and paper p, the model learns two sets of embeddings: $p_s^{\text{GMF}}, q_p^{\text{GMF}} \in \mathbb{R}^d$ and $p_s^{\text{MLP}}, q_p^{\text{MLP}} \in \mathbb{R}^m$. The two paths interact as:

$$\mathbf{z}_{\text{GMF}} = p_s^{\text{GMF}} \odot q_p^{\text{GMF}},$$
$$\mathbf{h}_{\text{MLP}} = \text{MLP}\big([p_s^{\text{MLP}};\ q_p^{\text{MLP}}]\big),$$
$$\hat{r}_{sp} = w^\top\left[\mathbf{z}_{\text{GMF}};\ \mathbf{h}_{\text{MLP}}\right] + b_s + b_p, \quad w \in \mathbb{R}^{d+m},$$

where b_s is the scientist bias (a per-scientist offset that represents the deviation of a particular scientist's ratings from the global mean) and, similarly, b_p is the paper bias (a

per-paper offset that models how a specific paper's ratings deviate from the global mean).

2.6. Ensemble

Ensembling is a common technique in machine learning, in which multiple models are combined to improve the overall performance and robustness of predictions. The main idea behind ensembling is that by leveraging the strengths of different models, we can reduce the likelihood of over- or underfitting, while improving generalization to unseen data. We use stacking, combining predictions from multiple models through a meta-learner that makes the final decision. Given the distinct structures of our models and their similar performance levels, we combined them into an ensemble to improve overall results. To construct our ensemble, we split the dataset into a training set and a validation set. The individual models were then trained using the training set. After training, each model made predictions on the validation set. These predictions were subsequently used to train a meta-learner, in our case ridge regression. Once the meta-learner was trained, we applied our trained models to predict ratings for the test set and aggregated their predictions using the meta model to evaluate the test performance.

3. Results

The results of our models, along with the baselines, are presented in Table I. We report the Root Mean Squared Error (RMSE) on the test set, which corresponds to the public leader board score on Kaggle. For training, we partitioned the provided dataset into a training set (75%) and a validation set (25%). We also used the additional data of the implicit interactions for training the models that support it. All our predictions have been clipped to the range $[1 - 5]$. The best performance among our individual models was achieved by the BFM approach. The most significant performance boost came from incorporating the implicit scientist-paper interactions. Additionally, using ordinal regression led to a slight improvement. Another method capable of leveraging implicit feedback is SVD++, which yielded the second-best results among the different approaches. Notably, we were able to further improve upon traditional SVD++ by integrating contrastive learning, where user and item embeddings were pulled closer in the presence of an interaction and pushed apart otherwise. These findings confirm that accounting for implicit feedback substantially enhances model performance. Interestingly, the ALS model, which only utilized explicit rating data, was able to achieve performance somewhat comparable to the better models. In contrast, the NCF model yielded the weakest results among the ensemble members. This may be attributed to the complexity and large parameter space typical of neural network-based models. However, despite being the worst individual model in the ensemble, the NCF model still managed to significantly outperform all baseline models. Finally, we observed a modest performance improvement by ensembling our models using ridge regression, leading to our best overall score.

Model	Public RMSE
Baseline Models	
Embedding Dot Product Model	0.94243
SVD	1.20645
SVP	1.16698
SVT	0.98827
Individual Models	
SVD++	0.85312
SVD++ + contrastive learning	0.84888
Bayesian Factorization Machines	0.84530
Bayesian Factorization Machines + implicit	0.83782
Bayesian Factorization Machines + ordinal regression	0.84670
Bayesian Factorization Machines + implicit + ordinal regression	0.83576
Alternating Least Squares	0.85771
Neural Collaborative Filtering	0.87299
Ensemble	
Ensemble (Ridge Regression)	**0.83415**

Table 1. Public RMSE scores of baseline methods, individual models, and the final ensemble model on the Kaggle test set

4. Discussion

Our approach combines multiple collaborative filtering algorithms using Ridge Regression, creating a robust and flexible recommendation framework. By integrating models trained on both explicit ratings and dual implicit feedback, the ensemble effectively adapts to various types of feedback and improves prediction precision. Notably, we enhanced traditional SVD++ by integrating contrastive learning, aligning user and item embeddings based on observed interactions. This modification led to improved predictive performance, demonstrating the potential of combining classical models with modern representation learning techniques. Overall, our method provides a practical, extensible strategy, and its strong performance sets the stage for future research bridging traditional and representation-based methods.

5. Summary

In this work, we develop an ensemble-based collaborative filtering system that combines a diverse set of algorithms, including SVD++, Alternating Least Squares (ALS), Neural Collaborative Filtering (NCF), and Bayesian Factorization Machines (BFMs), using Ridge regression. This approach effectively leverages both explicit ratings and implicit interaction data. In addition, we enhance SVD++ with contrastive learning to improve representation quality. Our results show that this approach yields accurate and reliable recommendations. Overall, our contributions offer a practical and extensible framework and point toward promising directions for future hybrid recommendation systems.

References

Cai, J.-F., Candès, E. J., and Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

Freudenthaler, C., Schmidt-Thieme, L., and Rendle, S. *Bayesian Factorization Machines*. 2011. URL https://hilpub.uni-hildesheim.de/handle/ubhi/17189.

Funk, S. Netflix update: Try this at home. http://sifter.org/~simon/journal/20061211.html, 2006. Accessed: 2025-05-07.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pp. 173–182. International World Wide Web Conferences Steering Committee, 2017. ISBN 978-1-4503-4913-0. doi: 10.1145/3038912.3052569. URL https://doi.org/10.1145/3038912.3052569.

Hsieh, C.-K., Yang, L., Cui, Y., Lin, T.-Y., Belongie, S., and Estrin, D. Collaborative metric learning. In *Proceedings of the 26th International Conference on World Wide Web*, pp. 193–201, 2017.

Jain, P., Meka, R., and Dhillon, I. S. Guaranteed rank minimization via singular value projection. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 23, 2010.

Koren, Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, 2008.

Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37, 2009.

McCullagh, P. Regression models for ordinal data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 42(2):109–142, 1980. ISSN 00359246. URL http://www.jstor.org/stable/2984952.

Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., Freeman, J., Tsai, D., Amde, M., Owen, S., et al. Mllib: Machine learning in apache spark. *Journal of Machine Learning Research*, 17(34): 1–7, 2016. URL https://spark.apache.org/mllib/. Accessed: 2025-05-08.

Ohtsuki, T. A python/c++ implementation of bayesian factorization machines. https://myfm.readthedocs.io/, 2025. Accessed: 2025-05-08.

Rendle, S. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pp. 995–1000, 2010. doi: 10.1109/ICDM.2010.127.

Rendle, S. Factorization machines with libfm. In *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2012.

Rendle, S., Zhang, L., and Koren, Y. On the difficulty of evaluating baselines: A study on recommender systems, 2019. URL https://arxiv.org/abs/1905.01395.

Salakhutdinov, R. and Mnih, A. Probabilistic matrix factorization. In *Advances in neural information processing systems*, volume 20, pp. 1257–1264, 2007.

Team, R. Neural collaborative filtering (single-node) implementation. https://github.com/recommenders-team/recommenders/tree/main/recommenders/models/ncf, 2023.

A. Appendix: Declaration of Originality

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

A RIDGE REGRESSION ENSEMBLE FRAMEWORK FOR COLLABORATIVE FILTERING WITH EXPLICIT AND IMPLICIT FEEDBACK

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
BARBIERI	KEVIN
JANKOSCHEK	LENA
KEIST	LORIS
PINTER	ELIAS

With my signature I confirm that

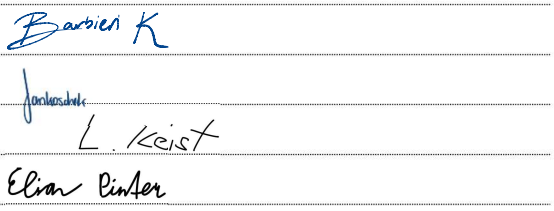
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date

08.05,2025

Signature(s)



For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.