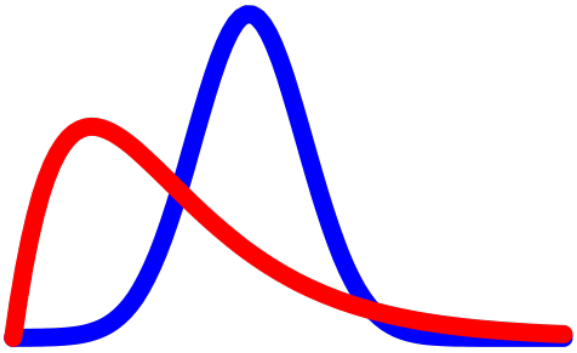
Data Science Lab - Project 1

Bayesian Probabilistic Matrix Factorisation

Anouk Ruer, Jacques Lachouque and Ruben Ifrah
Master 2 IASD - Dauphine-PSL — École Polytechnique



Contents

1 Project Guidelines and Setup

This assignment is part of the *Data Science Lab*, a practical course structured around three group projects designed to connect theoretical concepts with hands-on experimentation. As stated in the course material, each assignment explores a distinct machine learning problem through a combination of algorithmic implementation, literature exploration, and empirical evaluation.

1.1 Objectives and Expected Work

For this first project, the goal is to tackle a collaborative filtering problem using the MovieLens-like dataset provided by the instructors. The goal is to:

- implement at least one baseline method, in particular *Matrix Factorization* trained via Alternating Least Squares (ALS) or Gradient Descent;
- select and implement one or more alternative approaches among those discussed in lecture (e.g. Locality-Sensitive Hashing, PCA variants, Optimal Transport, or Neural Collaborative Filtering);
- analyse design choices such as hyperparameter tuning (e.g. choice of rank k, regularization parameters) and possible uses of side information such as genres.

1.2 Dataset and Evaluation Protocol

The dataset consists of three rating matrices: `train.npy`, `test.npy`, and `eval.npy`, each containing user–item ratings over a set of 600 users and 1600 movies. The models must be trained on the union of the training and test matrices, while performance is assessed on the held-out evaluation set.

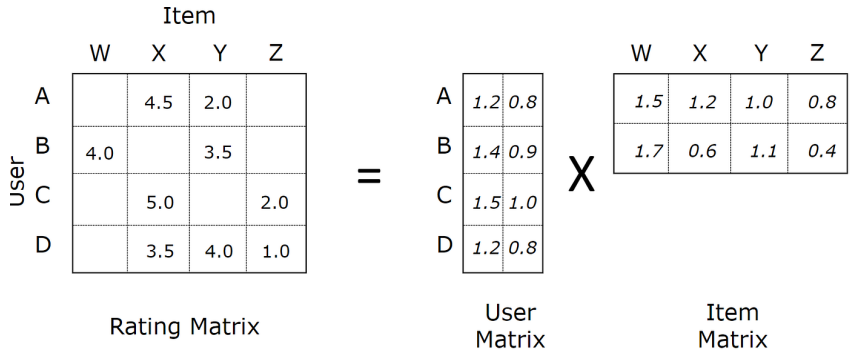


Figure 1: Problem setup: Matrix Factorization for collaborative filtering

The primary quantitative metric is the *Root Mean Squared Error* (RMSE) computed over the evaluation entries:

$$\mathrm{RMSE}(R, \hat{R}, T) = \sqrt{\frac{1}{|T|} \sum_{(u,i) \in T} \left(R_{ui} - \hat{R}_{ui}\right)^2},$$

where R denotes the ground-truth sparse matrix and $\hat{R}$ the completed prediction matrix. Secondary metrics include prediction accuracy on exact ratings and computational efficiency, both of which are taken into account by the evaluation platform.

2 Baseline Methods : ALS & Gradient Descent

Recommender systems often rely on matrix factorization, which decomposes the user–item rating matrix R into two latent matrices U and I, representing user preferences and item characteristics respectively. We implemented two baseline methods Alternated Least-Square(ALS) and Gradient Descent(GD). For both methods we try to solve the same optimisation problem :

$$\min_{I,U} ||R - IU^T||_F^2 + \lambda||I||_F^2 + \mu||U||_F^2$$

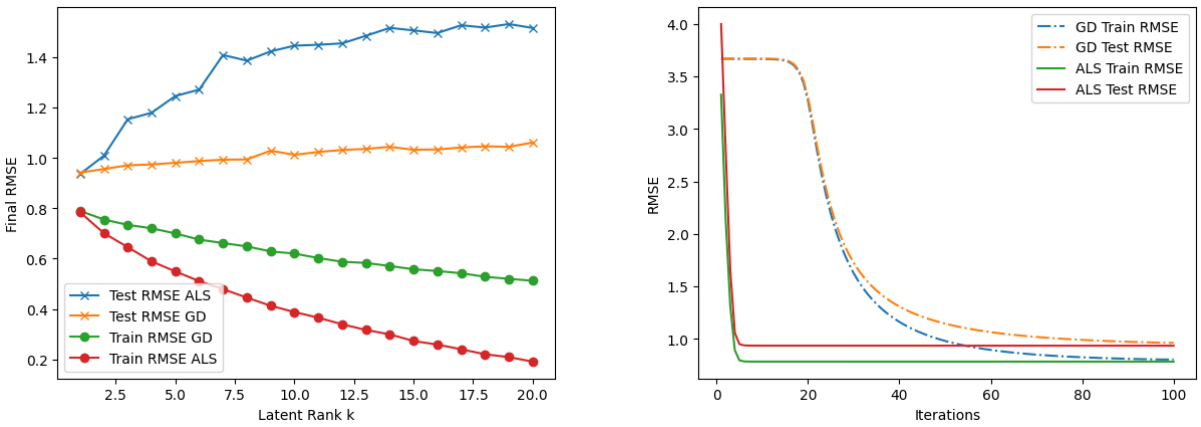
Where $I \in \mathbb{R}^{m,k}$, $U \in \mathbb{R}^{n,k}$.

For both methods, we consider the following hyperparameters: k the latent rank, μ and λ the regularization parameters, the number of iterations and γ the learning rate (only for the Gradient Descent)

We performed cross-validation to find the best parameters for both models and obtained the following results:

- $k = 1$, $\lambda = \mu = 0.1$, $\gamma = 0.001$ and 200 iterations for the Gradient Descent.
- $k = 1$, $\lambda = \mu = 0.1$ and 20 iterations for ALS.

A first observation is that both models share the same values for k, λ, μ and $\lambda = \mu$. The fact that the optimal latent rank is $k = 1$ may seem suprising, but when we examine the test and training RMSE fo $k \in [\![1, 20]\!]$, we can observe that both models begin to overfit for $k \geq 2$ [2a]. For the optimal parameters, both models yield similar RMSE values (around 0.94). However, ALS converges much faster than GD — approximately six times faster [2b].



(a) Influence of latent rank on both methods

(b) RMSE Evolution for GD and ALS

Figure 2: Comparison of the baseline methods

3 Bayesian Probabilistic Matrix Factorization (BPMF)

3.1 Motivation

Probabilistic Matrix Factorization (PMF) (Salakhutdinov and Mnih, 2007) represents user–item interactions by decomposing the rating matrix R into latent user and item matrices U and V. Each rating is modeled as

$$R_{ui} \sim \mathcal{N}(U_u^\top V_i, \alpha^{-1}),$$

where α controls observation noise. PMF learns single point estimates of U and V by maximizing the likelihood (or equivalently minimizing a regularized squared loss). However, this deterministic formulation cannot capture uncertainty: all users and items are treated as equally reliable, even when some have very few ratings. The Bayesian Probabilistic Matrix Factorization (BPMF) model (Salakhutdinov and Mnih, 2008) addresses this limitation by placing hierarchical Bayesian priors on the latent factors, allowing uncertainty and regularization to emerge naturally from the inference process.

3.2 Model Formulation

BPMF augments PMF with Gaussian–Wishart priors over user and item factors:

$$U_u \sim \mathcal{N}(\mu_U, \Lambda_U^{-1}), \quad V_i \sim \mathcal{N}(\mu_V, \Lambda_V^{-1}),$$

$$R_{ui} \sim \mathcal{N}(U_u^\top V_i, \alpha^{-1}),$$

$$(\mu_U, \Lambda_U), (\mu_V, \Lambda_V) \sim \text{Normal–Wishart}(\mu_0, \beta_0, W_0, \nu_0).$$

This hierarchical prior couples all user and item vectors, encouraging them to share statistical strength. In contrast to standard PMF, which provides a single maximum a posteriori (MAP) estimate, BPMF infers a full posterior distribution over all latent representations. This allows each user or item to have its own uncertainty level—broad posteriors for sparse users, sharper ones for dense users—resulting in adaptive regularization.

3.3 Inference via Gibbs Sampling

Exact inference of the posterior $p(U, V, \mu_U, \Lambda_U, \mu_V, \Lambda_V \mid R)$ is intractable. Instead, BPMF employs a Gibbs sampler, a Markov Chain Monte Carlo (MCMC) algorithm that iteratively samples from each conditional distribution:

1. Sample user vectors $U_u \mid V, R$ from multivariate Gaussians.
2. Sample item vectors $V_i \mid U, R$ analogously.
3. Update hyperparameters (μ_U, Λ_U) and (μ_V, Λ_V) using conjugate Normal–Wishart posteriors.

After a *burn-in* phase to reach stationarity, samples are averaged to form the posterior predictive mean:

$$R_{\text{pred}} = \frac{1}{T} \sum_{t > \text{burn-in}} U^{(t)} V^{(t)\top}.$$

This averaging integrates over multiple plausible factorizations, rather than relying on a single estimate, improving robustness and predictive stability.

3.4 Experimental Setup

We followed the configuration of Salakhutdinov and Mnih (2008), setting $\alpha = 2.0$, $\beta_0 = 2.0$, $W_0 = I_k$, $\nu_0 = k$, and varying the latent dimension $k \in \{2, 8, 12, 20\}$. All experiments were run with fixed noise precision and different Gibbs chain lengths (100–200 iterations, burn-in 10–20). Performance was evaluated using Root Mean Squared Error (RMSE) on the test set.

3.5 Results and Analysis

Across all explored configurations ($k \in \{2, 8, 12, 20\}$, $n_{\text{iter}} \in \{50, 100, 200\}$, burn_in $\in \{10, 20, 50\}$), the **test RMSE** consistently converged to a narrow plateau of ≈ 0.854–0.858, improving over our ALS/GD baselines (both around ~ 0.94). We observe:

- **Convergence speed.** Test RMSE stabilizes within ~ 10 Gibbs iterations which is very impressive. Yet better results (0.837 RMSE on the evaluation data points) are achieved when averaging over at least 50/100 iterations. This behavior indicates that satisfactory convergence can be achieved with significantly fewer sampling iterations than those originally recommended in the article, without loss in predictive RMSE. Given that BPMF via MCMC is computationally demanding—due to repeated sampling and matrix updates, reducing the total number of iterations offers a trade-off between runtime and model precision.

- **Capacity vs. generalization.** Increasing the latent dimension beyond $k \in [8, 12]$ does not improve generalization: additional capacity mainly reduces the *training* RMSE while the *test* RMSE saturates, suggesting diminishing returns under current sparsity.

- **Mild overfitting at the snapshot level.** Per-iteration (instantaneous) test RMSE curves are slightly higher than the final value. Averaging $\frac{1}{T}\sum_{t>\text{burn-in}} U^{(t)}V^{(t)\top}$ reduces variance and yields the lower, reported *posterior-mean* RMSE. That was expected following the theory presented in Salakhutdinov and Mnih (2008) and is compensated by the posterior average.

- **Best observed setting.** To obtain our best results we used is $k = 12$, $n_{\text{iter}} = 200$, burn_in $= 30$, with test RMSE near the lower end of the plateau (0.837). Yet computational cost of this setting needs to be addressed at it was one of the longest method to run in the results table of the class. A good trade-off is to lower both $k = 2$ and $n_{iter} = 50$ ($burn\_in = 20$) to already achieve a satisfying 0.86 RMSE while speeding the method by a factor of 10.

Figure 3 illustrates a representative run: the training curve decreases rapidly along with the test curve, both quickly plateau, with small oscillations due to sampling. This behavior is consistent with well-calibrated Bayesian averaging under a Gaussian likelihood.

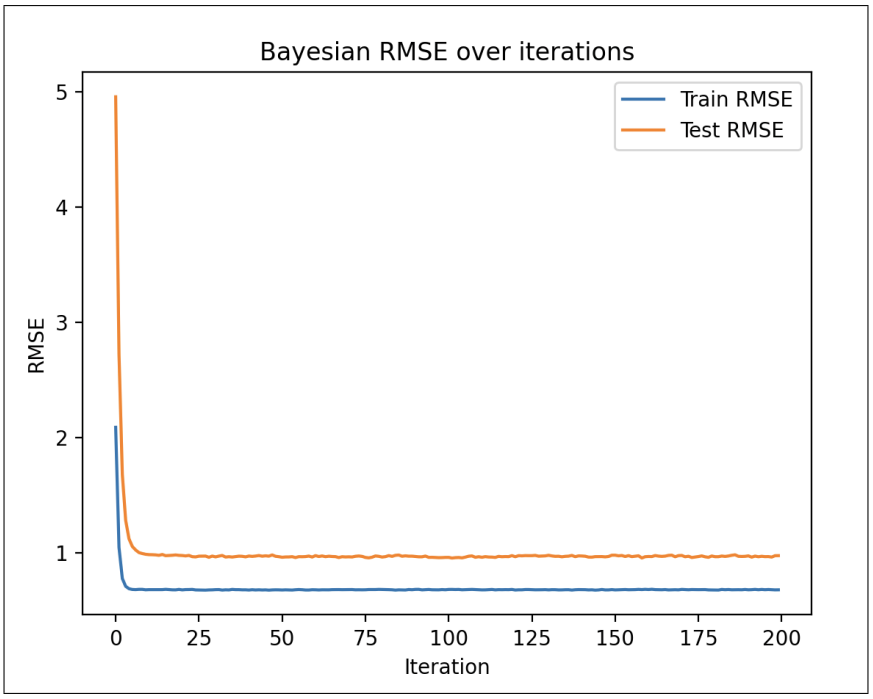


Figure 3: RMSE evolution during Gibbs sampling iterations.

Takeaway. Under our current likelihood and priors, BPMF reaches a stable accuracy regime quickly; further improvements are unlikely to come from more iterations or larger k, but from model choices (e.g., noise calibration, biases, or ordinal likelihood).

3.6 Discussion

Strengths. BPMF integrates over many plausible factorizations rather than committing to a single point estimate. This posterior averaging provides (i) adaptive regularization (users/items with few ratings have broader posteriors) and (ii) increased robustness to sparsity, which explains the consistent improvement over ALS/GD.

Limitations.

- The current model uses a *Gaussian* likelihood for discrete/ordinal ratings (1, 1.5, ..., 5), which caps accuracy. Clipping methods provides satisfying accuracy but degrade the RMSE;
- We keep the observation precision α fixed, instead of learning it (for computational purposes and simplicity);
- There are no explicit user/item bias terms, so part of the intercept structure is left to the factors;
- Gibbs sampling is costlier than ALS/SGD.

4 Results for BPMF

We adopted the same experimental configuration as described in the original paper: $k = 12,\ \alpha = 2.0,\ \beta_0 = 2.0,\ W_0 = I_k,\ \nu_0 = k.$, where k denotes the latent dimensionality. To assess model sensitivity, we experimented with different latent dimensions $k \in \{2, 8, 12, 20\}$ and varied the number of MCMC iterations. The test RMSE consistently converged to a plateau around $\approx$ 0.854–0.858 across all configurations [Figure 1], outperforming the ALS baseline.

Moreover, increasing the latent dimension k beyond a moderate size ($k = 2$) did not yield notable performance gains, suggesting that model complexity has diminishing returns for this dataset. The computational complexity being proportional to the latent dimension is also a discouraging factor.

5 Conclusion

We compared classical matrix factorization baselines (ALS and Gradient Descent) with a Bayesian extension (BPMF) on a sparse rating prediction task. The baselines reached ~ 0.94 RMSE with rapid training for ALS, while BPMF consistently improved test performance to a narrow plateau of $\approx$ 0.854–0.858 by averaging posterior samples. With longer chains and averaging, we observed a best run around 0.837 RMSE at a higher computational cost. These results validate the main advantage of BPMF: integrating over uncertainty yields better generalization than point-estimate PMF/ALS, especially under sparsity.

From a modeling perspective, the Gaussian likelihood and fixed noise precision α are the main bottlenecks now: they cap accuracy and shift the burden of calibration to the factors. Our experiments also show diminishing returns beyond $k \in [8, 12]$ and that ~ 100 Gibbs iterations already deliver stable posterior means, suggesting a practical trade-off between runtime and accuracy.

Takeaways. (i) Bayesian averaging materially improves RMSE over ALS/GD; (ii) short chains suffice for robust posterior means; (iii) capacity alone (higher k) does not translate into better test error on this split.

References

Salakhutdinov, R. and Mnih, A. (2007). Probabilistic Matrix Factorization. *Advances in Neural Information Processing Systems (NIPS)*.

Salakhutdinov, R. and Mnih, A. (2008). Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo. *Proceedings of the 25th International Conference on Machine Learning (ICML)*.