UNIVERSITÀ DEGLI STUDI DI PADOVA

Optimization Methods for Recommender Systems

K. Skurativska S. Zolghadr

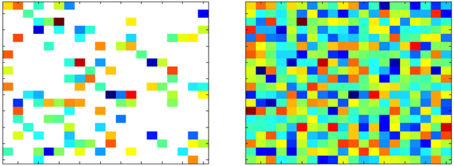
July 1, 2025

Introduction

Matrix completion aims to reconstruct a complete matrix from a subset of observed entries, typically under a low-rank assumption.

We focus on the Frank–Wolfe (FW) algorithm and its pairwise variant (PFW), which are projection-free methods suited for large-scale problems. FW and PFW are implemented with four step-size strategies and evaluated on the MovieLens-100k and X-Wines datasets.

Matrix Completion Problem



Example of a partially observed matrix U (left) and its recovered low-rank matrix X (right).

Let $U \in \mathbb{R}^{n_1 \times n_2}$ be a partially observed matrix, with known entries indexed by $\Omega \subseteq [1 : n_1] \times [1 : n_2]$.

The goal of matrix completion is to recover a low-rank matrix $X \in \mathbb{R}^{n_1 \times n_2}$ that agrees with U on the observed entries.

Matrix Completion Problem

We solve the nuclear-norm-constrained matrix completion problem:

$$\begin{aligned} \min_{X \in \mathbb{R}^{n_1 \times n_2}} \quad & \tfrac{1}{2}\|P_\Omega(X - U)\|_F^2 \\ \text{s.t.} \quad & \|X\|_* \le \tau \end{aligned} \tag{1}$$

- U: partially observed matrix
- Ω: indices of observed entries
- $P_\Omega(Z)_{ij} = \begin{cases} Z_{ij}, & (i,j) \in \Omega \\ 0, & \text{otherwise} \end{cases}$
- $\|X\|_*$: nuclear norm (sum of singular values)
- $\tau > 0$: bounds rank via convex relaxation

Frank–Wolfe for Matrix Completion

Algorithm 1 Frank–Wolfe for Matrix Completion

Require: $X_0 \in \mathcal{B}_*^\tau$, tolerance $\varepsilon > 0$

for $k = 0, 1, 2, \ldots$ **do**

  $G_k \leftarrow P_J(X_k - U)$

  $(u_1, v_1) \leftarrow$ top singular vectors of G_k

  $S_k \leftarrow -\tau\, u_1 v_1^\top$ ▷ LMO solution

  $g_k^{\mathrm{FW}} \leftarrow \langle G_k, X_k - S_k \rangle$ ▷ FW gap

  if $g_k^{\mathrm{FW}} \le \varepsilon$ **then return** X_k

  end if

  $\alpha_k \leftarrow$ line search or $\frac{2}{k+2}$

  $X_{k+1} \leftarrow X_k + \alpha_k (S_k - X_k)$

end for

Pairwise Frank–Wolfe for Matrix Completion

Algorithm 2 Pairwise Frank–Wolfe for Matrix Completion

Require: $X_0 \in \mathcal{B}_*^\tau$, active set $\mathcal{A}_0$, tolerance $\varepsilon > 0$

for $k = 0, 1, 2, \ldots$ **do**

$\quad G_k \leftarrow P_\Omega(X_k - U)$

$\quad (u_1, v_1) \leftarrow$ top singular vectors of G_k

$\quad S_k \leftarrow -\tau\, u_1 v_1^\top$ ▷ FW atom (LMO solution)

$\quad V_k \leftarrow \arg\max_{A \in \mathcal{A}_k} \langle G_k, A \rangle$ ▷ Away atom

$\quad d_k \leftarrow S_k - V_k$

$\quad g_k^{\mathrm{FW}} \leftarrow \langle G_k, X_k - S_k \rangle$ ▷ FW gap

$\quad$ **if** $g_k^{\mathrm{FW}} \leq \varepsilon$ **then return** X_k

$\quad$ **end if**

$\quad \gamma_{\max} \leftarrow$ weight of V_k in X_k

$\quad$ Choose $\alpha_k \in [0, \gamma_{\max}]$

$\quad X_{k+1} \leftarrow X_k + \alpha_k d_k$

Step-Size Strategies

Duality Gap: $g_k = -\langle \nabla f(X_k), d_k \rangle = -\langle G_k, d_k \rangle, \quad$ where $G_k = P_J(X_k - U)$.

- **Exact Line Search:** Closed-form step for quadratic loss:

$$\alpha_k = \frac{g_k}{\|P_J d_k\|^2}$$

- **Diminishing Step Size:** Simple decay rule:

$$\alpha_k = \frac{2}{k+2}$$

- **Armijo Backtracking:** Adaptive geometric decay:

$$f(X_k + \alpha_k d_k) \leq f(X_k) - c\alpha_k g_k$$

  where $c \in (0, 1)$, typically 10^{-4}

- **Lipschitz-Based Rule:** Gradient assumed L-Lipschitz (we use $L = 1$):

$$\alpha_k = \min\left\{1, \frac{g_k}{L\|P_J d_k\|^2}\right\}$$

Implementation

Initialization:

- Start from $X_0 = 0$
- If τ is not provided:

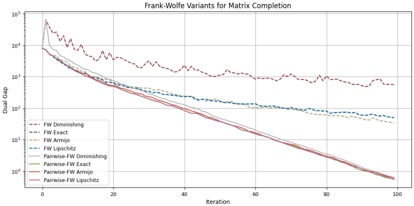
$$\tau = 0.1 \sum_{(i,j)\in\Omega} \sigma_{ij} \quad \text{(observed singular values)}$$

otherwise: use $\tau \in (0, 1)$

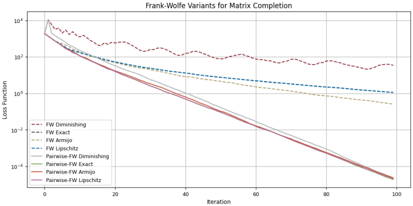
Stopping Criterion:

- Stop if dual gap $g_k < \texttt{TOL}$ or iteration count $k = T_{\max}$.
- Default tolerances:
  - $\texttt{TOL} = 10^{-8}$ (FW)
  - $\texttt{TOL} = 10^{-8}$ (PFW)

Implementation Results



(a) Objective convergence



(b) Dual-gap decay

Training curves on the Unreal dataset (solid: FW, dashed: PFW).

Datasets and Preparation

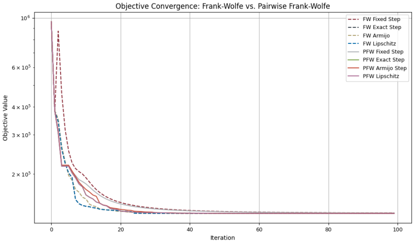
Datasets:

- **MovieLens-100K**:
  - 943 users, 1,682 movies, 100,000 ratings
- **XWine**:
  - 100,646 wines with sensory scores, 1,056,079 users and 21,013,536 ratings

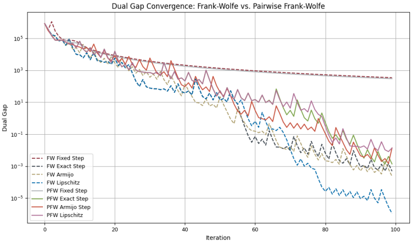
Dataset Preparation:

- Filter users to retain only:
  - MovieLens: users with > 50 ratings
  - XWines: users with > 20 ratings

Training results MovieLense



(a) Objective convergence



(b) Dual-gap decay

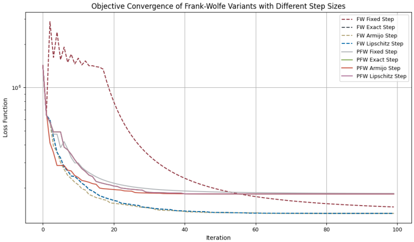
Results on MovieLens

Table: MovieLens Dataset – FW and PFW Variants

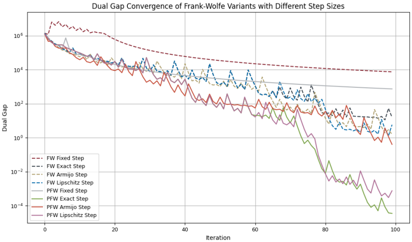
Alg.	Step	Obj.	Gap	Rank	RMSE / Acc (%)
FW	Fixed	133826.01	3.48×10^{2}	29	2.8285 / 98.17
FW	Exact	133132.17	4.89×10^{-4}	7	2.8320 / 98.17
FW	Armijo	133132.17	2.31×10^{-4}	2	2.8320 / 98.17
FW	Lipschitz	133132.17	1.10×10^{-6}	3	2.8320 / 98.17
PFW	Fixed	133731.07	3.00×10^{2}	28	2.8289 / 98.17
PFW	Exact	133132.17	1.37×10^{-3}	7	2.8320 / 98.17
PFW	Armijo	133132.17	1.39×10^{-2}	2	2.8320 / 98.17
PFW	Lipschitz	133132.18	1.36×10^{-2}	7	2.8320 / 98.17

Task: Completion of $563 \times 1\,681$ *rating matrix with nuclear-norm ball* $\tau = 1\,686$

Training results XWines



(a) Objective convergence



(b) Dual-gap decay

Results on XWines

Università degli Studi di Padova

Table: XWines Dataset – FW and PFW Variants

Alg.	**Step**	**Obj.**	**Gap**	**Rank**	**RMSE / Acc (%)**
FW	Fixed	147241.14	7.52×10^{3}	32	2.9934 / 99.66
FW	Exact	132719.08	1.58×10^{1}	13	3.0429 / 99.66
FW	Armijo	132703.82	1.58×10^{0}	1	3.0429 / 99.66
FW	Lipschitz	132702.88	5.77×10^{0}	13	3.0430 / 99.66
PFW	Fixed	182735.70	7.43×10^{2}	27	2.8361 / 99.66
PFW	Exact	181255.43	3.49×10^{-5}	4	2.8411 / 99.66
PFW	Armijo	181255.59	4.03×10^{-1}	3	2.8411 / 99.66
PFW	Lipschitz	181255.43	7.53×10^{-4}	4	2.8411 / 99.66

Conclusion

- Both Frank-Wolfe (FW) and Pairwise Frank-Wolfe (PFW) effectively solve the matrix completion task on the MovieLens and XWines datasets.
- While all step-size strategies achieve comparable RMSE and accuracy, Exact and Armijo steps consistently yield faster convergence and lower-rank solutions, especially when used with PFW.
- PFW demonstrates superior optimization efficiency over FW by enabling faster objective and dual gap reduction, without sacrificing predictive performance.
- For the best trade-off between accuracy, convergence speed, and solution sparsity, PFW with Exact or Armijo step size is recommended.

Questions?