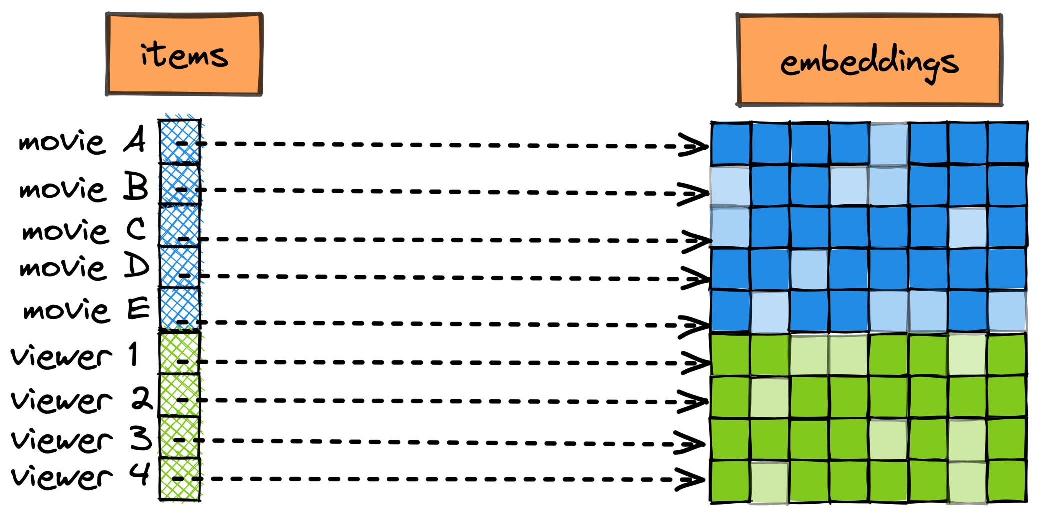
Frequency-aware SGD for Efficient Embedding Learning with Provable Benefits

Yan Li*, Dhruv Choudhary†, Xiaohan Wei†, Baichuan Yuan†, Bhargav Bhushanam†, Tuo Zhao*, Guanghui Lan*
*Georgia Tech †Meta

Embedding Learning

▷ Learning **continuous representation** for **discrete items**.



▷ Learning through historical item-interactions:

$$\min_{\Theta \in \mathbb{R}^{N\times d}} f(\Theta) = \mathbb{E}_{(i,j)\sim \mathcal{D}} [\ell(\theta_i, \theta_j; y_{ij})]$$
$$= \sum_{i\in U, j\in V} D(i,j)\ell(\theta_i,\theta_j; y_{ij})$$

(Nonconvex!)

- $D(i,j)$: occurrence probability of (i,j) item pair
  - $p_i = \sum_j D(i,j), p_j = \sum_i D(i,j)$ denote the occurrence probabilities of item i and j
- $y_{ij} \in \{-1,+1\}$: interaction label between item i, j
- θ_i, θ_j: embedding vector of item i, j, respectively
- N: number of items
- d: embedding dimension

How to learn embeddings, efficiently?

▷ **Standard Practice**

- Popular choices of methods: **Adagrad/Adam**.
- **SGD** gives *significantly (incomparably) worse* performance.
  - Liu et al. '20, Understanding the difficulty of training transformers.
  - Zhang et al. '19, Why are adaptive methods good for attention models?

SGD v.s. Adaptive Methods

▷ **Items distribution follows power-law.**

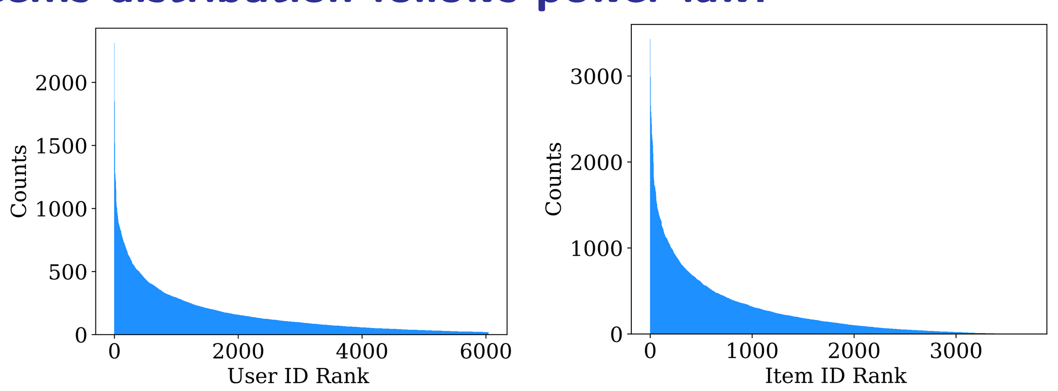


Figure 1: User and Movie occurences (descending) in Movielens.

▷ Adam/Adagrad uses **larger** learning rate for **infrequent** items, thus learns infrequent items faster than SGD.

▷ Existing convergence rate of Adaptive methods (Adagrad/Adam) compared to SGD:

- Convex setting:
  - Duchi et al. '11: better dimensional dependency.
- Nonconvex setting:
  - Ward et al. '18, Defossez et al. '20, Chen et al. '18, Zhou et al. '18:
    - ⋆ hardly matches SGD.
    - ⋆ improvement relies on strong assumptions.

Q1: Can we reconcile theory-practice gap?

Q2: Can we build better methods than existing ones?

Frequency-aware SGD

Frequency-aware SGD

Input: Total iterations T, token frequency $\{p_k\}_{k\in X}$, learning rate $\{\eta_k^t\}_{k\in X, t\in[T]}$, $\eta_k^t = \min\{1/(4L), \alpha/\sqrt{Tp_k}\}$.
Initialize: $\Theta^0 \in \mathbb{R}^{N\times d}$, sample $\tau \sim \mathrm{Unif}([T])$.
For $t = 0 \ldots \tau$:
Sample $(i_t, j_t) \sim \mathcal{D}$, calculate:
$g_{i_t}^t = \nabla_{\theta_{i_t}} \ell(\theta_{i_t}, \theta_{j_t}; y_{i_t,j_t})$, $g_{j_t}^t = \nabla_{\theta_{j_t}} \ell(\theta_{i_t}, \theta_{j_t}; y_{i_t,j_t})$
Update parameters:
$\theta_{i_t}^{t+1} = \theta_{i_t}^t - \eta_{i_t}^t g_{i_t}^t$, $\theta_i^{t+1} = \theta_i^t$, $\forall i \in U, i \neq i_t$
$\theta_{j_t}^{t+1} = \theta_{j_t}^t - \eta_{j_t}^t g_{j_t}^t$, $\theta_j^{t+1} = \theta_j^t$, $\forall j \in V, j \neq j_t$
Output: Θ^τ

▷ *SGD - but adaptive to item frequency. Convergence rate?*

Theorem 1 (FA-SGD). *Take proper α in FA-SGD, we have*

$$\mathbb{E}\|\nabla f_k^\tau\|^2 = \mathcal{O}\left(\frac{\sqrt{p_k}\sqrt{\sum_{l\in X} p_l \sigma_l^2 (f(\Theta^0) - f^*) L}}{\sqrt{T}}\right).$$

Theorem 2 (Standard SGD). *Take learning rate policy to be $\eta_k^t = \min\left\{\frac{1}{4L}, \frac{\alpha}{\sqrt{T}}\right\}$, where α is chosen properly, we have*

$$\mathbb{E}\|\nabla f_k^\tau\|^2 = \mathcal{O}\left(\frac{\sqrt{\sum_{l\in X} p_l^2 \sigma_l^2 (f(\Theta^0) - f^*) L}}{\sqrt{T}}\right).$$

Here $\nabla f_k^\tau = \partial f(\Theta^\tau)/\partial \theta_k$ denotes the partial gradient w.r.t the embedding of item k.

♠ **Convergence of each embedding is frequency-dependent** ♠

Provable Speed-up for Imbalanced Data

Corollary 3 (Exponential Tail). *Let $U = \{i_n\}_{n=1}^{|U|}$, $V = \{j_m\}_{m=1}^{|V|}$, where i_n denote the user with n-th largest frequency, j_m denote the item with the m-th largest frequency. Suppose $p_{i_n} \propto \exp(-\tau n), p_{j_m} \propto \exp(-\tau m)$, for some $\tau > 0$. Define U_T as the set of users whose frequencies are within e-factor from the highest frequency. Then given $|U|, |V| \geq \frac{1}{\tau}$, FA-SGD, compared to standard SGD:*

1. *Obtains the same rate of convergence, for the top users U_T and top items V_T;*
2. $\mathbb{E}\|\nabla f_{i_n}^\tau\|^2$ *can converge faster by a factor of* $\Omega\{\exp(\tau(n - |U_T|))\}$ *for* $i_n \in U \setminus U_T$ *;*
3. $\mathbb{E}\|\nabla f_{j_m}^\tau\|^2$ *can converge faster by a factor of* $\Omega\{\exp(\tau(m - |V_T|))\}$ *for* $j_m \in V \setminus V_T$.

(Q1) ♠ First theoretical speed-up of adaptive methods without algorithmic assumptions ♠

Additional Benefits

▷ **A fully online variant** of FA-SGD – named CF-SGD.

- No requirement for exact frequency $\{p_k\}_{k\in X}$, use online estimate $\{\widehat{p}_k\}_{k\in X}$.
- Maintains the same convergence properties as FA-SGD.

▷ **Memory efficient (Q2)**

- SGD - 1X model size
- Adagrad - 2X model size (second-order moment).
- Adam - 3X model size (first/second-order moment).
- FA-SGD - $(1+\epsilon)$-X model size ($\epsilon \ll 1$).

Experiments - Recommendation Systems

▷ **Movielens-1M dataset with FM and DeepFM model.**

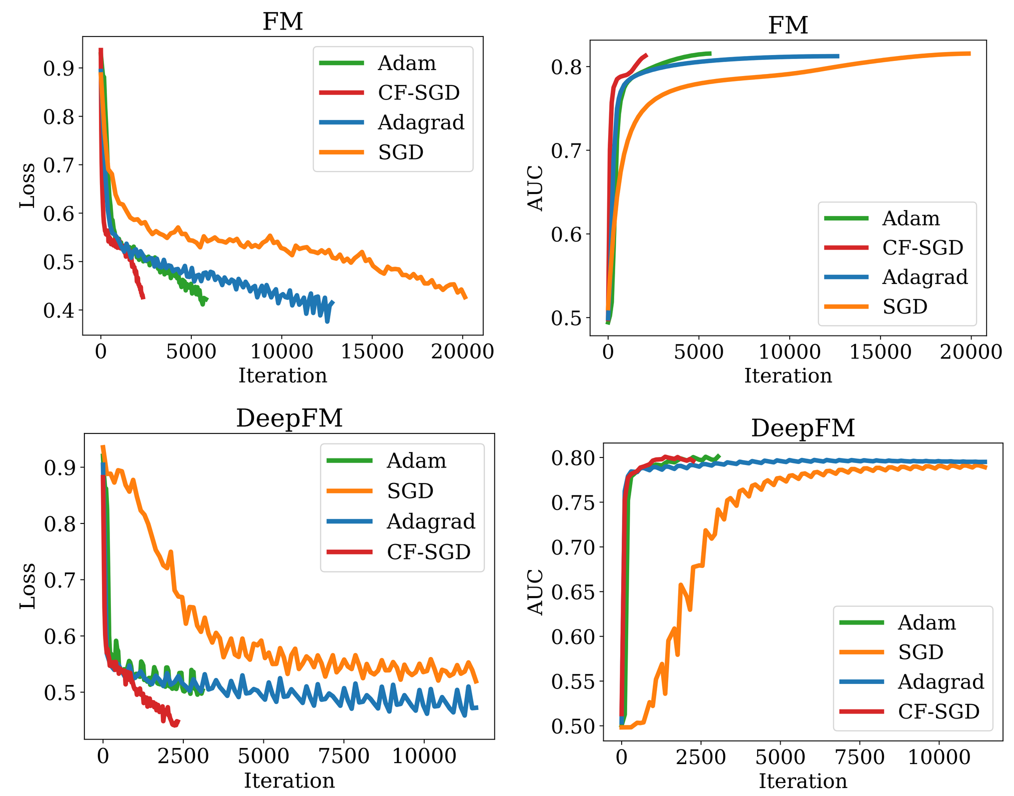


Figure 2: (Left). Training loss. (Right). Validation AUC.

CF-SGD significantly outperforms standard SGD, and is highly competitive against Adam, Adagrad.

Experiments - Recommendation Systems

▷ **Criteo 1TB Click Logs.**

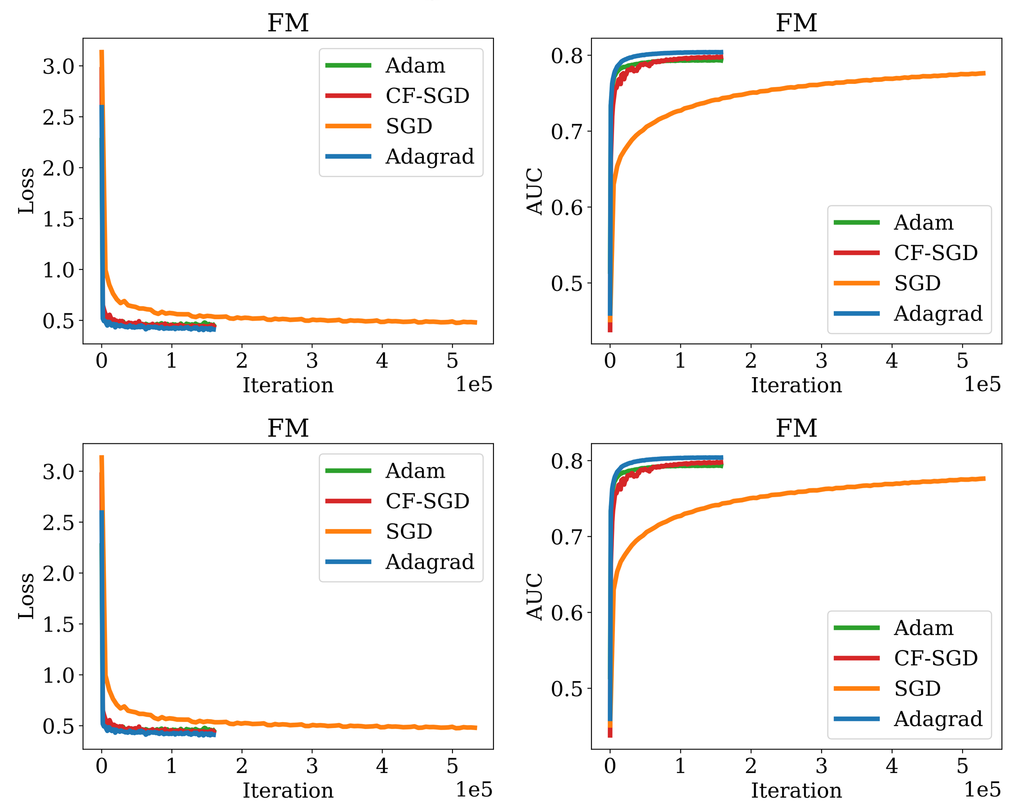


Figure 3: (Left). Training loss. (Right). Validation AUC.

▷ **(Ultra-large) Industrial Recommendation System.**

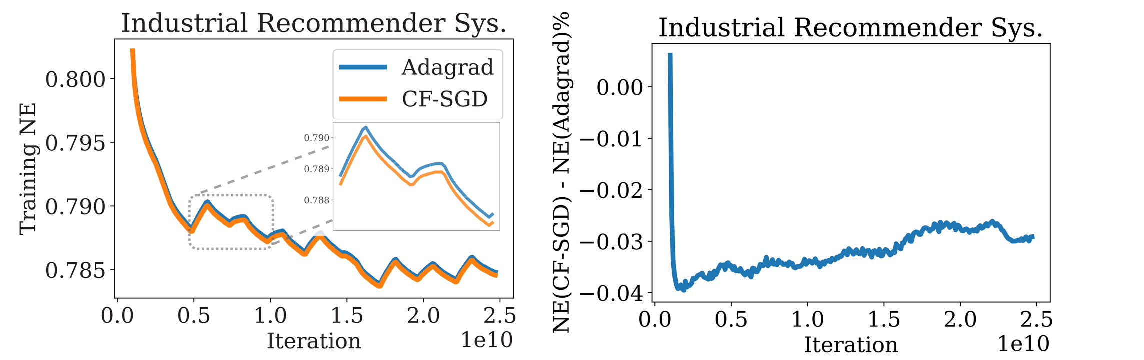


Figure 4: (Left). Train NE curve (lower is better); (Right). Train NE diff.

- ∼2.5 billion examples per day (*25 billion total*).
- ∼ 800 features, with ∼ 10^8 number of items per feature.
- huge memory savings compared to Adagrad/Adam.

Experiments - Word2Vec Embeddings

▷ **Word2Vec: CBOW and Skip-Gram model.**

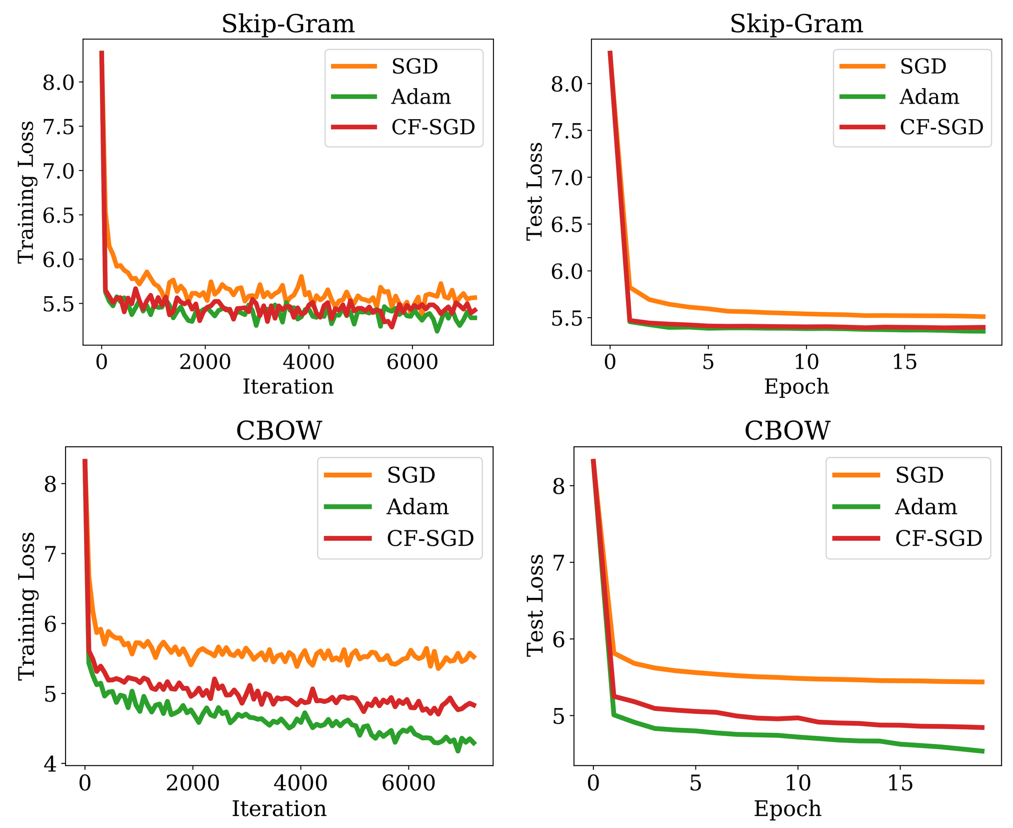


Figure 5: (Left). Training loss. (Right). Testing loss.

Benefits of FA(CF)-SGD

- **Provable speed-up** in nonconvex settings.
- **Consistent empirical strength** across various embedding learning tasks.
- **Huge memory savings** for large-scale problems.