Exponentially convergent algorithms for Supervised matrix factorization

Hanbaek Lyu

Department of Mathematics
University of Wisconsin – Madison
Institute for Foundations of Data Science

Joint work with
Joowon Lee (UW Madison Statistics)
Weixin Yao (UC Riverside Statistics)

SIAM MDS 2024
Efficient and Robust Optimization Techniques for Structured Data Learning

Oct. 22, 2024

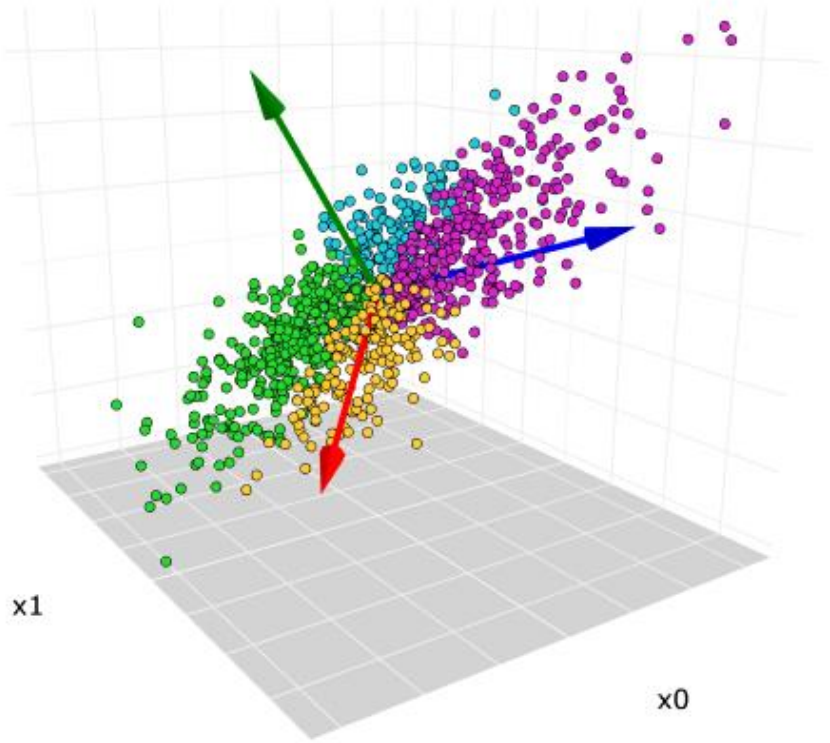
Supervised Matrix Factorization

Dimension Reduction

- $\mathbf{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^p$: high-dimensional data points, *presumably low-rank*
- How do we find the "best" low-dimensional representation?

  - e.g., low-rank matrix factorization, dictionary learning, subspace learning, PCA

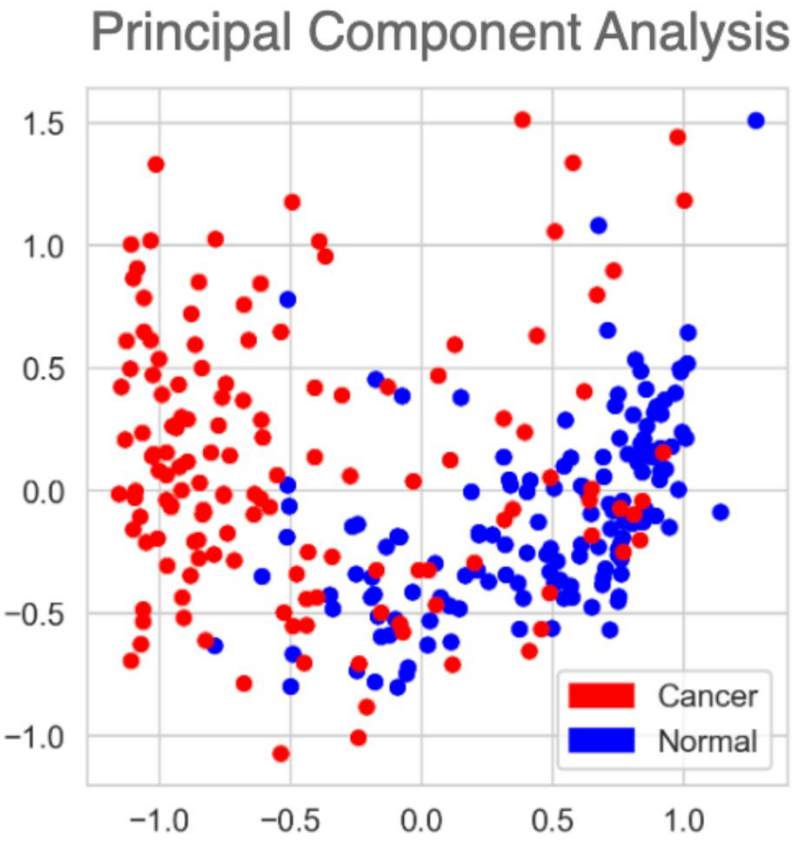
  - PCA: Subspace that explains the most amount of data variance!



Dimension Reduction

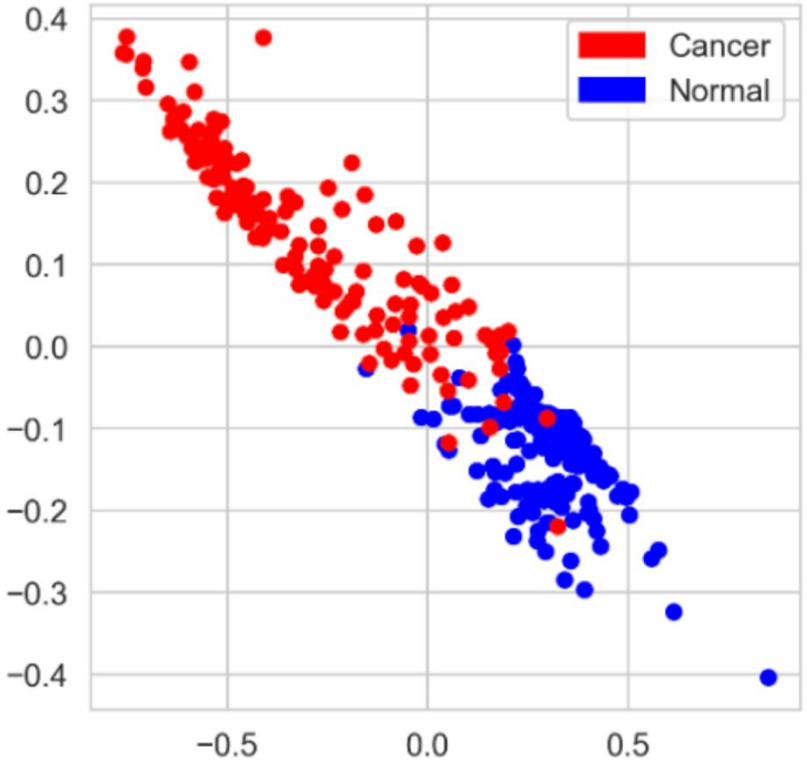
- $\mathbf{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^p$: high-dimensional data points, *presumably low-rank*
- $y_1, \ldots, y_n \in \{0,1\}$: corresponding labels (e.g., cancer vs. normal)

- $\mathbf{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^p$: high-dimensional data points, *presumably low-rank*
- $y_1, \dots, y_n \in \{0,1\}$: corresponding labels (e.g., cancer vs. normal)
- How do we find the "**best**" low-dimensional representation?
  - "Explaining the most amount of data variance" – may not be the right criterion!

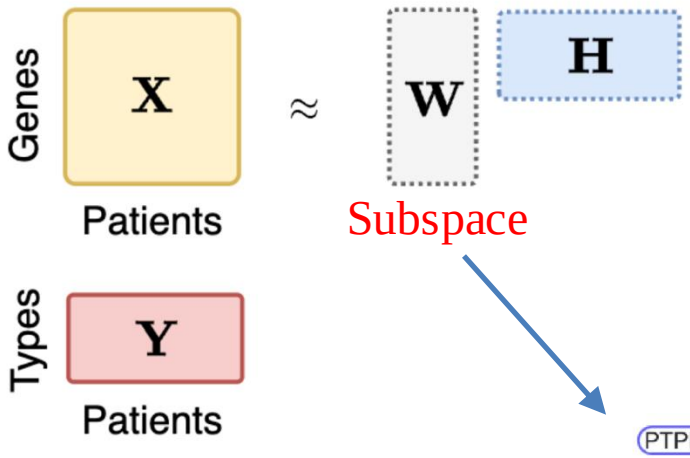


Supervised Dimension Reduction

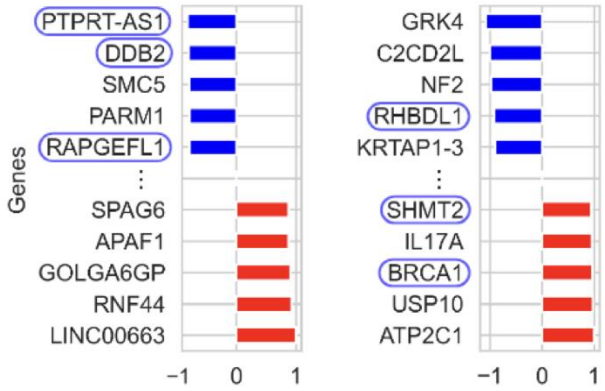
- $\mathbf{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^p$: high-dimensional data points, *presumably low-rank*
- $y_1, \dots, y_n \in \{0,1\}$: corresponding labels (e.g., cancer vs. normal)
- How do we find the "**best**" low-dimensional representation?
  - Define a notion of "**label-aware low-rankness**"



Supervised Matrix Factorization

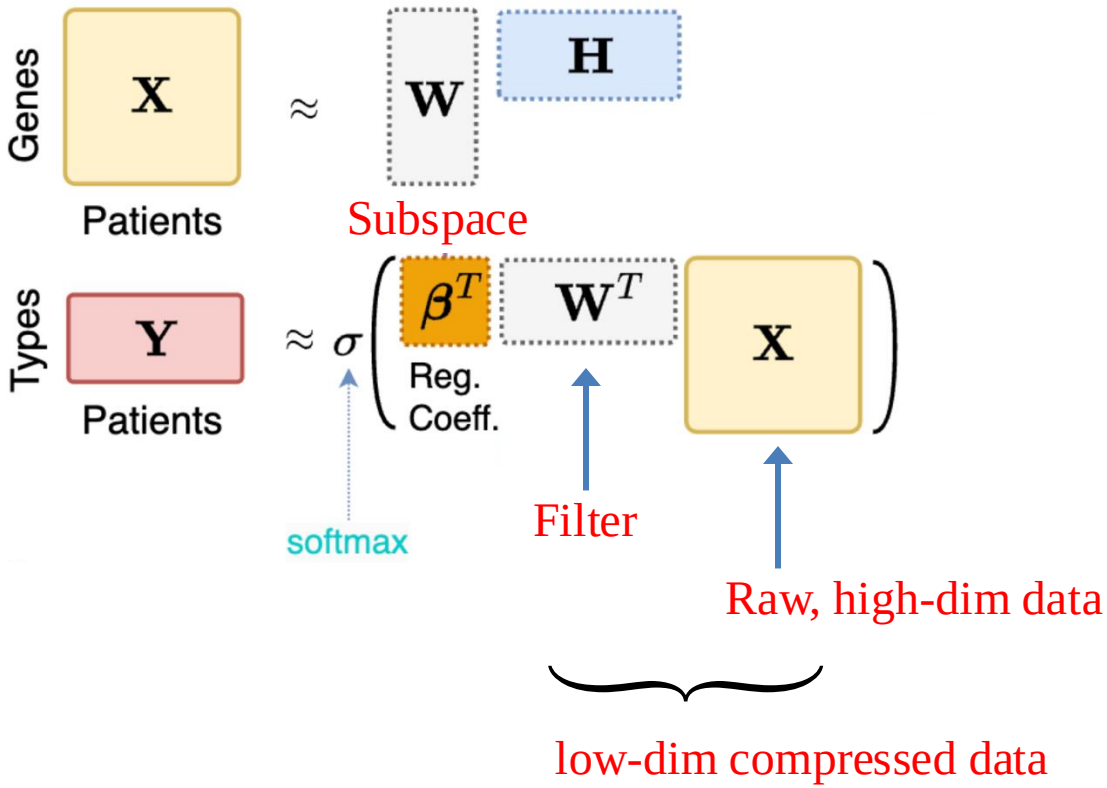


Subspace

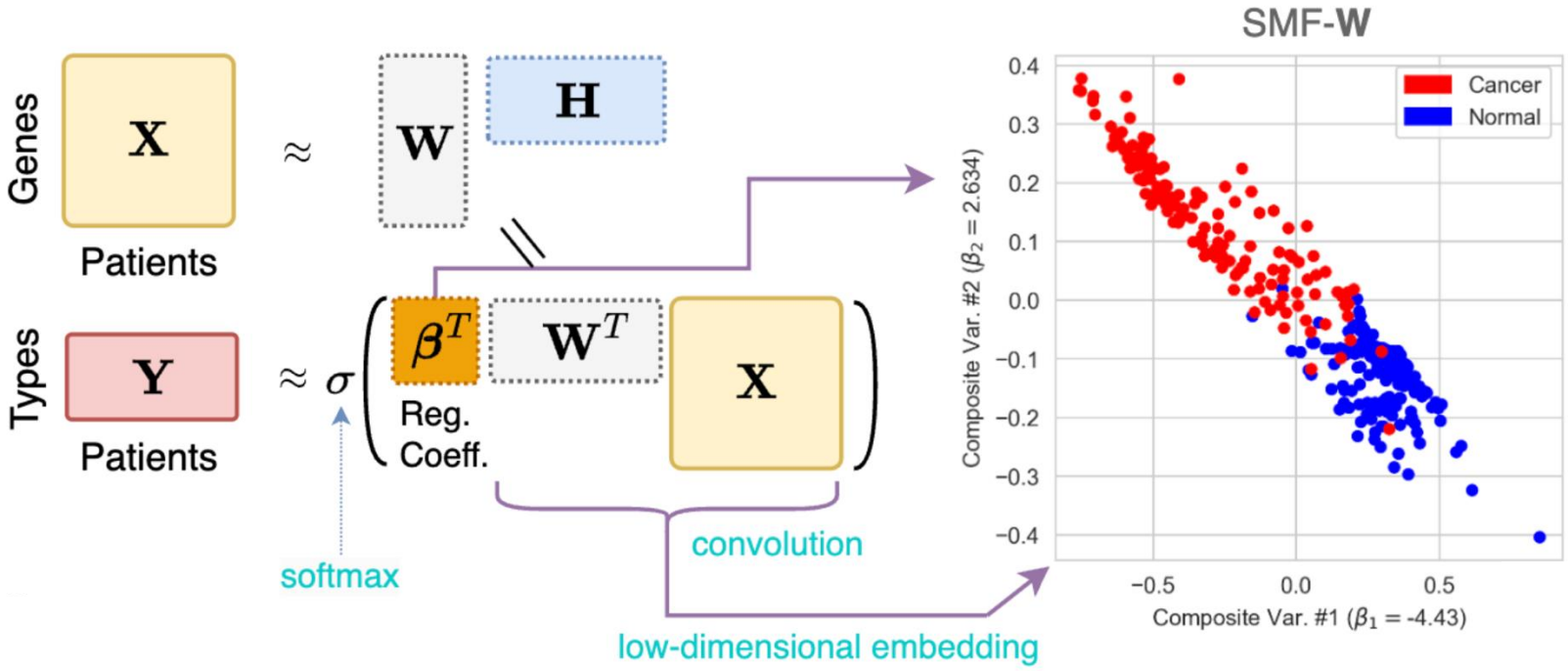


Latent gene groups

Supervised Matrix Factorization

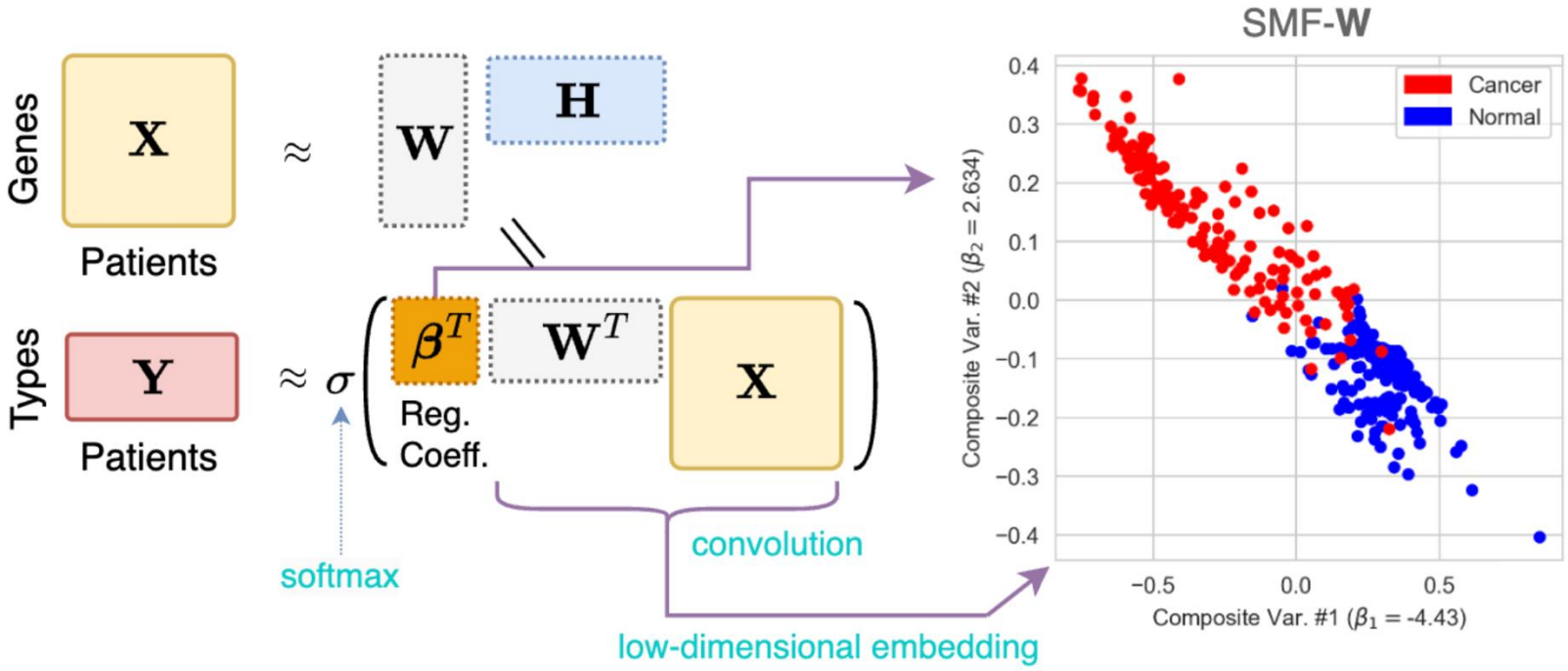


Supervised Matrix Factorization



Simultaneous dimension reduction and classification!

Supervised Matrix Factorization



Classification loss Dimension reduction loss

$$\min_{\mathbf{W},\mathbf{H},\boldsymbol{\beta}} f(\mathbf{W},\mathbf{H},\boldsymbol{\beta}) := \sum_{i=1}^{n} \ell(y_i, \boldsymbol{\beta}^T\mathbf{W}^T\mathbf{x}_i) + \xi\|\mathbf{X}-\mathbf{W}\mathbf{H}\|_F^2$$

Negative log-likelihood under logistic model

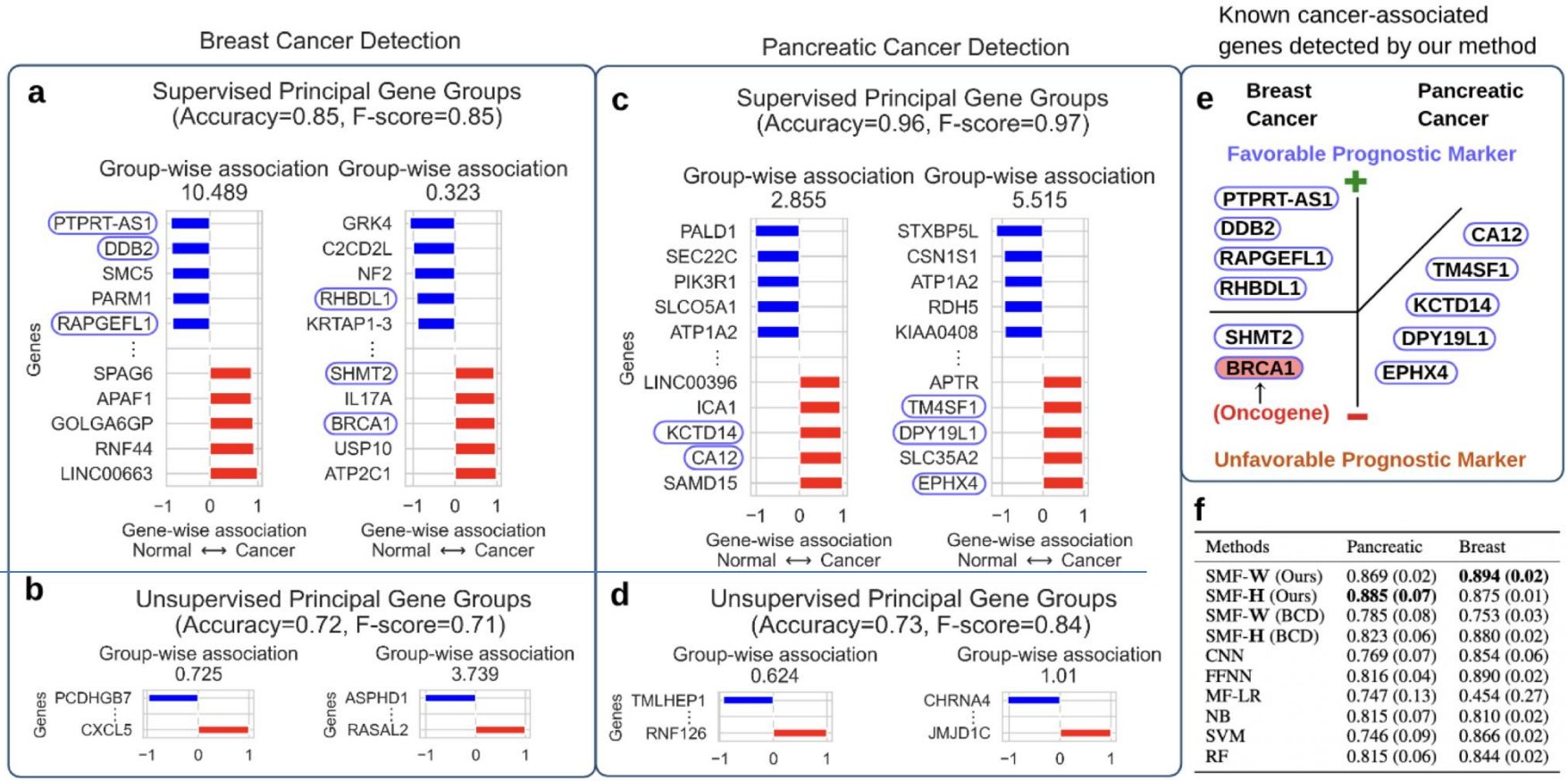
$$\ell(y,a) = \log(1+\exp(a)) - \mathbf{1}_{\{y=1\}}a$$

Parameters
= Three factor matrices $\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{\beta}$

Supervised Matrix Factorization

SMF

PCA



f

Methods	Pancreatic	Breast
SMF-W (Ours)	0.869 (0.02)	**0.894 (0.02)**
SMF-H (Ours)	**0.885 (0.07)**	0.875 (0.01)
SMF-W (BCD)	0.785 (0.08)	0.753 (0.03)
SMF-H (BCD)	0.823 (0.06)	0.880 (0.02)
CNN	0.769 (0.07)	0.854 (0.06)
FFNN	0.816 (0.04)	0.890 (0.02)
MF-LR	0.747 (0.13)	0.454 (0.27)
NB	0.815 (0.07)	0.810 (0.02)
SVM	0.746 (0.09)	0.866 (0.02)
RF	0.815 (0.06)	0.844 (0.02)

SMF identifies latent gene groups associated with cancers

It even identifies **known oncogenes** and **prognostic markers**!

- SMF has been around from ‘08 (Mairal et al.)
- Has been used in lots of applications, but with ad hoc algorithms
- Not much rigorous results have been known before (both computationally and statistically)

- SMF has been around from ‘08 (Mairal et al.)
- Has been used in lots of applications, but with ad hoc algorithms
- Not much rigorous results have been known before (both computationally and statistically)

Optimization for SMF

1. *Adaptive Block Coordinate Descent*
2. *Low-rank PGD*

Stepsizes in PGD

(PGD) $$\boldsymbol{\theta}_{n+1} \leftarrow \Pi_{\boldsymbol{\Theta}} \left(\boldsymbol{\theta}_n - \alpha_n \nabla f(\boldsymbol{\theta}_n) \right)$$

- α_n: Stepsizes. How to choose them?

Stepsizes in PGD

(PGD) $$\boldsymbol{\theta}_{n+1} \leftarrow \Pi_{\boldsymbol{\Theta}}\left(\boldsymbol{\theta}_n - \alpha_n \nabla f(\boldsymbol{\theta}_n)\right)$$

- α_n: Stepsizes. How to choose them?
  - "*Small enough stepsize*": $\alpha_n \leq 1/L$, where
    - L = **Lipschitz constant for ∇f over Θ**
      $\approx$ **Largest absolute eigenvalue of $\nabla^2 f$ over Θ**

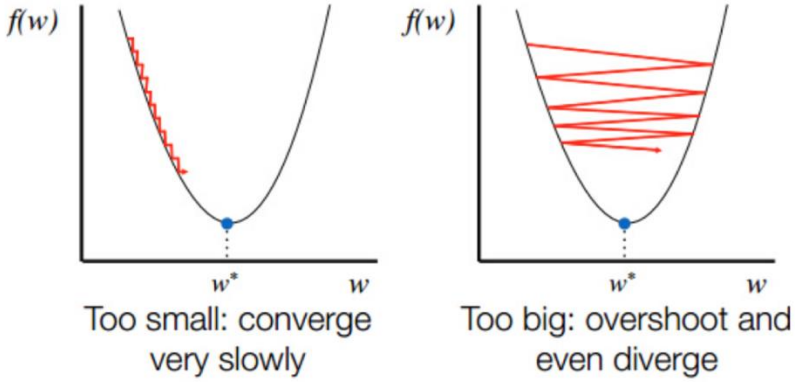
(PGD) $$\boldsymbol{\theta}_{n+1} \leftarrow \Pi_{\boldsymbol{\Theta}}\left(\boldsymbol{\theta}_n - \alpha_n \nabla f(\boldsymbol{\theta}_n)\right)$$

- α_n: Stepsizes. How to choose them?
  - "*Small enough stepsize*": $\alpha_n \leq 1/L$, where
    - $\boldsymbol{L}$ **= Lipschitz constant for ∇f over Θ**
      $\approx$ **Largest absolute eigenvalue of $\nabla^2 f$ over Θ**
  - But $\alpha_n = 1/L$ is TOO SMALL!
    - In practice use "line search" to find larger α_n that works
    - L might be unknown and hard to estimate

(PGD) $$\boldsymbol{\theta}_{n+1} \leftarrow \Pi_{\Theta}\left(\boldsymbol{\theta}_n - \alpha_n \nabla f(\boldsymbol{\theta}_n)\right)$$

- α_n: Stepsizes. How to choose them?
  - "*Small enough stepsize*": $\alpha_n \leq 1/L$, where
    - L = **Lipschitz constant for ∇f over Θ**
      $\approx$ **Largest absolute eigenvalue of $\nabla^2 f$ over Θ**
  - But $\alpha_n = 1/L$ is TOO SMALL!
    - In practice use "line search" to find larger α_n that works
    - L might be unknown and hard to estimate

In practice, performance of P(S)GD depends very sensitively on α_ns



Block PGD = BCD

Two-Block structure : $\boldsymbol{\theta} = (A, B)$, $\boldsymbol{\Theta} = \Theta_A \times \Theta_B$

(Block PGD)
(or BCD)

$$A_{n+1} \leftarrow \Pi_{\Theta_A}\left(A_n - \alpha_n \nabla_A f(A_n, B_n)\right)$$
$$B_{n+1} \leftarrow \Pi_{\Theta_B}\left(B_n - \beta_n \nabla_B f(A_{n+1}, B_n)\right)$$

Block PGD = BCD

Two-Block structure : $\boldsymbol{\theta} = (A, B)$, $\boldsymbol{\Theta} = \Theta_A \times \Theta_B$

(Block PGD) (or BCD)

$$A_{n+1} \leftarrow \Pi_{\Theta_A}\left(A_n - \alpha_n \nabla_A f(A_n, B_n)\right)$$
$$B_{n+1} \leftarrow \Pi_{\Theta_B}\left(B_n - \beta_n \nabla_B f(A_{n+1}, B_n)\right)$$

- Known to be much more robust against stepsize choices then PGD
  - e.g., low-rank matrix factorization, dictionary learning, tensor factorization, kernel learning

Block PGD = BCD

Two-Block structure : $\boldsymbol{\theta} = (A, B)$, $\boldsymbol{\Theta} = \Theta_A \times \Theta_B$

(Block PGD) (or BCD)

$$A_{n+1} \leftarrow \Pi_{\Theta_A}\left(A_n - \alpha_n \nabla_A f(A_n, B_n)\right)$$
$$B_{n+1} \leftarrow \Pi_{\Theta_B}\left(B_n - \beta_n \nabla_B f(A_{n+1}, B_n)\right)$$

- Known to be much more robust against stepsize choices then PGD
  - e.g., low-rank matrix factorization, dictionary learning, tensor factorization, kernel learning
- [Large $L \Leftrightarrow \nabla f$ changes wildly]

  → **Sensitive dependence on stepsize**

Block PGD = BCD

Two-Block structure : $\boldsymbol{\theta} = (A, B)$, $\boldsymbol{\Theta} = \Theta_A \times \Theta_B$

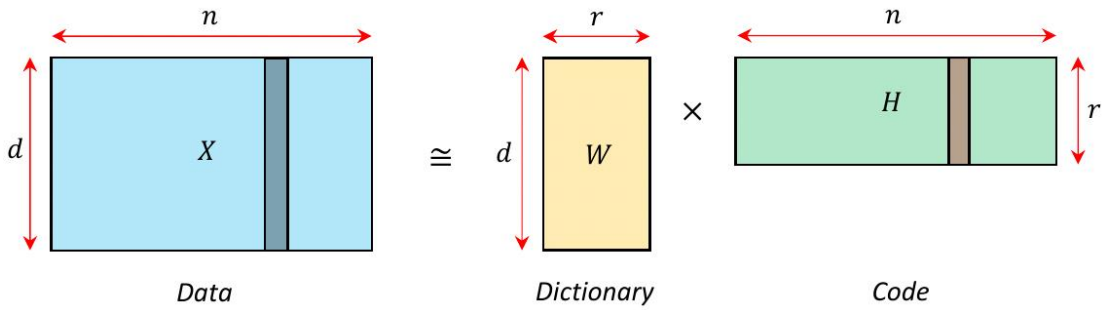
(Block PGD) (or BCD)

$$A_{n+1} \leftarrow \Pi_{\Theta_A}\left(A_n - \alpha_n \nabla_A f(A_n, B_n)\right)$$
$$B_{n+1} \leftarrow \Pi_{\Theta_B}\left(B_n - \beta_n \nabla_B f(A_{n+1}, B_n)\right)$$

- Known to be much more robust against stepsize choices then PGD
  - e.g., low-rank matrix factorization, dictionary learning, tensor factorization, kernel learning
- [Large $L \Leftrightarrow \nabla f$ changes wildly]

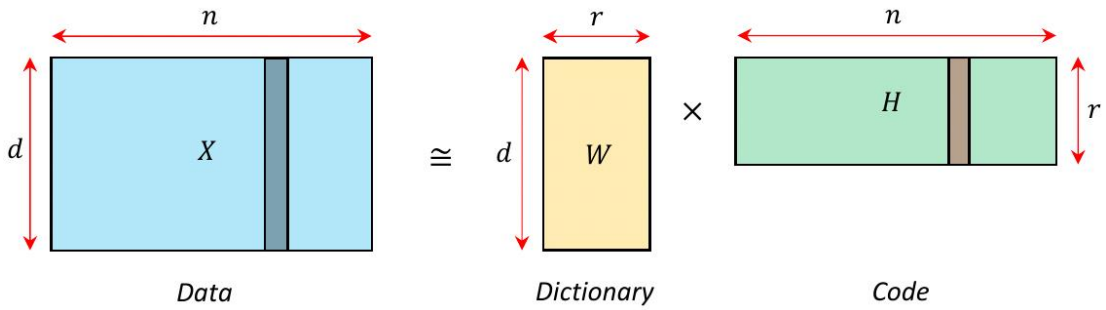
  → **Sensitive dependence on stepsize**
- **Exploiting block structure → The "effective L" is reduced**

Motivating example : **Nonnegative Matrix Factorization (Lee & Seung, Nature '99)**



$$\begin{cases} \text{minimize} & f(W,H) = \|X - WH\|_F^2 & \text{(Reconstruction error)} \\ \text{subject to} & W \in \mathbb{R}^{p\times r}_{\geq 0},\ H \in \mathbb{R}^{r\times n}_{\geq 0} & (\textit{Constraints}) \end{cases}$$

Motivating example : **Nonnegative Matrix Factorization (Lee & Seung, Nature '99)**



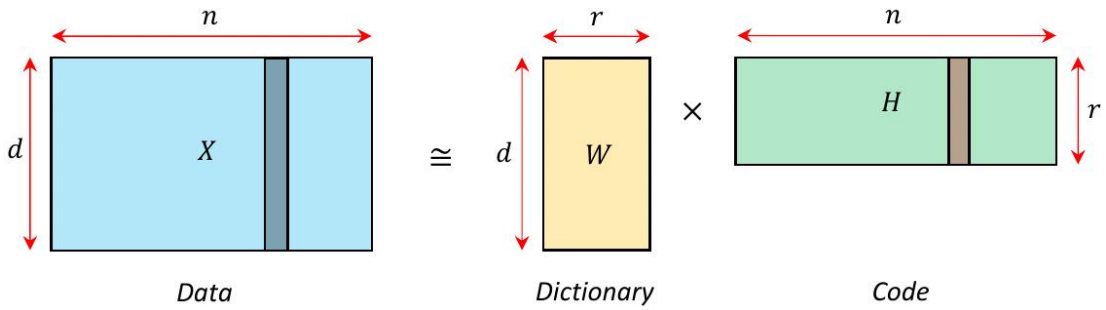
$$\begin{cases} \text{minimize} & f(W,H) = \|X - WH\|_F^2 & \text{(Reconstruction error)} \\ \text{subject to} & W \in \mathbb{R}^{p\times r}_{\geq 0},\ H \in \mathbb{R}^{r\times n}_{\geq 0} & \text{(Constraints)} \end{cases}$$

PGD $\begin{cases} (W_{n+1},H_{n+1}) \leftarrow (W_n,H_n) - \alpha_n(\nabla_W f(W_n,H_n), \nabla_H f(W_n,H_n)) \\ (W_{n+1},H_{n+1}) \leftarrow \max\big(0,(W_{n+1},H_{n+1})\big) \end{cases}$

(Almost no one uses this ☺)

- $f(W,H)$ = Bi-convex
- $\nabla_W f(W,H) = (WH - X)H^T$
- $\nabla_H f(W,H) = W^T(WH - X)$

Motivating example : **Nonnegative Matrix Factorization (Lee & Seung, Nature '99)**



$$\begin{cases} \text{minimize} & f(W,H) = \|X - WH\|_F^2 & \text{(Reconstruction error)} \\ \text{subject to} & W \in \mathbb{R}^{p\times r}_{\geq 0},\ H \in \mathbb{R}^{r\times n}_{\geq 0} & \text{(Constraints)} \end{cases}$$

PGD $\begin{cases} (W_{n+1},H_{n+1}) \leftarrow (W_n,H_n) - \alpha_n(\nabla_W f(W_n,H_n), \nabla_H f(W_n,H_n)) \\ (W_{n+1},H_{n+1}) \leftarrow \max\big(0,(W_{n+1},H_{n+1})\big) \end{cases}$

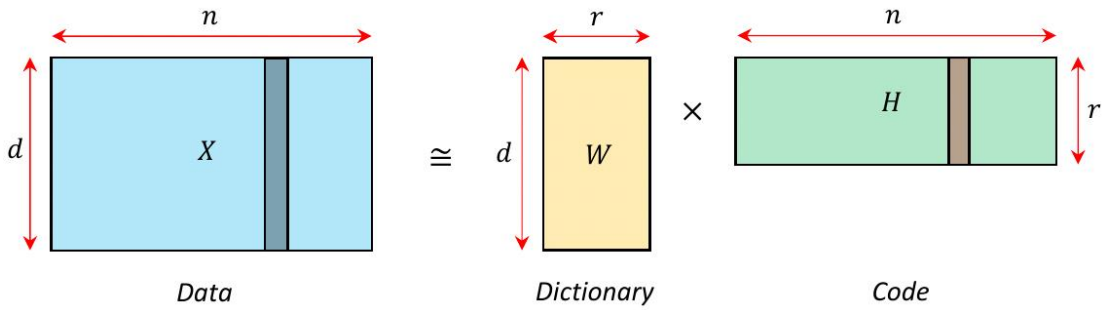
(Almost no one uses this ☺)

- $f(W,H)$ = Bi-convex
- $\nabla_W f(W,H) = (WH - X)H^T$
- $\nabla_H f(W,H) = W^T(WH - X)$

- $\nabla^2 f = \begin{matrix} \\ \text{vec}(\mathbf{W}) \\ \text{vec}(\mathbf{H}) \end{matrix} \begin{matrix} \begin{matrix} \text{vec}(\mathbf{W})^T & \text{vec}(\mathbf{H})^T \end{matrix} \\ \begin{bmatrix} \mathbf{H}\mathbf{H}^T \otimes \mathbf{I}_p & A_{12} \\ A_{12}^T & \mathbf{I}_n \otimes \mathbf{W}^T\mathbf{W} \end{bmatrix} \end{matrix}$

$$A_{12} = [(\mathbf{H} \otimes \mathbf{W}) + \mathbf{I}_r \otimes (\mathbf{W}\mathbf{H} - \mathbf{X})]\,\mathbf{C}^{(r,n)}$$

Motivating example : **Nonnegative Matrix Factorization (Lee & Seung, Nature '99)**



$$\begin{cases} \text{minimize} & f(W,H) = \|X - WH\|_F^2 & \text{(Reconstruction error)} \\ \text{subject to} & W \in \mathbb{R}^{p\times r}_{\geq 0},\ H \in \mathbb{R}^{r\times n}_{\geq 0} & \text{(Constraints)} \end{cases}$$

PGD $\begin{cases} (W_{n+1},H_{n+1}) \leftarrow (W_n,H_n) - \alpha_n(\nabla_W f(W_n,H_n), \nabla_H f(W_n,H_n)) \\ (W_{n+1},H_{n+1}) \leftarrow \max\big(0,(W_{n+1},H_{n+1})\big) \end{cases}$

(Almost no one uses this ☺)

- $f(W,H)$ = Bi-convex
- $\nabla_W f(W,H) = (WH - X)H^T$
- $\nabla_H f(W,H) = W^T(WH - X)$

- $\nabla^2 f = \begin{matrix} \\ \text{vec}(\mathbf{W}) \\ \text{vec}(\mathbf{H}) \end{matrix} \begin{matrix} \begin{matrix} \text{vec}(\mathbf{W})^T & \text{vec}(\mathbf{H})^T \end{matrix} \\ \begin{bmatrix} \mathbf{H}\mathbf{H}^T \otimes \mathbf{I}_p & A_{12} \\ A_{12}^T & \mathbf{I}_n \otimes \mathbf{W}^T\mathbf{W} \end{bmatrix} \end{matrix}$

$$A_{12} = [(\mathbf{H} \otimes \mathbf{W}) + \mathbf{I}_r \otimes (\mathbf{WH} - \mathbf{X})]\,\mathbf{C}^{(r,n)}$$

L = Max eval over all (W,H)

= *Unbounded!*

Case study: **Supervised Matrix Factorization (Lee, Lyu, Yao ICML '24)**

Adaptive BCD:

$$\mathbf{W} \leftarrow \Pi\left(\mathbf{W} - \frac{1}{\lambda_{\max}(\mathbf{H}\mathbf{H}^T) + \varepsilon}\,(\mathbf{W}\mathbf{H} - \mathbf{X})\mathbf{H}^T\right),$$

$$\mathbf{H} \leftarrow \Pi'\left(\mathbf{H} - \frac{1}{\lambda_{\max}(\mathbf{W}\mathbf{W}^T) + \varepsilon}\,\mathbf{W}^T(\mathbf{W}\mathbf{H} - \mathbf{X})\right)$$

Adaptive BCD:

$$\mathbf{W} \leftarrow \Pi\left(\mathbf{W} - \frac{1}{\lambda_{\max}(\mathbf{H}\mathbf{H}^T) + \varepsilon}(\mathbf{W}\mathbf{H} - \mathbf{X})\mathbf{H}^T\right),$$
$$\mathbf{H} \leftarrow \Pi'\left(\mathbf{H} - \frac{1}{\lambda_{\max}(\mathbf{W}\mathbf{W}^T) + \varepsilon}\mathbf{W}^T(\mathbf{W}\mathbf{H} - \mathbf{X})\right)$$

- $$\nabla^2 f = \begin{matrix} \\ \mathrm{vec}(\mathbf{W}) \\ \mathrm{vec}(\mathbf{H}) \end{matrix} \begin{matrix} \mathrm{vec}(\mathbf{W})^T & \mathrm{vec}(\mathbf{H})^T \end{matrix} \begin{bmatrix} \mathbf{H}\mathbf{H}^T \otimes \mathbf{I}_p & A_{12} \\ A_{12}^T & \mathbf{I}_n \otimes \mathbf{W}^T\mathbf{W} \end{bmatrix}$$

$\nabla^2_W f(\cdot, \mathbf{H})$ $\quad$ $\nabla^2_H f(\mathbf{W}, \cdot)$

Adaptive BCD:

$$\mathbf{W} \leftarrow \Pi\left(\mathbf{W} - \frac{1}{\lambda_{\max}(\mathbf{H}\mathbf{H}^T)+\varepsilon}(\mathbf{W}\mathbf{H}-\mathbf{X})\mathbf{H}^T\right),$$
$$\mathbf{H} \leftarrow \Pi'\left(\mathbf{H} - \frac{1}{\lambda_{\max}(\mathbf{W}\mathbf{W}^T)+\varepsilon}\mathbf{W}^T(\mathbf{W}\mathbf{H}-\mathbf{X})\right)$$

- $$\nabla^2 f = \begin{matrix}\\ \mathrm{vec}(\mathbf{W}) \\ \mathrm{vec}(\mathbf{H})\end{matrix}\begin{matrix}\mathrm{vec}(\mathbf{W})^T & \mathrm{vec}(\mathbf{H})^T \\ \left[\mathbf{H}\mathbf{H}^T \otimes \mathbf{I}_p \right. & A_{12} \\ A_{12}^T & \left. \mathbf{I}_n \otimes \mathbf{W}^T\mathbf{W}\right]\end{matrix}$$

$\nabla^2_W f(\cdot, \mathbf{H})$ $\quad$ $\nabla^2_H f(\mathbf{W}, \cdot)$

Largest Eval of *diagonal blocks* of the Hessian

$\ll$ **Largest Eval of the entire Hessian**

Supervised Matrix Factorization

Algorithm 1 BCD for SMF-$\mathbf{W}$

Input: $\mathbf{X} \in \mathbb{R}^{p\times n}$ (Data); $\mathbf{X}_{\text{aux}} \in \mathbb{R}^{q\times n}$ (Auxiliary covariate); $\mathbf{Y}_{\text{label}} \in \{0,\dots,\kappa\}^{1\times n}$ (Label);
Constraints: Convex subsets $\mathcal{C}_1 \subseteq \mathbb{R}^{p\times r}$, $\mathcal{C}_2 \subseteq \mathbb{R}^{r\times n}$, $\mathcal{C}_3 \subseteq \mathbb{R}^{r\times \kappa}$, $\mathcal{C}_4 \subseteq \mathbb{R}^{q\times \kappa}$
Parameters: $\xi \geq 0$ (Tuning parameter); $T \in \mathbb{N}$ (number of iterations); $(\eta_{k;i})_{k\geq 1, 1\leq i\leq 4}$ (step-sizes)
Initialize $\mathbf{W} \in \mathcal{C}_1, \mathbf{H} \in \mathcal{C}_2, \boldsymbol{\beta} \in \mathcal{C}_3, \boldsymbol{\Gamma} \in \mathcal{C}_4$
For $k = 1, 2, \dots, T$ **do:** ($\triangleright$ *For* α^+ *see 4.3.*)
  (Update $\mathbf{W}$)
  Update activation $\mathbf{a}_1, \dots, \mathbf{a}_n$ and $\mathbf{K}$
  $\nabla_{\mathbf{W}} f(\mathbf{Z}) \leftarrow \mathbf{X}\mathbf{K}^T\boldsymbol{\beta}^T + 2\xi(\mathbf{W}\mathbf{H} - \mathbf{X})\mathbf{H}^T$
  Choose $\eta_{k,1}^{-1} > L_1 := \alpha^+\|\boldsymbol{\beta}\|_2^2 \cdot \|\mathbf{X}\|_2^2 + 2\xi\|\mathbf{H}\|_2^2$
  $\mathbf{W} \leftarrow \Pi_{\mathcal{C}_1}(\mathbf{W} - \eta_{k;1}\nabla_{\mathbf{W}} f(\mathbf{Z}))$
  (Update $\mathbf{H}$)
  $\nabla_{\mathbf{H}} f(\mathbf{Z}) \leftarrow 2\xi\mathbf{W}^T(\mathbf{W}\mathbf{H} - \mathbf{X})$
  Choose $\eta_{k,2}^{-1} > L_2 := 2\xi\|\mathbf{W}\|_2^2$
  $\mathbf{H} \leftarrow \Pi_{\mathcal{C}_2}(\mathbf{H} - \eta_{k;2}\nabla_{\mathbf{H}} f(\mathbf{Z}))$
  (Update $\boldsymbol{\beta}$)
  Update activation $\mathbf{a}_1, \dots, \mathbf{a}_n$ and $\mathbf{K}$
  $\nabla_{\boldsymbol{\beta}} f(\mathbf{Z}) \leftarrow \mathbf{W}^T\mathbf{X}\mathbf{K}^T$
  Choose $\eta_{k,3}^{-1} > L_3 := \alpha^+\|\mathbf{W}\|_2^2 \cdot \|\mathbf{X}\|_2^2$
  $\boldsymbol{\beta} \leftarrow \Pi_{\mathcal{C}_3}(\boldsymbol{\beta} - \eta_{k;3}\nabla_{\boldsymbol{\beta}} f(\mathbf{Z}))$
  (Update $\boldsymbol{\Gamma}$)
  Update activation $\mathbf{a}_1, \dots, \mathbf{a}_n$ and $\mathbf{K}$
  $\nabla_{\boldsymbol{\Gamma}} f(\mathbf{Z}) \leftarrow \mathbf{X}_{\text{aux}}\mathbf{K}^T$
  Choose $\eta_{k,4}^{-1} > L_4 := \alpha^+\|\mathbf{X}_{\text{aux}}\|_2^2$
  $\boldsymbol{\Gamma} \leftarrow \Pi_{\mathcal{C}_4}(\boldsymbol{\Gamma} - \eta_{k;4}\nabla_{\boldsymbol{\Gamma}} f(\mathbf{Z}))$
End for
Output: $\mathbf{Z} = (\mathbf{W}, \mathbf{H}, \boldsymbol{\beta}, \boldsymbol{\Gamma})$

- $\nabla^2 f =$

	$\text{vec}(\mathbf{W})^T$	$\text{vec}(\mathbf{H})^T$	$\text{vec}(\boldsymbol{\beta})^T$	$\text{vec}(\boldsymbol{\Gamma})^T$
$\text{vec}(\mathbf{W})$	A_{11}	A_{12}	A_{13}	O
$\text{vec}(\mathbf{H})$	A_{21}	A_{22}	O	O
$\text{vec}(\boldsymbol{\beta})$	A_{31}	O	A_{33}	A_{34}
$\text{vec}(\boldsymbol{\Gamma})$	O	O	A_{43}	A_{44}

Supervised Matrix Factorization

Algorithm 1 BCD for SMF-W

Input: $\mathbf{X} \in \mathbb{R}^{p\times n}$ (Data); $\mathbf{X}_{\text{aux}} \in \mathbb{R}^{q\times n}$ (Auxiliary covariate); $\mathbf{Y}_{\text{label}} \in \{0,\dots,\kappa\}^{1\times n}$ (Label);
Constraints: Convex subsets $\mathcal{C}_1 \subseteq \mathbb{R}^{p\times r}$, $\mathcal{C}_2 \subseteq \mathbb{R}^{r\times n}$, $\mathcal{C}_3 \subseteq \mathbb{R}^{r\times \kappa}$, $\mathcal{C}_4 \subseteq \mathbb{R}^{q\times \kappa}$
Parameters: $\xi \geq 0$ (Tuning parameter); $T \in \mathbb{N}$ (number of iterations); $(\eta_{k;i})_{k\geq 1, 1\leq i\leq 4}$ (step-sizes)
Initialize $\mathbf{W} \in \mathcal{C}_1, \mathbf{H} \in \mathcal{C}_2, \boldsymbol{\beta} \in \mathcal{C}_3, \boldsymbol{\Gamma} \in \mathcal{C}_4$
For $k = 1, 2, \dots, T$ **do:** (▷ *For* α^+ *see 4.3.*)
  (Update $\mathbf{W}$)
  Update activation $\mathbf{a}_1, \dots, \mathbf{a}_n$ and $\mathbf{K}$
  $\nabla_{\mathbf{W}} f(\mathbf{Z}) \leftarrow \mathbf{X}\mathbf{K}^T\boldsymbol{\beta}^T + 2\xi(\mathbf{W}\mathbf{H} - \mathbf{X})\mathbf{H}^T$
  Choose $\eta_{k,1}^{-1} > L_1 := \alpha^+\|\boldsymbol{\beta}\|_2^2 \cdot \|\mathbf{X}\|_2^2 + 2\xi\|\mathbf{H}\|_2^2$
  $\mathbf{W} \leftarrow \Pi_{\mathcal{C}_1}(\mathbf{W} - \eta_{k;1}\nabla_{\mathbf{W}} f(\mathbf{Z}))$
  (Update $\mathbf{H}$)
  $\nabla_{\mathbf{H}} f(\mathbf{Z}) \leftarrow 2\xi\mathbf{W}^T(\mathbf{W}\mathbf{H} - \mathbf{X})$
  Choose $\eta_{k,2}^{-1} > L_2 := 2\xi\|\mathbf{W}\|_2^2$
  $\mathbf{H} \leftarrow \Pi_{\mathcal{C}_2}(\mathbf{H} - \eta_{k;2}\nabla_{\mathbf{H}} f(\mathbf{Z}))$
  (Update $\boldsymbol{\beta}$)
  Update activation $\mathbf{a}_1, \dots, \mathbf{a}_n$ and $\mathbf{K}$
  $\nabla_{\boldsymbol{\beta}} f(\mathbf{Z}) \leftarrow \mathbf{W}^T\mathbf{X}\mathbf{K}^T$
  Choose $\eta_{k,3}^{-1} > L_3 := \alpha^+\|\mathbf{W}\|_2^2 \cdot \|\mathbf{X}\|_2^2$
  $\boldsymbol{\beta} \leftarrow \Pi_{\mathcal{C}_3}(\boldsymbol{\beta} - \eta_{k;3}\nabla_{\boldsymbol{\beta}} f(\mathbf{Z}))$
  (Update $\boldsymbol{\Gamma}$)
  Update activation $\mathbf{a}_1, \dots, \mathbf{a}_n$ and $\mathbf{K}$
  $\nabla_{\boldsymbol{\Gamma}} f(\mathbf{Z}) \leftarrow \mathbf{X}_{\text{aux}}\mathbf{K}^T$
  Choose $\eta_{k,4}^{-1} > L_4 := \alpha^+\|\mathbf{X}_{\text{aux}}\|_2^2$
  $\boldsymbol{\Gamma} \leftarrow \Pi_{\mathcal{C}_4}(\boldsymbol{\Gamma} - \eta_{k;4}\nabla_{\boldsymbol{\Gamma}} f(\mathbf{Z}))$
End for
Output: $\mathbf{Z} = (\mathbf{W}, \mathbf{H}, \boldsymbol{\beta}, \boldsymbol{\Gamma})$

- $\nabla^2 f =$

	$\text{vec}(\mathbf{W})^T$	$\text{vec}(\mathbf{H})^T$	$\text{vec}(\boldsymbol{\beta})^T$	$\text{vec}(\boldsymbol{\Gamma})^T$
$\text{vec}(\mathbf{W})$	A_{11}	A_{12}	A_{13}	O
$\text{vec}(\mathbf{H})$	A_{21}	A_{22}	O	O
$\text{vec}(\boldsymbol{\beta})$	A_{31}	O	A_{33}	A_{34}
$\text{vec}(\boldsymbol{\Gamma})$	O	O	A_{43}	A_{44}

Theorem. (Lee., L., Yao ICML '24)

Adaptive BDC finds an ϵ-stationary pt. of SMF-W within $O(\epsilon^{-2})$ iterations

Depends only on the block-wise condition number

Supervised Matrix Factorization

Algorithm 1 BCD for SMF-W

Input: $\mathbf{X} \in \mathbb{R}^{p\times n}$ (Data); $\mathbf{X}_{\text{aux}} \in \mathbb{R}^{q\times n}$ (Auxiliary covariate); $\mathbf{Y}_{\text{label}} \in \{0,\dots,\kappa\}^{1\times n}$ (Label);
Constraints: Convex subsets $\mathcal{C}_1 \subseteq \mathbb{R}^{p\times r}$, $\mathcal{C}_2 \subseteq \mathbb{R}^{r\times n}$, $\mathcal{C}_3 \subseteq \mathbb{R}^{r\times \kappa}$, $\mathcal{C}_4 \subseteq \mathbb{R}^{q\times \kappa}$
Parameters: $\xi \geq 0$ (Tuning parameter); $T \in \mathbb{N}$ (number of iterations); $(\eta_{k;i})_{k\geq 1, 1\leq i\leq 4}$ (step-sizes)
Initialize $\mathbf{W} \in \mathcal{C}_1, \mathbf{H} \in \mathcal{C}_2, \boldsymbol{\beta} \in \mathcal{C}_3, \boldsymbol{\Gamma} \in \mathcal{C}_4$
For $k = 1, 2, \dots, T$ **do:** (▷ *For* α^+ *see 4.3.*)
  (Update $\mathbf{W}$)
  Update activation $\mathbf{a}_1, \dots, \mathbf{a}_n$ and $\mathbf{K}$
  $\nabla_{\mathbf{W}} f(\mathbf{Z}) \leftarrow \mathbf{X}\mathbf{K}^T\boldsymbol{\beta}^T + 2\xi(\mathbf{W}\mathbf{H} - \mathbf{X})\mathbf{H}^T$
  Choose $\eta_{k,1}^{-1} > L_1 := \alpha^+\|\boldsymbol{\beta}\|_2^2 \cdot \|\mathbf{X}\|_2^2 + 2\xi\|\mathbf{H}\|_2^2$
  $\mathbf{W} \leftarrow \Pi_{\mathcal{C}_1}(\mathbf{W} - \eta_{k;1}\nabla_{\mathbf{W}} f(\mathbf{Z}))$
  (Update $\mathbf{H}$)
  $\nabla_{\mathbf{H}} f(\mathbf{Z}) \leftarrow 2\xi\mathbf{W}^T(\mathbf{W}\mathbf{H} - \mathbf{X})$
  Choose $\eta_{k,2}^{-1} > L_2 := 2\xi\|\mathbf{W}\|_2^2$
  $\mathbf{H} \leftarrow \Pi_{\mathcal{C}_2}(\mathbf{H} - \eta_{k;2}\nabla_{\mathbf{H}} f(\mathbf{Z}))$
  (Update $\boldsymbol{\beta}$)
  Update activation $\mathbf{a}_1, \dots, \mathbf{a}_n$ and $\mathbf{K}$
  $\nabla_{\boldsymbol{\beta}} f(\mathbf{Z}) \leftarrow \mathbf{W}^T\mathbf{X}\mathbf{K}^T$
  Choose $\eta_{k,3}^{-1} > L_3 := \alpha^+\|\mathbf{W}\|_2^2 \cdot \|\mathbf{X}\|_2^2$
  $\boldsymbol{\beta} \leftarrow \Pi_{\mathcal{C}_3}(\boldsymbol{\beta} - \eta_{k;3}\nabla_{\boldsymbol{\beta}} f(\mathbf{Z}))$
  (Update $\boldsymbol{\Gamma}$)
  Update activation $\mathbf{a}_1, \dots, \mathbf{a}_n$ and $\mathbf{K}$
  $\nabla_{\boldsymbol{\Gamma}} f(\mathbf{Z}) \leftarrow \mathbf{X}_{\text{aux}}\mathbf{K}^T$
  Choose $\eta_{k,4}^{-1} > L_4 := \alpha^+\|\mathbf{X}_{\text{aux}}\|_2^2$
  $\boldsymbol{\Gamma} \leftarrow \Pi_{\mathcal{C}_4}(\boldsymbol{\Gamma} - \eta_{k;4}\nabla_{\boldsymbol{\Gamma}} f(\mathbf{Z}))$
End for
Output: $\mathbf{Z} = (\mathbf{W}, \mathbf{H}, \boldsymbol{\beta}, \boldsymbol{\Gamma})$

- $\nabla^2 f =$

	$\text{vec}(\mathbf{W})^T$	$\text{vec}(\mathbf{H})^T$	$\text{vec}(\boldsymbol{\beta})^T$	$\text{vec}(\boldsymbol{\Gamma})^T$
$\text{vec}(\mathbf{W})$	A_{11}	A_{12}	A_{13}	O
$\text{vec}(\mathbf{H})$	A_{21}	A_{22}	O	O
$\text{vec}(\boldsymbol{\beta})$	A_{31}	O	A_{33}	A_{34}
$\text{vec}(\boldsymbol{\Gamma})$	O	O	A_{43}	A_{44}

Theorem. (Lee., L., Yao ICML '24)

Adaptive BDC finds an ϵ-stationary pt. of SMF-W within $O(\epsilon^{-2})$ iterations

Depends only on the block-wise condition number

Hessian analysis + Block Majorization-Minimization guarantees (w/ Yuchen Li (→Job market))

Supervised Matrix Factorization

Algorithm 1 BCD for SMF-W

Input: $\mathbf{X} \in \mathbb{R}^{p\times n}$ (Data); $\mathbf{X}_{\text{aux}} \in \mathbb{R}^{q\times n}$ (Auxiliary covariate); $\mathbf{Y}_{\text{label}} \in \{0,\dots,\kappa\}^{1\times n}$ (Label);
Constraints: Convex subsets $\mathcal{C}_1 \subseteq \mathbb{R}^{p\times r}$, $\mathcal{C}_2 \subseteq \mathbb{R}^{r\times n}$, $\mathcal{C}_3 \subseteq \mathbb{R}^{r\times \kappa}$, $\mathcal{C}_4 \subseteq \mathbb{R}^{q\times \kappa}$
Parameters: $\xi \geq 0$ (Tuning parameter); $T \in \mathbb{N}$ (number of iterations); $(\eta_{k;i})_{k\geq 1, 1\leq i\leq 4}$ (step-sizes)
Initialize $\mathbf{W} \in \mathcal{C}_1, \mathbf{H} \in \mathcal{C}_2, \boldsymbol{\beta} \in \mathcal{C}_3, \boldsymbol{\Gamma} \in \mathcal{C}_4$
For $k = 1, 2, \dots, T$ **do:** (▷ *For* α^+ *see 4.3.*)
  (Update $\mathbf{W}$)
  Update activation $\mathbf{a}_1, \dots, \mathbf{a}_n$ and $\mathbf{K}$
  $\nabla_{\mathbf{W}} f(\mathbf{Z}) \leftarrow \mathbf{X}\mathbf{K}^T\boldsymbol{\beta}^T + 2\xi(\mathbf{W}\mathbf{H} - \mathbf{X})\mathbf{H}^T$
  Choose $\eta_{k,1}^{-1} > L_1 := \alpha^+\|\boldsymbol{\beta}\|_2^2 \cdot \|\mathbf{X}\|_2^2 + 2\xi\|\mathbf{H}\|_2^2$
  $\mathbf{W} \leftarrow \Pi_{\mathcal{C}_1}(\mathbf{W} - \eta_{k;1}\nabla_{\mathbf{W}} f(\mathbf{Z}))$
  (Update $\mathbf{H}$)
  $\nabla_{\mathbf{H}} f(\mathbf{Z}) \leftarrow 2\xi\mathbf{W}^T(\mathbf{W}\mathbf{H} - \mathbf{X})$
  Choose $\eta_{k,2}^{-1} > L_2 := 2\xi\|\mathbf{W}\|_2^2$
  $\mathbf{H} \leftarrow \Pi_{\mathcal{C}_2}(\mathbf{H} - \eta_{k;2}\nabla_{\mathbf{H}} f(\mathbf{Z}))$
  (Update $\boldsymbol{\beta}$)
  Update activation $\mathbf{a}_1, \dots, \mathbf{a}_n$ and $\mathbf{K}$
  $\nabla_{\boldsymbol{\beta}} f(\mathbf{Z}) \leftarrow \mathbf{W}^T\mathbf{X}\mathbf{K}^T$
  Choose $\eta_{k,3}^{-1} > L_3 := \alpha^+\|\mathbf{W}\|_2^2 \cdot \|\mathbf{X}\|_2^2$
  $\boldsymbol{\beta} \leftarrow \Pi_{\mathcal{C}_3}(\boldsymbol{\beta} - \eta_{k;3}\nabla_{\boldsymbol{\beta}} f(\mathbf{Z}))$
  (Update $\boldsymbol{\Gamma}$)
  Update activation $\mathbf{a}_1, \dots, \mathbf{a}_n$ and $\mathbf{K}$
  $\nabla_{\boldsymbol{\Gamma}} f(\mathbf{Z}) \leftarrow \mathbf{X}_{\text{aux}}\mathbf{K}^T$
  Choose $\eta_{k,4}^{-1} > L_4 := \alpha^+\|\mathbf{X}_{\text{aux}}\|_2^2$
  $\boldsymbol{\Gamma} \leftarrow \Pi_{\mathcal{C}_4}(\boldsymbol{\Gamma} - \eta_{k;4}\nabla_{\boldsymbol{\Gamma}} f(\mathbf{Z}))$
End for
Output: $\mathbf{Z} = (\mathbf{W}, \mathbf{H}, \boldsymbol{\beta}, \boldsymbol{\Gamma})$

- $\nabla^2 f =$

	$\text{vec}(\mathbf{W})^T$	$\text{vec}(\mathbf{H})^T$	$\text{vec}(\boldsymbol{\beta})^T$	$\text{vec}(\boldsymbol{\Gamma})^T$
$\text{vec}(\mathbf{W})$	A_{11}	A_{12}	A_{13}	O
$\text{vec}(\mathbf{H})$	A_{21}	A_{22}	O	O
$\text{vec}(\boldsymbol{\beta})$	A_{31}	O	A_{33}	A_{34}
$\text{vec}(\boldsymbol{\Gamma})$	O	O	A_{43}	A_{44}

Theorem. (Lee., L., Yao ICML '24)

Adaptive BDC finds an ϵ-stationary pt. of SMF-W within $O(\epsilon^{-2})$ iterations

Depends only on the block-wise condition number

Hessian analysis + Block Majorization-Minimization guarantees (w/ Yuchen Li)

Theorem. (Lee., L., Yao ICML '24)

With n i.i.d. samples from a generative model, ∃ a local max. of the likelihood function within $O(n^{-1/2})$ of the true parameter*

Needs bound on the evals of the *entire* Hessian

* Some true factors needs L2-reg.

Exponentially fast optimization algorithm?

Classification loss Dimension reduction loss

$$\min_{\mathbf{W},\mathbf{H},\boldsymbol{\beta}} f(\mathbf{W},\mathbf{H},\boldsymbol{\beta}) := \sum_{i=1}^{n} \ell(y_i, \boldsymbol{\beta}^T \mathbf{W}^T \mathbf{x}_i) + \xi \|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2$$

Negative log-likelihood under logistic model

$$\ell(y,a) = \log(1+\exp(a)) - \mathbf{1}_{\{y=1\}} a$$

Parameters
= Three factor matrices $\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{\beta}$

- Highly non-convex, possibly constrained optimization problem
- Adaptive BCD gives standard $O(\epsilon^{-2})$ polynomial convergence rate
- No hope for any **exponentially fast** algorithm? Yes there is!

Exponentially fast optimization algorithm?

Classification loss Dimension reduction loss

$$\min_{\mathbf{W},\mathbf{H},\boldsymbol{\beta}} f(\mathbf{W},\mathbf{H},\boldsymbol{\beta}) := \sum_{i=1}^{n} \ell(y_i, \boldsymbol{\beta}^T \mathbf{W}^T \mathbf{x}_i) + \xi \|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2$$

Negative log-likelihood under logistic model

$$\ell(y, a) = \log(1 + \exp(a)) - \mathbf{1}_{\{y=1\}} a$$

Parameters
= Three factor matrices $\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{\beta}$

- Highly non-convex, possibly constrained optimization problem
- Adaptive BCD gives standard $O(\epsilon^{-2})$ polynomial convergence rate
- No hope for any **exponentially fast** algorithm? Yes there is!

Reformulate it as a low-rank matrix estimation problem

$$\min_{\mathrm{rk}(Z) \le r} F\left(\begin{array}{|c|cc|} \hline \multirow{2}{*}{W} & \beta & H \\ \cline{2-3} & \multicolumn{2}{c|}{Z} \\ \hline \end{array} \right)$$

"lifted" objective: **strongly convex and smooth**

Exponentially fast optimization algorithm?

$$\min_{\mathrm{rk}(Z) \le r} F\left(\begin{array}{|c|c|c|} \hline \multicolumn{1}{|c|}{\multirow{2}{*}{W}} & \beta & H \\ \cline{2-3} \multicolumn{1}{|c|}{} & \multicolumn{2}{c|}{Z} \\ \hline \end{array} \right)$$

"lifted" objective: **strongly convex and smooth**

(Low-rk PGD) $\quad Z_{k+1} \leftarrow \mathrm{SVD}_r\left(Z_k - \eta \nabla \mathrm{F}(Z_k)\right)$

Exponentially fast optimization algorithm?

$$\min_{\mathrm{rk}(Z)\le r} F\left(\begin{array}{|c|c|c|}\hline W & \beta & H \\ \hline & & Z \\ \hline\end{array}\right)$$

"lifted" objective: **strongly convex and smooth**

(Low-rk PGD) $\quad Z_{k+1} \leftarrow \mathrm{SVD}_r\left(Z_k - \eta\, \nabla \mathrm{F}(Z_k)\right)$

NOT a Riemannian optimization alg.!

F is SC only in the Euclidean geometry!!

Exponentially fast optimization algorithm?

$$\min_{\mathrm{rk}(Z)\le r} F\left(\begin{array}{cc} W & \beta \\ & Z \end{array} H\right)$$

"lifted" objective: **strongly convex and smooth**

(Low-rk PGD) $\quad Z_{k+1} \leftarrow \mathrm{SVD}_r\left(Z_k - \eta\, \nabla \mathrm{F}(Z_k)\right)$

NOT a Riemannian optimization alg.!

F is SC only in the Euclidean geometry!!

Theorem. (Lee., L., Yao NeurIPS '23)

If F is well-conditioned (cn<3) and $\eta \ll 1$,
LPGD converges exponentially fast to the global minimizer

Exponentially fast optimization algorithm?

$$\min_{\mathrm{rk}(Z)\le r} F\left(\begin{bmatrix} W & \beta H \\ & Z \end{bmatrix}\right)$$

"lifted" objective: **strongly convex and smooth**

(Low-rk PGD) $\quad Z_{k+1} \leftarrow \mathrm{SVD}_r\left(Z_k - \eta\, \nabla \mathrm{F}(Z_k)\right)$

NOT a Riemannian optimization alg.!

F is SC only in the Euclidean geometry!!

Theorem. (Lee., L., Yao NeurIPS '23)

If F is well-conditioned (cn<3) and $\eta \ll 1$,
LPGD converges exponentially fast to the global minimizer

F is well-conditioned, $\mathrm{SVD}_r \approx Subspace\ Proj.$

→2-Lipschitz, so stronger then ½ contraction suffices

Exponentially fast optimization algorithm?

$$\min_{\mathrm{rk}(Z)\le r} F\left(\begin{bmatrix} W & \beta & H \\ & & Z \end{bmatrix}\right)$$

"lifted" objective: **strongly convex and smooth**

$$\sum_{i=1}^{n} \overbrace{\ell(y_i, \beta^T W^T x_i)}^{\text{Classification loss}} + \overbrace{\xi \|X - WH\|_F^2}^{\text{Dimension reduction loss}}$$

Better Condition #

(Low-rk PGD) $Z_{k+1} \leftarrow \mathrm{SVD}_r\left(Z_k - \eta \nabla F(Z_k)\right)$

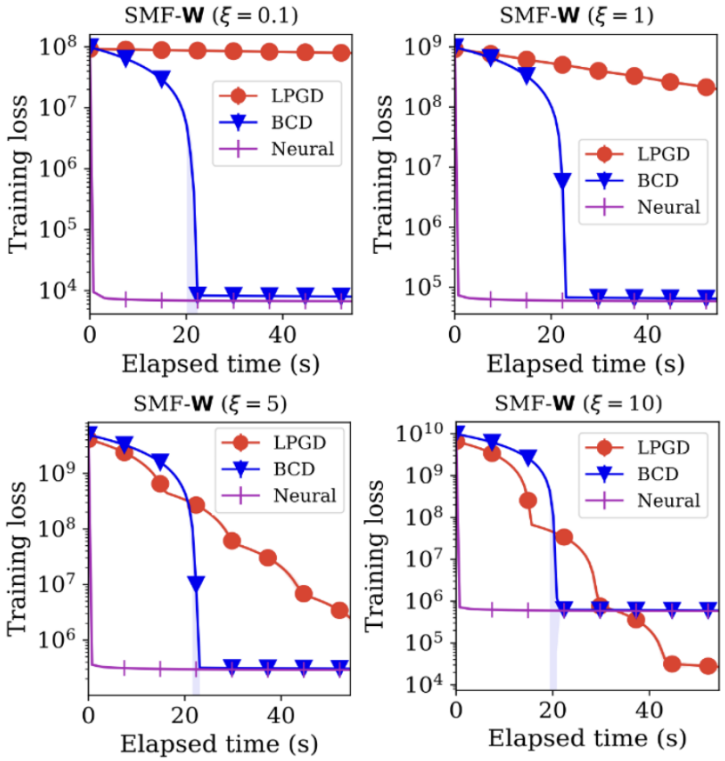
NOT a Riemannian optimization alg.!

F is SC only in the Euclidean geometry!!

Theorem. (Lee., L., Yao NeurIPS '23)

If F is well-conditioned (cn<3) and $\eta \ll 1$,
LPGD converges exponentially fast to the global minimizer

F is well-conditioned, $\mathrm{SVD}_r \approx \mathit{Subspace\ Proj}$.



Summary

SMF provides an interpretable low-dimensional, label-aware compression

- **Various applications where interpretability matters!**
  - Identifying:
    - cancer-related gene groups
    - survival-related gene groups (German Cancer Research Center)
    - driving patterns for auto insurance customers (AmFam)
    - climate patterns indicating future El Nino/La Nina (N. Chen)
- **Adaptive BCD**
  - Robust nonconvex constrained optimization
  - Depends only on the block-diagonal Hessian
  - Sublinear convergence
- **Low-rank PGD**
  - Lift non-convex problem to convex low-rank opt problem
  - Exponentially fast convergence with good condition number
- **Non-asymptotic Statistical Guarantees**

Thank you very much!

References

1. Joowon Lee, Hanbaek Lyu, and Weixin Yao, "*Exponentially Convergent Algorithms for Supervised Matrix Factorization*", NeurIPS 2023

2. Joowon Lee, Hanbaek Lyu, and Weixin Yao, "*Supervised Matrix Factorization: Local Landscape Analysis and Applications*" ICML 2024