Recent Advances in Nonnegative Matrix Factorization

Part II: Extensions of NMF

Cédric Févotte
CNRS, Toulouse, France

Vincent Y. F. Tan
National University of Singapore

ICASSP Tutorial
Singapore, May 2022

Outline

Outline

Nonnegative Rank Selection by Automatic Relevance Determination (ARD)

Tan and Févotte (2013)

Nonnegative Rank Selection by Automatic Relevance Determination (ARD)

Tan and Févotte (2013)

- Recall that in NMF, one is given a data matrix $\mathbf{V} \in \mathbb{R}_+^{F \times N}$ and tries to find a dictionary matrix $\mathbf{W} \in \mathbb{R}_+^{F \times K}$ and coefficient matrix $\mathbf{H} \in \mathbb{R}_+^{K \times N}$ such that

$$\mathbf{V} \approx \hat{\mathbf{V}} = \mathbf{WH}.$$

Nonnegative Rank Selection by Automatic Relevance Determination (ARD)

Tan and Févotte (2013)

- Recall that in NMF, one is given a data matrix $\mathbf{V} \in \mathbb{R}_+^{F \times N}$ and tries to find a dictionary matrix $\mathbf{W} \in \mathbb{R}_+^{F \times K}$ and coefficient matrix $\mathbf{H} \in \mathbb{R}_+^{K \times N}$ such that

$$\mathbf{V} \approx \hat{\mathbf{V}} = \mathbf{WH}.$$

- Usually solved using a constrained minimization problem

$$\min_{\mathbf{W},\mathbf{H} \geq \mathbf{0}} D(\mathbf{V} \mid \mathbf{WH}) = \sum_{f=1}^{F} \sum_{n=1}^{N} d\big([\mathbf{V}]_{fn} \mid [\mathbf{WH}]_{fn}\big).$$

Nonnegative Rank Selection by Automatic Relevance Determination (ARD)

Tan and Févotte (2013)

- Recall that in NMF, one is given a data matrix $\mathbf{V} \in \mathbb{R}_+^{F \times N}$ and tries to find a dictionary matrix $\mathbf{W} \in \mathbb{R}_+^{F \times K}$ and coefficient matrix $\mathbf{H} \in \mathbb{R}_+^{K \times N}$ such that

$$\mathbf{V} \approx \hat{\mathbf{V}} = \mathbf{WH}.$$

- Usually solved using a constrained minimization problem

$$\min_{\mathbf{W},\mathbf{H} \geq \mathbf{0}} D(\mathbf{V} \mid \mathbf{WH}) = \sum_{f=1}^{F} \sum_{n=1}^{N} d\big([\mathbf{V}]_{fn} \mid [\mathbf{WH}]_{fn}\big).$$

- How to find the common/latent dimension K?

Nonnegative Rank Selection by Automatic Relevance Determination (ARD)

Tan and Févotte (2013)

- Recall that in NMF, one is given a data matrix $\mathbf{V} \in \mathbb{R}_+^{F \times N}$ and tries to find a dictionary matrix $\mathbf{W} \in \mathbb{R}_+^{F \times K}$ and coefficient matrix $\mathbf{H} \in \mathbb{R}_+^{K \times N}$ such that

$$\mathbf{V} \approx \hat{\mathbf{V}} = \mathbf{WH}.$$

- Usually solved using a constrained minimization problem

$$\min_{\mathbf{W},\mathbf{H} \geq \mathbf{0}} D(\mathbf{V} \mid \mathbf{WH}) = \sum_{f=1}^{F} \sum_{n=1}^{N} d\big([\mathbf{V}]_{fn} \mid [\mathbf{WH}]_{fn}\big).$$

- How to find the common/latent dimension K?
- If K is too large $\implies$ Overfitting! K too small $\implies$ Poor fit to model!

Nonnegative Rank Selection by Automatic Relevance Determination (ARD)

Tan and Févotte (2013)

- Recall that in NMF, one is given a data matrix $\mathbf{V} \in \mathbb{R}_+^{F \times N}$ and tries to find a dictionary matrix $\mathbf{W} \in \mathbb{R}_+^{F \times K}$ and coefficient matrix $\mathbf{H} \in \mathbb{R}_+^{K \times N}$ such that

$$\mathbf{V} \approx \hat{\mathbf{V}} = \mathbf{WH}.$$

- Usually solved using a constrained minimization problem

$$\min_{\mathbf{W},\mathbf{H} \geq \mathbf{0}} D(\mathbf{V} \mid \mathbf{WH}) = \sum_{f=1}^{F} \sum_{n=1}^{N} d\big([\mathbf{V}]_{fn} \mid [\mathbf{WH}]_{fn}\big).$$

- How to find the common/latent dimension K?
- If K is too large $\implies$ Overfitting! K too small $\implies$ Poor fit to model!
- Solve this by automatic relevance determination (Bishop, 1999; Tipping, 2001)

Nonnegative Rank Selection by Automatic Relevance Determination (ARD)

Tan and Févotte (2013)

- Recall that in NMF, one is given a data matrix $\mathbf{V} \in \mathbb{R}_+^{F \times N}$ and tries to find a dictionary matrix $\mathbf{W} \in \mathbb{R}_+^{F \times K}$ and coefficient matrix $\mathbf{H} \in \mathbb{R}_+^{K \times N}$ such that

$$\mathbf{V} \approx \hat{\mathbf{V}} = \mathbf{WH}.$$

- Usually solved using a constrained minimization problem

$$\min_{\mathbf{W},\mathbf{H} \geq \mathbf{0}} D(\mathbf{V} \mid \mathbf{WH}) = \sum_{f=1}^{F} \sum_{n=1}^{N} d\big([\mathbf{V}]_{fn} \mid [\mathbf{WH}]_{fn}\big).$$

- How to find the common/latent dimension K?
- If K is too large $\implies$ Overfitting! K too small $\implies$ Poor fit to model!
- Solve this by automatic relevance determination (Bishop, 1999; Tipping, 2001)
- Natural extension of regularization ideas discussed by Cédric.

Probabilistic Model for ARD in NMF

- Assign each column of $\mathbf{W}$ and each row of $\mathbf{H}$ priors

$$\mathbf{W} = \begin{bmatrix} | & | & & | \\ \mathbf{w}_1 & \mathbf{w}_2 & \dots & \mathbf{w}_K \\ | & | & & | \end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix} - & \underline{h}_1 & - \\ - & \underline{h}_2 & - \\ & \vdots & \\ - & \underline{h}_K & - \end{bmatrix}$$

Probabilistic Model for ARD in NMF

- Assign each column of $\mathbf{W}$ and each row of $\mathbf{H}$ priors

$$\mathbf{W} = \begin{bmatrix} | & | & & | \\ \mathbf{w}_1 & \mathbf{w}_2 & \dots & \mathbf{w}_K \\ | & | & & | \end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix} - & \underline{h}_1 & - \\ - & \underline{h}_2 & - \\ & \vdots & \\ - & \underline{h}_K & - \end{bmatrix}$$

- Tie the k^{th} column $\mathbf{w}_k$ and the k^{th} row $\underline{h}_k$ together through a common relevance weight $\lambda_k \geq 0$.

Probabilistic Model for ARD in NMF

- Assign each column of $\mathbf{W}$ and each row of $\mathbf{H}$ priors

$$\mathbf{W} = \begin{bmatrix} | & | & & | \\ \mathbf{w}_1 & \mathbf{w}_2 & \cdots & \mathbf{w}_K \\ | & | & & | \end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix} - & \underline{h}_1 & - \\ - & \underline{h}_2 & - \\ & \vdots & \\ - & \underline{h}_K & - \end{bmatrix}$$

- Tie the k^{th} column $\mathbf{w}_k$ and the k^{th} row $\underline{h}_k$ together through a common relevance weight $\lambda_k \geq 0$.

- Maintain nonnegativity by choosing nonnegative priors, e.g.,
  - Half Gaussian, i.e.,

$$p(w_{fk} \mid \lambda_k) = \Big(\frac{2}{\pi\lambda_k}\Big)^{1/2} \exp\Big(-\frac{w_{fk}^2}{2\lambda_k}\Big) \qquad p(h_{kn} \mid \lambda_k) = \Big(\frac{2}{\pi\lambda_k}\Big)^{1/2} \exp\Big(-\frac{h_{kn}^2}{2\lambda_k}\Big).$$

  - Exponential

$$p(w_{fk} \mid \lambda_k) = \frac{1}{\lambda_k} \exp\Big(-\frac{w_{fk}}{\lambda_k}\Big) \qquad p(h_{kn} \mid \lambda_k) = \frac{1}{\lambda_k} \exp\Big(-\frac{h_{kn}}{\lambda_k}\Big)$$

  - Both these distributions are supported on $\mathbb{R}_+$.

Half Gaussian and Exponential

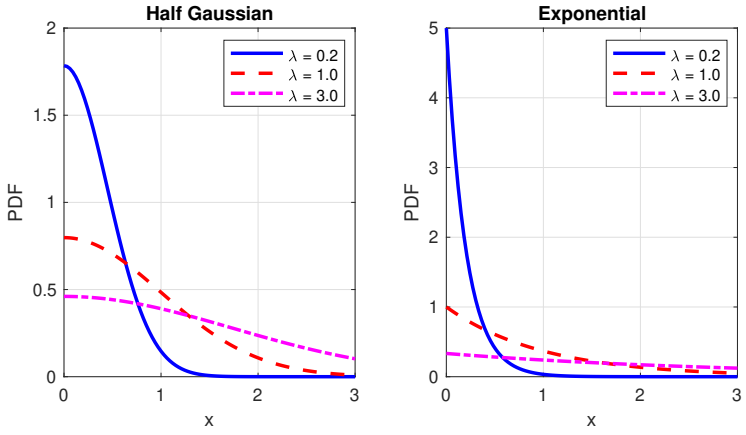
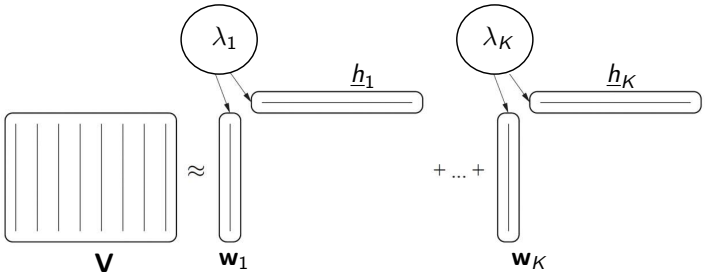


Figure: Half Gaussian and Exponential Distributions

Probabilistic Model for ARD in NMF



- λ_k is a common variance-like quantity.
- When $\lambda_k \downarrow 0$, $\|\mathbf{w}_k\|$ and $\|\underline{h}_k\|$ both tend to 0.
- The k^{th} component can be removed without compromising data fidelity.

Probabilistic Model for ARD in NMF

- Prior on common variance-like parameter λ_k is inverse-Gamma

$$p(\lambda_k; a, b) = \frac{b^a}{\Gamma(a)} \lambda_k^{-(a+1)} \exp\Big(- \frac{b}{\lambda_k} \Big),$$

where a and b are the shape and scale hyperparameters, respectively.

Probabilistic Model for ARD in NMF

- Prior on common variance-like parameter λ_k is inverse-Gamma

$$p(\lambda_k; a, b) = \frac{b^a}{\Gamma(a)} \lambda_k^{-(a+1)} \exp\Big(- \frac{b}{\lambda_k} \Big),$$

  where a and b are the shape and scale hyperparameters, respectively.

- Set a and b to be the same for all k.

Probabilistic Model for ARD in NMF

- Prior on common variance-like parameter λ_k is inverse-Gamma

$$p(\lambda_k; a, b) = \frac{b^a}{\Gamma(a)} \lambda_k^{-(a+1)} \exp\Big(- \frac{b}{\lambda_k} \Big),$$

  where a and b are the shape and scale hyperparameters, respectively.

- Set a and b to be the same for all k.

- The inverse-Gamma prior is chosen because it is conjugate to the variance-parameter in the Half Gaussian and the inverse rate parameter in the Exponential.

Probabilistic Model for ARD in NMF

- Prior on common variance-like parameter λ_k is inverse-Gamma

$$p(\lambda_k; a, b) = \frac{b^a}{\Gamma(a)} \lambda_k^{-(a+1)} \exp\Big(- \frac{b}{\lambda_k} \Big),$$

  where a and b are the shape and scale hyperparameters, respectively.

- Set a and b to be the same for all k.
- The inverse-Gamma prior is chosen because it is conjugate to the variance-parameter in the Half Gaussian and the inverse rate parameter in the Exponential.
- Leads to closed-form updates.

Probabilistic Model for ARD in NMF

- Prior on common variance-like parameter λ_k is inverse-Gamma

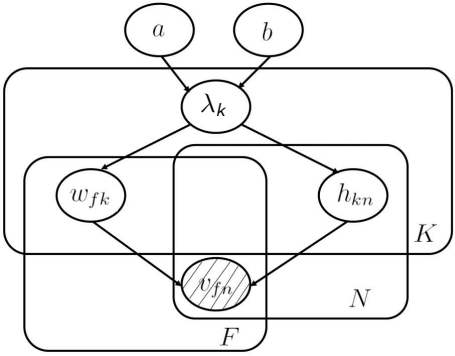
$$p(\lambda_k; a, b) = \frac{b^a}{\Gamma(a)} \lambda_k^{-(a+1)} \exp\Big(- \frac{b}{\lambda_k} \Big),$$

  where a and b are the shape and scale hyperparameters, respectively.

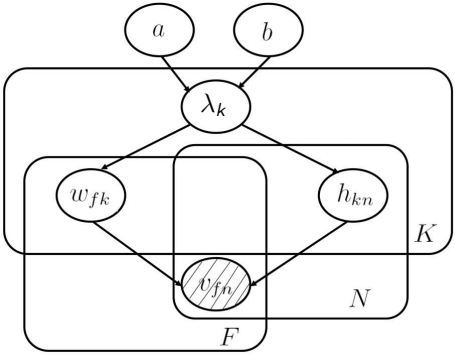
- Set a and b to be the same for all k.
- The inverse-Gamma prior is chosen because it is conjugate to the variance-parameter in the Half Gaussian and the inverse rate parameter in the Exponential.
- Leads to closed-form updates.
- Assume independence

$$p(\boldsymbol{\lambda}; a, b) = \prod_{k=1}^{K} p(\lambda_k; a, b).$$

Probabilistic Model for ARD in NMF



Probabilistic Model for ARD in NMF



- $\mathbf{V} = [v_{fn}]$ are observed;
- a, b are hyperparameters;
- Want to learn $\mathbf{W} = [w_{fn}]$ and $\mathbf{H} = [h_{kn}]$ and implicitly K, i.e.,

$$K = |\{k \in [K] : \lambda_k > \text{threshold}\}| .$$

Objective function for ARD in NMF

- Combining the prior and likelihood, the objective function (log-posterior) can be written as

$$
\begin{aligned}
C(\mathbf{W}, \mathbf{H}, \boldsymbol{\lambda}) &= -\log p(\mathbf{W}, \mathbf{H}, \boldsymbol{\lambda} \mid \mathbf{V}) \\
&\stackrel{c}{=} \frac{1}{\phi} D_\beta(\mathbf{V} \mid \mathbf{WH}) + \sum_{k=1}^{K} \frac{1}{\lambda_k} \big(f(\mathbf{w}_k) + f(\underline{h}_k) + b \big) + c \log \lambda_k .
\end{aligned}
$$

Objective function for ARD in NMF

- Combining the prior and likelihood, the objective function (log-posterior) can be written as

$$C(\mathbf{W}, \mathbf{H}, \boldsymbol{\lambda}) = -\log p(\mathbf{W}, \mathbf{H}, \boldsymbol{\lambda} \mid \mathbf{V})$$
$$\stackrel{c}{=} \frac{1}{\phi} D_\beta(\mathbf{V} \mid \mathbf{WH}) + \sum_{k=1}^{K} \frac{1}{\lambda_k} \big(f(\mathbf{w}_k) + f(\underline{h}_k) + b\big) + c \log \lambda_k.$$

- Constant ϕ is the dispersion parameter (of the Tweedie distribution):
  - $\beta = 2$: Gaussian distribution and $\phi = \sigma^2$;
  - $\beta = 1$: Poisson distribution and $\phi = 1$;
  - $\beta = 0$: Gamma distribution and $\phi = 1/\alpha$ where α is the shape parameter;

Objective function for ARD in NMF

- Combining the prior and likelihood, the objective function (log-posterior) can be written as

$$C(\mathbf{W}, \mathbf{H}, \boldsymbol{\lambda}) = -\log p(\mathbf{W}, \mathbf{H}, \boldsymbol{\lambda} \mid \mathbf{V})$$
$$\stackrel{c}{=} \frac{1}{\phi} D_\beta(\mathbf{V} \mid \mathbf{WH}) + \sum_{k=1}^{K} \frac{1}{\lambda_k} \big(f(\mathbf{w}_k) + f(\underline{h}_k) + b\big) + c \log \lambda_k.$$

- Constant ϕ is the dispersion parameter (of the Tweedie distribution):
  - $\beta = 2$: Gaussian distribution and $\phi = \sigma^2$;
  - $\beta = 1$: Poisson distribution and $\phi = 1$;
  - $\beta = 0$: Gamma distribution and $\phi = 1/\alpha$ where α is the shape parameter;

- Constant c and function f depend on the likelihood model:
  - Half Gaussian model: $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2$ and $c = (F + N)/2 + a + 1$;
  - Exponent model: $f(\mathbf{x}) = \|\mathbf{x}\|_1$ and $c = F + N + a + 1$

Objective function for ARD in NMF

- Combining the prior and likelihood, the objective function (log-posterior) can be written as

$$C(\mathbf{W},\mathbf{H},\boldsymbol{\lambda}) = -\log p(\mathbf{W},\mathbf{H},\boldsymbol{\lambda} \mid \mathbf{V})$$
$$\stackrel{c}{=} \frac{1}{\phi} D_\beta(\mathbf{V} \mid \mathbf{WH}) + \sum_{k=1}^{K} \frac{1}{\lambda_k}\big(f(\mathbf{w}_k) + f(\underline{h}_k) + b\big) + c\log\lambda_k.$$

- Constant ϕ is the dispersion parameter (of the Tweedie distribution):
  - $\beta = 2$: Gaussian distribution and $\phi = \sigma^2$;
  - $\beta = 1$: Poisson distribution and $\phi = 1$;
  - $\beta = 0$: Gamma distribution and $\phi = 1/\alpha$ where α is the shape parameter;

- Constant c and function f depend on the likelihood model:
  - Half Gaussian model: $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2$ and $c = (F+N)/2 + a + 1$;
  - Exponent model: $f(\mathbf{x}) = \|\mathbf{x}\|_1$ and $c = F + N + a + 1$

- This cost has connections to reweighted ℓ_1 minimization (Candès et al., 2008) and group LASSO (Yuan and Lin, 2007).

Majorization-Minimization Algorithms for ℓ_2-ARD-NMF

- Using the MM ideas discussed by Cédric, we can derive updates for $\mathbf{W}$ and $\mathbf{H}$:

$$\mathbf{H} \longleftarrow \mathbf{H} \cdot \left(\frac{\mathbf{W}^\top [(\mathbf{WH})^{\cdot(\beta-2)} \cdot \mathbf{V}]}{\mathbf{W}^\top [(\mathbf{WH})]^{\cdot(\beta-1)} + \phi \mathbf{H} / \mathrm{repmat}(\boldsymbol{\lambda}, 1, N)} \right)^{\xi(\beta)}$$

$$\mathbf{W} \longleftarrow \mathbf{W} \cdot \left(\frac{[(\mathbf{WH})^{\cdot(\beta-2)} \cdot \mathbf{V}]\mathbf{H}^\top}{[(\mathbf{WH})]^{\cdot(\beta-1)}\mathbf{H}^\top + \phi \mathbf{W} / \mathrm{repmat}(\boldsymbol{\lambda}, F, 1)} \right)^{\xi(\beta)}$$

where

$$\xi(\beta) = \begin{cases} 1/(3-\beta) & \beta \leq 2 \\ 1/(\beta-1) & \beta > 2 \end{cases} .$$

Majorization-Minimization Algorithms for ℓ_2-ARD-NMF

- Using the MM ideas discussed by Cédric, we can derive updates for $\mathbf{W}$ and $\mathbf{H}$:

$$\mathbf{H} \longleftarrow \mathbf{H} \cdot \left(\frac{\mathbf{W}^\top [(\mathbf{WH})^{\cdot(\beta-2)} \cdot \mathbf{V}]}{\mathbf{W}^\top [(\mathbf{WH})]^{\cdot(\beta-1)} + \phi \mathbf{H} / \mathrm{repmat}(\boldsymbol{\lambda}, 1, N)} \right)^{\xi(\beta)}$$

$$\mathbf{W} \longleftarrow \mathbf{W} \cdot \left(\frac{[(\mathbf{WH})^{\cdot(\beta-2)} \cdot \mathbf{V}]\mathbf{H}^\top}{[(\mathbf{WH})]^{\cdot(\beta-1)}\mathbf{H}^\top + \phi \mathbf{W} / \mathrm{repmat}(\boldsymbol{\lambda}, F, 1)} \right)^{\xi(\beta)}$$

where

$$\xi(\beta) = \begin{cases} 1/(3-\beta) & \beta \leq 2 \\ 1/(\beta-1) & \beta > 2 \end{cases} .$$

- The update for $\boldsymbol{\lambda}$ is

$$\lambda_k \longleftarrow \frac{\frac{1}{2}\|\mathbf{w}_k\|^2 + \frac{1}{2}\|\underline{h}_k\|^2 + b}{c} \qquad \forall\, k \in [K].$$

Estimating Hyperparameter b via the Method of Moments

- By the law of large numbers

$$\hat{\mu}_{\mathbf{V}} = \frac{1}{FN} \sum_{f',n'} v_{f'n'} \approx \mathbb{E}[v_{fn}] = \mathbb{E}[\hat{v}_{fn}] = \sum_{k} \mathbb{E}[w_{fk} h_{kn}].$$

Estimating Hyperparameter b via the Method of Moments

- By the law of large numbers

$$\hat{\mu}_{\mathbf{V}} = \frac{1}{FN} \sum_{f',n'} v_{f'n'} \approx \mathbb{E}[v_{fn}] = \mathbb{E}[\hat{v}_{fn}] = \sum_{k} \mathbb{E}[w_{fk} h_{kn}].$$

- Can compute $\mathbb{E}[\hat{v}_{fn}] = \sum_k \mathbb{E}[w_{fk} h_{kn}]$ in closed-form for the Half Gaussian and Exponential models using their moments:

$$\mathbb{E}[\hat{v}_{fn}] = \begin{cases} \dfrac{2Kb}{\pi(a-1)} & \text{Half Gaussian} \\ \dfrac{Kb^2}{(a-1)(a-2)} & \text{Exponential} \end{cases}$$

Estimating Hyperparameter b via the Method of Moments

- By the law of large numbers

$$\hat{\mu}_{\mathbf{V}} = \frac{1}{FN} \sum_{f',n'} v_{f'n'} \approx \mathbb{E}[v_{fn}] = \mathbb{E}[\hat{v}_{fn}] = \sum_{k} \mathbb{E}[w_{fk} h_{kn}].$$

- Can compute $\mathbb{E}[\hat{v}_{fn}] = \sum_k \mathbb{E}[w_{fk} h_{kn}]$ in closed-form for the Half Gaussian and Exponential models using their moments:

$$\mathbb{E}[\hat{v}_{fn}] = \begin{cases} \dfrac{2Kb}{\pi(a-1)} & \text{Half Gaussian} \\ \dfrac{Kb^2}{(a-1)(a-2)} & \text{Exponential} \end{cases}$$

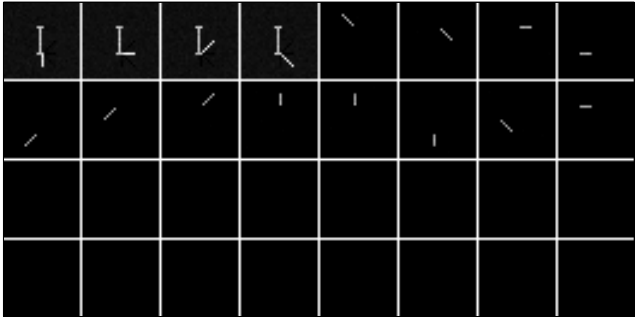
- Can "invert" these relations to yield

$$\hat{b} = \begin{cases} \dfrac{\pi(a-1)\hat{\mu}_{\mathbf{V}}}{2K} & \text{Half Gaussian} \\ \sqrt{\dfrac{(a-1)(a-2)\hat{\mu}_{\mathbf{V}}}{K}} & \text{Exponential} \end{cases}$$

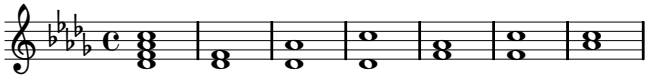
Swimmer Decomposition Results

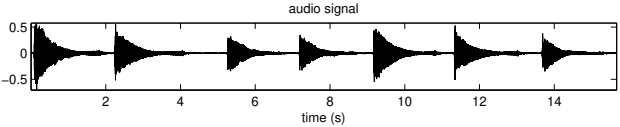
8 data samples (among 256)

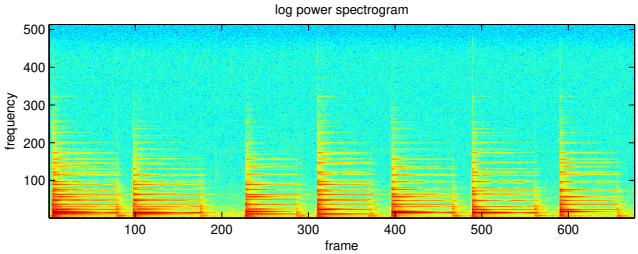
Estimated $\mathbf{W}$ using exponential priors/ℓ_1 penalization



Audio Decomposition Results







Audio Decomposition Results

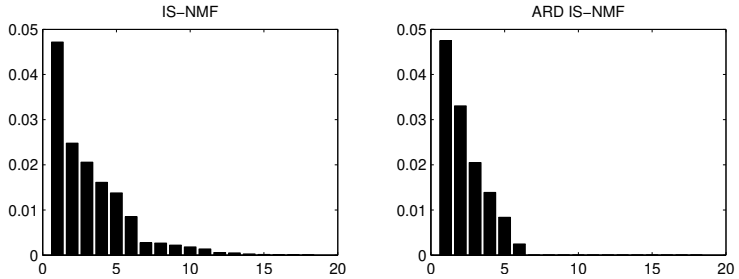


Figure: Histograms of standard deviation values of all $K = 18$ components produced by Itakura–Saito NMF and ARD Itakura–Saito NMF (with ℓ_2 penalization). ARD IS-NMF only retains the 6 "right" components

Audio Decomposition Results

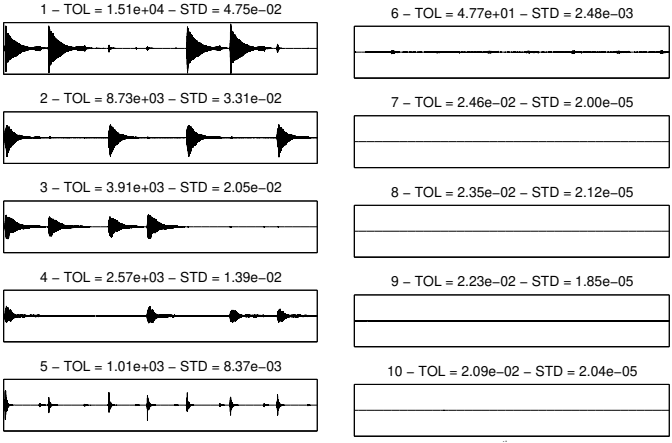


Figure: First 4 components extract the individual notes and the next 2 components extract the sound of hammer hitting the strings and the sound produced by the sustain pedal

Concluding Remarks from using ARD on NMF

- Introduced an Automatic Relevance Determination framework for learning the common/latent dimension K in NMF.

Concluding Remarks from using ARD on NMF

- Introduced an Automatic Relevance Determination framework for learning the common/latent dimension K in NMF.
- Simple, cheap and intuitive.

Concluding Remarks from using ARD on NMF

- Introduced an Automatic Relevance Determination framework for learning the common/latent dimension K in NMF.
- Simple, cheap and intuitive.
- Since its publication, ARD NMF (Tan and Févotte, 2013) has been used successfully in biology and genomics, among other scientific fields, e.g.,

[HTML] Comprehensive molecular characterization of muscle-invasive bladder cancer
AG Robertson, J Kim, H Al-Ahmadie, J Bellmunt, G Guo… - Cell, 2017 - Elsevier
We report a comprehensive analysis of 412 muscle-invasive bladder cancers characterized by multiple TCGA analytical platforms. Fifty-eight genes were significantly mutated, and the …
☆ Save 99 Cite Cited by 1453 Related articles All 24 versions

[HTML] Comprehensive and integrative genomic characterization of hepatocellular carcinoma
A Ally, M Balasundaram, R Carlsen, E Chuah, A Clarke… - Cell, 2017 - Elsevier
Liver cancer has the second highest worldwide cancer mortality rate and has limited therapeutic options. We analyzed 363 hepatocellular carcinoma (HCC) cases by whole …
☆ Save 99 Cite Cited by 1153 Related articles All 17 versions

[HTML] The repertoire of mutational signatures in human cancer
LB Alexandrov, J Kim, NJ Haradhvala, MN Huang… - Nature, 2020 - nature.com
Somatic mutations in cancer genomes are caused by multiple mutational processes, each of which generates a characteristic mutational signature 1. Here, as part of the Pan-Cancer …
☆ Save 99 Cite Cited by 1073 Related articles All 19 versions

Outline

Distributionally Robust Nonnegative Matrix Factorization

(Gillis, Hien, Leplat, and Tan, 2022)

Distributionally Robust Nonnegative Matrix Factorization

(Gillis, Hien, Leplat, and Tan, 2022)

- The parameter β in D_β controls the noise statistics on **WH** (Tweedie distn);

Distributionally Robust Nonnegative Matrix Factorization

(Gillis, Hien, Leplat, and Tan, 2022)

- The parameter β in D_β controls the noise statistics on $\mathbf{WH}$ (Tweedie distn);
- If $v_{fn} = [\mathbf{WH}]_{fn} + \text{Gaussian noise}$ ($\beta = 2$), then

$$-\log p(v_{fn} \mid [\mathbf{WH}]_{fn}) \overset{c}{=} \frac{1}{2\sigma^2}\left([\mathbf{WH}]_{fn} - v_{fn}\right)^2,$$

then maximizing the log-likelihood $\equiv$ minimizing D_2 (Frobenius-NMF).

Distributionally Robust Nonnegative Matrix Factorization

(Gillis, Hien, Leplat, and Tan, 2022)

- The parameter β in D_β controls the noise statistics on $\mathbf{WH}$ (Tweedie distn);
- If $v_{fn} = [\mathbf{WH}]_{fn} + \text{Gaussian noise}$ ($\beta = 2$), then

$$-\log p(v_{fn} \mid [\mathbf{WH}]_{fn}) \overset{c}{=} \frac{1}{2\sigma^2}\big([\mathbf{WH}]_{fn} - v_{fn}\big)^2,$$

  then maximizing the log-likelihood $\equiv$ minimizing D_2 (Frobenius-NMF).

- If $v_{fn} \sim \text{Poisson}([\mathbf{WH}]_{fn})$ ($\beta = 1$), then

$$-\log p(v_{fn} \mid [\mathbf{WH}]_{fn}) = v_{fn} \log \frac{v_{fn}}{[\mathbf{WH}]_{fn}} + [\mathbf{WH}]_{fn},$$

  then maximizing the log-likelihood $\equiv$ minimizing D_1 (KL-NMF).

Distributionally Robust Nonnegative Matrix Factorization

(Gillis, Hien, Leplat, and Tan, 2022)

- The parameter β in D_β controls the noise statistics on $\mathbf{WH}$ (Tweedie distn);
- If $v_{fn} = [\mathbf{WH}]_{fn} + \text{Gaussian noise}$ ($\beta = 2$), then

$$-\log p(v_{fn} \mid [\mathbf{WH}]_{fn}) \overset{c}{=} \frac{1}{2\sigma^2}\big([\mathbf{WH}]_{fn} - v_{fn}\big)^2,$$

  then maximizing the log-likelihood $\equiv$ minimizing D_2 (Frobenius-NMF).

- If $v_{fn} \sim \text{Poisson}([\mathbf{WH}]_{fn})$ ($\beta = 1$), then

$$-\log p(v_{fn} \mid [\mathbf{WH}]_{fn}) = v_{fn} \log \frac{v_{fn}}{[\mathbf{WH}]_{fn}} + [\mathbf{WH}]_{fn},$$

  then maximizing the log-likelihood $\equiv$ minimizing D_1 (KL-NMF).

- If $v_{fn} \sim \text{Gamma}(\alpha, [\mathbf{WH}]_{fn}/\alpha)$ ($\beta = 0$), then

$$-\log p(v_{fn} \mid [\mathbf{WH}]_{fn}) = \frac{v_{fn}}{[\mathbf{WH}]_{fn}} - \log \frac{v_{fn}}{[\mathbf{WH}]_{fn}} - 1,$$

  then maximizing the log-likelihood $\equiv$ minimizing D_0 (IS-NMF).

Applications, MO-NMF and DR-NMF

- Audio signal processing (Févotte et al., 2009; Virtanen, 2007): $\beta \in \{0, 1\}$

Applications, MO-NMF and DR-NMF

- Audio signal processing (Févotte et al., 2009; Virtanen, 2007): $\beta \in \{0, 1\}$
- Sparse document datasets (Chi and Kolda, 2012): $\beta \in \{1, 2\}$

Applications, MO-NMF and DR-NMF

- Audio signal processing (Févotte et al., 2009; Virtanen, 2007): $\beta \in \{0, 1\}$
- Sparse document datasets (Chi and Kolda, 2012): $\beta \in \{1, 2\}$
- How to choose an appropriate β when given a new task? Say we only consider $\beta \in \Omega$ where $\Omega \subset \mathbb{R}$ is a finite set, e.g., $\Omega = \{0, 1, 2\}$.

Applications, MO-NMF and DR-NMF

- Audio signal processing (Févotte et al., 2009; Virtanen, 2007): $\beta \in \{0, 1\}$
- Sparse document datasets (Chi and Kolda, 2012): $\beta \in \{1, 2\}$
- How to choose an appropriate β when given a new task? Say we only consider $\beta \in \Omega$ where $\Omega \subset \mathbb{R}$ is a finite set, e.g., $\Omega = \{0, 1, 2\}$.
- Multi-Objective NMF (MO-NMF)

$$\min_{\mathbf{W},\mathbf{H} \geq \mathbf{0}} \{D_\beta(\mathbf{V}, \mathbf{WH})\}_{\beta \in \Omega}$$

which is solved for a given probability vector $\boldsymbol{\lambda} = (\lambda_\beta)_{\beta \in \Omega}$ using

$$\min_{\mathbf{W},\mathbf{H} \geq \mathbf{0}} \left[D_\Omega^{\boldsymbol{\lambda}}(\mathbf{V}, \mathbf{WH}) = \sum_{\beta \in \Omega} \lambda_\beta D_\beta(\mathbf{V}, \mathbf{WH}) \right]$$

Applications, MO-NMF and DR-NMF

- Audio signal processing (Févotte et al., 2009; Virtanen, 2007): $\beta \in \{0, 1\}$
- Sparse document datasets (Chi and Kolda, 2012): $\beta \in \{1, 2\}$
- How to choose an appropriate β when given a new task? Say we only consider $\beta \in \Omega$ where $\Omega \subset \mathbb{R}$ is a finite set, e.g., $\Omega = \{0, 1, 2\}$.
- Multi-Objective NMF (MO-NMF)

$$\min_{\mathbf{W},\mathbf{H} \geq \mathbf{0}} \{D_\beta(\mathbf{V}, \mathbf{WH})\}_{\beta \in \Omega}$$

  which is solved for a given probability vector $\boldsymbol{\lambda} = (\lambda_\beta)_{\beta \in \Omega}$ using

$$\min_{\mathbf{W},\mathbf{H} \geq \mathbf{0}} \left[D_\Omega^{\boldsymbol{\lambda}}(\mathbf{V}, \mathbf{WH}) = \sum_{\beta \in \Omega} \lambda_\beta D_\beta(\mathbf{V}, \mathbf{WH}) \right]$$

- Distributionally Robust NMF (DR-NMF)

$$\min_{\mathbf{W},\mathbf{H} \geq \mathbf{0}} \max_{\beta \in \Omega} D_\beta(\mathbf{V}, \mathbf{WH})$$

Scaling of the Objective

- For the family of β-divergences,

$$D_\beta(\alpha \mathbf{V}, \alpha \mathbf{WH}) = \alpha^\beta D_\beta(\mathbf{V}, \mathbf{WH}) \qquad \forall\, \alpha > 0.$$

Scaling of the Objective

- For the family of β-divergences,

$$D_\beta(\alpha \mathbf{V}, \alpha \mathbf{WH}) = \alpha^\beta D_\beta(\mathbf{V}, \mathbf{WH}) \qquad \forall\, \alpha > 0.$$

- Not desirable in practice as datasets are not properly scaled.

Scaling of the Objective

- For the family of β-divergences,

$$D_\beta(\alpha \mathbf{V}, \alpha \mathbf{WH}) = \alpha^\beta D_\beta(\mathbf{V}, \mathbf{WH}) \qquad \forall\, \alpha > 0.$$

- Not desirable in practice as datasets are not properly scaled.
- Compute an approximate solution

$$(\mathbf{W}_\beta, \mathbf{H}_\beta) \approx \underset{\mathbf{W},\mathbf{H} \geq \mathbf{0}}{\arg\min}\, D_\beta(\mathbf{V}, \mathbf{WH}) \quad \text{with error} \quad e_\beta = D_\beta(\mathbf{V}, \mathbf{W}_\beta \mathbf{H}_\beta)$$

and define

$$\overline{D}_\beta(\mathbf{V}, \mathbf{WH}) = \frac{D_\beta(\mathbf{V}, \mathbf{WH})}{e_\beta} \quad \text{so that} \quad \overline{D}_\beta(\mathbf{V}, \mathbf{W}_\beta \mathbf{H}_\beta) = 1.$$

Scaling of the Objective

- For the family of β-divergences,

$$D_\beta(\alpha \mathbf{V}, \alpha \mathbf{WH}) = \alpha^\beta D_\beta(\mathbf{V}, \mathbf{WH}) \qquad \forall\, \alpha > 0.$$

- Not desirable in practice as datasets are not properly scaled.

- Compute an approximate solution

$$(\mathbf{W}_\beta, \mathbf{H}_\beta) \approx \underset{\mathbf{W},\mathbf{H} \geq \mathbf{0}}{\arg\min}\, D_\beta(\mathbf{V}, \mathbf{WH}) \quad \text{with error} \quad e_\beta = D_\beta(\mathbf{V}, \mathbf{W}_\beta \mathbf{H}_\beta)$$

and define

$$\overline{D}_\beta(\mathbf{V}, \mathbf{WH}) = \frac{D_\beta(\mathbf{V}, \mathbf{WH})}{e_\beta} \quad \text{so that} \quad \overline{D}_\beta(\mathbf{V}, \mathbf{W}_\beta \mathbf{H}_\beta) = 1.$$

- Consider the optimization problem

$$\min_{\mathbf{W},\mathbf{H} \geq \mathbf{0}} \overline{D}^{\boldsymbol{\lambda}}_\Omega(\mathbf{V}, \mathbf{WH}) \quad \text{where} \quad \overline{D}^{\boldsymbol{\lambda}}_\Omega(\mathbf{V}, \mathbf{WH}) = \sum_{\beta \in \Omega} \lambda_\beta \overline{D}_\beta(\mathbf{V}, \mathbf{WH}).$$

Multiplicative Update Algorithm

- Let $f : \mathbb{R}^d \to \mathbb{R}$ be a differentiable function. Consider the general optimization problem with nonnegativity constraints

$$\min\{f(\mathbf{x}) \ : \ \mathbf{x} \geq \mathbf{0}\}.$$

Multiplicative Update Algorithm

- Let $f : \mathbb{R}^d \to \mathbb{R}$ be a differentiable function. Consider the general optimization problem with nonnegativity constraints

$$\min\{f(\mathbf{x}) \ : \ \mathbf{x} \geq \mathbf{0}\}.$$

- Rescaled gradient descent method (with rescaling matrix $\mathbf{B}$)

$$\mathbf{x}^+ = \mathbf{x} - \mathbf{B}\nabla f(\mathbf{x})$$

Multiplicative Update Algorithm

- Let $f : \mathbb{R}^d \to \mathbb{R}$ be a differentiable function. Consider the general optimization problem with nonnegativity constraints

$$\min\{f(\mathbf{x}) \ : \ \mathbf{x} \geq \mathbf{0}\}.$$

- Rescaled gradient descent method (with rescaling matrix $\mathbf{B}$)

$$\mathbf{x}^+ = \mathbf{x} - \mathbf{B}\nabla f(\mathbf{x})$$

- Say that $\nabla f(\mathbf{x}) = \nabla_+ f(\mathbf{x}) - \nabla_- f(\mathbf{x})$ where $\nabla_+ f(\mathbf{x}) \geq \mathbf{0}$ and $\nabla_- f(\mathbf{x}) > \mathbf{0}$.

Multiplicative Update Algorithm

- Let $f : \mathbb{R}^d \to \mathbb{R}$ be a differentiable function. Consider the general optimization problem with nonnegativity constraints

$$\min\{f(\mathbf{x}) \ : \ \mathbf{x} \geq \mathbf{0}\}.$$

- Rescaled gradient descent method (with rescaling matrix $\mathbf{B}$)

$$\mathbf{x}^+ = \mathbf{x} - \mathbf{B}\nabla f(\mathbf{x})$$

- Say that $\nabla f(\mathbf{x}) = \nabla_+ f(\mathbf{x}) - \nabla_- f(\mathbf{x})$ where $\nabla_+ f(\mathbf{x}) \geq \mathbf{0}$ and $\nabla_- f(\mathbf{x}) > \mathbf{0}$.

- Taking $B_{ii} = x_i / [\nabla_+ f(\mathbf{x})]_i$, we obtain

$$\mathbf{x}^+ = \mathbf{x} - \frac{[\mathbf{x}]}{[\nabla_+ f(\mathbf{x})]} \big(\nabla_+ f(\mathbf{x}) - \nabla_- f(\mathbf{x})\big) = \mathbf{x} \cdot \frac{\nabla_- f(\mathbf{x})}{\nabla_+ f(\mathbf{x})}$$

Multiplicative Update Algorithm

- Let $f : \mathbb{R}^d \to \mathbb{R}$ be a differentiable function. Consider the general optimization problem with nonnegativity constraints

$$\min\{f(\mathbf{x}) \ : \ \mathbf{x} \geq \mathbf{0}\}.$$

- Rescaled gradient descent method (with rescaling matrix $\mathbf{B}$)

$$\mathbf{x}^+ = \mathbf{x} - \mathbf{B}\nabla f(\mathbf{x})$$

- Say that $\nabla f(\mathbf{x}) = \nabla_+ f(\mathbf{x}) - \nabla_- f(\mathbf{x})$ where $\nabla_+ f(\mathbf{x}) \geq \mathbf{0}$ and $\nabla_- f(\mathbf{x}) > \mathbf{0}$.

- Taking $B_{ii} = x_i / [\nabla_+ f(\mathbf{x})]_i$, we obtain

$$\mathbf{x}^+ = \mathbf{x} - \frac{[\mathbf{x}]}{[\nabla_+ f(\mathbf{x})]}\big(\nabla_+ f(\mathbf{x}) - \nabla_- f(\mathbf{x})\big) = \mathbf{x} \cdot \frac{\nabla_- f(\mathbf{x})}{\nabla_+ f(\mathbf{x})}$$

- No tuning of step-sizes. If $\mathbf{x} \geq \mathbf{0}$, then $\mathbf{x}^+ \geq \mathbf{0}$ as well.

Application of MU Algorithm to DR-NMF

- Recall that for a fixed probability vector $\boldsymbol{\lambda}$, we want to solve

$$\min_{\mathbf{W},\mathbf{H}\geq 0} \overline{D}_{\Omega}^{\boldsymbol{\lambda}}(\mathbf{V},\mathbf{WH}), \quad \text{where} \quad \overline{D}_{\Omega}^{\boldsymbol{\lambda}}(\mathbf{V},\mathbf{WH}) = \sum_{\beta\in\Omega} \lambda_{\beta}\overline{D}_{\beta}(\mathbf{V},\mathbf{WH}).$$

Application of MU Algorithm to DR-NMF

- Recall that for a fixed probability vector $\boldsymbol{\lambda}$, we want to solve

$$\min_{\mathbf{W},\mathbf{H}\geq 0} \overline{D}_{\Omega}^{\boldsymbol{\lambda}}(\mathbf{V},\mathbf{WH}), \quad \text{where} \quad \overline{D}_{\Omega}^{\boldsymbol{\lambda}}(\mathbf{V},\mathbf{WH}) = \sum_{\beta\in\Omega} \lambda_{\beta} \overline{D}_{\beta}(\mathbf{V},\mathbf{WH}).$$

- Alternating minimization procedure: Minimize over $\mathbf{H}$, then over $\mathbf{W}$.

Application of MU Algorithm to DR-NMF

- Recall that for a fixed probability vector $\boldsymbol{\lambda}$, we want to solve

$$\min_{\mathbf{W},\mathbf{H}\geq 0} \overline{D}_{\Omega}^{\boldsymbol{\lambda}}(\mathbf{V},\mathbf{WH}), \quad \text{where} \quad \overline{D}_{\Omega}^{\boldsymbol{\lambda}}(\mathbf{V},\mathbf{WH}) = \sum_{\beta\in\Omega} \lambda_\beta \overline{D}_\beta(\mathbf{V},\mathbf{WH}).$$

- Alternating minimization procedure: Minimize over $\mathbf{H}$, then over $\mathbf{W}$.
- For all β,

$$\nabla^{\mathbf{H}} D_\beta(\mathbf{V},\mathbf{WH}) = \nabla_+^{\mathbf{H}} D_\beta(\mathbf{V},\mathbf{WH}) - \nabla_-^{\mathbf{H}} D_\beta(\mathbf{V},\mathbf{WH}),$$

where $\nabla^{\mathbf{H}}$ means gradient w.r.t. $\mathbf{H}$.

Application of MU Algorithm to DR-NMF

- Recall that for a fixed probability vector $\boldsymbol{\lambda}$, we want to solve

$$\min_{\mathbf{W},\mathbf{H}\geq 0} \overline{D}_{\Omega}^{\boldsymbol{\lambda}}(\mathbf{V},\mathbf{WH}), \quad \text{where} \quad \overline{D}_{\Omega}^{\boldsymbol{\lambda}}(\mathbf{V},\mathbf{WH}) = \sum_{\beta\in\Omega} \lambda_\beta \overline{D}_\beta(\mathbf{V},\mathbf{WH}).$$

- Alternating minimization procedure: Minimize over $\mathbf{H}$, then over $\mathbf{W}$.

- For all β,

$$\nabla^{\mathbf{H}} D_\beta(\mathbf{V},\mathbf{WH}) = \nabla_+^{\mathbf{H}} D_\beta(\mathbf{V},\mathbf{WH}) - \nabla_-^{\mathbf{H}} D_\beta(\mathbf{V},\mathbf{WH}),$$

  where $\nabla^{\mathbf{H}}$ means gradient w.r.t. $\mathbf{H}$.

- After some tedious calculation,

$$\nabla_+^{\mathbf{H}} D_\beta(\mathbf{V},\mathbf{WH}) = \mathbf{W}^\top (\mathbf{WH})^{\cdot(\beta-1)} \quad \text{and}$$
$$\nabla_-^{\mathbf{H}} D_\beta(\mathbf{V},\mathbf{WH}) = \mathbf{W}^\top \left((\mathbf{WH})^{\cdot(\beta-2)} \cdot \mathbf{V} \right),$$

Application of MU Algorithm to DR-NMF

- Update $\mathbf{H}$ as follows:

$$\mathbf{H}^{+} = \mathbf{H} \cdot \frac{\sum_{\beta \in \Omega} \lambda_{\beta}\left(\nabla_{-}^{\mathbf{H}} \overline{D}_{\beta}(\mathbf{V}, \mathbf{WH})\right)}{\sum_{\beta \in \Omega} \lambda_{\beta}\left(\nabla_{+}^{\mathbf{H}} \overline{D}_{\beta}(\mathbf{V}, \mathbf{WH})\right)}.$$

Application of MU Algorithm to DR-NMF

- Update $\mathbf{H}$ as follows:

$$\mathbf{H}^{+} = \mathbf{H} \cdot \frac{\sum_{\beta \in \Omega} \lambda_{\beta} \left(\nabla_{-}^{\mathbf{H}} \overline{D}_{\beta}(\mathbf{V}, \mathbf{WH}) \right)}{\sum_{\beta \in \Omega} \lambda_{\beta} \left(\nabla_{+}^{\mathbf{H}} \overline{D}_{\beta}(\mathbf{V}, \mathbf{WH}) \right)}.$$

- Sometimes this may not result in a decrease in the objective, so we set $\gamma = 1$ and $\mathbf{H}_1^{+} = \mathbf{H}^{+}$ and successively find γ such that while

$$\overline{D}_{\Omega}^{\boldsymbol{\lambda}}(\mathbf{V}, \mathbf{WH}_{\gamma}^{+}) > \overline{D}_{\Omega}^{\boldsymbol{\lambda}}(\mathbf{V}, \mathbf{WH})$$

we reduce

$$\gamma \longleftarrow \frac{\gamma}{2}$$

and set

$$\mathbf{H}_{\gamma}^{+} = (1 - \gamma)\mathbf{H} + \gamma \mathbf{H}^{+}.$$

Application of MU Algorithm to DR-NMF

- Update $\mathbf{H}$ as follows:

$$\mathbf{H}^{+} = \mathbf{H} \cdot \frac{\sum_{\beta\in\Omega} \lambda_\beta \left(\nabla^{\mathbf{H}}_{-} \overline{D}_\beta(\mathbf{V}, \mathbf{WH})\right)}{\sum_{\beta\in\Omega} \lambda_\beta \left(\nabla^{\mathbf{H}}_{+} \overline{D}_\beta(\mathbf{V}, \mathbf{WH})\right)}.$$

- Sometimes this may not result in a decrease in the objective, so we set $\gamma = 1$ and $\mathbf{H}_1^{+} = \mathbf{H}^{+}$ and successively find γ such that while

$$\overline{D}^{\boldsymbol{\lambda}}_{\Omega}(\mathbf{V}, \mathbf{WH}^{+}_{\gamma}) > \overline{D}^{\boldsymbol{\lambda}}_{\Omega}(\mathbf{V}, \mathbf{WH})$$

we reduce

$$\gamma \longleftarrow \frac{\gamma}{2}$$

and set

$$\mathbf{H}^{+}_{\gamma} = (1-\gamma)\mathbf{H} + \gamma\mathbf{H}^{+}.$$

- But this tweak of γ is rarely needed.

Algorithm for DR-NMF

- For fixed λ, we have an MU algorithm to solve

$$\min_{\mathbf{W},\mathbf{H}\geq\mathbf{0}} \overline{D}_{\Omega}^{\lambda}(\mathbf{V},\mathbf{WH}), \quad \text{where} \quad \overline{D}_{\Omega}^{\lambda}(\mathbf{V},\mathbf{WH}) = \sum_{\beta\in\Omega} \lambda_{\beta}\overline{D}_{\beta}(\mathbf{V},\mathbf{WH}).$$

Algorithm for DR-NMF

- For fixed λ, we have an MU algorithm to solve

$$\min_{\mathbf{W},\mathbf{H}\geq\mathbf{0}} \overline{D}_{\Omega}^{\lambda}(\mathbf{V},\mathbf{WH}), \quad \text{where} \quad \overline{D}_{\Omega}^{\lambda}(\mathbf{V},\mathbf{WH}) = \sum_{\beta\in\Omega} \lambda_{\beta}\overline{D}_{\beta}(\mathbf{V},\mathbf{WH}).$$

- But we want to solve for $\mathbf{W},\mathbf{H}\geq\mathbf{0}$ that minimizes

$$\max_{\beta\in\Omega} \overline{D}_{\beta}(\mathbf{V},\mathbf{WH}) = \max_{\boldsymbol{\lambda}\geq\mathbf{0}:\|\boldsymbol{\lambda}\|_1=1} \sum_{\beta\in\Omega} \lambda_{\beta}\overline{D}_{\beta}(\mathbf{V},\mathbf{WH}).$$

Algorithm for DR-NMF

- For fixed λ, we have an MU algorithm to solve

$$\min_{\mathbf{W},\mathbf{H}\geq\mathbf{0}} \overline{D}_{\Omega}^{\lambda}(\mathbf{V},\mathbf{WH}), \quad \text{where} \quad \overline{D}_{\Omega}^{\lambda}(\mathbf{V},\mathbf{WH}) = \sum_{\beta\in\Omega} \lambda_{\beta}\overline{D}_{\beta}(\mathbf{V},\mathbf{WH}).$$

- But we want to solve for $\mathbf{W},\mathbf{H}\geq\mathbf{0}$ that minimizes

$$\max_{\beta\in\Omega} \overline{D}_{\beta}(\mathbf{V},\mathbf{WH}) = \max_{\boldsymbol{\lambda}\geq\mathbf{0}:\|\boldsymbol{\lambda}\|_1=1} \sum_{\beta\in\Omega} \lambda_{\beta}\overline{D}_{\beta}(\mathbf{V},\mathbf{WH}).$$

- So we want to solve

$$\min_{\mathbf{W},\mathbf{H}\geq\mathbf{0}} \max_{\boldsymbol{\lambda}\geq\mathbf{0}:\|\boldsymbol{\lambda}\|_1=1} \sum_{\beta\in\Omega} \lambda_{\beta}\overline{D}_{\beta}(\mathbf{V},\mathbf{WH})$$

  which is a min-max optimization problem.

Algorithm for DR-NMF

- For fixed λ, we have an MU algorithm to solve

$$\min_{\mathbf{W},\mathbf{H}\geq\mathbf{0}} \overline{D}_{\Omega}^{\lambda}(\mathbf{V},\mathbf{WH}), \quad \text{where} \quad \overline{D}_{\Omega}^{\lambda}(\mathbf{V},\mathbf{WH}) = \sum_{\beta\in\Omega} \lambda_{\beta}\overline{D}_{\beta}(\mathbf{V},\mathbf{WH}).$$

- But we want to solve for $\mathbf{W},\mathbf{H}\geq\mathbf{0}$ that minimizes

$$\max_{\beta\in\Omega} \overline{D}_{\beta}(\mathbf{V},\mathbf{WH}) = \max_{\boldsymbol{\lambda}\geq\mathbf{0}:\|\boldsymbol{\lambda}\|_1=1} \sum_{\beta\in\Omega} \lambda_{\beta}\overline{D}_{\beta}(\mathbf{V},\mathbf{WH}).$$

- So we want to solve

$$\min_{\mathbf{W},\mathbf{H}\geq\mathbf{0}} \max_{\boldsymbol{\lambda}\geq\mathbf{0}:\|\boldsymbol{\lambda}\|_1=1} \sum_{\beta\in\Omega} \lambda_{\beta}\overline{D}_{\beta}(\mathbf{V},\mathbf{WH})$$

  which is a min-max optimization problem.

- There are dual subgradient methods to solve this with convergence guarantees, but we found them to be slow.

Frank–Wolfe-type Algorithm for DR-NMF

- Initialize $\lambda_\beta = 1/|\Omega|$ for all $\beta \in \Omega$.

Frank–Wolfe-type Algorithm for DR-NMF

- Initialize $\lambda_\beta = 1/|\Omega|$ for all $\beta \in \Omega$.
- For each $t = 1, 2, \ldots$, we obtain $\mathbf{H}^{(t+1)}$ using the MU algorithm with $\mathbf{W} = \mathbf{W}^{(t)}$ and $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{(t)}$.

Frank–Wolfe-type Algorithm for DR-NMF

- Initialize $\lambda_\beta = 1/|\Omega|$ for all $\beta \in \Omega$.
- For each $t = 1, 2, \ldots$, we obtain $\mathbf{H}^{(t+1)}$ using the MU algorithm with $\mathbf{W} = \mathbf{W}^{(t)}$ and $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{(t)}$.
- We obtain $\mathbf{W}^{(t+1)}$ using the MU algorithm with $\mathbf{H} = \mathbf{H}^{(t+1)}$ and $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{(t)}$.

Frank–Wolfe-type Algorithm for DR-NMF

- Initialize $\lambda_\beta = 1/|\Omega|$ for all $\beta \in \Omega$.
- For each $t = 1, 2, \ldots$, we obtain $\mathbf{H}^{(t+1)}$ using the MU algorithm with $\mathbf{W} = \mathbf{W}^{(t)}$ and $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{(t)}$.
- We obtain $\mathbf{W}^{(t+1)}$ using the MU algorithm with $\mathbf{H} = \mathbf{H}^{(t+1)}$ and $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{(t)}$.
- Let $\beta^* \in \arg\max_{\beta \in \Omega} \overline{D}_\beta(\mathbf{V}, \mathbf{W}^{(t+1)}\mathbf{H}^{(t+1)})$ and

$$[\lambda_*^{(t)}]_\beta = \begin{cases} 1 & \text{if } \beta = \beta^*, \\ 0 & \text{if } \beta \neq \beta^*. \end{cases}$$

Update

$$\boldsymbol{\lambda}^{(t+1)} = (1 - \rho_t)\boldsymbol{\lambda}^{(t)} + \rho_t \boldsymbol{\lambda}_*^{(t)},$$

where $\rho_t = 1/t$.

Frank–Wolfe-type Algorithm for DR-NMF

- Initialize $\lambda_\beta = 1/|\Omega|$ for all $\beta \in \Omega$.
- For each $t = 1, 2, \ldots$, we obtain $\mathbf{H}^{(t+1)}$ using the MU algorithm with $\mathbf{W} = \mathbf{W}^{(t)}$ and $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{(t)}$.
- We obtain $\mathbf{W}^{(t+1)}$ using the MU algorithm with $\mathbf{H} = \mathbf{H}^{(t+1)}$ and $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{(t)}$.
- Let $\beta^* \in \arg\max_{\beta \in \Omega} \overline{D}_\beta(\mathbf{V}, \mathbf{W}^{(t+1)}\mathbf{H}^{(t+1)})$ and

$$[\lambda_*^{(t)}]_\beta = \begin{cases} 1 & \text{if } \beta = \beta^*, \\ 0 & \text{if } \beta \neq \beta^*. \end{cases}$$

  Update

$$\boldsymbol{\lambda}^{(t+1)} = (1 - \rho_t)\boldsymbol{\lambda}^{(t)} + \rho_t \boldsymbol{\lambda}_*^{(t)},$$

  where $\rho_t = 1/t$.

- This is a Frank–Wolfe-type algorithm (FW would use $\rho_t = 2/(t+2)$).

Remarks on our Algorithm for DR-NMF

- Updates for $\mathbf{W}$ and $\mathbf{H}$ are meant to approximately minimize

$$(\mathbf{W}, \mathbf{H}) \mapsto \overline{D}_{\Omega}^{\boldsymbol{\lambda}^{(t)}}(\mathbf{V}, \mathbf{WH})$$

Remarks on our Algorithm for DR-NMF

- Updates for $\mathbf{W}$ and $\mathbf{H}$ are meant to approximately minimize

$$(\mathbf{W}, \mathbf{H}) \mapsto \overline{D}_{\Omega}^{\boldsymbol{\lambda}^{(t)}}(\mathbf{V}, \mathbf{WH})$$

- For the update of $\boldsymbol{\lambda}$, notice that for all $\beta \in \Omega$

$$\overline{D}_{\beta^*}(\mathbf{V}, \mathbf{W}^{(t+1)}\mathbf{H}^{(t+1)}) \geq \overline{D}_{\beta}(\mathbf{V}, \mathbf{W}^{(t+1)}\mathbf{H}^{(t+1)}),$$

and since $\boldsymbol{\lambda} \mapsto \overline{D}_{\beta}^{\boldsymbol{\lambda}}$ is linear, we have

$$\boldsymbol{\lambda}_*^{(t)} = \arg\max \left\{ \overline{D}_{\beta}^{\boldsymbol{\lambda}}(\mathbf{V}, \mathbf{W}^{(t+1)}\mathbf{H}^{(t+1)}) : \boldsymbol{\lambda} \geq \mathbf{0}, \|\boldsymbol{\lambda}\|_1 = 1 \right\}.$$

Remarks on our Algorithm for DR-NMF

- Updates for $\mathbf{W}$ and $\mathbf{H}$ are meant to approximately minimize

$$(\mathbf{W}, \mathbf{H}) \mapsto \overline{D}_{\Omega}^{\boldsymbol{\lambda}^{(t)}}(\mathbf{V}, \mathbf{WH})$$

- For the update of $\boldsymbol{\lambda}$, notice that for all $\beta \in \Omega$

$$\overline{D}_{\beta^*}(\mathbf{V}, \mathbf{W}^{(t+1)}\mathbf{H}^{(t+1)}) \geq \overline{D}_{\beta}(\mathbf{V}, \mathbf{W}^{(t+1)}\mathbf{H}^{(t+1)}),$$

and since $\boldsymbol{\lambda} \mapsto \overline{D}_{\beta}^{\boldsymbol{\lambda}}$ is linear, we have

$$\boldsymbol{\lambda}_*^{(t)} = \arg\max \left\{ \overline{D}_{\beta}^{\boldsymbol{\lambda}}(\mathbf{V}, \mathbf{W}^{(t+1)}\mathbf{H}^{(t+1)}) : \boldsymbol{\lambda} \geq \mathbf{0}, \|\boldsymbol{\lambda}\|_1 = 1 \right\}.$$

- The β^*-divergence is given the most importance at the next iteration

Remarks on our Algorithm for DR-NMF

- Updates for $\mathbf{W}$ and $\mathbf{H}$ are meant to approximately minimize

$$(\mathbf{W}, \mathbf{H}) \mapsto \overline{D}_{\Omega}^{\boldsymbol{\lambda}^{(t)}}(\mathbf{V}, \mathbf{WH})$$

- For the update of $\boldsymbol{\lambda}$, notice that for all $\beta \in \Omega$

$$\overline{D}_{\beta^*}(\mathbf{V}, \mathbf{W}^{(t+1)}\mathbf{H}^{(t+1)}) \geq \overline{D}_{\beta}(\mathbf{V}, \mathbf{W}^{(t+1)}\mathbf{H}^{(t+1)}),$$

and since $\boldsymbol{\lambda} \mapsto \overline{D}_{\beta}^{\boldsymbol{\lambda}}$ is linear, we have

$$\boldsymbol{\lambda}_*^{(t)} = \arg\max \left\{ \overline{D}_{\beta}^{\boldsymbol{\lambda}}(\mathbf{V}, \mathbf{W}^{(t+1)}\mathbf{H}^{(t+1)}) : \boldsymbol{\lambda} \geq \mathbf{0}, \|\boldsymbol{\lambda}\|_1 = 1 \right\}.$$

- The β^*-divergence is given the most importance at the next iteration
- Forcing all β-divergences to decrease as well.

Sparse Document Data Sets

- For sparse data sets, one often chooses $\beta \in \Omega = \{1, 2\}$

Sparse Document Data Sets

- For sparse data sets, one often chooses $\beta \in \Omega = \{1, 2\}$
- For sparse word-count datasets, Poisson noise is the most appropriate

Sparse Document Data Sets

- For sparse data sets, one often chooses $\beta \in \Omega = \{1, 2\}$
- For sparse word-count datasets, Poisson noise is the most appropriate
- But say we do not know this, we can compare DR-NMF, KL-NMF and Frobenius-NMF

Sparse Document Data Sets

- For sparse data sets, one often chooses $\beta \in \Omega = \{1, 2\}$
- For sparse word-count datasets, Poisson noise is the most appropriate
- But say we do not know this, we can compare DR-NMF, KL-NMF and Frobenius-NMF
- Use these NMF methods for clustering (topic modeling)

Sparse Document Data Sets

- For sparse data sets, one often chooses $\beta \in \Omega = \{1, 2\}$
- For sparse word-count datasets, Poisson noise is the most appropriate
- But say we do not know this, we can compare DR-NMF, KL-NMF and Frobenius-NMF
- Use these NMF methods for clustering (topic modeling)
- Clustering accuracy

$$\text{accuracy}(\{\tilde{C}_i\}_{i=1}^r) := \min_{\pi:[r]\to[r]} \frac{1}{r} \sum_{i=1}^{r} \left| C_i \cap \tilde{C}_{\pi(i)} \right|$$

Sparse Document Data Sets

data set	number of classes	Clustering accuracy (%) KL-NMF	Fro-NMF	DR-NMF
NG20	20	**50.15**	17.78	27.60
NG3SIM	3	59.07	34.29	**68.05**
classic	4	**65.53**	49.21	58.98
ohscal	10	**41.54**	35.71	40.23
k1b	6	54.40	**73.50**	62.35
hitech	6	41.03	**48.28**	41.68
reviews	5	**78.10**	45.24	75.33
sports	7	53.48	49.24	**62.60**
la1	6	**70.69**	45.47	66.67
la12	6	**71.24**	47.91	67.75
la2	6	**70.34**	51.58	68.62
tr11	9	**52.90**	46.38	46.62
tr23	6	30.39	**39.71**	34.80
tr41	10	**60.25**	35.31	49.20
tr45	10	**56.67**	38.12	31.59
Average		57.05	43.85	53.47

Figure: Clustering accuracies of various methods

Dense Time-Frequency Matrices of Audio Signals

- Use the data set piano_Mary

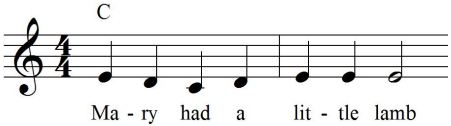


Figure: Musical score of "Mary had a little lamb". The notes are activated as follows: E_4, D_4, C_4, D_4, E_4, E_4, E_4.

Dense Time-Frequency Matrices of Audio Signals

- Use the data set piano_Mary

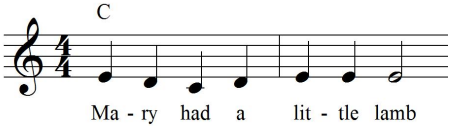


Figure: Musical score of "Mary had a little lamb". The notes are activated as follows: E_4, D_4, C_4, D_4, E_4, E_4, E_4.

- Considered no added noise and adding Poisson noise to the music piece

Dense Time-Frequency Matrices of Audio Signals

- Use the data set piano_Mary

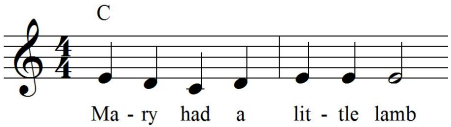


Figure: Musical score of "Mary had a little lamb". The notes are activated as follows: E_4, D_4, C_4, D_4, E_4, E_4, E_4.

- Considered no added noise and adding Poisson noise to the music piece
- Tested in DR-NMF (with $\Omega = \{0, 1\}$), IS-NMF ($\beta = 0$) and KL-NMF ($\beta = 1$)

No Added Noise

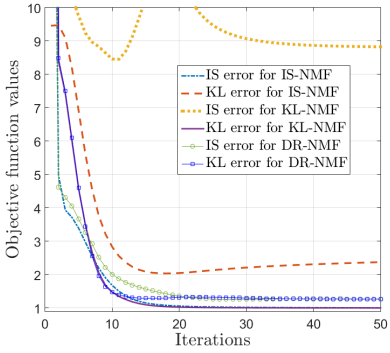


Figure: Evolution of scaled β-divergences

No Added Noise

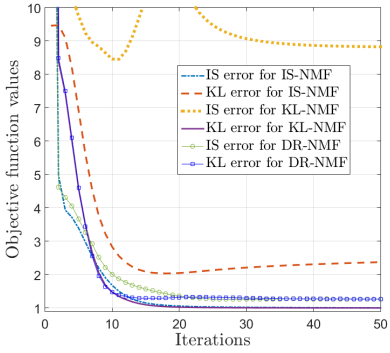


Figure: Evolution of scaled β-divergences

- DR-NMF is able to compute a model with low IS- and KL-error

No Added Noise

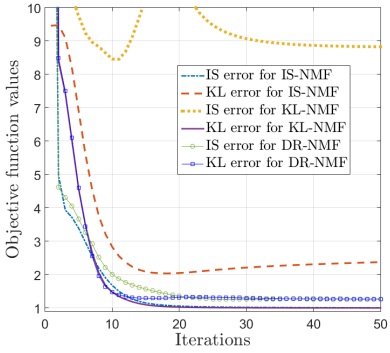


Figure: Evolution of scaled β-divergences

- DR-NMF is able to compute a model with low IS- and KL-error
- KL-NMF has IS-error 9 times that of IS-NMF

Added Poisson Noise

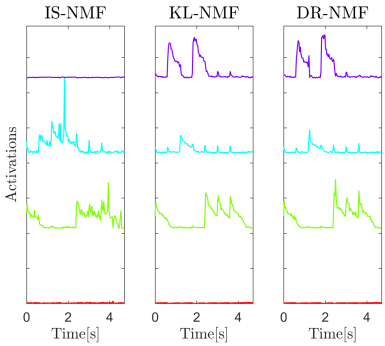


Figure: IS-NMF, KL-NMF, and DR-NMF with $\Omega = \{0, 1\}$ in Poisson noise.

Added Poisson Noise

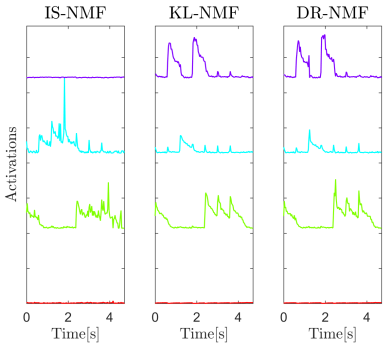


Figure: IS-NMF, KL-NMF, and DR-NMF with $\Omega = \{0, 1\}$ in Poisson noise.

- Rows of $\mathbf{H}$ are recovered successfully.
- C_4 is activated once, D_4 twice and E_4 four times.

Outline

Using Nonnegative Matrix Factorization in Ranking Models for Sports Analytics

(Xia, Tan, Filstroff, and Févotte, 2019)

Using Nonnegative Matrix Factorization in Ranking Models for Sports Analytics

(Xia, Tan, Filstroff, and Févotte, 2019)

Rod Laver
3 Grand Slam

Roger Federer
3 Grand Slam

Novak Djokovic
5 Grand Slam

1960s

2005 2006

2010 2011

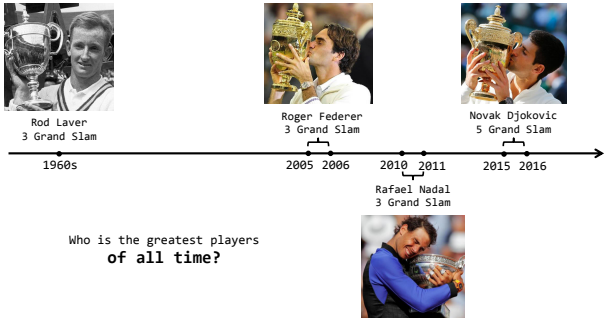
2015 2016

Rafael Nadal
3 Grand Slam

Who is the greatest players
of all time?

Using Nonnegative Matrix Factorization in Ranking Models for Sports Analytics

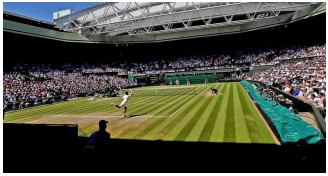
(Xia, Tan, Filstroff, and Févotte, 2019)



Who is the greatest of all time (GOAT)?

What could be a Pertinent Latent Variable?

What could be a Pertinent Latent Variable?



Wimbledon
Grass Outdoors

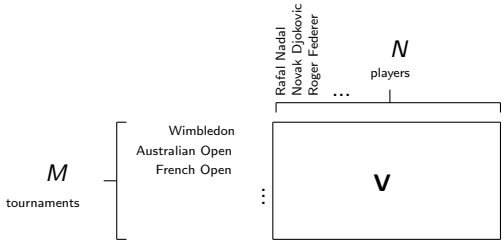
Australian Open
Hard Outdoors

French Open
Clay Outdoors

US Open
Hard Outdoors

Ranking Tennis Players with Latent Variables

Ranking Tennis Players with Latent Variables



Ranking Tennis Players with Latent Variables

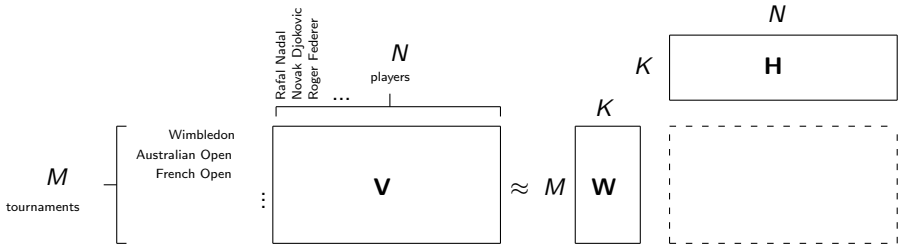


Figure: The hybrid BTL-NMF Model

Ranking Tennis Players with Latent Variables

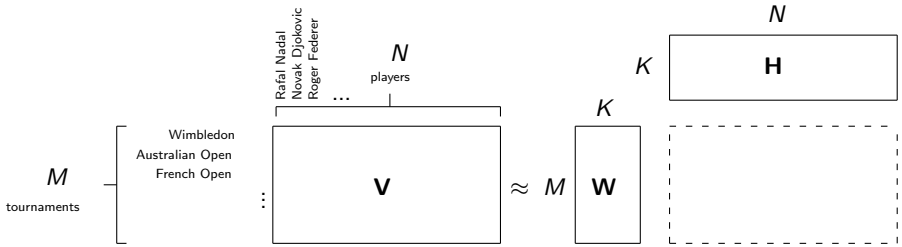


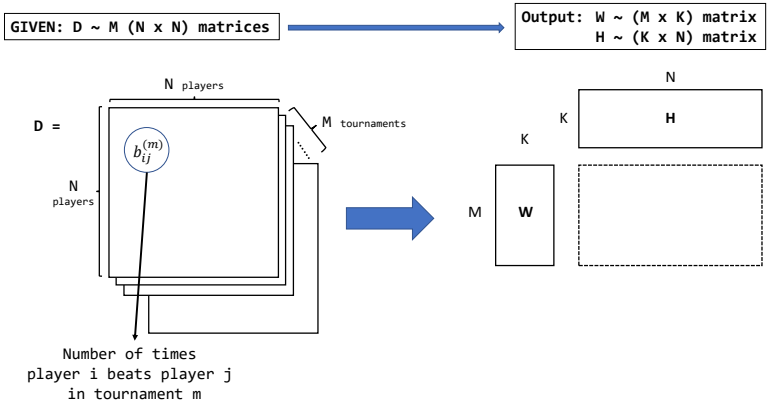
Figure: The hybrid BTL-NMF Model

- Bradley–Terry–Luce (Bradley and Terry, 1952; Luce, 1959) ranking model:

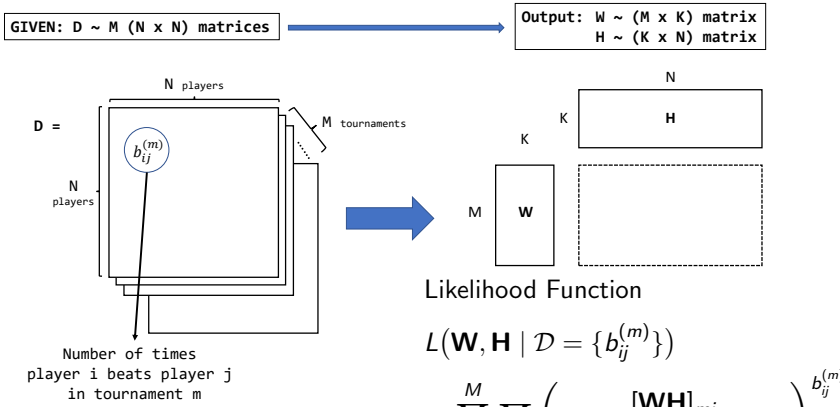
$$\Pr\left(\text{player } i \text{ beats player } j \mid \text{tournament } m\right) = \frac{\lambda_{mi}}{\lambda_{mi} + \lambda_{mj}}$$

- λ_{mi}: Skill level of player i in tournament m.

Data Collected and Likelihood Function



Data Collected and Likelihood Function



Likelihood Function

$$L(\mathbf{W}, \mathbf{H} \mid \mathcal{D} = \{b_{ij}^{(m)}\})$$

$$= \prod_{m=1}^{M} \prod_{(i,j)} \left(\frac{[\mathbf{WH}]_{mi}}{\underbrace{[\mathbf{WH}]_{mi} + [\mathbf{WH}]_{mj}}_{\text{Prob. } i \text{ beats } j \text{ in tourn. } m}} \right)^{b_{ij}^{(m)}}$$

Objective Function to be Minimized

- Take the negative log of the likelihood to get the following objective function

$$\begin{aligned}
&\arg\min_{\mathbf{W},\mathbf{H}\geq\mathbf{0}} f(\mathbf{W},\mathbf{H} \mid \mathcal{D}) = -\log L\big(\mathbf{W},\mathbf{H} \mid \mathcal{D}\big) \\
\equiv &\arg\min_{\mathbf{W},\mathbf{H}\geq\mathbf{0}} \sum_{m=1}^{M} \sum_{(i,j)\in\mathcal{P}_m} b_{ij}^{(m)} \left[-\log([\mathbf{WH}]_{mi}) + \log([\mathbf{WH}]_{mi} + [\mathbf{WH}]_{mj})\right],
\end{aligned}$$

where $\mathcal{P}_m$ is the set of games that i and j played in tournament m.

Objective Function to be Minimized

- Take the negative log of the likelihood to get the following objective function

$$\begin{aligned}
&\arg\min_{\mathbf{W},\mathbf{H}\geq\mathbf{0}} f(\mathbf{W},\mathbf{H} \mid \mathcal{D}) = -\log L\big(\mathbf{W},\mathbf{H} \mid \mathcal{D}\big) \\
\equiv &\arg\min_{\mathbf{W},\mathbf{H}\geq\mathbf{0}} \sum_{m=1}^{M} \sum_{(i,j)\in\mathcal{P}_m} b_{ij}^{(m)} \left[-\log([\mathbf{WH}]_{mi}) + \log([\mathbf{WH}]_{mi} + [\mathbf{WH}]_{mj})\right],
\end{aligned}$$

  where $\mathcal{P}_m$ is the set of games that i and j played in tournament m.

- Unfortunately, this objective function is not convex in $(\mathbf{W},\mathbf{H})$.

Objective Function to be Minimized

- Take the negative log of the likelihood to get the following objective function

$$\begin{aligned}
&\arg\min_{\mathbf{W},\mathbf{H}\geq\mathbf{0}} f(\mathbf{W},\mathbf{H} \mid \mathcal{D}) = -\log L\big(\mathbf{W},\mathbf{H} \mid \mathcal{D}\big) \\
\equiv &\arg\min_{\mathbf{W},\mathbf{H}\geq\mathbf{0}} \sum_{m=1}^{M} \sum_{(i,j)\in\mathcal{P}_m} b_{ij}^{(m)} \left[-\log([\mathbf{WH}]_{mi}) + \log([\mathbf{WH}]_{mi} + [\mathbf{WH}]_{mj})\right],
\end{aligned}$$

  where $\mathcal{P}_m$ is the set of games that i and j played in tournament m.

- Unfortunately, this objective function is not convex in $(\mathbf{W},\mathbf{H})$.
- Majorization-Minimization (MM) comes to the rescue again!

Objective Function to be Minimized

- Take the negative log of the likelihood to get the following objective function

$$\begin{aligned}&\underset{\mathbf{W},\mathbf{H}\geq\mathbf{0}}{\arg\min}\, f(\mathbf{W},\mathbf{H}\mid\mathcal{D}) = -\log L\big(\mathbf{W},\mathbf{H}\mid\mathcal{D}\big)\\ \equiv &\underset{\mathbf{W},\mathbf{H}\geq\mathbf{0}}{\arg\min}\sum_{m=1}^{M}\sum_{(i,j)\in\mathcal{P}_m} b_{ij}^{(m)}\left[-\log([\mathbf{WH}]_{mi}) + \log([\mathbf{WH}]_{mi}+[\mathbf{WH}]_{mj})\right],\end{aligned}$$

  where $\mathcal{P}_m$ is the set of games that i and j played in tournament m.

- Unfortunately, this objective function is not convex in $(\mathbf{W},\mathbf{H})$.
- Majorization-Minimization (MM) comes to the rescue again!
- Main ideas: For any concave function g (tangent inequality),

$$g(\mathbf{y}) \leq g(\mathbf{x}) + \nabla g(\mathbf{x})^\top(\mathbf{y}-\mathbf{x})$$

  and Jensen's inequality for the convex function $t \mapsto -\log t$.

Majorization-Minimization Updates

- After some straightforward but tedious algebra, we can construct two auxiliary functions $u_1(\mathbf{W}, \tilde{\mathbf{W}} \mid \mathbf{H})$ and $u_2(\mathbf{H}, \tilde{\mathbf{H}} \mid \mathbf{W})$ that majorize the objective function

$$f(\mathbf{W}, \mathbf{H} \mid \mathcal{D}) = -\log L\big(\mathbf{W}, \mathbf{H} \mid \mathcal{D}\big).$$

Majorization-Minimization Updates

- After some straightforward but tedious algebra, we can construct two auxiliary functions $u_1(\mathbf{W}, \tilde{\mathbf{W}} \mid \mathbf{H})$ and $u_2(\mathbf{H}, \tilde{\mathbf{H}} \mid \mathbf{W})$ that majorize the objective function

$$f(\mathbf{W}, \mathbf{H} \mid \mathcal{D}) = -\log L\big(\mathbf{W}, \mathbf{H} \mid \mathcal{D}\big).$$

- Implement

$$\mathbf{W}^{(t+1)} = \underset{\mathbf{W} \geq \mathbf{0}}{\arg\min}\, u_1(\mathbf{W}, \mathbf{W}^{(t)} \mid \mathbf{H}^{(t)})$$
$$\mathbf{H}^{(t+1)} = \underset{\mathbf{H} \geq \mathbf{0}}{\arg\min}\, u_2(\mathbf{H}, \mathbf{H}^{(t)} \mid \mathbf{W}^{(t+1)})$$

Majorization-Minimization Updates

- Update for $\mathbf{W}$:

$$w_{mk} \longleftarrow \frac{\sum_{(i,j)\in\mathcal{P}_m} b_{ij}^{(m)} \frac{w_{mk} h_{ki}}{[\mathbf{WH}]_{mi}}}{\sum_{(i,j)\in\mathcal{P}_m} b_{ij}^{(m)} \frac{h_{ki}+h_{kj}}{[\mathbf{WH}]_{mi}+[\mathbf{WH}]_{mj}}}.$$

Majorization-Minimization Updates

- Update for $\mathbf{W}$:

$$w_{mk} \longleftarrow \frac{\sum\limits_{(i,j)\in\mathcal{P}_m} b_{ij}^{(m)} \frac{w_{mk}h_{ki}}{[\mathbf{WH}]_{mi}}}{\sum\limits_{(i,j)\in\mathcal{P}_m} b_{ij}^{(m)} \frac{h_{ki}+h_{kj}}{[\mathbf{WH}]_{mi}+[\mathbf{WH}]_{mj}}}.$$

- Update for $\mathbf{H}$:

$$h_{ki} \longleftarrow \frac{\sum\limits_{m}\sum\limits_{j\neq i:(i,j)\in\mathcal{P}_m} b_{ij}^{(m)} \frac{w_{mk}h_{ki}}{[\mathbf{WH}]_{mi}}}{\sum\limits_{m}\sum\limits_{j\neq i:(i,j)\in\mathcal{P}_m} \left(b_{ij}^{(m)}+b_{ji}^{(m)}\right) \frac{w_{mk}}{[\mathbf{WH}]_{mi}+[\mathbf{WH}]_{mj}}}.$$

Majorization-Minimization Updates

- Update for $\mathbf{W}$:

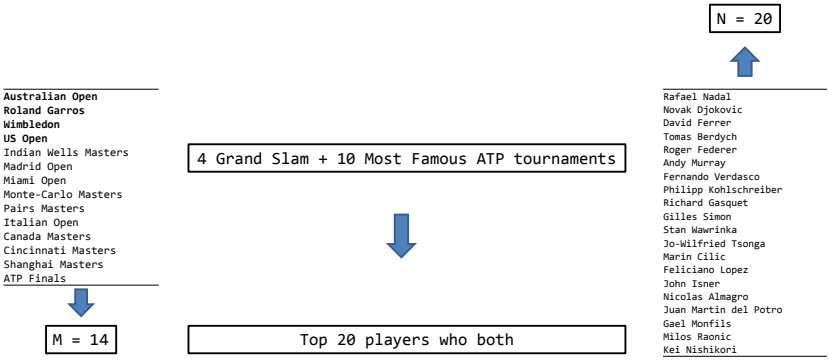
$$w_{mk} \longleftarrow \frac{\sum\limits_{(i,j)\in\mathcal{P}_m} b_{ij}^{(m)} \frac{w_{mk}h_{ki}}{[\mathbf{WH}]_{mi}}}{\sum\limits_{(i,j)\in\mathcal{P}_m} b_{ij}^{(m)} \frac{h_{ki}+h_{kj}}{[\mathbf{WH}]_{mi}+[\mathbf{WH}]_{mj}}}.$$

- Update for $\mathbf{H}$:

$$h_{ki} \longleftarrow \frac{\sum\limits_{m}\sum\limits_{j\neq i:(i,j)\in\mathcal{P}_m} b_{ij}^{(m)} \frac{w_{mk}h_{ki}}{[\mathbf{WH}]_{mi}}}{\sum\limits_{m}\sum\limits_{j\neq i:(i,j)\in\mathcal{P}_m} \left(b_{ij}^{(m)}+b_{ji}^{(m)}\right) \frac{w_{mk}}{[\mathbf{WH}]_{mi}+[\mathbf{WH}]_{mj}}}.$$

- Simple, fuss-free updates.
- Used a few other hacks to ensure normalization and no divide by 0 errors.
- Under the right conditions, can prove convergence guarantees to "stationary points" (Zhao and Tan, 2018).

Data Collection



Data Collection

Name	Rafael Nadal	Novak Djokovic	David Ferrer	Tomas Berdych	Roger Federer	Andy Murray	Fernando Verdasco	Philipp Kohlschreiber	Richard Gasquet
Rafael Nadal	0	0	0	1	3	1	1	2	0
Novak Djokovic	1	0	3	2	3	5	2	0	0
David Ferrer	1	0	0	0	0	0	0	0	1
Tomas Berdych	1	0	0	0	0	0	1	1	1
Roger Federer	1	1	0	4	0	2	0	0	0
Andy Murray	1	0	2	1	1	0	1	0	0
Fernando Verdasco	1	0	0	0	0	1	0	0	0
Philipp Kohlschreiber	0	0	0	0	0	0	0	0	0
Richard Gasquet	0	0	0	0	0	0	0	0	0

Zeros on the diagonal
$(b_{ii}^{(m)} = 0)$

Non-zero

True zeros
$(b_{ij}^{(m)} = 0, b_{ji}^{(m)} > 0)$

Missing data
$(b_{ij}^{(m)} = b_{ji}^{(m)} = 0)$

	Male	
Total Entries	$14 \times 20 \times 20 = 5600$	
	Number	Percentage
Non-zero	1024	18.30%
Zeros on the diagonal	280	5.00%
Missing data	3478	62.10%
True zeros	818	14.60%

Results on Tournaments for Men's Dataset

Tournaments	Row Normalization		Column Normalization		
	non-clay	clay	non-clay	clay	
Australian Open	5.77E-01	4.23E-01	1.15E-01	7.66E-02	
French Open	3.44E-01	6.56E-01	8.66E-02	1.50E-01	1
Wimbledon	6.43E-01	3.57E-01	6.73E-02	3.38E-02	
US Open	5.07E-01	4.93E-01	4.62E-02	4.06E-02	
Indian Wells Masters	6.52E-01	3.48E-01	1.34E-01	6.50E-02	
Madrid Open	3.02E-01	6.98E-01	6.43E-02	1.34E-01	3
Miami Open	5.27E-01	4.73E-01	4.95E-02	4.02E-02	
Monte-Carlo Masters	1.68E-01	8.32E-01	2.24E-02	1.01E-01	4
Paris Masters	1.68E-01	8.32E-01	1.29E-02	5.76E-02	
Italian Open	0.00E-00	1.00E-00	1.82E-104	1.36E-01	2
Canadian Open	1.00E-00	0.00E-00	1.28E-01	1.78E-152	
Cincinnati Masters	5.23E-01	4.77E-01	1.13E-01	9.36E-02	
Shanghai Masters	7.16E-01	2.84E-01	1.13E-01	4.07E-02	
The ATP Finals	5.72E-01	4.28E-01	4.59E-02	3.11E-02	

Latent variable discovered to be "surface type"

Results on Player Rankings by Latent Variable

	Players	non-clay (matrix $\mathbf{H}^T$)	clay (matrix $\mathbf{H}^T$)	Total Matches
Hard Court player →	Novak Djokovic	1.20E-01	9.98E-02	283
Clay player →	Rafael Nadal	2.48E-02	1.55E-01	241
Grass player →	Roger Federer	1.15E-01	2.34E-02	229
Non-clay player →	Andy Murray	7.57E-02	8.43E-03	209
	Tomas Berdych	0.00E-00	3.02E-02	154
	David Ferrer	6.26E-40	3.27E-02	147
Clay player →	Stan Wawrinka	2.93E-55	4.08E-02	141
	Jo-Wilfried Tsonga	3.36E-02	2.71E-03	121
	Richard Gasquet	5.49E-03	1.41E-02	102
	Juan Martin del Potro	2.90E-02	1.43E-02	101
	Marin Cilic	2.12E-02	0.00E-00	100
	Fernando Verdasco	1.36E-02	8.79E-03	96
	Kei Nishikori	7.07E-03	2.54E-02	94
	Gilles Simon	1.32E-02	4.59E-03	83
	Milos Raonic	1.45E-02	7.25E-03	78
	Philipp Kohlschreiber	2.18E-06	5.35E-03	76
	John Isner	2.70E-03	1.43E-02	78
	Feliciano Lopez	1.43E-02	3.31E-03	75
	Gael Monfils	3.86E-21	1.33E-02	70
	Nicolas Almagro	6.48E-03	6.33E-06	60

Figure: Players rankings according to discovered latent variable – "surface type"

Results on Player Rankings by Tournament

Tournament	Novak Djokovic	Rafael Nadal	Roger Federer	Andy Murray	Stan Wawrinka
Australian Open	2.16E-02	1.54E-02	1.47E-02	9.13E-03	3.34E-03
French Open	1.39E-02	1.43E-02	7.12E-03	4.11E-03	3.48E-03
Wimbledon	2.63E-02	1.66E-02	1.91E-02	1.20E-02	3.39E-03
US Open	1.17E-02	9.42E-03	7.38E-03	4.51E-03	2.13E-03
Indian Wells Masters	2.29E-02	1.42E-02	1.68E-02	1.06E-02	2.88E-03
Madrid Open	1.38E-02	1.51E-02	6.63E-03	3.75E-03	3.72E-03
Miami Open	2.95E-02	2.30E-02	1.90E-02	1.17E-02	5.15E-03
Monte-Carlo Masters	1.19E-02	1.53E-02	4.46E-03	2.27E-03	3.92E-03
Paris Masters	7.29E-03	9.37E-03	2.73E-03	1.39E-03	2.40E-03
Italian Open	1.19E-02	1.84E-02	2.78E-03	1.00E-03	4.87E-03
Canadian Open	1.16E-02	2.40E-03	1.11E-02	7.32E-03	2.42E-51
Cincinnati Masters	1.82E-02	1.43E-02	1.17E-02	7.17E-03	3.20E-03
Shanghai Masters	8.12E-03	4.38E-03	6.29E-03	4.01E-03	8.24E-04
The ATP Finals	1.13E-02	8.13E-03	7.63E-03	4.74E-03	1.77E-03

Figure: Players' skill levels according to tournaments

Concluding Remarks for NMF for Sports Analytics

- Developed a statistical model that is a hybrid between the BTL ranking/pairwise comparison model

$$\Pr\left(\text{player } i \text{ beats player } j \mid \text{tournament } m\right) = \frac{\lambda_{mi}}{\lambda_{mi} + \lambda_{mj}}$$

and nonnegative matrix factorization (NMF).

Concluding Remarks for NMF for Sports Analytics

- Developed a statistical model that is a hybrid between the BTL ranking/pairwise comparison model

$$\Pr\left(\text{player } i \text{ beats player } j \mid \text{tournament } m\right) = \frac{\lambda_{mi}}{\lambda_{mi} + \lambda_{mj}}$$

  and nonnegative matrix factorization (NMF).

- Developed simple update rules based on MM that are easy to implement and have convergence guarantees.

Concluding Remarks for NMF for Sports Analytics

- Developed a statistical model that is a hybrid between the BTL ranking/pairwise comparison model

$$\Pr\left(\text{player } i \text{ beats player } j \mid \text{tournament } m\right) = \frac{\lambda_{mi}}{\lambda_{mi} + \lambda_{mj}}$$

  and nonnegative matrix factorization (NMF).

- Developed simple update rules based on MM that are easy to implement and have convergence guarantees.

- Confirms our intuition that court surface is a pertinent latent variable.

Concluding Remarks for NMF for Sports Analytics

- Developed a statistical model that is a hybrid between the BTL ranking/pairwise comparison model

$$\Pr\left(\text{player } i \text{ beats player } j \mid \text{tournament } m\right) = \frac{\lambda_{mi}}{\lambda_{mi} + \lambda_{mj}}$$

  and nonnegative matrix factorization (NMF).

- Developed simple update rules based on MM that are easy to implement and have convergence guarantees.

- Confirms our intuition that court surface is a pertinent latent variable.

- Ranked players according to the discovered latent variable (court surface) over a time window of 10 years.

Outline

Positive Semidefinite Matrix Factorization (PSDMF)

(Fiorini, Massar, Pokutta, Tiwary, and Wolf, 2012; Gouveia, Parrilo, and Thomas, 2013)

Positive Semidefinite Matrix Factorization (PSDMF)

(Fiorini, Massar, Pokutta, Tiwary, and Wolf, 2012; Gouveia, Parrilo, and Thomas, 2013)

- Given a nonnegative matrix $\mathbf{V} \in \mathbb{R}_+^{F \times N}$, find (symmetric) $K \times K$ positive semidefinite (PSD) matrices $\mathbf{W}_f, f = 1, \ldots, F$ and $\mathbf{H}_n, n = 1, \ldots, N$ such that

$$v_{fn} = [\mathbf{V}]_{fn} = \underbrace{\langle \mathbf{W}_f, \mathbf{H}_n \rangle}_{\text{matrix inner product}} = \underbrace{\operatorname{Tr}(\mathbf{W}_f \mathbf{H}_n)}_{\text{trace}}.$$

Positive Semidefinite Matrix Factorization (PSDMF)

(Fiorini, Massar, Pokutta, Tiwary, and Wolf, 2012; Gouveia, Parrilo, and Thomas, 2013)

- Given a nonnegative matrix $\mathbf{V} \in \mathbb{R}_+^{F \times N}$, find (symmetric) $K \times K$ positive semidefinite (PSD) matrices $\mathbf{W}_f, f = 1, \ldots, F$ and $\mathbf{H}_n, n = 1, \ldots, N$ such that

$$v_{fn} = [\mathbf{V}]_{fn} = \underbrace{\langle \mathbf{W}_f, \mathbf{H}_n \rangle}_{\text{matrix inner product}} = \underbrace{\operatorname{Tr}(\mathbf{W}_f \mathbf{H}_n)}_{\text{trace}}.$$

- The PSD rank of $\mathbf{V}$ is smallest K such that $\mathbf{V}$ admits an exact PSD factorization.

Positive Semidefinite Matrix Factorization (PSDMF)

(Fiorini, Massar, Pokutta, Tiwary, and Wolf, 2012; Gouveia, Parrilo, and Thomas, 2013)

- Given a nonnegative matrix $\mathbf{V} \in \mathbb{R}_+^{F \times N}$, find (symmetric) $K \times K$ positive semidefinite (PSD) matrices $\mathbf{W}_f, f = 1, \ldots, F$ and $\mathbf{H}_n, n = 1, \ldots, N$ such that

$$v_{fn} = [\mathbf{V}]_{fn} = \underbrace{\langle \mathbf{W}_f, \mathbf{H}_n \rangle}_{\text{matrix inner product}} = \underbrace{\operatorname{Tr}(\mathbf{W}_f \mathbf{H}_n)}_{\text{trace}}.$$

- The PSD rank of $\mathbf{V}$ is smallest K such that $\mathbf{V}$ admits an exact PSD factorization.

- If $\{\mathbf{W}_f\}$ and $\{\mathbf{H}_n\}$ are diagonal, let

$$\mathbf{w}_f = \operatorname{diag}(\mathbf{W}_f) \in \mathbb{R}_+^K, \quad \text{and} \quad \mathbf{h}_n = \operatorname{diag}(\mathbf{H}_n) \in \mathbb{R}_+^K,$$

then

$$v_{fn} = [\mathbf{V}]_{fn} = \underbrace{\langle \mathbf{w}_f, \mathbf{h}_n \rangle}_{\text{vector inner product}} = \sum_k w_{fk} h_{kn}.$$

PSDMF reduces to NMF!

PSDMF and PSD Rank

- Extension linking NMF with geometric and linear constraints in linear programming (Yannakakis, 1991)

PSDMF and PSD Rank

- Extension linking NMF with geometric and linear constraints in linear programming (Yannakakis, 1991)
- The smallest number K such that a polytope can be written as a projection (a "shadow") of a spectrahedron of size K (an affine slice of the cone of $K \times K$ positive semidefnite matrices $\mathbb{S}_+^K$) is equal to the PSD rank of a slack matrix of the original polytope.

PSDMF and PSD Rank

- Extension linking NMF with geometric and linear constraints in linear programming (Yannakakis, 1991)
- The smallest number K such that a polytope can be written as a projection (a "shadow") of a spectrahedron of size K (an affine slice of the cone of $K \times K$ positive semidefnite matrices $\mathbb{S}^K_+$) is equal to the PSD rank of a slack matrix of the original polytope.
- Example: Slack matrix of the square.

$$S_4 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix},$$

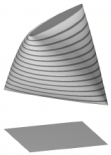


Figure: From Averkov et al. (2018)

$\text{rank}(S_4) = 3$, $\text{nn-rank}(S_4) = 4$ and $\text{psd-rank}(S_4) = 3$, a spectrahedron in $\mathbb{S}^3_+$.

Other Motivations for PSDMF

- Of fundamental importance in various fields:
  - Combinatorial optimization (Gouveia et al., 2013; Fawzi et al., 2015);
  - Quantum information theory (Fiorini et al., 2012; Fawzi et al., 2015);
  - Quantum communications and quantum computing (Jain et al., 2013; van Apeldoorn et al., 2020);
  - Probabilistic modeling (Glasser et al., 2019);
  - Quantum-based recommendation systems (Stark, 2016).

Other Motivations for PSDMF

- Of fundamental importance in various fields:
  - Combinatorial optimization (Gouveia et al., 2013; Fawzi et al., 2015);
  - Quantum information theory (Fiorini et al., 2012; Fawzi et al., 2015);
  - Quantum communications and quantum computing (Jain et al., 2013; van Apeldoorn et al., 2020);
  - Probabilistic modeling (Glasser et al., 2019);
  - Quantum-based recommendation systems (Stark, 2016).

- Connection to quantum is because quantum measurements $\{\mathbf{M}_i\}$, known as positve operator valued measures (POVMs) are PSD and sum to the identity

$$\sum_i \mathbf{M}_i = \mathbf{I}.$$

Other Motivations for PSDMF

- Of fundamental importance in various fields:
  - Combinatorial optimization (Gouveia et al., 2013; Fawzi et al., 2015);
  - Quantum information theory (Fiorini et al., 2012; Fawzi et al., 2015);
  - Quantum communications and quantum computing (Jain et al., 2013; van Apeldoorn et al., 2020);
  - Probabilistic modeling (Glasser et al., 2019);
  - Quantum-based recommendation systems (Stark, 2016).

- Connection to quantum is because quantum measurements $\{\mathbf{M}_i\}$, known as positve operator valued measures (POVMs) are PSD and sum to the identity

$$\sum_i \mathbf{M}_i = \mathbf{I}.$$

- We are mainly concerned with algorithms and approximate factorization.

Objective Function

- Consider the PSMDF model

$$v_{fn} = [\mathbf{V}]_{fn} = \langle \mathbf{W}_f, \mathbf{H}_n \rangle = \mathrm{Tr}(\mathbf{W}_f \mathbf{H}_n)$$

Objective Function

- Consider the PSMDF model

$$v_{fn} = [\mathbf{V}]_{fn} = \langle \mathbf{W}_f, \mathbf{H}_n \rangle = \mathrm{Tr}(\mathbf{W}_f \mathbf{H}_n)$$

- PSD matrices $\{\mathbf{W}_f\}_{f=1}^F$ and $\{\mathbf{H}_n\}_{n=1}^N$ can be estimated by minimizing a quadratic objective function (Stark, 2016; Vandaele et al., 2018):

$$g(\{\mathbf{W}_f\}_{f=1}^F, \{\mathbf{H}_n\}_{n=1}^N) = \frac{1}{2} \sum_{f,n} \left(v_{fn} - \mathrm{Tr}(\mathbf{W}_f \mathbf{H}_n)\right)^2$$

Objective Function

- Consider the PSMDF model

$$v_{fn} = [\mathbf{V}]_{fn} = \langle \mathbf{W}_f, \mathbf{H}_n \rangle = \mathrm{Tr}(\mathbf{W}_f \mathbf{H}_n)$$

- PSD matrices $\{\mathbf{W}_f\}_{f=1}^F$ and $\{\mathbf{H}_n\}_{n=1}^N$ can be estimated by minimizing a quadratic objective function (Stark, 2016; Vandaele et al., 2018):

$$g(\{\mathbf{W}_f\}_{f=1}^F, \{\mathbf{H}_n\}_{n=1}^N) = \frac{1}{2} \sum_{f,n} \left(v_{fn} - \mathrm{Tr}(\mathbf{W}_f \mathbf{H}_n)\right)^2$$

- For fixed $\{\mathbf{W}_f\}_{f=1}^F$, g is convex in $\{\mathbf{H}_n\}_{n=1}^N$ and vice versa (Vandaele et al., 2018).

Objective Function

- Consider the PSMDF model

$$v_{fn} = [\mathbf{V}]_{fn} = \langle \mathbf{W}_f, \mathbf{H}_n \rangle = \mathrm{Tr}(\mathbf{W}_f \mathbf{H}_n)$$

- PSD matrices $\{\mathbf{W}_f\}_{f=1}^F$ and $\{\mathbf{H}_n\}_{n=1}^N$ can be estimated by minimizing a quadratic objective function (Stark, 2016; Vandaele et al., 2018):

$$g(\{\mathbf{W}_f\}_{f=1}^F, \{\mathbf{H}_n\}_{n=1}^N) = \frac{1}{2} \sum_{f,n} \left(v_{fn} - \mathrm{Tr}(\mathbf{W}_f \mathbf{H}_n)\right)^2$$

- For fixed $\{\mathbf{W}_f\}_{f=1}^F$, g is convex in $\{\mathbf{H}_n\}_{n=1}^N$ and vice versa (Vandaele et al., 2018).

- Other objective functions are possible (Glasser et al., 2019; Basu et al., 2016; Lahat and Févotte, 2021)

Alternating Minimization

- Minimize the objective function g w.r.t. $\{\mathbf{H}_n\}_{n=1}^N$ for fixed $\{\mathbf{W}_f\}_{f=1}^F$, i.e.,

$$\{\mathbf{H}_n^+\}_{n=1}^N = \underset{\{\mathbf{H}_n\}_{n=1}^N}{\arg\min}\, g(\{\mathbf{W}_f\}_{f=1}^F, \{\mathbf{H}_n\}_{n=1}^N)$$

Alternating Minimization

- Minimize the objective function g w.r.t. $\{\mathbf{H}_n\}_{n=1}^N$ for fixed $\{\mathbf{W}_f\}_{f=1}^F$, i.e.,

$$\{\mathbf{H}_n^+\}_{n=1}^N = \underset{\{\mathbf{H}_n\}_{n=1}^N}{\arg\min}\, g(\{\mathbf{W}_f\}_{f=1}^F, \{\mathbf{H}_n\}_{n=1}^N)$$

- Change roles, i.e.,

$$\{\mathbf{W}_f^+\}_{f=1}^F = \underset{\{\mathbf{W}_f\}_{f=1}^F}{\arg\min}\, g(\{\mathbf{W}_f\}_{f=1}^F, \{\mathbf{H}_n^+\}_{n=1}^N)$$

Alternating Minimization

- Minimize the objective function g w.r.t. $\{\mathbf{H}_n\}_{n=1}^N$ for fixed $\{\mathbf{W}_f\}_{f=1}^F$, i.e.,

$$\{\mathbf{H}_n^+\}_{n=1}^N = \underset{\{\mathbf{H}_n\}_{n=1}^N}{\arg\min}\, g(\{\mathbf{W}_f\}_{f=1}^F, \{\mathbf{H}_n\}_{n=1}^N)$$

- Change roles, i.e.,

$$\{\mathbf{W}_f^+\}_{f=1}^F = \underset{\{\mathbf{W}_f\}_{f=1}^F}{\arg\min}\, g(\{\mathbf{W}_f\}_{f=1}^F, \{\mathbf{H}_n^+\}_{n=1}^N)$$

- Repeat until a stopping criterion is achieved;

Alternating Minimization

- Minimize the objective function g w.r.t. $\{\mathbf{H}_n\}_{n=1}^N$ for fixed $\{\mathbf{W}_f\}_{f=1}^F$, i.e.,

$$\{\mathbf{H}_n^+\}_{n=1}^N = \underset{\{\mathbf{H}_n\}_{n=1}^N}{\arg\min}\, g(\{\mathbf{W}_f\}_{f=1}^F, \{\mathbf{H}_n\}_{n=1}^N)$$

- Change roles, i.e.,

$$\{\mathbf{W}_f^+\}_{f=1}^F = \underset{\{\mathbf{W}_f\}_{f=1}^F}{\arg\min}\, g(\{\mathbf{W}_f\}_{f=1}^F, \{\mathbf{H}_n^+\}_{n=1}^N)$$

- Repeat until a stopping criterion is achieved;

- Several other algorithms had been independently developed by Vandaele et al. (2018), Basu et al. (2016), Glasser et al. (2019) and Stark (2016) based on this alternating approach.

Decrease Objective Separately w.r.t. each $\mathbf{H}_n$

- Focus on the first problem:

$$\{\mathbf{H}_n^+\}_{n=1}^N = \underset{\{\mathbf{H}_n\}_{n=1}^N}{\arg\min}\, g(\{\mathbf{W}_f\}_{f=1}^F, \{\mathbf{H}_n\}_{n=1}^N)$$

Decrease Objective Separately w.r.t. each $\mathbf{H}_n$

- Focus on the first problem:

$$\{\mathbf{H}_n^+\}_{n=1}^N = \underset{\{\mathbf{H}_n\}_{n=1}^N}{\arg\min}\, g(\{\mathbf{W}_f\}_{f=1}^F, \{\mathbf{H}_n\}_{n=1}^N)$$

- Objective function can be written as a sum of N terms

$$g(\{\mathbf{W}_f\}_{f=1}^F, \{\mathbf{H}_n\}_{n=1}^N) = \sum_{n=1}^N g_n(\{\mathbf{W}_f\}_{f=1}^F, \mathbf{H}_n)$$

where

$$g_n(\{\mathbf{W}_f\}_{f=1}^F, \mathbf{H}_n) = \frac{1}{2}\sum_{f=1}^F (v_{fn} - \mathrm{Tr}(\mathbf{W}_f\mathbf{H}_n))^2 = \frac{1}{2}\left\|\mathbf{v}_n - \mathcal{W}(\mathbf{H}_n)\right\|^2$$

and $\mathcal{W}(\mathbf{H}) = [\langle \mathbf{W}_1, \mathbf{H}\rangle, \ldots, \langle \mathbf{W}_F, \mathbf{H}\rangle]^\top \in \mathbb{R}^F$ for any matrix $\mathbf{H}$.

Link with Affine Rank Minimization and Phase Retrieval

(Lahat, Lang, Tan, and Févotte, 2021; Lahat and Févotte, 2021)

- For a specific $n = 1, \ldots, N$, estimating $\mathbf{H}_n$ is tantamount to

$$\min_{\mathbf{H}_n} \ \|\mathbf{v}_n - \mathcal{W}(\mathbf{H}_n)\|^2 \quad \text{subject to} \quad \mathbf{H}_n \in \mathbb{S}_+^K$$

  and possibly a rank constraint on $\mathbf{H}_n$;

Link with Affine Rank Minimization and Phase Retrieval

(Lahat, Lang, Tan, and Févotte, 2021; Lahat and Févotte, 2021)

- For a specific $n = 1, \ldots, N$, estimating $\mathbf{H}_n$ is tantamount to

$$\min_{\mathbf{H}_n} \ \|\mathbf{v}_n - \mathcal{W}(\mathbf{H}_n)\|^2 \quad \text{subject to} \quad \mathbf{H}_n \in \mathbb{S}_+^K$$

  and possibly a rank constraint on $\mathbf{H}_n$;

- Because in PSDMF, one typically imposes low rank constraints on $\mathbf{W}_f$ and $\mathbf{H}_n$ too ("inner ranks" are small);

Link with Affine Rank Minimization and Phase Retrieval

(Lahat, Lang, Tan, and Févotte, 2021; Lahat and Févotte, 2021)

- For a specific $n = 1, \ldots, N$, estimating $\mathbf{H}_n$ is tantamount to

$$\min_{\mathbf{H}_n} \ \|\mathbf{v}_n - \mathcal{W}(\mathbf{H}_n)\|^2 \quad \text{subject to} \quad \mathbf{H}_n \in \mathbb{S}_+^K$$

  and possibly a rank constraint on $\mathbf{H}_n$;

- Because in PSDMF, one typically imposes low rank constraints on $\mathbf{W}_f$ and $\mathbf{H}_n$ too ("inner ranks" are small);

- This optimization is known as affine rank minimization (Recht et al., 2010; Jain et al., 2010);

Link with Affine Rank Minimization and Phase Retrieval

(Lahat, Lang, Tan, and Févotte, 2021; Lahat and Févotte, 2021)

- For a specific $n = 1, \ldots, N$, estimating $\mathbf{H}_n$ is tantamount to

$$\min_{\mathbf{H}_n} \ \|\mathbf{v}_n - \mathcal{W}(\mathbf{H}_n)\|^2 \quad \text{subject to} \quad \mathbf{H}_n \in \mathbb{S}_+^K$$

  and possibly a rank constraint on $\mathbf{H}_n$;

- Because in PSDMF, one typically imposes low rank constraints on $\mathbf{W}_f$ and $\mathbf{H}_n$ too ("inner ranks" are small);

- This optimization is known as affine rank minimization (Recht et al., 2010; Jain et al., 2010);

- If $\text{rank}(\mathbf{W}_f) = 1$ for all $f = 1, \ldots, F$, and $\text{rank}(\mathbf{H}_n) = 1$, this is known as phase retrieval (Candès et al., 2015).

Affine Rank Minimization and Hard Thresholding

(Lahat, Lang, Tan, and Févotte, 2021)

- Many algorithms for affine rank minimization and phase retrieval;

Affine Rank Minimization and Hard Thresholding

(Lahat, Lang, Tan, and Févotte, 2021)

- Many algorithms for affine rank minimization and phase retrieval;
- Use them for PSDMF to solve the subproblem to update $\{\mathbf{H}_n\}_{n=1}^N$ given $\mathcal{W}$.

Affine Rank Minimization and Hard Thresholding

(Lahat, Lang, Tan, and Févotte, 2021)

- Many algorithms for affine rank minimization and phase retrieval;
- Use them for PSDMF to solve the subproblem to update $\{\mathbf{H}_n\}_{n=1}^N$ given $\mathcal{W}$.
- Projection of a matrix onto the set of $K \times K$ PSD matrices of rank $\leq R$ is denoted as $\mathrm{H}_{\mathbb{S}_+^K,R}(\cdot)$, also known as hard thresholding;

Affine Rank Minimization and Hard Thresholding

(Lahat, Lang, Tan, and Févotte, 2021)

- Many algorithms for affine rank minimization and phase retrieval;
- Use them for PSDMF to solve the subproblem to update $\{\mathbf{H}_n\}_{n=1}^N$ given $\mathcal{W}$.
- Projection of a matrix onto the set of $K \times K$ PSD matrices of rank $\leq R$ is denoted as $\mathrm{H}_{\mathbb{S}_+^K,R}(\cdot)$, also known as hard thresholding;
- Can be computed as (Jain et al., 2010; Tu et al., 2016)

$$\mathrm{H}_{\mathbb{S}_+^K,R}(\mathbf{H}) = \mathbf{U}\mathbf{\Lambda}_R\mathbf{U}^\top$$

where

- $\mathbf{\Lambda}_R \in \mathbb{R}^{R\times R}$ is a diagonal nonnegative matrix with the R largest nonnegative eigenvalues of $\mathbf{H}$ on its main diagonal;
- the columns of $\mathbf{U} \in \mathbb{R}^{K\times R}$ are the eigenvectors of $\mathbf{W}$ associated with these R largest nonnegative eigenvalues

Affine Rank Minimization and Hard Thresholding

(Lahat, Lang, Tan, and Févotte, 2021)

- Many algorithms for affine rank minimization and phase retrieval;
- Use them for PSDMF to solve the subproblem to update $\{\mathbf{H}_n\}_{n=1}^N$ given $\mathcal{W}$.
- Projection of a matrix onto the set of $K \times K$ PSD matrices of rank $\leq R$ is denoted as $\mathrm{H}_{\mathbb{S}_+^K,R}(\cdot)$, also known as hard thresholding;
- Can be computed as (Jain et al., 2010; Tu et al., 2016)

$$\mathrm{H}_{\mathbb{S}_+^K,R}(\mathbf{H}) = \mathbf{U}\mathbf{\Lambda}_R\mathbf{U}^\top$$

where

  - $\mathbf{\Lambda}_R \in \mathbb{R}^{R\times R}$ is a diagonal nonnegative matrix with the R largest nonnegative eigenvalues of $\mathbf{H}$ on its main diagonal;
  - the columns of $\mathbf{U} \in \mathbb{R}^{K\times R}$ are the eigenvectors of $\mathbf{W}$ associated with these R largest nonnegative eigenvalues

- Can also use singular value projection; see Lahat et al. (2021) for details.

Majorization-Minimization Algorithm for PSDMF

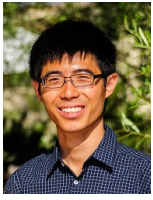
(Soh and Varvitsiotis, 2021)

- While links to phase retrieval and affine rank minimization are nice, theoretical guarantees (e.g., convergence guarantees) are lacking. ☹

Majorization-Minimization Algorithm for PSDMF

(Soh and Varvitsiotis, 2021)

- While links to phase retrieval and affine rank minimization are nice, theoretical guarantees (e.g., convergence guarantees) are lacking. ☹
- Would be good to develop a multiplicative update-type algorithm based on majorization-minimization (MM). ☺



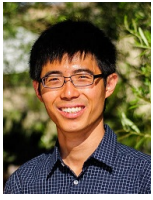
Y. S. Soh (NUS Math)

A. Vavitsiotis (SUTD)

Majorization-Minimization Algorithm for PSDMF

(Soh and Varvitsiotis, 2021)

- While links to phase retrieval and affine rank minimization are nice, theoretical guarantees (e.g., convergence guarantees) are lacking. ☹
- Would be good to develop a multiplicative update-type algorithm based on majorization-minimization (MM). ☺



Y. S. Soh (NUS Math)

A. Vavitsiotis (SUTD)

"Slides" below borrowed from Y. S. Soh with permission and with thanks.

From NMF to PSDMF

Lee and Seung (1999) update rule writes

$$\mathbf{h} \longleftarrow \mathbf{h} \cdot \frac{\mathbf{W}^\top \mathbf{v}}{\mathbf{W}^\top \mathbf{W} \mathbf{v}}.$$

From NMF to PSDMF

Lee and Seung (1999) update rule writes

$$\mathbf{h} \longleftarrow \mathbf{h} \cdot \frac{\mathbf{W}^\top \mathbf{v}}{\mathbf{W}^\top \mathbf{W}\mathbf{v}}.$$

Embed $\mathbf{h}$ as a diagonal matrix.

$$\begin{pmatrix} \ddots & & \\ & h_k & \\ & & \ddots \end{pmatrix} \longleftarrow \begin{pmatrix} \ddots & & \\ & \frac{(\mathbf{W}^\top \mathbf{v})_k}{(\mathbf{W}^\top \mathbf{W}\mathbf{h})_k} & \\ & & \ddots \end{pmatrix} \begin{pmatrix} \ddots & & \\ & h_k & \\ & & \ddots \end{pmatrix}$$

nng vectors $\cong$ diagonal PSD matrices

From NMF to PSDMF

Re-arrange

$$\begin{pmatrix} \ddots & & \\ & h_k & \\ & & \ddots \end{pmatrix} \longleftarrow \begin{pmatrix} \ddots & & \\ & \frac{(\mathbf{W}^\top \mathbf{v})_k}{(\mathbf{W}^\top \mathbf{W}\mathbf{h})_k} & \\ & & \ddots \end{pmatrix} \begin{pmatrix} \ddots & & \\ & h_k & \\ & & \ddots \end{pmatrix}$$

$$= \begin{pmatrix} \ddots & & \\ & \frac{h_k}{(\mathbf{W}^\top \mathbf{W}\mathbf{h})_k} & \\ & & \ddots \end{pmatrix} \begin{pmatrix} \ddots & & \\ & (\mathbf{W}^\top \mathbf{v})_k & \\ & & \ddots \end{pmatrix}$$

PSD Factorization

$$\underbrace{\mathbf{H}}_{\text{matrix}} \leftarrow \underbrace{T}_{\text{operator}} (\underbrace{\mathcal{W}^\top \mathbf{v}}_{\text{matrix}})$$

From NMF to PSDMF

Re-arrange

$$\begin{pmatrix} \ddots & & \\ & h_k & \\ & & \ddots \end{pmatrix} \leftarrow \begin{pmatrix} \ddots & & \\ & \frac{(\mathbf{W}^\top \mathbf{v})_k}{(\mathbf{W}^\top \mathbf{W}\mathbf{h})_k} & \\ & & \ddots \end{pmatrix} \begin{pmatrix} \ddots & & \\ & h_k & \\ & & \ddots \end{pmatrix}$$

$$= \begin{pmatrix} \ddots & & \\ & \frac{h_k}{(\mathbf{W}^\top \mathbf{W}\mathbf{h})_k} & \\ & & \ddots \end{pmatrix} \begin{pmatrix} \ddots & & \\ & (\mathbf{W}^\top \mathbf{v})_k & \\ & & \ddots \end{pmatrix}$$

PSD Factorization

$$\underbrace{\mathbf{H}}_{\text{matrix}} \leftarrow \underbrace{T}_{\text{operator}} (\underbrace{\mathcal{W}^\top \mathbf{v}}_{\text{matrix}})$$

Find: operator T that is (i) simple, (ii) preserves PSD-ness, (iii) generalizes averaging of $\mathbf{H}$ and $([\mathcal{W}^T \mathcal{W}](\mathbf{H}))^{-1}$.

From NMF to PSDMF

- The analogue of diagonal scaling is conjugation

$$\mathbf{W} \longleftarrow \mathbf{M}(\mathcal{W}^{\top}\mathbf{v})\mathbf{M}$$

where

$$\mathbf{M} = \text{Geometric mean}\left(\mathbf{H}, ([\mathcal{W}^{T}\mathcal{W}](\mathbf{H}))^{-1}\right)$$

From NMF to PSDMF

- The analogue of diagonal scaling is conjugation

$$\mathbf{W} \longleftarrow \mathbf{M}(\mathcal{W}^{\top}\mathbf{v})\mathbf{M}$$

where

$$\mathbf{M} = \text{Geometric mean}\big(\mathbf{H}, ([\mathcal{W}^{T}\mathcal{W}](\mathbf{H}))^{-1}\big)$$

- Preserves PSD-ness and is simple.

Geometric Mean of Two Positive Definite (PD) Matrices

- **Definition**: The matrix geometric mean of two PD matrices $\mathbf{C}$ and $\mathbf{D}$ is

$$\mathbf{C}\#\mathbf{D} := \mathbf{C}^{1/2}(\mathbf{C}^{-1/2}\mathbf{D}\mathbf{C}^{-1/2})^{1/2}\mathbf{C}^{1/2}$$

  Generalizes the geometric mean $\sqrt{cd}$ of two positive numbers c and d.

Geometric Mean of Two Positive Definite (PD) Matrices

- **Definition**: The matrix geometric mean of two PD matrices $\mathbf{C}$ and $\mathbf{D}$ is

$$\mathbf{C}\#\mathbf{D} := \mathbf{C}^{1/2}(\mathbf{C}^{-1/2}\mathbf{D}\mathbf{C}^{-1/2})^{1/2}\mathbf{C}^{1/2}$$

  Generalizes the geometric mean $\sqrt{cd}$ of two positive numbers c and d.

- **Equivalent Definition**: Unique PD solution $\mathbf{X}^*$ to the Riccati equation

$$\mathbf{X}\mathbf{C}^{-1}\mathbf{X} = \mathbf{D}.$$

Geometric Mean of Two Positive Definite (PD) Matrices

- **Definition**: The matrix geometric mean of two PD matrices $\mathbf{C}$ and $\mathbf{D}$ is

$$\mathbf{C}\#\mathbf{D} := \mathbf{C}^{1/2}(\mathbf{C}^{-1/2}\mathbf{D}\mathbf{C}^{-1/2})^{1/2}\mathbf{C}^{1/2}$$

  Generalizes the geometric mean $\sqrt{cd}$ of two positive numbers c and d.

- **Equivalent Definition**: Unique PD solution $\mathbf{X}^*$ to the Riccati equation

$$\mathbf{X}\mathbf{C}^{-1}\mathbf{X} = \mathbf{D}.$$

- $\mathbf{C}\#\mathbf{D}$ is the midpoint of the geodesic joining $\mathbf{C}$ and $\mathbf{D}$ on the manifold of PD matrices.

Geometric Mean of Two Positive Definite (PD) Matrices

- **Definition**: The matrix geometric mean of two PD matrices $\mathbf{C}$ and $\mathbf{D}$ is

$$\mathbf{C}\#\mathbf{D} := \mathbf{C}^{1/2}(\mathbf{C}^{-1/2}\mathbf{D}\mathbf{C}^{-1/2})^{1/2}\mathbf{C}^{1/2}$$

  Generalizes the geometric mean $\sqrt{cd}$ of two positive numbers c and d.

- **Equivalent Definition**: Unique PD solution $\mathbf{X}^*$ to the Riccati equation

$$\mathbf{X}\mathbf{C}^{-1}\mathbf{X} = \mathbf{D}.$$

- $\mathbf{C}\#\mathbf{D}$ is the midpoint of the geodesic joining $\mathbf{C}$ and $\mathbf{D}$ on the manifold of PD matrices.

- Fun facts:

$$\mathbf{C}\#\mathbf{D} = \mathbf{D}\#\mathbf{C} \quad \text{and} \quad (\mathbf{C}\#\mathbf{D})^{-1} = \mathbf{C}^{-1}\#\mathbf{D}^{-1}.$$

Multiplicative-Type Algorithm for PSDMF

Recall for fixed $\mathbf{W}_f \in \mathbb{S}_+^K, f = 1, \ldots, F$, we aim to solve

$$\min_{\mathbf{H}} \quad \|\mathbf{v} - \mathcal{W}(\mathbf{H})\|^2 \quad \text{subject to} \quad \mathbf{H} \in \mathbb{S}_+^K$$

where $\mathcal{W}(\mathbf{H}) = [\langle \mathbf{W}_1, \mathbf{H} \rangle, \ldots, \langle \mathbf{W}_F, \mathbf{H} \rangle]^\top \in \mathbb{R}^F$.

Multiplicative-Type Algorithm for PSDMF

Recall for fixed $\mathbf{W}_f \in \mathbb{S}_+^K, f = 1, \ldots, F$, we aim to solve

$$\min_{\mathbf{H}} \quad \|\mathbf{v} - \mathcal{W}(\mathbf{H})\|^2 \quad \text{subject to} \quad \mathbf{H} \in \mathbb{S}_+^K$$

where $\mathcal{W}(\mathbf{H}) = [\langle \mathbf{W}_1, \mathbf{H} \rangle, \ldots, \langle \mathbf{W}_F, \mathbf{H} \rangle]^\top \in \mathbb{R}^F$.

Theorem (Soh and Varvitsiotis (2021))

The objective function $\|\mathbf{v} - \mathcal{W}(\mathbf{H})\|$ is non-increasing under the update rule

$$\mathbf{H}^+ = \mathbf{M}(\mathcal{W}^\top \mathbf{v})\mathbf{M}, \quad \textit{where} \quad \mathbf{M} = \left([\mathcal{W}^\top \mathcal{W}](\mathbf{H})\right)^{-1} \#(\mathbf{H})$$

Furthermore, if initialized with a PD matrix, the subsequent iterates remain PD.

Multiplicative-Type Algorithm for PSDMF

Recall for fixed $\mathbf{W}_f \in \mathbb{S}_+^K, f = 1, \ldots, F$, we aim to solve

$$\min_{\mathbf{H}} \quad \|\mathbf{v} - \mathcal{W}(\mathbf{H})\|^2 \quad \text{subject to} \quad \mathbf{H} \in \mathbb{S}_+^K$$

where $\mathcal{W}(\mathbf{H}) = [\langle \mathbf{W}_1, \mathbf{H} \rangle, \ldots, \langle \mathbf{W}_F, \mathbf{H} \rangle]^\top \in \mathbb{R}^F$.

Theorem (Soh and Varvitsiotis (2021))

The objective function $\|\mathbf{v} - \mathcal{W}(\mathbf{H})\|$ is non-increasing under the update rule

$$\mathbf{H}^+ = \mathbf{M}(\mathcal{W}^\top \mathbf{v})\mathbf{M}, \quad \textit{where} \quad \mathbf{M} = \left([\mathcal{W}^\top \mathcal{W}](\mathbf{H})\right)^{-1} \#(\mathbf{H})$$

Furthermore, if initialized with a PD matrix, the subsequent iterates remain PD.

Reduces to Lee and Seung (1999) update in the diagonal case, i.e.,

$$\mathbf{h}^+ = \mathbf{h} \cdot \frac{\mathbf{W}^\top \mathbf{v}}{\mathbf{W}^\top \mathbf{W} \mathbf{v}}.$$

Multiplicative-Type Algorithm for PSDMF

Matrix Multiplicative Update (MMU) Algorithm (Soh and Varvitsiotis, 2021):

- Input: A matrix $\mathbf{V} \in \mathbb{R}_{+}^{F \times N}$ and parameter $K \in \mathbb{N}$;

Multiplicative-Type Algorithm for PSDMF

Matrix Multiplicative Update (MMU) Algorithm (Soh and Varvitsiotis, 2021):

- Input: A matrix $\mathbf{V} \in \mathbb{R}_+^{F \times N}$ and parameter $K \in \mathbb{N}$;
- Output $\{\mathbf{W}_f\}_{f=1}^F, \{\mathbf{H}_n\}_{n=1}^N \subset \mathbb{S}_+^K$ such that $v_{fn} \approx \langle \mathbf{W}_f, \mathbf{H}_n \rangle$ for all f, n;

Multiplicative-Type Algorithm for PSDMF

Matrix Multiplicative Update (MMU) Algorithm (Soh and Varvitsiotis, 2021):

- Input: A matrix $\mathbf{V} \in \mathbb{R}_+^{F \times N}$ and parameter $K \in \mathbb{N}$;
- Output $\{\mathbf{W}_f\}_{f=1}^F, \{\mathbf{H}_n\}_{n=1}^N \subset \mathbb{S}_+^K$ such that $v_{fn} \approx \langle \mathbf{W}_f, \mathbf{H}_n \rangle$ for all f, n;
- While stopping criterion not satisfied, do

$$\mathbf{W}_f \longleftarrow \mathbf{N}_f(\mathcal{H}^\top \mathbf{v}_{f,:})\mathbf{N}_f \quad \text{where} \quad \mathbf{N}_f = \big([\mathcal{H}^\top \mathcal{H}](\mathbf{W}_f)\big)^{-1} \# (\mathbf{W}_f)$$

and

$$\mathbf{H}_n \longleftarrow \mathbf{M}_n(\mathcal{W}^\top \mathbf{v}_{:,n})\mathbf{M}_n \quad \text{where} \quad \mathbf{M}_n = \big([\mathcal{W}^\top \mathcal{W}](\mathbf{H}_n)\big)^{-1} \# (\mathbf{H}_n).$$

Multiplicative-Type Algorithm for PSDMF

Matrix Multiplicative Update (MMU) Algorithm (Soh and Varvitsiotis, 2021):

- Input: A matrix $\mathbf{V} \in \mathbb{R}_+^{F \times N}$ and parameter $K \in \mathbb{N}$;
- Output $\{\mathbf{W}_f\}_{f=1}^F, \{\mathbf{H}_n\}_{n=1}^N \subset \mathbb{S}_+^K$ such that $v_{fn} \approx \langle \mathbf{W}_f, \mathbf{H}_n \rangle$ for all f, n;
- While stopping criterion not satisfied, do

$$\mathbf{W}_f \longleftarrow \mathbf{N}_f(\mathcal{H}^\top \mathbf{v}_{f,:})\mathbf{N}_f \quad \text{where} \quad \mathbf{N}_f = \big([\mathcal{H}^\top \mathcal{H}](\mathbf{W}_f)\big)^{-1} \# (\mathbf{W}_f)$$

and

$$\mathbf{H}_n \longleftarrow \mathbf{M}_n(\mathcal{W}^\top \mathbf{v}_{:,n})\mathbf{M}_n \quad \text{where} \quad \mathbf{M}_n = \big([\mathcal{W}^\top \mathcal{W}](\mathbf{H}_n)\big)^{-1} \# (\mathbf{H}_n).$$

Properties of MMU:

- Always operates in interior of PSD cone (no projection needed);
- Geometric interpretation of trajectory;
- Recovers classical MU (Lee and Seung, 1999) if matrices are diagonal.

Proof Idea

- Write down candidate auxiliary function inspired by Taylor's theorem;

Proof Idea

- Write down candidate auxiliary function inspired by Taylor's theorem;
- Show it dominates square loss, this reduces to

$$\mathbf{M} \otimes \mathbf{M} - \mathcal{W}^\top \mathcal{W} \succcurlyeq 0,$$

where $\mathbf{M} = \left([\mathcal{W}^\top \mathcal{W}](\mathbf{H})\right)^{-1} \#(\mathbf{H})$ is the matrix geometric mean;

Proof Idea

- Write down candidate auxiliary function inspired by Taylor's theorem;
- Show it dominates square loss, this reduces to

$$\mathbf{M} \otimes \mathbf{M} - \mathcal{W}^\top \mathcal{W} \succcurlyeq 0,$$

  where $\mathbf{M} = \left([\mathcal{W}^\top \mathcal{W}](\mathbf{H})\right)^{-1} \#(\mathbf{H})$ is the matrix geometric mean;
- Pre-multiply with $\mathbf{H}^{-1/2}$, reduces to $\mathbf{H} = \mathbf{I}$;

Proof Idea

- Write down candidate auxiliary function inspired by Taylor's theorem;
- Show it dominates square loss, this reduces to

$$\mathbf{M} \otimes \mathbf{M} - \mathcal{W}^\top \mathcal{W} \succcurlyeq 0,$$

where $\mathbf{M} = \left([\mathcal{W}^\top \mathcal{W}](\mathbf{H})\right)^{-1} \# (\mathbf{H})$ is the matrix geometric mean;

- Pre-multiply with $\mathbf{H}^{-1/2}$, reduces to $\mathbf{H} = \mathbf{I}$;
- Apply Cauchy–Schwarz inequality

$$\mathrm{Tr}(\mathbf{X}^2)\mathrm{Tr}(\mathbf{Y}^2) \geq \mathrm{Tr}(\mathbf{X}\mathbf{Y})^2$$

and a consequence of Lieb's concavity theorem (Lieb, 1973)

$$\left(\sum_i \mathbf{X}_i^{1/2}\right) \otimes \left(\sum_i \mathbf{X}_i^{1/2}\right) \preccurlyeq \left(\sum_i \mathbf{X}_i\right)^{1/2} \otimes \left(\sum_i \mathbf{X}_i\right)^{1/2}.$$

Concluding Remarks for PSDMF

- PSDMF (Gouveia et al., 2013; Fiorini et al., 2012; Vandaele et al., 2018) is a generalization of NMF

$$\begin{aligned} &\text{(PSDMF)} && v_{fn} = \langle \mathbf{W}_f, \mathbf{H}_n \rangle, && \mathbf{W}_f, \mathbf{H}_n \succcurlyeq 0, \\ &\text{(NMF)} && v_{fn} = \langle \mathbf{w}_f, \mathbf{h}_n \rangle, && \mathbf{w}_f, \mathbf{h}_n \geq \mathbf{0}. \end{aligned}$$

Concluding Remarks for PSDMF

- PSDMF (Gouveia et al., 2013; Fiorini et al., 2012; Vandaele et al., 2018) is a generalization of NMF

$$\begin{aligned} &\text{(PSDMF)} \quad v_{fn} = \langle \mathbf{W}_f, \mathbf{H}_n \rangle, \quad \mathbf{W}_f, \mathbf{H}_n \succcurlyeq 0, \\ &\text{(NMF)} \quad v_{fn} = \langle \mathbf{w}_f, \mathbf{h}_n \rangle, \quad \mathbf{w}_f, \mathbf{h}_n \geq \mathbf{0}. \end{aligned}$$

- Can use signal processing primitives such as phase retrieval and affine rank minimization within an alternating minimzation framework to find $\{\mathbf{W}_f\}$ and $\{\mathbf{H}_n\}$ (Lahat et al., 2021);

Concluding Remarks for PSDMF

- PSDMF (Gouveia et al., 2013; Fiorini et al., 2012; Vandaele et al., 2018) is a generalization of NMF

$$\begin{aligned} &\text{(PSDMF)} \quad v_{fn} = \langle \mathbf{W}_f, \mathbf{H}_n \rangle, \quad \mathbf{W}_f, \mathbf{H}_n \succcurlyeq 0, \\ &\text{(NMF)} \quad v_{fn} = \langle \mathbf{w}_f, \mathbf{h}_n \rangle, \quad \mathbf{w}_f, \mathbf{h}_n \geq \mathbf{0}. \end{aligned}$$

- Can use signal processing primitives such as phase retrieval and affine rank minimization within an alternating minimzation framework to find $\{\mathbf{W}_f\}$ and $\{\mathbf{H}_n\}$ (Lahat et al., 2021);

- Even better, use majorization-minimization (MM) in the space of PD matrices (Soh and Varvitsiotis, 2021);

Concluding Remarks for PSDMF

- PSDMF (Gouveia et al., 2013; Fiorini et al., 2012; Vandaele et al., 2018) is a generalization of NMF

$$\begin{aligned} &\text{(PSDMF)} \quad v_{fn} = \langle \mathbf{W}_f, \mathbf{H}_n \rangle, \qquad \mathbf{W}_f, \mathbf{H}_n \succcurlyeq 0, \\ &\text{(NMF)} \quad v_{fn} = \langle \mathbf{w}_f, \mathbf{h}_n \rangle, \qquad \mathbf{w}_f, \mathbf{h}_n \geq \mathbf{0}. \end{aligned}$$

- Can use signal processing primitives such as phase retrieval and affine rank minimization within an alternating minimzation framework to find $\{\mathbf{W}_f\}$ and $\{\mathbf{H}_n\}$ (Lahat et al., 2021);
- Even better, use majorization-minimization (MM) in the space of PD matrices (Soh and Varvitsiotis, 2021);
- Other extensions to symmetric cones, including SOCPs.

References I

G. Averkov, V. Kaibel, and S. Weltge. Maximum semidefinite and linear extension complexity of families of polytopes. Math. Program., 167(2):381–394, 2018.

A. Basu, M. Dinitz, and X. Li. Computing approximate PSD factorizations. In Proc. APPROX/RANDOM, volume 60, pages 2:1–2:12, 2016.

C. M. Bishop. Bayesian PCA. In Advances of Neural Information Processing Systems (NIPS), 1999.

R. Bradley and M. Terry. Rank analysis of incomplete block designs I: The method of paired comparisons. Biometrika, 35:324–345, 1952.

E. Candès, Y. C. Eldar, T. Strohmer, and V. Voroninski. Phase retrieval via matrix completion. SIAM Review, 57(2):225–251, 2015.

E. J. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. J. Fourier Analysis and Applications, 14:877–905, 2008.

E. C. Chi and T. G. Kolda. On tensors, sparsity, and nonnegative factorizations. SIAM Journal on Matrix Analysis and Applications, 33(4):1272–1299, 2012.

H. Fawzi, J. Gouveia, P. A. Parrilo, R. Z. Robinson, and R. R. Thomas. Positive semidefinite rank. Math. Program., 153(1):133–177, 2015.

C. Févotte, N. Bertin, and J.-L. Durrieu. Nonnegative matrix factorization with the Itakura-Saito divergence. With application to music analysis. Neural Computation, 21(3):793–830, Mar. 2009. doi: 10.1162/neco.2008.04-08-771. URL `https://www.irit.fr/~Cedric.Fevotte/publications/journals/neco09_is-nmf.pdf`.

S. Fiorini, S. Massar, S. Pokutta, H. R. Tiwary, and R. D. Wolf. Linear vs. semidefinite extended formulations: exponential separation and strong lower bounds. In Proc. STOC, pages 95–106, 2012.

References II

N. Gillis, L. T. K. Hien, V. Leplat, and V. Y. F. Tan. Distributionally Robust and Multi-Objective Nonnegative Matrix Factorization . IEEE Trans. on Pattern Analysis and Machine Intelligence, 2022.

I. Glasser, R. Sweke, N. Pancotti, J. Eisert, and I. Cirac. Expressive power of tensor-network factorizations for probabilistic modeling. In Proc. Adv. Neural Inf. Process. Syst., pages 1496–1508, 2019.

J. Gouveia, P. A. Parrilo, and R. R. Thomas. Lifts of convex sets and cone factorizations. Mathematics of Operations Research, 38(2):248–264, 2013.

P. Jain, R. Meka, and I. S. Dhillon. Guaranteed rank minimization via singular value projection. In Advances of Neural Information Processing Systems (NIPS), pages 935–945, 2010.

R. Jain, Y. Shi, Z. Wei, and S. Zhang. Efficient protocols for generating bipartite classical distributions and quantum states. IEEE Transactions on Information Theory, 59(8):5171–5178, 2013.

D. Lahat and C. Févotte. Positive semidefinite matrix factorization based on truncated Wirtinger flow. In Proc. Eusipco, 2021.

D. Lahat, Y. Lang, V. Y. F. Tan, and C. Févotte. Positive semidefinite matrix factorization: A connection with phase retrieval and affine rank minimization. IEEE Transactions on Signal Processing, 69:3059–3074, 2021. doi: 10.1109/TSP.2021.3071293. URL https://arxiv.org/pdf/2007.12364.

D. D. Lee and H. S. Seung. Learning the parts of objects with nonnegative matrix factorization. Nature, 401: 788–791, 1999.

E. H. Lieb. Convex trace functions and the Wigner–Yanase–Dyson conjecture. Advances in Mathematics, 11 (3):267–288, 1973.

R. Luce. Individual choice behavior: A theoretical analysis. Wiley, 1959.

References III

B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM Review, 52(3):471–501, 2010.

Y. S. Soh and A. Varvitsiotis. A non-commutative extension of Lee-Seung's algorithm for positive semidefinite factorizations. In Advances in Neural Processing Systems (NeurIPS), 2021.

C. J. Stark. Recommender systems inspired by the structure of quantum theory. arXiv 1691.06035, 2016.

V. Y. F. Tan and C. Févotte. Automatic relevance determination in nonnegative matrix factorization with the β-divergence. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(7):1592–1605, 2013.

M. E. Tipping. Sparse Bayesian Learning and the Relevance Vector Machine. Journal of Machine Learning Research, 1:211–244, 2001.

S. Tu, R. Boczar, M. Simchowitz, M. Soltanolkotabi, and B. Recht. Low-rank solutions of linear matrix equations via Procrustes flow. In Proc. Intl. Conf. on Machine Learning (ICML), pages 964–073, 2016.

J. van Apeldoorn, A. Gilyén, S. Gribling, and R. deWolf. Quantum SDPSolvers: Better upper and lower bounds. Quantum, 4(230), Feb 2020.

A. Vandaele, F. Glineur, and N. Gillis. Algorithms for positive semidefinite factorization. Computational Optimization and Applications, 71(1):193–219, 2018.

T. Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. IEEE Transactions on Audio, Speech, and Language Processing, 15(3):1066–1074, 2007.

References IV

R. Xia, V. Y. F. Tan, L. Filstroff, and C. Févotte. A ranking model motivated by nonnegative matrix factorization with applications to tennis tournaments. In Proc. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD), Sep. 2019. URL https://arxiv.org/abs/1903.06500.

M. Yannakakis. Expressing combinatorial optimization problems by linear programs. J. Comput. Syst. Sci, 43 (3):441–466, 1991.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. J. Royal Statistical Soc., 68(1):49–67, 2007.

R. Zhao and V. Y. F. Tan. A unified convergence analysis of the multiplicative update algorithm for regularized nonnegative matrix factorization. IEEE Transactions on Signal Processing, 66(1):129–138, Jan 2018. ISSN 1053-587X. doi: 10.1109/TSP.2017.2757914.