Provable Sparse Tensor Decomposition for Personalized Recommendation and High-dimensional Latent Variable Models

Wei Sun
Yahoo Labs

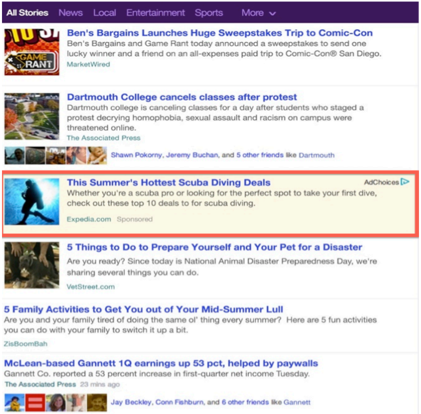
December 3, 2015
Big Data Theory Group Meeting
Purdue University

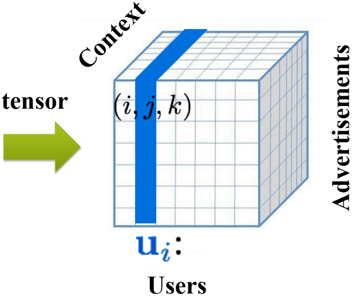
Joint work with Junwei Lu (Princeton), Han Liu (Princeton), Guang Cheng (Purdue)

Outline

- Motivation Examples
- Sparse Tensor Decomposition
- Local and Global Convergence Analysis
- Experiments
- Future Work on Statistical-and-Computational Tradeoffs

Motivation: Personalized Recommendation

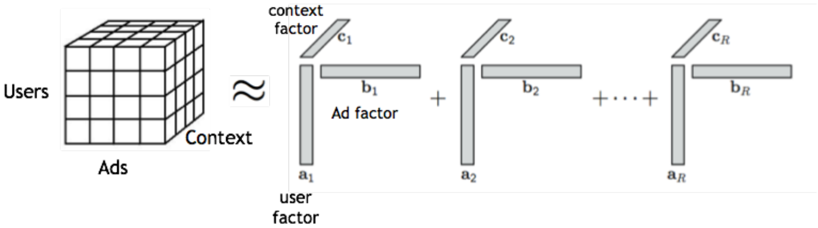




$(i,\ j,\ k)$

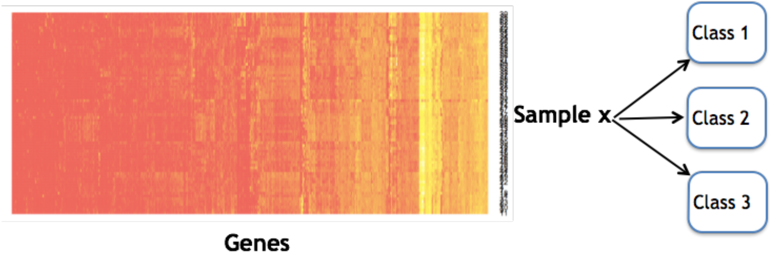
Whether *user* i, under *context* j will click *ad.* k ?

Motivation: Personalized Recommendation



- Goal: Given the observed tensor, compute the factors to recover the whole tensor.
- Difficulty: the tensor is sparse and the factors are sparse.

Motivation: High-dimensional Latent Variable Model



- Gaussian mixture: $\boldsymbol{x} \sim \sum_{k=1}^{K} w_k N(\boldsymbol{\mu}_k, \sigma^2 \mathbb{1})$

Motivation: High-dimensional Latent Variable Model

- (Hsu and Kakade, 2013) Define

$$\mathcal{M} := \mathbb{E}[\boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{x}] - \sigma^2 f(\mathbb{E}[\boldsymbol{x}]),$$

  then

$$\mathcal{M} = \sum_{k=1}^{K} w_k \boldsymbol{\mu}_k \otimes \boldsymbol{\mu}_k \otimes \boldsymbol{\mu}_k.$$

- Goal: Recover w_k and $\boldsymbol{\mu}_k$ from empirical tensor $\widehat{\mathcal{M}}$.
- Difficulty: many genes contain no information about clustering structure. Require sparse $\boldsymbol{\mu}_k$'s!

Sparse Tensor Decomposition

- Assume tensor $\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ to be sparse and have rank K,

$$\mathcal{T} = \sum_{i=1}^{K} w_i \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i,$$

  where $w_i \in \mathbb{R}$, $\mathbf{a}_i \in \mathbb{R}^{d_1}$, $\mathbf{b}_i \in \mathbb{R}^{d_2}$, $\mathbf{c}_i \in \mathbb{R}^{d_3}$, and $\mathbf{a}_i, \mathbf{b}_i, \mathbf{c}_i \in \mathcal{S}_{d_0} := \{\mathbf{v} : \|\mathbf{v}\|_2 = 1, \|\mathbf{v}\|_0 \leq d_0\}$ for any i.
- It generalizes matrix SVD to tensor. For a matrix A

$$A = UDV = \sum_i \sigma_i \boldsymbol{u}_i \otimes \boldsymbol{v}_i.$$

Existing Tensor Decomposition Methods

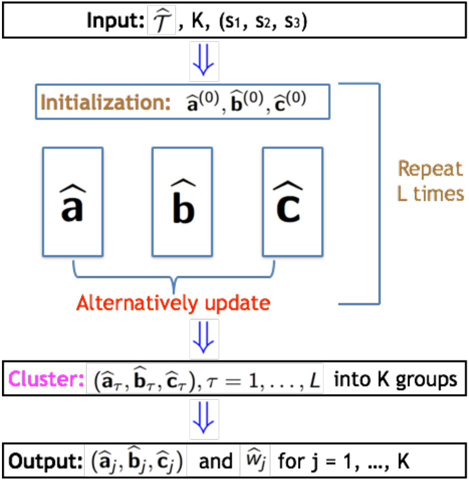
- Allen (2012) imposed an lasso penalty on $\mathbf{a}, \mathbf{b}, \mathbf{c}$ for rank-1 tensor recovery, but without theoretical guarantees,

$$\min_{\|\mathbf{a}\|=\|\mathbf{b}\|=\|\mathbf{c}\|=1} \|\mathcal{T} - w\mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c}\|_F + \lambda_1 \|\mathbf{a}\|_1 + \lambda_2 \|\mathbf{b}\|_1 + \lambda_3 \|\mathbf{c}\|_1.$$

- Anandkumar et al. (2014) proposed a non-sparse tensor decomposition method with guaranteed rates of convergence.

Our focus: propose a sparse tensor decomposition via l_0 optimization with theoretical guarantees of estimation accuracy.

Main Algorithm: Outline



Tensor Operations

- For $\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ and $\mathbf{u} \in \mathbb{R}^{d_1}, \mathbf{v} \in \mathbb{R}^{d_2}, \mathbf{w} \in \mathbb{R}^{d_3}$, define

$$\mathcal{T} \times_2 \mathbf{v} \times_3 \mathbf{w} := \sum_{j,l} \mathbf{v}_j \mathbf{w}_l [\mathcal{T}]_{:,j,l}$$

$$\mathcal{T} \times_1 \mathbf{u} \times_2 \mathbf{v} \times_3 \mathbf{w} := \sum_{i,j,l} \mathbf{u}_i \mathbf{v}_j \mathbf{w}_l [\mathcal{T}]_{i,j,l}$$

- Define $\mathrm{Norm}(\mathbf{v}) = \mathbf{v}/\|\mathbf{v}\|$.
- Define a truncation operator as

$$[\mathrm{Truncate}(\mathbf{v}, s)]_i = \begin{cases} \mathbf{v}_i, & \text{if } i \in \mathrm{supp}(\mathbf{v}, s) \\ 0, & \text{otherwise} \end{cases}.$$

- $\mathrm{Truncate}\big(\underbrace{(0.1, 0.3, -0.2, -0.6)}_{\mathbf{v}}, 2\big) = (0, 0.2, 0, -0.6).$

Main Algorithm: Continued

- Key: alternative update steps

$$\bar{\mathbf{a}} = \mathrm{Norm}\left(\widehat{\mathcal{T}} \times_2 \widehat{\mathbf{b}} \times_3 \widehat{\mathbf{c}}\right); \ \check{\mathbf{a}} = \mathrm{Truncate}\left(\bar{\mathbf{a}}, s_1\right); \ \widehat{\mathbf{a}} = \mathrm{Norm}\left(\check{\mathbf{a}}\right)$$

$$\bar{\mathbf{b}} = \mathrm{Norm}\left(\widehat{\mathcal{T}} \times_1 \widehat{\mathbf{a}} \times_3 \widehat{\mathbf{c}}\right); \ \check{\mathbf{b}} = \mathrm{Truncate}\left(\bar{\mathbf{b}}, s_2\right); \ \widehat{\mathbf{b}} = \mathrm{Norm}\left(\check{\mathbf{b}}\right)$$

$$\bar{\mathbf{c}} = \mathrm{Norm}\left(\widehat{\mathcal{T}} \times_1 \widehat{\mathbf{a}} \times_2 \widehat{\mathbf{b}}\right); \ \check{\mathbf{c}} = \mathrm{Truncate}\left(\bar{\mathbf{c}}, s_3\right); \ \widehat{\mathbf{c}} = \mathrm{Norm}\left(\check{\mathbf{c}}\right)$$

- Initialization: Random (fast) or via sparse SVD (provable)

Tuning Procedure

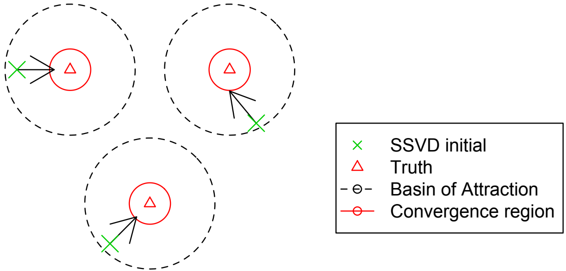
- Find exact tensor rank is an NP hard problem (Kolda, 2009).
- Tune (K, s_1, s_2, s_3) by minimizing BIC (Allen, 2012),

$$\mathrm{BIC} := \underbrace{\log\left(\frac{\|\widehat{\mathcal{E}}\|_F^2}{d_1 d_2 d_3}\right)}_{Model \;\; fitting} + \underbrace{\frac{\log(d_1 d_2 d_3)}{d_1 d_2 d_3}\left[K(s_1 + s_2 + s_3)\right]}_{Sparsity \;\; control}$$

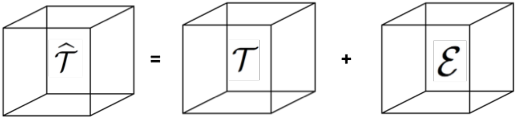
with $\widehat{\mathcal{E}} = \widehat{\mathcal{T}} - \sum_{i=1}^{K} \widehat{w}_i \widehat{\mathbf{a}}_i \circ \widehat{\mathbf{b}}_i \circ \widehat{\mathbf{c}}_i$.

Theoretical Analysis: Local and Global Convergence

- Goal: Quantify the rates of convergence of the estimators $\widehat{\mathbf{a}}_j$, $\widehat{\mathbf{b}}_j$, $\widehat{\mathbf{c}}_j$, and $\widehat{w}_j$ for each $j = 1, \ldots, K$.



Theoretical Analysis: Noisy Tensor Decomposition



- Observe the noisy tensor $\widehat{\mathcal{T}} = \mathcal{T} + \mathcal{E}$ where

$$\mathcal{T} = \sum_{i=1}^{K} w_i \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i$$

- Require assumptions on true tensor $\mathcal{T}$ and error tensor $\mathcal{E}$.

Theoretical Analysis: Key Assumptions

(A1) **Incoherence:** The decomposition components are incoherent s.t.

$$\max_{i \neq j} \{ |\langle \mathbf{a}_i, \mathbf{a}_j \rangle|, |\langle \mathbf{b}_i, \mathbf{b}_j \rangle|, |\langle \mathbf{c}_i, \mathbf{c}_j \rangle| \} \leq \frac{C}{\sqrt{d_0}},$$

for some constant C.

(A2) **Bounded error:** Define the sparse norm of $\mathcal{E}$ as

$$\rho(\mathcal{E}, m) := \sup_{\substack{\|\mathbf{u}\|=\|\mathbf{v}\|=\|\mathbf{w}\|=1 \\ \|\mathbf{u}\|_0 \leq m, \|\mathbf{v}\|_0 \leq m, \|\mathbf{w}\|_0 \leq m}} \left| \mathcal{E} \times_1 \mathbf{u} \times_2 \mathbf{v} \times_3 \mathbf{w} \right|.$$

Let $s = \max\{s_1, s_2, s_3\}$. For some constant C_0, assume

$$\rho(\mathcal{E}, d_0 + s) \leq \min \left\{ \frac{w_{\min}}{6}, \frac{w_{\min}\sqrt{\log K}}{C_0 \sqrt{d_0}} \right\}.$$

Theoretical Analysis: Local Convergence Analysis

$$\epsilon_R := \underbrace{C_1 \rho(\mathcal{E}, d_0 + s)}_{Sample\ \ error} + \underbrace{C_2 \frac{\sqrt{K}}{d_0}}_{Model\ \ error}$$

Theorem

If the initializations $\widehat{a}^{(0)}, \widehat{b}^{(0)}, \widehat{c}^{(0)}$ satisfy

$$\max\left\{ dist(\widehat{a}^{(0)}, a_j), dist(\widehat{b}^{(0)}, b_j) \right\} = O\left(\frac{w_{\min}}{w_{\max}}\right),$$

then our algorithm with $s \geq d_0$ satisfies w.h.p., for some $j \in [K]$,

$$\begin{aligned} \max\left\{ dist(\widehat{a}, a_j), dist(\widehat{b}, b_j), dist(\widehat{c}, c_j) \right\} &\leq O(\epsilon_R) \\ |\widehat{w} - w_j| &\leq O(\epsilon_R). \end{aligned}$$

Theoretical Analysis: Global Convergence Analysis

$$\epsilon_R := \underbrace{C_1 \rho(\mathcal{E}, d_0 + s)}_{Sample\ \ error} + \underbrace{C_2 \frac{\sqrt{K}}{d_0}}_{Model\ \ error}$$

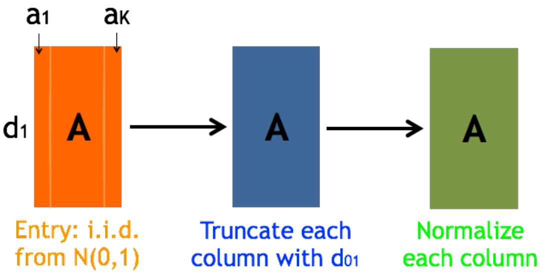
Theorem

For any $j \in [k]$, the output of our algorithm with $s \geq d_0$ using sparse SVD initialization satisfies, w.h.p.,

$$\begin{aligned} \max\left\{ dist(\widehat{a}_j, a_j), dist(\widehat{b}_j, b_j), dist(\widehat{c}_j, c_j) \right\} &\leq O(\epsilon_R), \\ |\widehat{w} - w_j| &\leq O(\epsilon_R). \end{aligned}$$

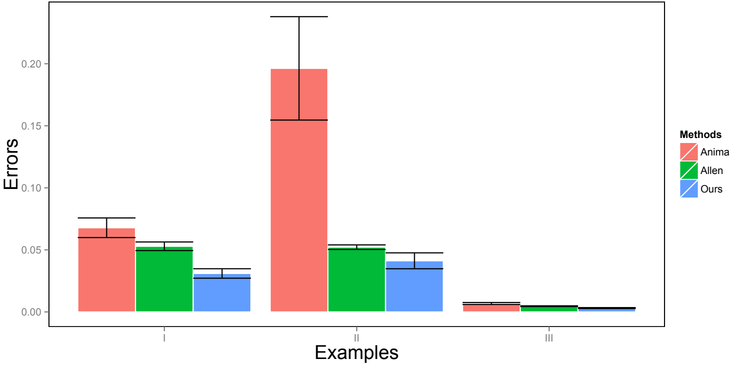
- Non-sparse tensor decomposition (Anandkumar et al., 2014) obtained an estimation error $O(\rho(\mathcal{E}, d) + \sqrt{K}/d)$.
- In high-dim regime, it is slower than ours.

Simulation 1: Sparse Tensor Recovery



- Generate $\widehat{\mathcal{T}} = \mathcal{T} + \mathcal{E}$
- $(d_1, d_2, d_3) = (1000, 100, 10)$ and $d_{0j} = 0.2 * d_j$.
  Example I: $[\mathcal{E}]_{i,j,k} \sim N(0,1), \quad K = 1;$
  Example II: $[\mathcal{E}]_{i,j,k} \sim N(0,1), \quad K = 2;$
  Example III: $[\mathcal{E}]_{i,j,k} \sim N(0,0.1), K = 1.$

Simulation 1: Estimation Accuracy



$$\epsilon_R := \underbrace{C_1 \rho(\mathcal{E}, d_0 + s)}_{Sample\ \ error} + \underbrace{C_2 \sqrt{K} / d_0}_{Model\ \ error}$$

Simulation 1: Variable Selection

Examples	Methods	TPR	FPR
I	Anima	1_0	1_0
	Allen	1_0	$0.003_{0.0022}$
	Ours	1_0	$0.016_{0.0130}$
II	Anima	1_0	1_0
	Allen	1_0	$0.002_{0.0016}$
	Ours	1_0	$0.067_{0.0311}$
III	Anima	1_0	1_0
	Allen	1_0	$0.002_{0.0022}$
	Ours	1_0	0_0

Simulation 2: Sparse Gaussian Mixture Model

- $\boldsymbol{x}_i \sim \sum_k w_k N(\boldsymbol{\mu}_k, 0.1 * \mathbb{1}) : n = 1000, d = 10, K = 4, w_k = \frac{1}{4}$
  $\boldsymbol{\mu}_1 = \mathbf{e}_1 + 0.2\mathbf{e}_2, \boldsymbol{\mu}_2 = \mathbf{e}_2 + 0.2\mathbf{e}_3$
  $\boldsymbol{\mu}_3 = \mathbf{e}_3 + 0.2\mathbf{e}_4, \boldsymbol{\mu}_4 = \mathbf{e}_4 + 0.2\mathbf{e}_1$

- **Step 1:** Estimate $\mathbb{E}[\boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{x}]$ and $\mathbb{E}[\boldsymbol{x}]$ to obtain $\widehat{\mathcal{M}}$ for

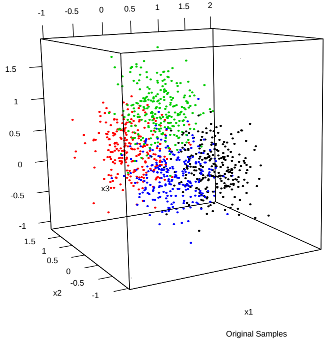
$$\mathcal{M} = \mathbb{E}[\boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{x}] - \sigma^2 f(\mathbb{E}[\boldsymbol{x}])$$

- **Step 2:** Apply sparse tensor decomposition on $\widehat{\mathcal{M}}$ to solve

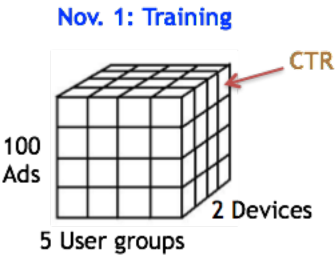
$$\widehat{\mathcal{M}} \approx \sum_{k=1}^{K} \widehat{w}_k \widehat{\boldsymbol{\mu}}_k \otimes \widehat{\boldsymbol{\mu}}_k \otimes \widehat{\boldsymbol{\mu}}_k.$$

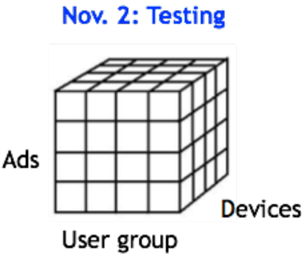
Simulation 2: Reconstruction Performance

- Left: original samples; Right: reconstructed samples.



Real Application 1: Click-through Rate Prediction

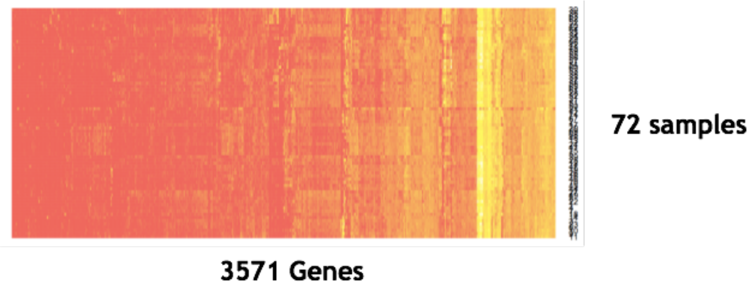




Methods	Training error	Testing error
Linear regression	0.189	0.534
Gradient boosting machine	0.190	0.533
Ours	**0.141**	**0.511**

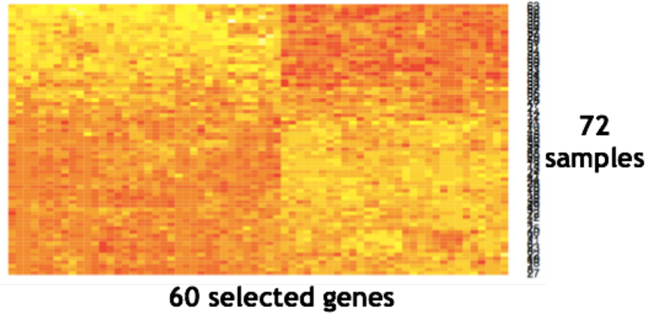
Real Application 2: High-dim Gene Clustering

- Leukemia data: cluster samples into 2 groups.

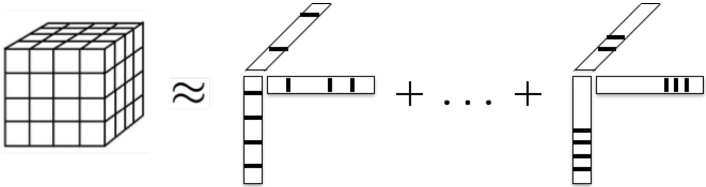


Real Application 2: High-dim Gene Clustering

Methods	No. genes	cluster error
K-means	3571	2/72
Reg. k-means (S. et al., 2012)	211	2/72
Ours	**60**	2/72



Summary

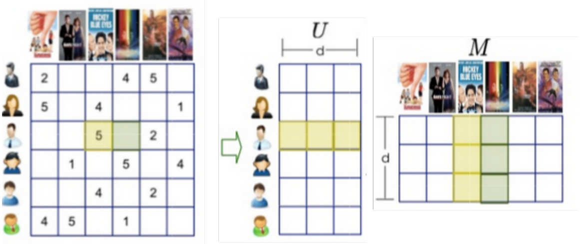


- new sparse tensor decomposition algorithm via ℓ_0 truncation
- local/global rates of convergence, faster than non-sparse one
- personalized recommendation, high-dim latent variable models

Future Work: Statistical-and-Computational Tradeoffs

- A function $g(\boldsymbol{a}, \mathbf{b}) : \mathcal{A} \times \mathcal{B} \to \mathbb{R}$ is biconvex if $g(\boldsymbol{a}, \mathbf{b})$ is convex in $\boldsymbol{a}$ for fixed $\mathbf{b} \in \mathcal{B}$, and convex in $\mathbf{b}$ for fixed $\boldsymbol{a} \in \mathcal{A}$.
- Biconvex optimization:

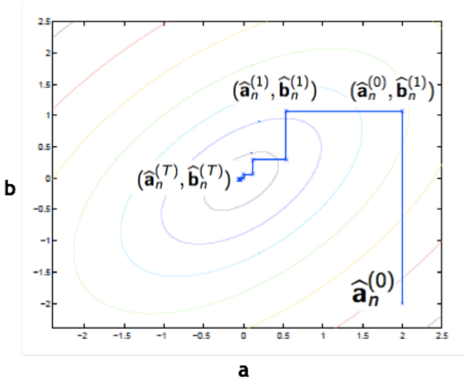
$$
\begin{aligned}
&\min \;\; g(\boldsymbol{a}, \mathbf{b}) \\
&s.t. \;\; \boldsymbol{a} \in \mathcal{A}, \mathbf{b} \in \mathcal{B}
\end{aligned}
$$



Source: A. Karatzoglou, ESSIR 2013 Recommender Systems tutorial

Statistical-and-Computational Tradeoffs: Problem

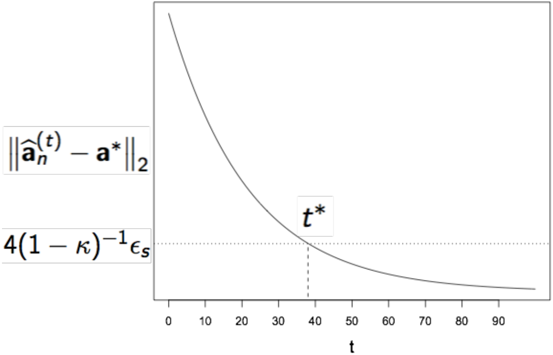
- Population version: $(\mathbf{a}^*, \mathbf{b}^*) = \arg\min_{\mathbf{a}\in\mathcal{A}, \mathbf{b}\in\mathcal{B}} \; g(\mathbf{a}, \mathbf{b})$
- Goal: Find $(\widehat{\mathbf{a}}, \widehat{\mathbf{b}})$ via sample $g_n(\mathbf{a}, \mathbf{b})$ s.t. $\|\widehat{\mathbf{a}} - \mathbf{a}^*\|_2$ and $\|\widehat{\mathbf{b}} - \mathbf{b}^*\|_2$ are small given limited computational resources.



Statistical-and-Computational Tradeoffs: Main Result

$$\left\|\widehat{\mathbf{a}}_n^{(t)} - \mathbf{a}^*\right\|_2 \le \underbrace{2(1-\kappa)^{-1}\epsilon_s}_{\text{Statistical Error}} + \underbrace{\kappa^t \epsilon_0}_{\text{Optimization Error}}$$

- ϵ_s: error due to sample function; ϵ_0: initialization error.
- Constant $\kappa < 1$.



Sparse Tensor Graphical Model (S. et al., 2015, NIPS)

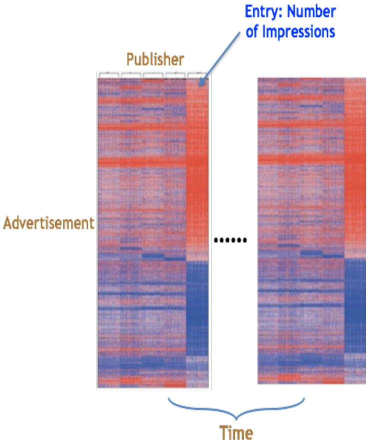


Figure: Tensor data

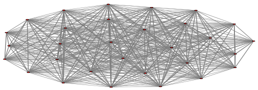


Figure: Advertisement network

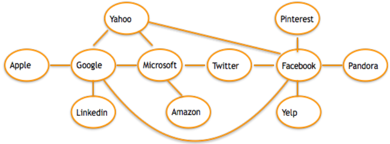


Figure: Publisher network

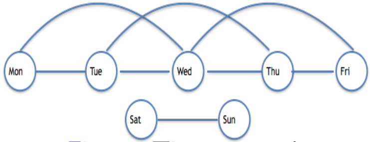


Figure: Time network

Wei Sun
Yahoo Labs
sunweisurrey@yahoo-inc.com

Backup Slides

- Backup slides start from here!

Initalization via Sparse SVD

Input: tensor $\widehat{\mathcal{T}}$, cardinality parameter (s_1, s_2, s_3)

Step 1: Generate a d_3-dim standard Gaussian vector $\boldsymbol{\theta}$.

Step 2: $\check{\boldsymbol{\theta}} = \mathrm{Truncate}(\boldsymbol{\theta}, \max\{s_1, s_2, s_3\})$.

Step 3: Calculate top left (right) singular vectors $\mathbf{u}_1$ $(\mathbf{v}_1)$ of $\widehat{\mathcal{T}} \times_3 \check{\boldsymbol{\theta}}$.

Step 4: $\check{\mathbf{u}}_1 = \mathrm{Truncate}(\mathbf{u}_1, s_1)$ and $\check{\mathbf{v}}_1 = \mathrm{Truncate}(\mathbf{v}_1, s_2)$.

Step 5: $\widehat{\mathbf{a}}_\tau^{(0)} = \mathrm{Norm}(\check{\mathbf{u}}_1)$, $\widehat{\mathbf{b}}_\tau^{(0)} = \mathrm{Norm}(\check{\mathbf{v}}_1)$, and update $\widehat{\mathbf{c}}_\tau^{(0)}$.

Output: $(\widehat{\mathbf{a}}_\tau^{(0)}, \widehat{\mathbf{b}}_\tau^{(0)}, \widehat{\mathbf{c}}_\tau^{(0)})$.

- Intuition:

$$\mathcal{T} \times_3 \check{\boldsymbol{\theta}} = \sum_{i \in [K]} \underbrace{w_i(\mathbf{c}_i^\top \check{\boldsymbol{\theta}})}_{singular\ \ value} \underbrace{\mathbf{a}_i \mathbf{b}_i^\top}_{singular\ \ vectors} \in \mathbb{R}^{d_1 \times d_2}$$

Clustering Procedure

Input: tensor $\widehat{\mathcal{T}}$, set $\left\{(\widehat{\mathbf{a}}_\tau, \widehat{\mathbf{b}}_\tau, \widehat{\mathbf{c}}_\tau), \tau \in [L]\right\}$.

For $j = 1$ **to** K **Do**

Step 1: Find $(\widehat{\mathbf{a}}, \widehat{\mathbf{b}}, \widehat{\mathbf{c}}) = \arg\max_{(\mathbf{a},\mathbf{b},\mathbf{c}) \in S} |\widehat{\mathcal{T}} \times_1 \mathbf{a} \times_2 \mathbf{b} \times_3 \mathbf{c}|$.

Step 2: Perform alternative update steps with initialization $(\widehat{\mathbf{a}}, \widehat{\mathbf{b}}, \widehat{\mathbf{c}})$.

Step 3: Output the cluster center as the final update in Step 2.

Step 4: Remove tupes with $\min\{\|\widehat{\mathbf{a}}_\tau \pm \widehat{\mathbf{a}}\|, \|\widehat{\mathbf{b}}_\tau \pm \widehat{\mathbf{b}}\|, \|\widehat{\mathbf{c}}_\tau \pm \widehat{\mathbf{c}}\|\} \leq 0.5$.

End For

Output: $\{(\widehat{\mathbf{a}}_j, \widehat{\mathbf{b}}_j, \widehat{\mathbf{c}}_j), j \in [K]\}$.

- Intuition 1: if $|\widehat{\mathcal{T}} \times_1 \mathbf{a} \times_2 \mathbf{b} \times_3 \mathbf{c}|$ is large for some $(\mathbf{a}, \mathbf{b}, \mathbf{c})$, then it is close to some $(\mathbf{a}_j, \mathbf{b}_j, \mathbf{c}_j)$.
- Intuition 2: if $(\widehat{\mathbf{a}}_\tau, \widehat{\mathbf{b}}_\tau, \widehat{\mathbf{c}}_\tau)$ is close to $(\mathbf{a}, \mathbf{b}, \mathbf{c})$, then their distance is *very* small; otherwise their distance is *very* large.

Illustration of Clustering Procedure

- $d_1 = d_2 = d_3 = 100, d_{01} = d_{02} = d_{03} = 50, K = 5.$
- Distance: $\min\{\|\widehat{\mathbf{a}}_\tau \pm \widehat{\mathbf{a}}\|, \|\widehat{\mathbf{b}}_\tau \pm \widehat{\mathbf{b}}\|, \|\widehat{\mathbf{c}}_\tau \pm \widehat{\mathbf{c}}\|\}.$

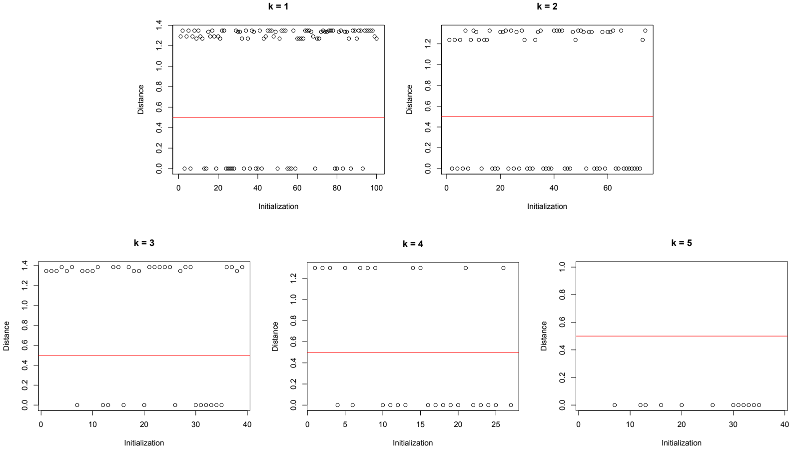


Illustration of Tuning Procedure

- $(d_1, d_2, d_3) = (40, 30, 20)$, $d_{0j} = 0.2 * d_j$, and $K = 3$