Generalized Tensor Decompositions for Non-Normal Data

Tamara G. Kolda, Sandia Natl. Labs.
Joint work with David Hong (Michigan), Jed Duersch (SNL)
A Tensor is an Multi-Way Array

Vector
\( d = 1 \)

Matrix
\( d = 2 \)

3\(^{rd}\)-order Tensor
\( d = 3 \)

\( d^{th}\)-order Tensor
\( d > 3 \)
Tensors Come From Many Applications

- **Chemometrics**: Emission x Excitation x Samples (Fluorescence Spectroscopy)
- **Neuroscience**: Neuron x Time x Trial (Calcium Imaging)
- **Criminology**: Day x Hour x Location x Crime (Chicago Crime Reports)
- **Medicine**: Channel x Wavelength x Time (EEG measurements)
- **Sports**: Player x Statistic x Season
- **Cyber-Traffic**: IP x IP x Port x Time
- **Social Network**: Person x Person x Time x Interaction-Type
Tensor Decomposition: A Mathematical & Statistical Tool for Analysis of Tensor Data

Express the tensor as the sum of meaningful parts, which is the starting point for data analysis activities

Includes visualization, clustering, filling in missing entries, etc.

Mathematics/Statistics play a role in....
• Defining the error metric
• Developing efficient algorithms

Related Concepts for Matrices
• Singular value decomposition (SVD)
• Principal component analysis (PCA)
• Independent component analysis (ICA)
• Nonnegative matrix factorization (NMF)
• Sparse matrix factorization
• Matrix completion
Break Tensor into Understandable Parts...

\[ \chi \approx \mathcal{M} = \text{Part 1} + \text{Part 2} + \cdots + \text{Part } r \]

**Data Tensor** 
\( n_1 \times n_2 \times n_3 \)

**Model Tensor** 
\( n_1 \times n_2 \times n_3 \)

**Part** 
\( n_1 \times n_2 \times n_3 \)

**Key:** The parts have structure!
Rank-1 Tensors are the “Parts”

Given \( d \) vectors:

\[
a_k \in \mathbb{R}^{n_k} \text{ for } k = 1, \ldots, d
\]

The outer product is

\[
P = a_1 \circ a_2 \cdots \circ a_d \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}
\]

\[
P(i_1, i_2, i_3) = a_1(i_1) a_2(i_2) a_3(i_3)
\]
CANDECOMP/PARAFAC (CP) Tensor Factorization Uncovers the Rank-1 Parts

Images are three-way ($d = 3$), but assume all tensors are of size $n_1 \times n_2 \times \cdots \times n_d$

WLOG, $n = n_1 = \cdots = n_d$

$X \approx M$ where $M = \sum_{j=1}^{r} A_1(:,j) \odot A_2(:,j) \odot \cdots \odot A_d(:,j)$

Low-rank: $\text{rank}(M) \leq r \ll n^d$

Factor matrices: $A_k \in \mathbb{R}^{n_k \times r}$ for $k \in \{1, \ldots, d\}$

Hitchcock, 1927; Carroll and Chang, 1970; Harshman, 1970
CP first invented in 1927

F. L. Hitchcock, *The Expression of a Tensor or a Polyadic as a Sum of Products*, Journal of Mathematics and Physics, 1927

2. Polyadic form of a tensor.

Any covariant tensor $A_{i_1 \cdots i_p}$ can be expressed as the sum of a finite number of tensors each of which is the product of $p$ covariant vectors.

\[ A_{i_1 \cdots i_p} = \sum_{j=1}^{j=h} a_{i_1j} a_{i_2j} \cdots a_{i pj} 
\]

where $a_{ij}$, $i_j$, etc., are a set of $kp$ covariant vectors. When the indices $i_1 \cdots i_p$ can be omitted this may be written

\[ A = \sum_{j=1}^{j=h} a_{ij} a_{2j} \cdots a_{pj}. \]
Many thanks to the following persons for helping me learn about Jih-Jie Chang: Fan Chung, Ron Graham, Shen Lin (husband), May Chang (niece), Lili Bruer (daughter).

J. Douglas Carroll
Bell Labs
(1939-2011)

Jih-Jie Chang
Bell Labs
(1927-2007)

Richard A. Harshman
Univ. Ontario
(1943-2008)

In 2000, Henk Kiers proposed this compromise name.

2010: Pierre Comon, Lieven DeLathauwer, and others reverse-engineered CP, revising some of Hitchcock’s terminology.

CP Independently Reinvented (twice) in 1970
Standard CP: Sum of Squares Error (SSE)

Shorthand for element of data tensor:

\[ x_i \equiv x(i_1, i_2, \ldots, i_d) \]

Element of model low-rank tensor:

\[ m_i \equiv \sum_{j=1}^{r} \prod_{k=1}^{d} A_{k}(i_k, j) \]

(defined in terms of factor matrices)

\[ \Omega = \text{set of all } n^d \text{ elements in tensor} \]

\[
\min_{\mathcal{M}} F(\mathcal{X}, \mathcal{M}) \equiv \sum_{i \in \Omega} (x_i - m_i)^2
\]

s.t. \( \text{rank}(\mathcal{M}) \leq r \)

Hitchcock, 1927; Carroll and Chang, 1970; Harshman, 1970
Generalized CP (GCP)

Why?

• SSE: maximum likelihood estimate (MLE) for Gaussian distribution
  \[ x_i = m_i + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma) \]
  \[ x_i \sim \mathcal{N}(m_i, \sigma) \]

• Different MLEs for different distributions
  – Poisson (counts)
  – Bernoulli (binary)

Hong, Kolda, Duersch, SIAM Review, 2019
Probability Distribution ⇒ Maximum Likelihood Estimator

\[ x_i \sim p(x_i | \theta_i) \text{ where } \ell(\theta_i) = m_i \]

Data Value  \quad “Natural” Parameter  \quad Model Value  \quad Link Function

Probability Distribution Function (PDF) or Probability Mass Function (PMF)

Maximize Likelihood of Data Tensor
\[ \prod_{i \in \Omega} p(x_i, \theta_i) \]

Maximize Log-Likelihood
\[ \sum_{i \in \Omega} \log p(x_i, \theta_i) \]

\[ \text{min } F(\mathbf{X}, \mathbf{M}) \equiv \sum_{i \in \Omega} f(x_i, m_i) \]

s.t. \( \text{rank} (\mathbf{M}) \leq r \)

Given PDF/PMF \( p(x | \theta) \) and link function \( \ell(\theta) \), GCP MLE by minimizing
\[ f(x, m) = - \log p(x, \ell^{-1}(m)) \]

Hong, Kolda, Duersch, SIAM Review, 2019
Gaussian MLE (Standard CP)

PDF for Normal Distribution

\[ p(x \mid \mu, \sigma) = \frac{e^{-(x-\mu)^2 / 2\sigma^2}}{\sqrt{2\pi\sigma^2}} \]

and

Link Function

\[ m = \mu \]

\sigma \text{ constant}

Negative log-likelihood:

\[ -\log p(x \mid \mu, \sigma) = \frac{(x-\mu)^2}{2\sigma^2} + \frac{1}{2} \log(2\pi\sigma^2) \]

Eliminate natural parameter via link function:

\[ f(x, m) = \frac{(x-m)^2}{2\sigma^2} + \frac{1}{2} \log(2\pi\sigma^2) \]

Eliminate constants:

\[ f(x, m) = (x - m)^2 \]

Hong, Kolda, Duersch, SIAM Review, 2019
Bernoulli random variable

\[ x \in \{0,1\} \]

\[ \rho = \text{probability of a 1} \]

\[ p(x | \rho) = \rho^x (1 - \rho)^{1-x}, \quad x \in \{0,1\} \]

**PMF for Bernoulli Distribution**

\[ p(x | \rho) = \rho^x (1 - \rho)^{1-x} \]

\[ x \in \{0,1\} \]

**Link Function**

\[ m = \frac{\rho}{(1 - \rho)} \]

**Odds Link**

\[ \ell(\rho) = \frac{\rho}{1 - \rho} \]

\[ \ell^{-1}(m) = \frac{m}{1 + m} \]

**Negative log-likelihood:**

\[ -\log p(x | \rho) = \log \frac{1}{1 - \rho} - x \log \frac{\rho}{1 - \rho} \]

**Eliminate natural parameter via link function:**

\[ f(x, m) = \log(1 + m) - x \log m \quad \text{for} \quad m > 0 \]

Hong, Kolda, Duersch, SIAM Review, 2019
Bernoulli MLE with Odds Link (Binary Data)

Bernoulli random variable
\[ x \in \{0,1\} \]
\[ \rho = \text{probability of a } 1 \]

\[ p(x \mid \rho) = \rho^x (1 - \rho)^{(1-x)}, \quad x \in \{0,1\} \]

PMF for Bernoulli Distribution
\[ p(x \mid \rho) = \rho^x (1 - \rho)^{(1-x)}, \quad x \in \{0,1\} \]

Link Function
\[ m = \frac{\rho}{1 - \rho} \]

Negative log-likelihood:
\[ -\log p(x \mid \rho) = \log \frac{1}{1 - \rho} - x \log \frac{\rho}{1 - \rho} \]

Eliminate natural parameter via link function:
\[ f(x, m) = \log(1 + m) - x \log m \quad \text{for} \quad m > 0 \]

Hong, Kolda, Duersch, SIAM Review, 2019
Bernoulli MLE with Logit Link (Binary Data)

**Bernoulli random variable**

\[ x \in \{0,1\} \]

\[ \rho = \text{probability of a 1} \]

\[ p(x \mid \rho) = \rho^x (1 - \rho)^{(1-x)}, \quad x \in \{0,1\} \]

**PMF for Bernoulli Distribution**

\[ p(x \mid \rho) = \rho^x (1 - \rho)^{(1-x)} \quad x \in \{0,1\} \]

**Link Function**

\[ m = \log \left( \frac{\rho}{1 - \rho} \right) \]

**Logit (Log-Odds) Link**

\[ \ell(\rho) = \log \left( \frac{\rho}{1 - \rho} \right) \]

\[ \ell^{-1}(m) = \frac{e^m}{1 + e^m} \]

**Negative log-likelihood:**

\[ -\log p(x \mid \rho) = \log \frac{1}{1 - \rho} - x \log \frac{\rho}{1 - \rho} \]

**Eliminate natural parameter via link function:**

\[ f(x, m) = \log (1 + e^m) - xm \quad \text{for} \quad m \in \mathbb{R} \]

*Hong, Kolda, Duersch, SIAM Review, 2019*
Bernoulli MLE with Logit Link (Binary Data)

Bernoulli random variable
\[ x \in \{0,1\} \]
\[ \rho = \text{probability of a 1} \]

PMF for Bernoulli Distribution
\[ p(x \mid \rho) = \rho^x (1 - \rho)^{(1-x)} \]
\[ x \in \{0,1\} \]

Link Function
\[ m = \log \left( \frac{\rho}{1-\rho} \right) \]

Negative log-likelihood:
\[ -\log p(x \mid \rho) = \log \left( \frac{1}{1-\rho} \right) - x \log \left( \frac{\rho}{1-\rho} \right) \]

Eliminate natural parameter via link function:
\[ f(x, m) = \log(1 + e^m) - xm \quad \text{for} \quad m \in \mathbb{R} \]

Hong, Kolda, Duersch, SIAM Review, 2019
Sampling of Loss Functions

- **Standard CP**
  - Binary (Odds Link)
  - Count Data (Identity Link)
  - Nonnegative Data
  - Nonnegative Data (not MLE)
- **“Robust”**
  - Binary (Logit Link)
- **Nonnegative Data**
  - $m \geq 0$
- **Failure**
  - Count Data (Odds Link)
- **Count Data**
  - $m \geq 0$

Hong, Kolda, Duersch, SIAM Review, 2019
Example Tensor from Neuroscience

Activity of Single Neuron Measured Over Time Produces Vector Data

Thanks to Schnitzer Group @ Stanford
Mark Schnitzer, Fori Wang, Tony Kim

Microscope by Inscopix
mouse in maze
neural activity via calcium imaging

Williams et al., Neuron, 2018
Multiple Neurons Measured Over Time Produces Matrix

Thanks to Schnitzer Group @ Stanford
Mark Schnitzer, Fori Wang, Tony Kim

Microscope by Inscopix

mouse in “maze”

neural activity

Williams et al., Neuron, 2018

282 neurons × 111 time bins
Multiple Trials Produces 3-way Tensor

300 Trials over 5 Days
\- Start West
\- Conditions Swap Twice
\- Turn South
\- Turn North
\- Turn South

282 neurons $\times$ 111 time bins $\times$ 300 trials

Williams et al., Neuron, 2018
Example Neuron Activity

Thin lines show 300 individual trials

Thick line is average

Hong, Kolda, Duersch, SIAM Review, 2019
Neuron Factor Vector Visualized as Bar Chart

Hong, Kolda, Duersch, SIAM Review, 2019
Time Factor Vector Visualized as Line

Time (within trial) Plotted as a Line
(Dashed Line is Zero)

Hong, Kolda, Duersch, SIAM Review, 2019
Trial Factor Vector Visualized as Color-Coded Scatter Plot

Trial Plotted as Scatter Graph
Right turn = Green
Left turn = Orange
Filled = Reward

Rule Change

Hong, Kolda, Duersch, SIAM Review, 2019
Visualization of CP Tensor Decomposition
Shows the Factors (Vectors)

\[ a_1 \approx \text{neuron} \]

\[ b_1 \approx \text{time} \]

\[ c_1 \approx \text{Trial} \ (\text{Green/Orange} = \text{Turn Right/Left}, \text{Reward} = \text{Filled}) \]

Hong, Kolda, Duersch, SIAM Review, 2019
“Standard” CP Decomposition of Mouse Data, aka Gaussian $f(x, m) = (x - m)^2$
CP Tensor Decomposition “Sees” Reward

Neuron (scaled)  Time  Trial (Green/Orange = Turn Right/Left, Reward = Filled)
CP Tensor Decomposition “Sees” Turn Direction
CP Tensor Decomposition Can be Tough to Interpret due to Negative Entries

Reward!
Turn left
Turn right
Turn
GCP Decomposition with Beta Divergence

\( \beta = 0.5, f(x, m) = \sqrt{m} + x/\sqrt{m} \)
Trial Factor Matrix is $300 \times 8$

Look at predicting turn and reward.
Split into two groups of 150 trials.
Train regression model with 1\textsuperscript{st} group.
Test with 2\textsuperscript{nd} group.
Repeat 100 times.

\[
\min_{\beta} \| A_3^{\text{train}} \beta - y^{\text{train}} \| \\
\hat{y}^{\text{test}} = [A_3^{\text{test}} \beta \geq 0.5]
\]
Optimization Formulation for GCP Tensor Decomposition

\[
\min F(\mathbf{X}, \mathbf{M}) \equiv \sum_{i \in \Omega} f(x_i, m_i)
\]

\[
\text{s.t. } \text{rank}(\mathbf{M}) \leq r
\]

\(i\) = multi-index \hspace{1cm} \Omega = \text{all indices}

- Standard CP [Hitchcock, 1927; Carrol & Chang, 1970; Harshman, 1970]
  \[f(x, m) = (x - m)^2\]

- Poisson CP (Identity Link) [Welling & Webber, 2001; Chi & Kolda, 2009]
  \[f(x, m) = m - x \log m\]

- Logistic CP, etc. [Hong, Kolda, Duersch, 2018]
  \[f(x, m) = \log(m + 1) - x \log(m)\]

\(\mathbf{X} \approx \mathbf{M}\) where \(\mathbf{M} = \sum_{j=1}^{r} \mathbf{A}_1(:, j) \circ \mathbf{A}_2(:, j) \circ \cdots \circ \mathbf{A}_d(:, j)\)

Low-rank: \(\text{rank}(\mathbf{M}) \leq r \ll n^d\)

Factor matrices: \(\mathbf{A}_k \in \mathbb{R}^{n_k \times r}\) for \(k \in \{1, \ldots, d\}\)

\(\text{WLOG, } n = n_1 = \cdots = n_d\)
Gradient-based Optimization for Fitting the GCP Model

\[
\min \ F(\mathbf{X}, \mathbf{M}) = \sum_{i \in \Omega} f(x_i, m_i)
\]

\[
\text{s.t. } \text{rank}(\mathbf{M}) \leq r
\]

Define: Elementwise partial gradient tensor, same size as data tensor = \( n^d \)

\[
y_i = \frac{\partial f}{\partial m}(x_i, m_i)
\]

Define: Khatri-Rao product in all modes but one of size \( n^{d-1} \times r \)

\[
\mathbf{Z}_k = \mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1
\]

Gradients computed via a sequence of matricized-tensor times Khatri-Rao product (MTTKRPs):

\[
\mathbf{G}_k = \frac{\partial F}{\partial \mathbf{A}_k} = \mathbf{Y}(k) \mathbf{Z}_k \text{ for } k = 1, \ldots, d
\]

MTTKRPs can be computed efficiently...
- Bader & Kolda, SISC, 2007 – Dense and sparse
- Phan, Tichavsky, Cichocki, 2013 – Sequence
- Smith et al., IPDPS 2015 – Sparse
- Kaya & Ucar, SC 2015 – Sparse
- Li et al., IPDPS 2017 – Sparse
- Hayashi et al., 2017 – Dense
- Ballard, Knight, Rouse, 2017 – Dense
Stochastic Gradient Descent (SGD) for GCP

30-Second Tutorial on SGD

\[ \min F(x) \]

Gradient Descent (GD)
\[ \alpha = \text{learning rate} \]
\[ x^{(t+1)} = x^{(t)} - \alpha g^{(t)} \]

Stochastic Gradient Descent (SGD)
\[ x^{(t+1)} = x^{(t)} - \alpha \hat{g}^{(t)} \]
\[ \mathbb{E}[\hat{g}^{(t)}] = g^{(t)} \equiv \nabla F(x^{(t)}) \]

Adam (Kingma & Ba, 2015)
Adaptive momentum SGD

\[ \mathbf{G}_k = \mathbf{Y}_k \mathbf{Z}_k \quad \text{Cost: } O(\text{rn}^d) \text{ flops} \]

\[ y_i = \frac{\partial f}{\partial m}(x_i, m_i) \]

\[ \mathbf{\tilde{G}}_k = \mathbf{\tilde{Y}}_k \mathbf{Z}_k \quad \text{Cost: } O(\text{rs}) \text{ flops} \]

Choose stochastic sparse \( Y \)-tensor
\[ \mathbb{E}[\mathbf{\tilde{Y}}] = \mathbf{y} \]
such that
\[ \text{nnz}(\mathbf{\tilde{Y}}) \leq s \ll n^d \]

By linearity of expectation:
\[ \mathbb{E}[\mathbf{\tilde{G}}_k] = \mathbf{G}_k \]
Uniform Sampling

Goal: Random sparse tensor of size $n^d$ that equals the “Y-tensor” in expectation

Sample $s \ll n^d$ random tensor entries (with replacement)

\[ \tilde{s}_i = \# \text{ times } i \text{ sampled} \]

\[ \tilde{y}_i = \frac{n^d}{s} \cdot y_i \]

\[ \sum_{i \in \Omega} \tilde{s}_i = s \]

\[ y_i = \frac{\partial f}{\partial m}(x_i, m_i) \]

Claim: \[ \mathbb{E}[\tilde{y}] = y \]

Proof: \[ \mathbb{E}[\tilde{s}_i] = \frac{s}{n^d} \]

\[ \mathbb{E}[\tilde{y}_i] = \mathbb{E}[\tilde{s}_i] \cdot \frac{n^d}{s} \cdot y_i = y_i \]

Choosing $s$, the number of sampled elements...

- Choose $s = O(n)$
- Gradient = $O(rs) = O(rn)$ versus $O(rn^d)$

Downside...

- If data tensor is sparse, few entries corresponding to nonzeros will be chosen
Stratified 0/1 Sampling

**Goal**: Random *sparse* tensor of size $n^d$ that equals the “Y-tensor” in expectation

Sample $p$ nonzeros and $q$ zeros.

\[
\tilde{p}_i = \# \text{ times nonzero } i \text{ sampled} \quad \eta = \# \text{ nonzeros} \\
\tilde{q}_i = \# \text{ times zero } i \text{ sampled} \quad \zeta = \# \text{ zeros} \\
\tilde{y}_i = \left( \frac{\tilde{p}_i \cdot \eta}{p} + \frac{\tilde{q}_i \cdot \zeta}{q} \right) \cdot y_i \quad y_i = \frac{\partial f}{\partial m}(x_i, m_i)
\]

**Claim**: \( \mathbb{E}[\tilde{y}] = y \)

**Proof**:

\[
\mathbb{E}[\tilde{p}_i] = \frac{p}{\eta}, \quad \mathbb{E}[\tilde{q}_i] = \frac{q}{\zeta} \\
x_i = 1 \Rightarrow \mathbb{E}[\tilde{y}_i] = \mathbb{E}[\tilde{p}_i] \cdot \frac{\eta}{p} \cdot y_i = y_i \\
x_i = 0 \Rightarrow \mathbb{E}[\tilde{y}_i] = \mathbb{E}[\tilde{q}_i] \cdot \frac{\zeta}{q} \cdot y_i = y_i
\]

Needell, Srebro, and Ward (2013) justify **biased sampling** toward functionals with higher Lipschitz smoothness constants to reduce the variance in the stochastic gradient.
Semi-Stratified 0/1 Sampling

**Goal:** Random **sparse** tensor of size $n^d$ that equals the “Y-tensor” in expectation

Sample $p$ nonzeros and $q$ **assumed** zeros.

$$\tilde{p}_i = \# \text{ times nonzero } i \text{ sampled} \quad \eta = \# \text{ nonzeros}$$

$$\tilde{q}_i = \# \text{ times “zero” } i \text{ sampled} \quad \zeta = \# \text{ zeros}$$

$$\tilde{y}_i = \tilde{p}_i \cdot \frac{\eta}{p} \cdot (y_i - c_i) + \tilde{q}_i \cdot \frac{(\eta + \zeta)}{q} \cdot c_i \quad \text{with } c_i = \frac{\partial f}{\partial m} (0, m_i)$$

$$y_i = \frac{\partial f}{\partial m} (x_i, m_i)$$

---

**Claim:** $\mathbb{E}[\tilde{y}] = y$

**Proof:**

$$\mathbb{E}[\tilde{p}_i] = \frac{p}{\eta}, \quad \mathbb{E}[\tilde{q}_i] = \frac{q}{(\zeta + \eta)}$$

$$x_i = 0 \Rightarrow \mathbb{E}[\tilde{y}_i] = \mathbb{E}[\tilde{q}_i] \cdot \frac{q}{(\zeta + \eta)} \cdot y_i = y_i$$

$$x_i = 1 \Rightarrow \mathbb{E}[\tilde{y}_i] = \mathbb{E}[\tilde{p}_i] \cdot \frac{\eta}{p} \cdot (y_i - c_i) + \mathbb{E}[\tilde{q}_i] \cdot \frac{\eta + \zeta}{q} \cdot c_i = y_i$$
GCP with Stochastic Optimization

- Nonconvex problem
  - No guarantees of finding minimizer
- Using Adam (Kingma & Ba, 2015)
  - Default parameters
  - Some tweaks for checking convergence
- Past work on recommender systems uses SGD but ignores zeros
  - Gemulla, Nijkamp, Hass, Sismanis, KDD'11
  - Zhuang, Chin, Juan, and Lin, RecSys'13
- Past work on streaming uses SGD but data appears one slice at a time
  - Mardani, Mateos, Giannakis, IEEE TSP 2015
  - Maehara, Hayashi, Kawarabayashi,
Example on Gamma-Distributed Data

200 × 150 × 100 × 50 Tensor with low-rank \((r = 5)\) structure based on Gamma distribution \((k = 1, \theta\) from model). Gamma loss: \(f(x, m) = \frac{x}{m} + \log m\). Running stochastic GCP with 25 random starts and varying numbers of samples.

Dashed lines: Individual runs, Solid lines: Median, Epoch: Asterisk (success), Dot (fail).

Success at Recovering Underlying Generative Factors

\[ \text{estimated loss (100,000 samples)} \]

\[ \times 10^7 \]

\[ \begin{align*}
\text{samples} &= 125, \\
\text{samples} &= 250, \\
\text{samples} &= 500, \\
\text{samples} &= 1000, \\
\text{samples} &= 2000 \\
\text{nominal (true solution)}
\end{align*} \]

\[ \begin{align*}
\text{number of true solution recoveries} \\
\text{gradient samples} &= 125, 250, 500, 1000, 2000
\end{align*} \]
200 × 150 × 100 × 50 Tensor with low-rank \((r = 5)\) structure based on Gamma distribution \( (k = 1, \theta \text{ from model})\).

Gamma loss: 
\[
f(x, m) = \frac{x}{m} + \log m.
\]

Running stochastic GCP with 25 random starts.

Each asterisk is an iteration.

Same as prior slide, but rescaled x-axis.
Example on Bernoulli-Distributed Data

200 × 150 × 100 × 50 Tensor with low-rank ($r = 5$) structure based on Bernoulli distribution (odds from model).

Sparse tensor, less than 0.35% dense (~500K nonzeros).

Bernoulli loss: $f(x, m) = \log(m + 1) - x \log m$. Running stochastic GCP with 25 random starts, varying # of samples.

Success at Recovering Underlying Generative Factors
Uniform Sampling is Worse than Stratified for Sparse Tensors

Same set-up as binary experiments, but bigger tensor: $400 \times 300 \times 200 \times 100$, 0.38% dense (9M nonzeroes). Using $s = 1000$ samples in every case.
Chicago Crime Data

- 4-way count tensor
  - 6,186 Days
  - 24 Hours of the Day
  - 77 Community Areas
  - 32 Crime Types
- Non-zeros: 5,330,673
  - Storage: 0.21GB for sparse tensor
- Distribution of entries
  - 0: 98.54%
  - 1: 1.33%
  - ≥ 2: 0.12%
- Data originally from Chicago Data Portal ([https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2](https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2))
Application to Sparse Crime Binary Tensor (Semi-stratified results)

Date (Tick = 1 Year)

Hour

Neighborhood

Crime
Component #3

![Graphs and maps showing data analysis and crime rates.](image)

- Date: Various data points are shown over the years 2001 to 2017.
- Hour of Day: A histogram indicating activity levels.
- Top Crimes: A bar chart showing the frequency of different crimes.
- Areas: A map with color gradients representing crime rates.
Component #6

Date

Hour of Day

Top Crimes

deceptive practice
theft
other offense
offense involving children
battery
criminal damage

Areas

Kolda - SDSS 2019, Bellevue, WA
Aside: Estimating Higher-Order Moments via Symmetric Tensor Factorization

Given a set of $p$ observations: $a_i \in \mathbb{R}^n$, $i = 1, 2, \ldots, p$

First-order moment (mean): \[
\frac{1}{p} \sum_{i=1}^{p} a_i
\]

Second-order moment: \[
\frac{1}{p} \sum_{i=1}^{p} a_i \odot a_i
\]

Third-order moment: \[
\frac{1}{p} \sum_{i=1}^{p} a_i \odot a_i \odot a_i
\]

Fourth-order moment: \[
\frac{1}{p} \sum_{i=1}^{p} a_i \odot a_i \odot a_i \odot a_i
\]

We can compute low-rank ($r \ll p$) symmetric tensor estimated to higher-order moments...

What are good applications, if any?

Joint work with Sam Sherman, Notre Dame
SIAM JOURNAL ON
Mathematics of Data Science

SIAM Journal on Mathematics of Data Science (SIMODS) publishes work that advances mathematical, statistical, and computational methods in the realm of data and information sciences.

We invite papers that present significant advances in this context, including applications to science, engineering, business, and medicine.

journals.siam.org/simods
References & Collaborators

My department is hiring statisticians! Talk to me to learn more.


- **Stochastic GCP** - D. Hong, T. G. Kolda. Stochastic Gradients for Large-Scale Tensor Decomposition, to appear on arXiv very soon!


- **LDRD project team** - Cliff Anderson-Bergman (LLNL), Grey Ballard (Wake Forrest), Jed Duersch (SNL), Karen Devine (SNL), Srinivas Eswar (Georgia Tech), David Hong (Michigan), Jiajia Li (PNNL), Eric Phipps (SNL), Rich Vuduc (Georgia Tech), Jeff Young (Georgia Tech)

For more information and references: [www.kolda.net](http://www.kolda.net)