SAND2017-8553C

Sandia National Laboratories

Illustration by Chris Brigman

Generalized Canonical Tensor Factorization for Binary, Count, Nonnegative, or Real-Valued Data

Tamara G. Kolda & Cliff Anderson-Bergman
Sandia National Labs, CA

David Hong
University of Michigan

SIAM Annual Meting (AN17), Pittsburgh, PA
MS69: High Performance Tensor Computations
July 13, 2017

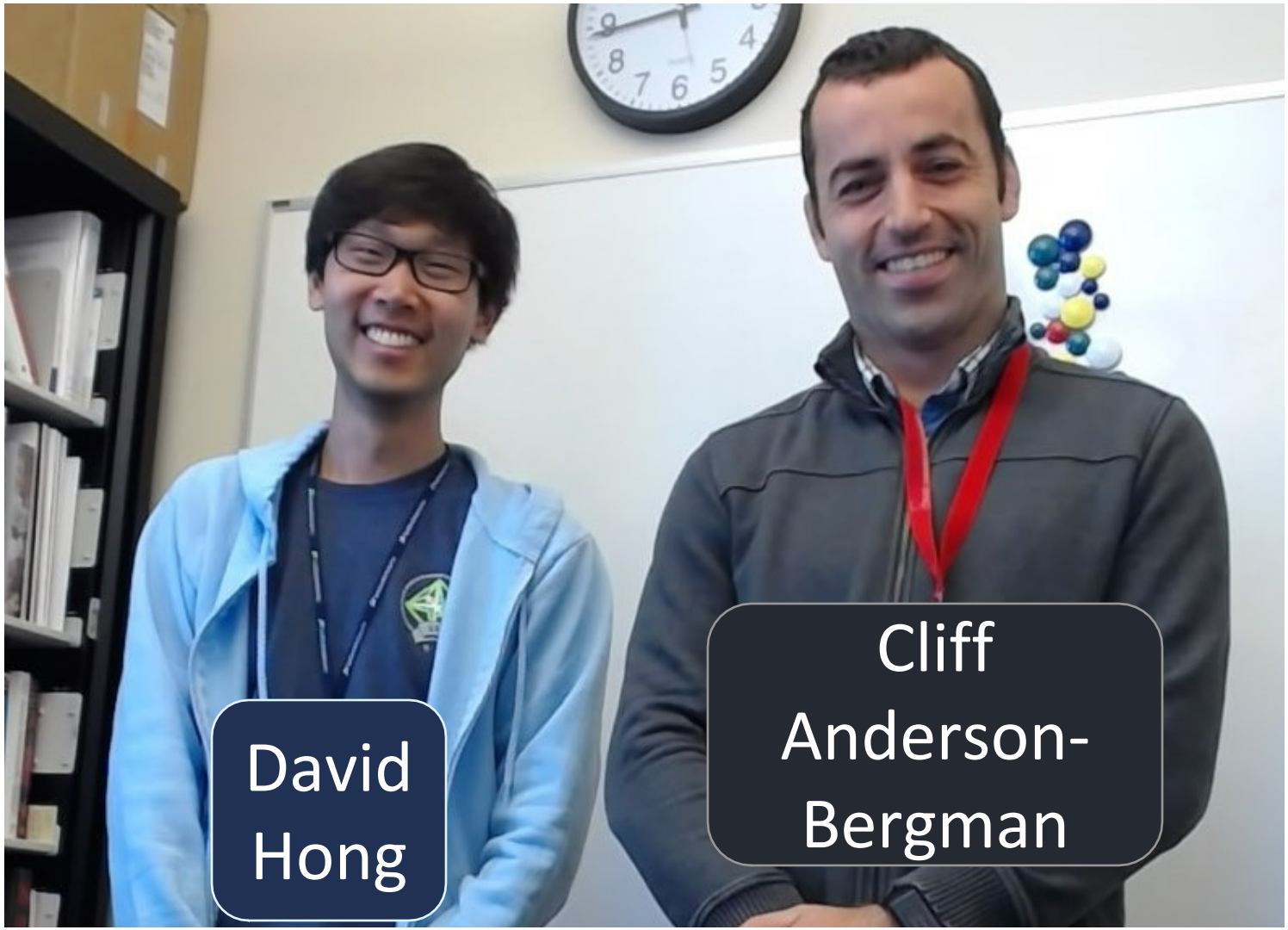
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

U.S. DEPARTMENT OF ENERGY

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

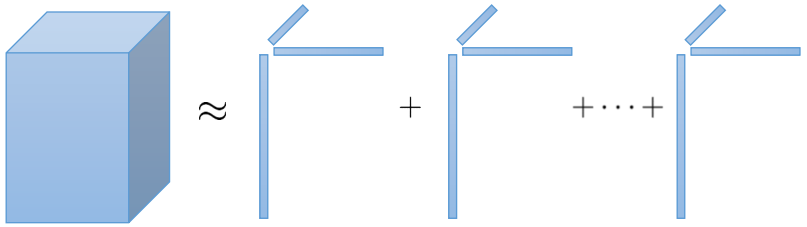
David Hong and Cliff Anderson-Bergman

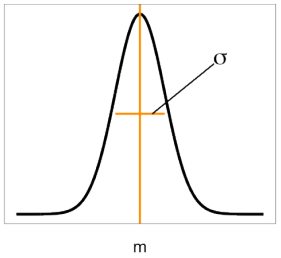
Sandia National Laboratories

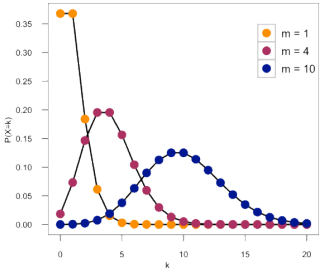


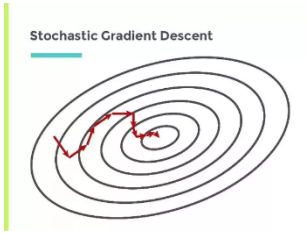
Summary: CP with Data Flexibility

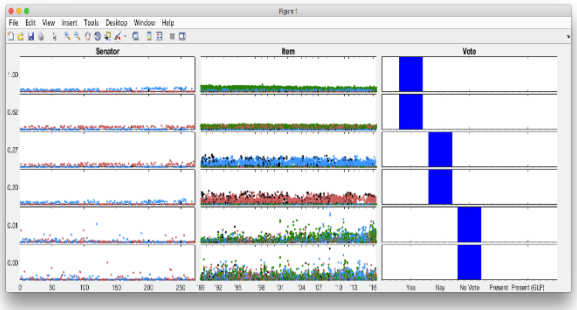
- Standard CP makes *implicit* statistical assumption
  - Normal-distributed
- <u>New</u> generalized CP allows other statistical assumptions
  - Bernoulli (0/1)
  - Poisson (counts)
- Elegant formula for gradients
  - Amenable to stochastic optimization methods
  - Takes advantage of existing kernels for sparse tensors
- Experimental results show advantages in interpretation





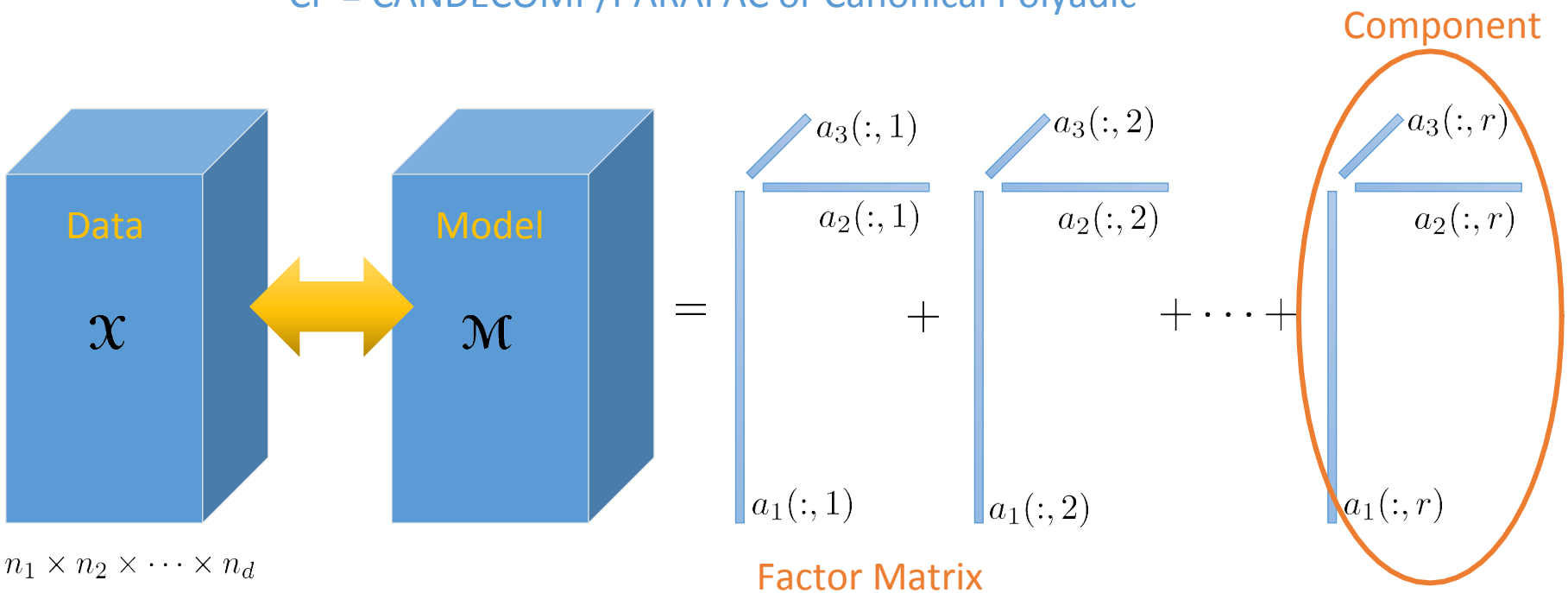






CP Tensor Decomposition

CP = CANDECOMP/PARAFAC or Canonical Polyadic



$n_1 \times n_2 \times \cdots \times n_d$

d = order

Factor Matrix

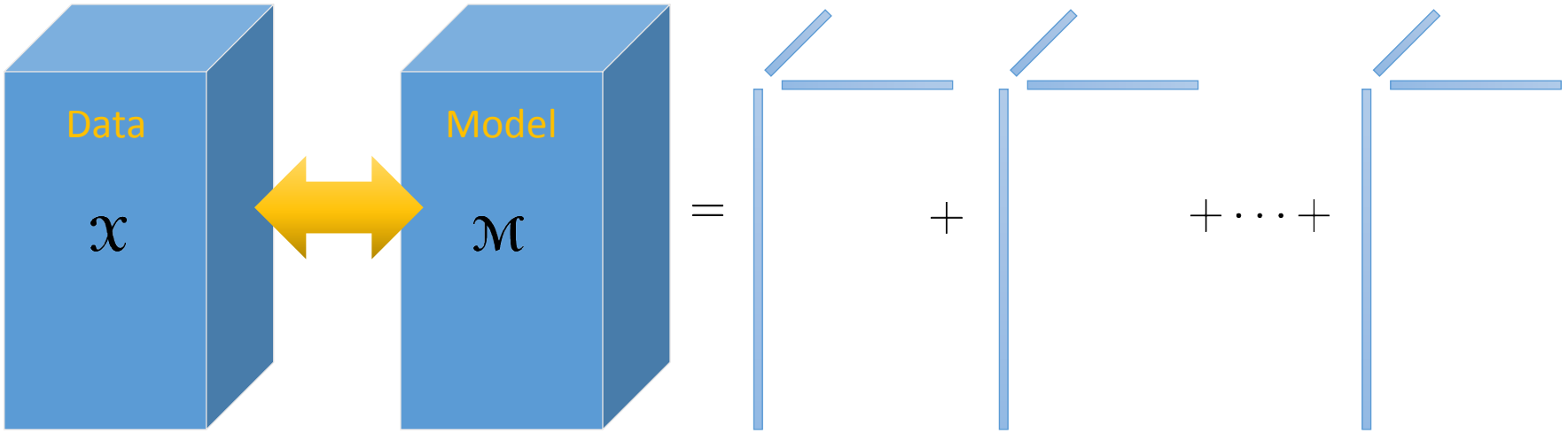
$$\mathcal{M} = [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]$$

r = # components

$$x(i_1, i_2, \ldots, i_d) \Leftrightarrow m(i_1, i_2 \ldots, i_d) = \sum_{j=1}^{r} a_1(i_1, j)\, a_2(i_2, j) \cdots a_d(i_d, j)$$

Hitchcock 1927, Harshman 1970, Carroll & Chang 1970

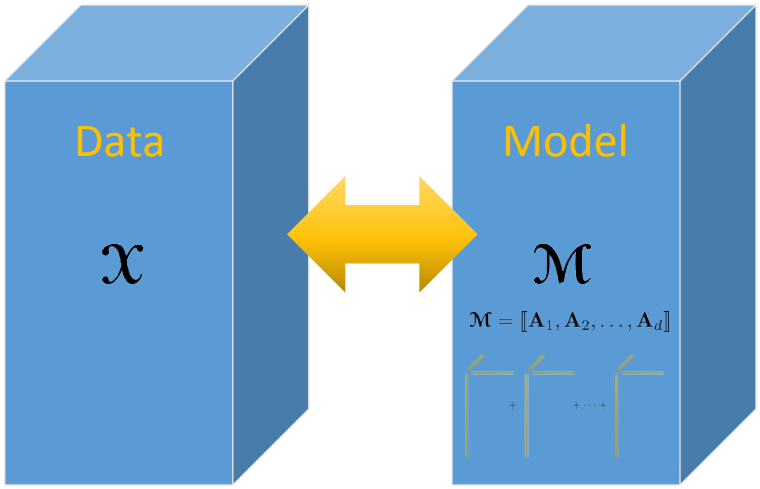
CP Tensor Decomposition



$$\mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$$

$$\underbrace{x(i_1, i_2, \ldots, i_d)}_{x_i} \Leftrightarrow \underbrace{m(i_1, i_2 \ldots, i_d)}_{m_i} = \sum_{j=1}^{r} a_1(i_1, j)\, a_2(i_2, j) \cdots a_d(i_d, j)$$

"Standard" CP



Typically: Consider data to be low-rank plus "white noise"

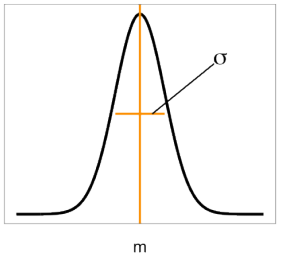
$$x_i \sim m_i + \mathcal{N}(0, \sigma)$$

Equivalently, Gaussian with mean m_i

$$x_i \sim \mathcal{N}(m_i, \sigma)$$

Gaussian Probability Density Function (PDF)

$$\frac{e^{-(x-m)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}}$$



Want to maximize likelihood of model:

$$L(\mathcal{M}) = \prod_i \frac{e^{-(x_i-m_i)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}}$$

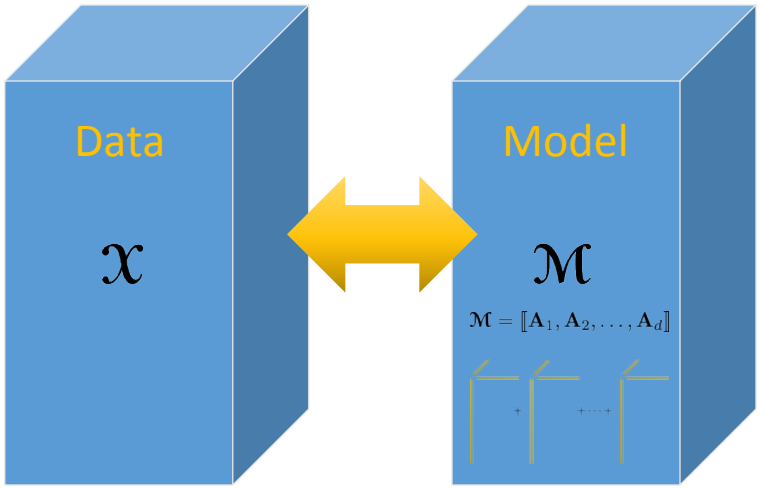
Equivalent to minimizing negative log likelihood:

$$-\log(L(\mathcal{M})) = \sum_i \frac{(x_i - m_i)^2}{2\sigma^2} + \log\sqrt{2\pi\sigma^2}$$

Results in the "standard" objective:

$$\min F(\mathcal{X}, \mathcal{M}) = \sum_i \underbrace{(x_i - m_i)^2}_{f(x_i, m_i)}$$

Generalized CP



$$\min \quad F(\mathcal{X}, \mathcal{M}) = \sum_i f(x_i, m_i)$$

$$\text{s.t.} \quad \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$$

"Standard" CP uses:

$$f(x, m) = (x - m)^2$$

Poisson CP (Chi-Kolda 2012) uses:

$$f(x, m) = m - x \log m \quad (x \in \mathbb{N}, m \geq 0)$$

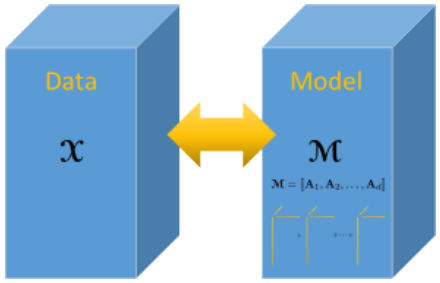
Bernoulli CP uses:

$$f(x, m) = \log(m + 1) - x \log m \quad (x \in \{0, 1\}, m \geq 0)$$

or:

$$f(x, m) = \log(1 + e^m) - xm \quad (x \in \{0, 1\})$$

Generalized CP Gradient



$$\min \quad F(\mathcal{X}, \mathcal{M}) = \sum_i f(x_i, m_i)$$

$$\text{s.t.} \quad \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$$

Define a tensor $\mathcal{G}$ such that $g(i_1, \ldots, i_d) = g_i = \dfrac{\partial f}{\partial m}(x_i, m_i)$

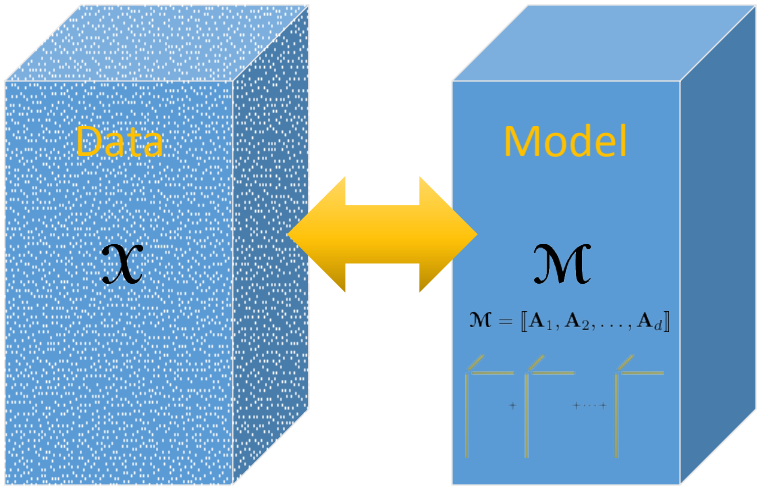
Recall $m_i = \sum_{j=1}^{r} a_1(i_1, j)\, a_2(i_2, j) \cdots a_d(i_d, j)$

Then $$\frac{\partial F}{\partial a_k(i_k, j)} = \underbrace{g(i_1, \ldots, i_d)}_{\text{No dependency on model form}} \underbrace{a_1(i_1, j) \cdots a_{k-1}(i_{k-1}, j)\, a_{k+1}(i_{k+1}, j) \cdots a_d(i_d, j)}_{\text{No dependency of function}}$$

Matricized tensor times Khatri-Rao product (MT

Matrix Version: $$\frac{\partial F}{\partial \mathbf{A}_k} = \mathbf{G}_{(k)}(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1)$$

Generalized CP Gradient with Missing Data



Ω = Set of Known Entries in Data Tensor

$$\min \quad F(\mathcal{X}, \mathcal{M}) = \sum_{i \in \Omega} f(x_i, m_i)$$

$$\text{s.t.} \quad \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$$

Define a tensor $\mathcal{G}$ such that

$$g_i = \begin{cases} \frac{\partial f}{\partial m}(x_i, m_i) & \text{if } i \in \Omega, \\ 0 & \text{if } i \notin \Omega. \end{cases}$$

Then (no change except G):

$$\frac{\partial F}{\partial \mathbf{A}_k} = \mathbf{G}_{(k)}(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1)$$

Acar, Dunlavy, Kolda, Morup 2011

Fitting Generalized CP

Given data tensor $\mathcal{X}$ and initial guess $\mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$.

1: $F_0 \leftarrow \sum_i f(x_i, m_i)$
2: **for** $\ell = 1, 2, \ldots$ **do**
3: Compute $\mathcal{G}$ based on current $\mathcal{M}$
4: **for** $k = 1, \ldots, d$ **do**
5: $\Delta_k \leftarrow \mathbf{G}_{(k)}(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1)$
6: **end for**

Compute Gradient

7: Determine step length α
8: **for** $k = 1, \ldots, d$ **do**
9: $\mathbf{A}_k \leftarrow \mathbf{A}_k - \alpha\Delta_k$
10: **end for**

Compute optimization step and take it

11: $F_\ell \leftarrow \sum_i f(x_i, m_i)$ using updated $\mathcal{M}$
12: Check convergence

Use either checks on function value or gradient

13: **end for**

Acar, Dunlavy, Kolda, J Chemometrics 2011

Observations on Generalized CP

- Can use any optimization method to solve the optimization problem

$$\min \quad F(\mathcal{X}, \mathcal{M}) = \sum_{i \in \Omega} f(x_i, m_i) \qquad \text{s.t.} \quad \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$$

- Can easily add regularization

$$\min \quad F(\mathcal{X}, \mathcal{M}) = \sum_{i \in \Omega} f(x_i, m_i) + \sum_{k} \rho_k(\mathbf{A}_k) \quad \text{s.t.} \quad \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$$

- If data tensor (X) is sparse, there is no guarantee that the gradient will be efficient
  - Standard CP is a special case that has exploitable structure
  - Scalability is potentially a major problem

- If known data is sparse (Omega), then gradient will be efficient
  - Doesn't require any sparsity in data tensor
  - Perhaps this can be exploited? Yes – stochastic algorithms.

"Stochastic" Generalized CP

Given data tensor $\mathcal{X}$ and initial guess $\mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$.

1: $F_0 \leftarrow \sum_{i \in \hat{\Omega}} f(x_i, m_i)$ ← Same sample every time in line 1 & 13
2: **for** $\ell = 1, 2, \ldots$ **do**
3: **for** $t = 1, \ldots, T$ **do**
4: $\Omega \leftarrow N$ random entries in the data tensor ← Different samples every time
5: Compute $\mathcal{G}$ *only at entries in* Ω (everywhere else is zero)
6: **for** $k = 1, \ldots, d$ **do**
7: $\Delta_k \leftarrow \mathbf{G}_{(k)}(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1)$ ← Sparse MTTKRP, super cheap!
8: **end for** ← Compute Stochastic Gradient
9: **for** $k = 1, \ldots, d$ **do**
10: $\mathbf{A}_k \leftarrow \mathbf{A}_k - \alpha \Delta_k$ ← Take tiny step (alpha is small)
11: **end for**
12: **end for** ← Epoch
13: $F_\ell \leftarrow \sum_{i \in \hat{\Omega}} f(x_i, m_i)$ using updated $\mathcal{M}$
14: Check convergence ← Using *approximate* function values
15: **end for**

ADAM SGD for CP

Given data tensor $\mathcal{X}$ and initial guess $\mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$.

for $k = 1, \ldots, d$ **do**
  $\Phi_k \leftarrow 0$, $\Psi_k \leftarrow 0$
end for
$F_0 \leftarrow \sum_{i \in \hat{\Omega}} f(x_i, m_i)$
for $\ell = 1, 2, \ldots$ **do**
  for $t = 1, \ldots, T$ **do**
    $\Omega \leftarrow N$ random entries in the data tensor
    Compute $\mathcal{G}$ *only at entries in* Ω (everywhere else is zero)
    for $k = 1, \ldots, d$ **do**
      $\Delta_k \leftarrow \mathbf{G}_{(k)}(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1)$
    end for
    for $k = 1, \ldots, d$ **do**
      $\Phi_k \leftarrow \beta_1 \Phi_k + (1 - \beta_1)\Delta_k$, $\hat{\Phi}_k \leftarrow \Phi_k / (1 - \beta_1^{(\ell-1)T+t})$
      $\Psi_k \leftarrow \beta_2 \Psi_k + (1 - \beta_2)\Delta_k^2$, $\hat{\Psi}_k \leftarrow \Psi_k / (1 - \beta_2^{(\ell-1)T+t})$
      $\mathbf{A}_k \leftarrow \mathbf{A}_k - \alpha \hat{\Phi}_k / (\sqrt{\hat{\Psi}_k} + \varepsilon)$
    end for
  end for
  $F_\ell \leftarrow \sum_{i \in \hat{\Omega}} f(x_i, m_i)$ using updated $\mathcal{M}$
  Check convergence
end for

Moment Correction Terms

Kingma and Ba, 2015

Example Results: Senate Voting Data

- Data on U.S. Senate: 1989-2016
- 271 Senators
  - Two senators per state (100 total)
  - Six-year term, can serve multiple terms
  - Two main parties: Republicans (red) & Democrats (blue)
- 9044 Items
  - Roll calls from 1989 to 2016
- 3 possible Votes
  - Yea, Nay, No Vote (NA)
- Tensor is 3-way & binary
  - *63% missing*
  - *Stochastic method only samples from known entries in data tensor*

Algorithm Details

Given data tensor $\mathcal{X}$ and initial guess $\mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$.

for $k = 1, \ldots, d$ **do**
  $\Phi_k \leftarrow 0$, $\Psi_k \leftarrow 0$
end for
$F_0 \leftarrow \sum_{i \in \hat{\Omega}} f(x_i, m_i)$
for $\ell = 1, 2, \ldots$ **do**
  for $t = 1, \ldots, T$ **do**
    $\Omega \leftarrow N$ random entries in the data tensor
    Compute $\mathcal{G}$ *only at entries in* Ω (everywhere else is zero)
    for $k = 1, \ldots, d$ **do**
      $\Delta_k \leftarrow \mathbf{G}_{(k)}(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1)$
    end for
    for $k = 1, \ldots, d$ **do**
      $\Phi_k \leftarrow \beta_1 \Phi_k + (1 - \beta_1)\Delta_k$, $\hat{\Phi}_k \leftarrow \Phi_k / (1 - \beta_1^{(\ell-1)T+t})$
      $\Psi_k \leftarrow \beta_2 \Psi_k + (1 - \beta_2)\Delta_k^2$, $\hat{\Psi}_k \leftarrow \Psi_k / (1 - \beta_2^{(\ell-1)T+t})$
      $\mathbf{A}_k \leftarrow \mathbf{A}_k - \alpha \hat{\Phi}_k / (\sqrt{\hat{\Psi}_k} + \varepsilon)$
    end for
  end for
  $F_\ell \leftarrow \sum_{i \in \hat{\Omega}} f(x_i, m_i)$ using updated $\mathcal{M}$
  Check convergence
end for

$$|\mathcal{X}| \sim 10^7$$

$$|\mathrm{nnz}(\mathcal{X})| \sim 10^7$$

$$|\hat{\Omega}| = 200,000$$

$$|\Omega| = 10,000$$

$$T = 2000$$

$$\alpha = .001$$

$$\beta_1 = 0.9$$

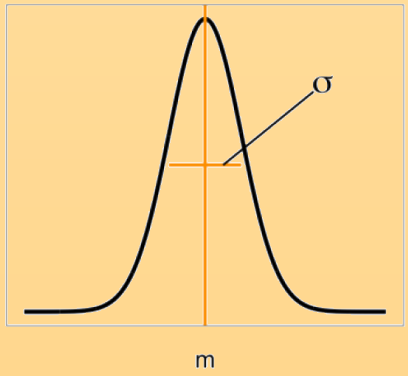
$$\beta_2 = 0.999$$

$$\epsilon = 10^{-8}$$

Different Statistical Assumptions and Interpretation

Sandia National Laboratories

Normally-Distributed

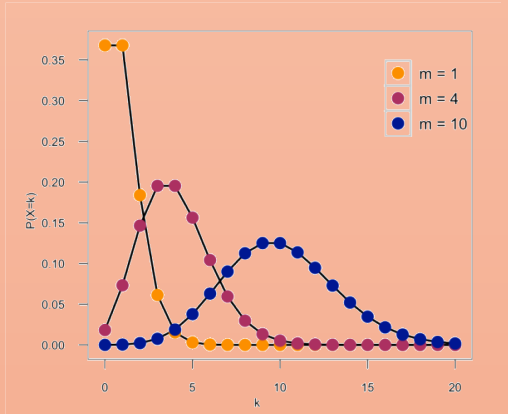


$$X_i \sim \mathcal{N}(m_i, \sigma)$$

$$m_i = \mathbb{E}(X_i)$$

$$f(x, m) = (x - m)^2$$

Poisson Distributed



$$X_i \sim \text{Poisson}(m_i)$$

$$m_i = \mathbb{E}(X_i)$$

$$f(x, m) = m - x \log m$$

Bernoulli Distributed

$$X_i \sim \mathcal{B}(1, {}^{m_i}/_{1 + m_i})$$

$$m_i = ({}^{\mathbb{E}(X_i)}/_{1 - \mathbb{E}(X_i)})$$

$$f(x, m) =$$

$$\log(m + 1) - x \log m$$

$$X_i \sim \mathcal{B}(1, e^{{}^{m_i}/_{1 + m_i}})$$

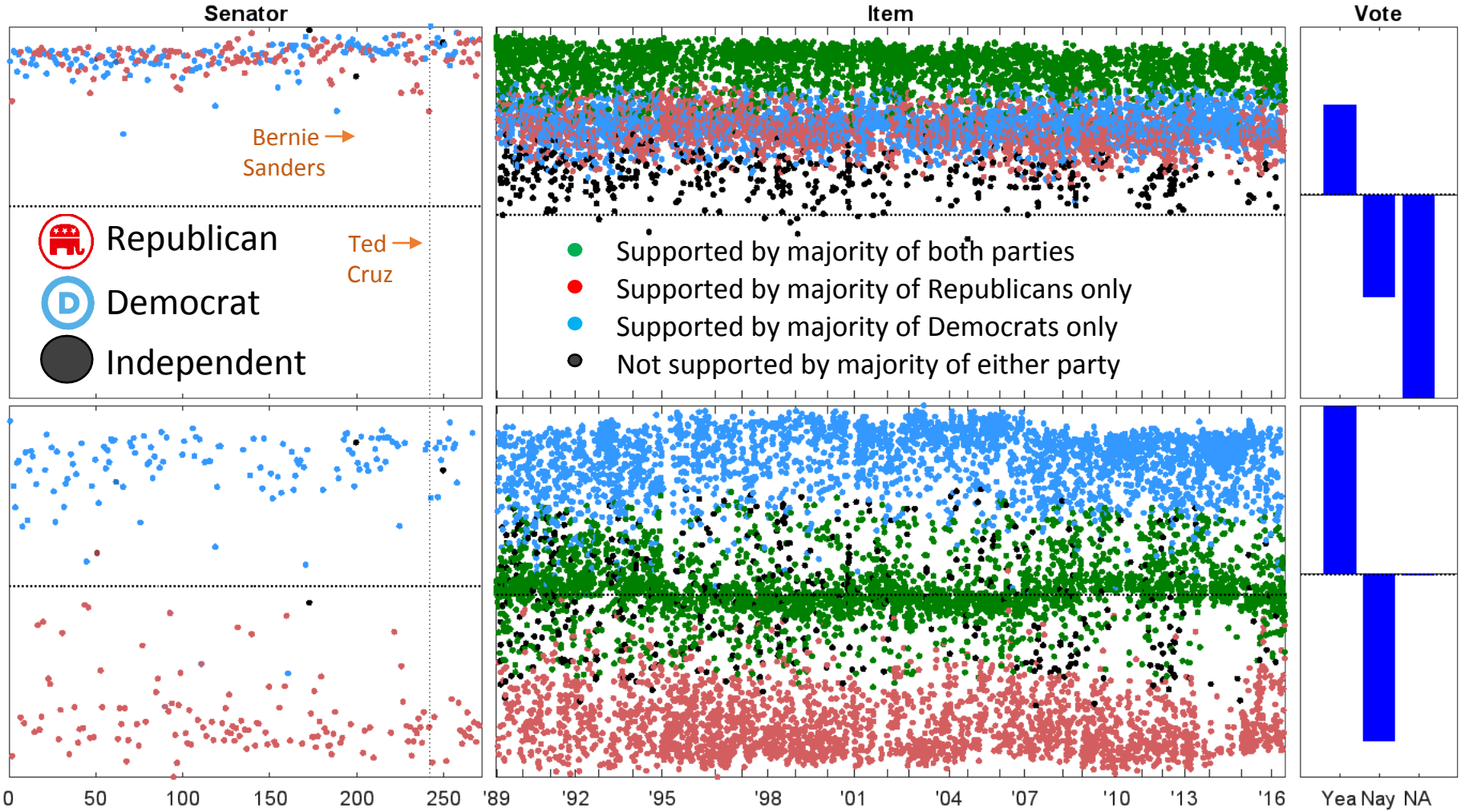
$$m_i = \log({}^{\mathbb{E}(X_i)}/_{1 - \mathbb{E}(X_i)})$$

$$f(x, m) =$$

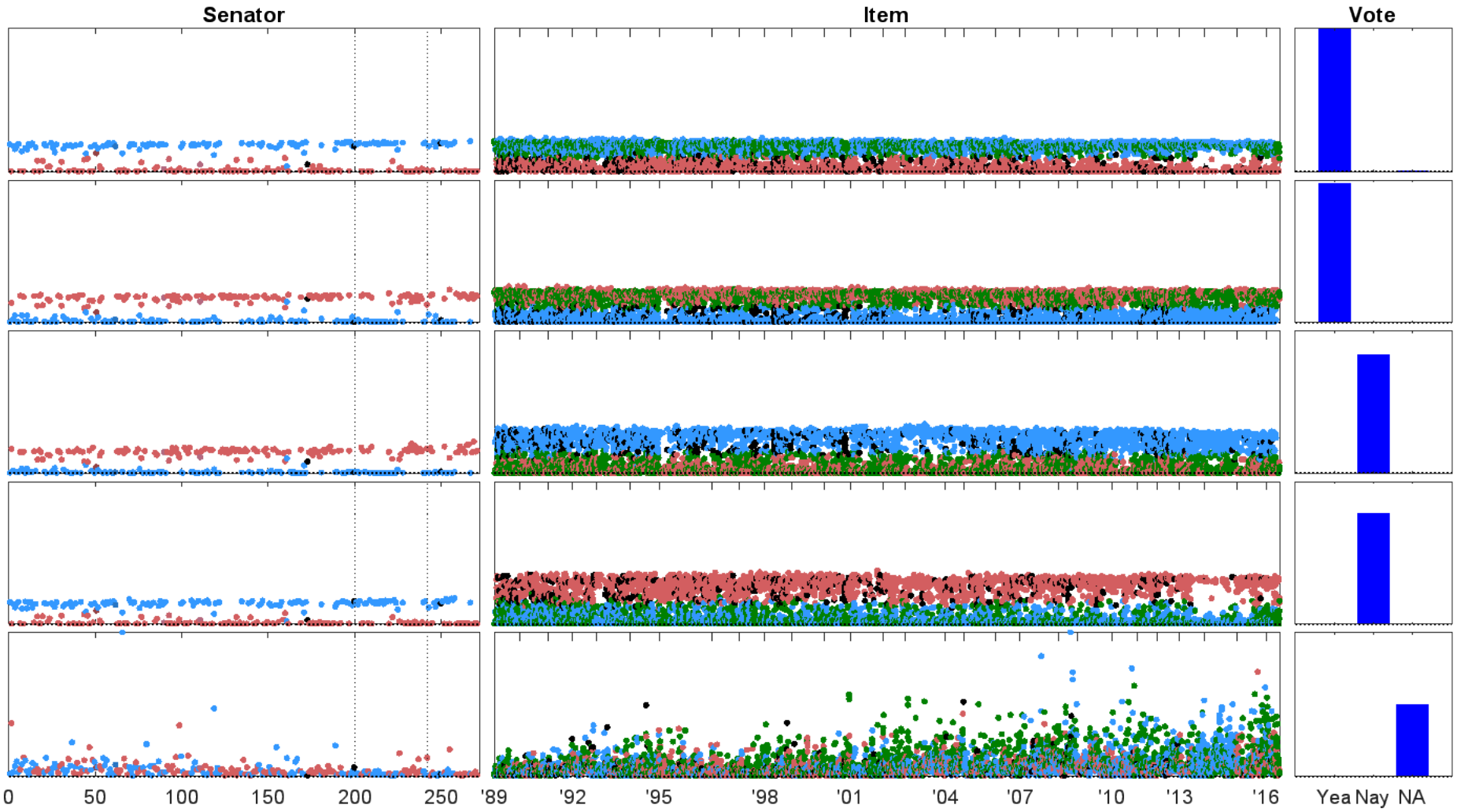
$$\log(e^m + 1) - xm$$

All models are wrong. Some models are useful. – Box

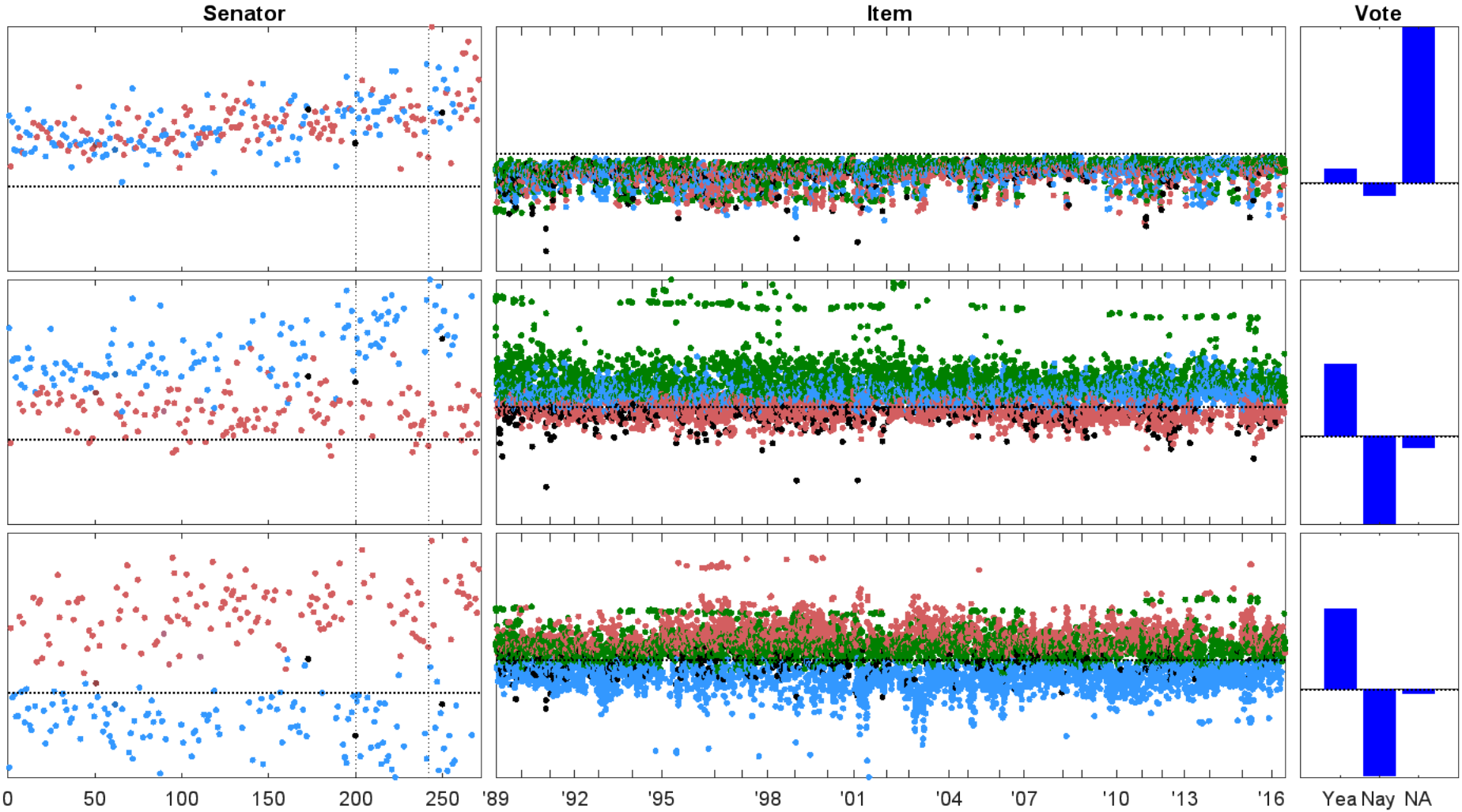
Gaussian (+/- 1 Data)



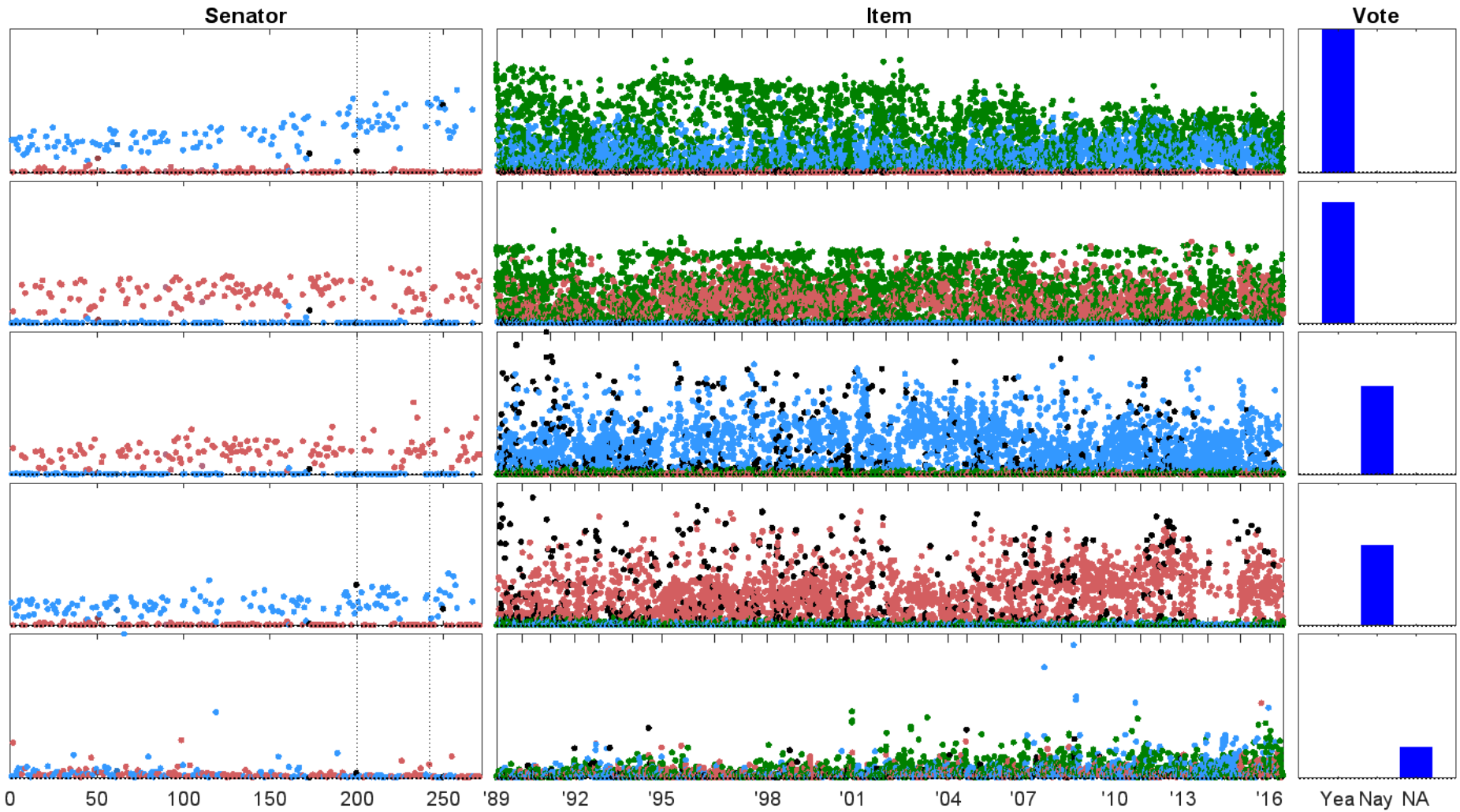
Poisson


Senator
Item
Vote
0
50
100
150
200
250
'89
'92
'95
'98
'01
'04
'07
'10
'13
'16
Yea
Nay
NA

Bernoulli – Log Odds

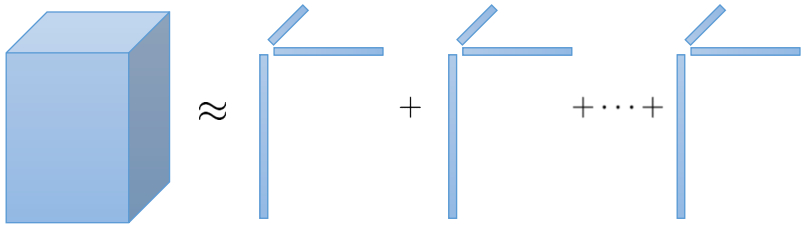


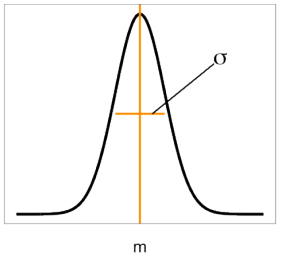
Bernoulli

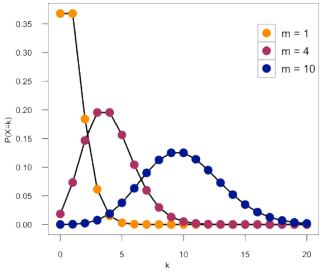

Senator
Item
Vote
0
50
100
150
200
250
'89
'92
'95
'98
'01
'04
'07
'10
'13
'16
Yea
Nay
NA

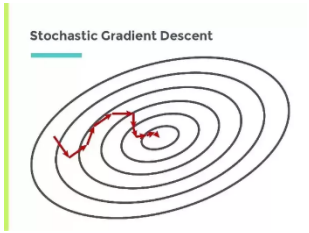
Summary: CP with Data Flexibility

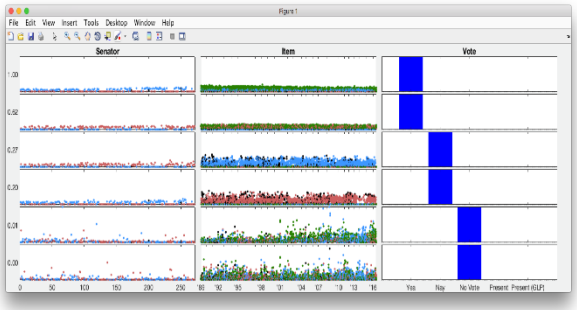
- Standard CP makes *implicit* statistical assumption
  - Normal-distributed
- New generalized CP allows other statistical assumptions
  - Bernoulli (0/1)
  - Poisson (counts)
- Elegant formula for gradients
  - Amenable to stochastic optimization methods
  - Takes advantage of existing kernels for sparse tensors
- Experimental results show advantages in interpretation
- What's still missing
  - Performance of ADAM compared to CP-ALS, CPRAND, CP-APR (Poisson)
  - Understanding effect of # iterations per epoch, batch size per iteration
  - Theoretical analysis











For more info: Tammy Kolda, tgkolda@sandia.gov

Illustration by Chris Brigman

Backup Slides

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Generalized CP Gradient in the Standard Case

$$\begin{aligned} \min \quad & F(\mathcal{X}, \mathcal{M}) = \sum_i f(x_i, m_i) = {}^1\!/_2 \sum_i (x_i - m_i)^2 \\ \text{s.t.} \quad & \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!] \end{aligned}$$

$$g_i = \frac{\partial f}{\partial m}(x_i, m_i) = m_i - x_i \quad \Leftrightarrow \quad \mathcal{G} = \mathcal{M} - \mathcal{X}$$

$$\frac{\partial F}{\partial \mathbf{A}_k} = \mathbf{G}_{(k)}(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1) = 0$$

$$\mathcal{X}(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1) = \mathbf{A}_k(\mathbf{A}_d'\mathbf{A}_d * \cdots * \mathbf{A}_{k+1}'\mathbf{A}_{k+1} * \mathbf{A}_{k-1}'\mathbf{A}_{k-1} * \cdots * \mathbf{A}_1'\mathbf{A}_1)$$

$$\mathbf{A}_k = \mathcal{X}(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1)(\mathbf{A}_d'\mathbf{A}_d * \cdots * \mathbf{A}_{k+1}'\mathbf{A}_{k+1} * \mathbf{A}_{k-1}'\mathbf{A}_{k-1} * \cdots * \mathbf{A}_1'\mathbf{A}_1)^{-1}$$