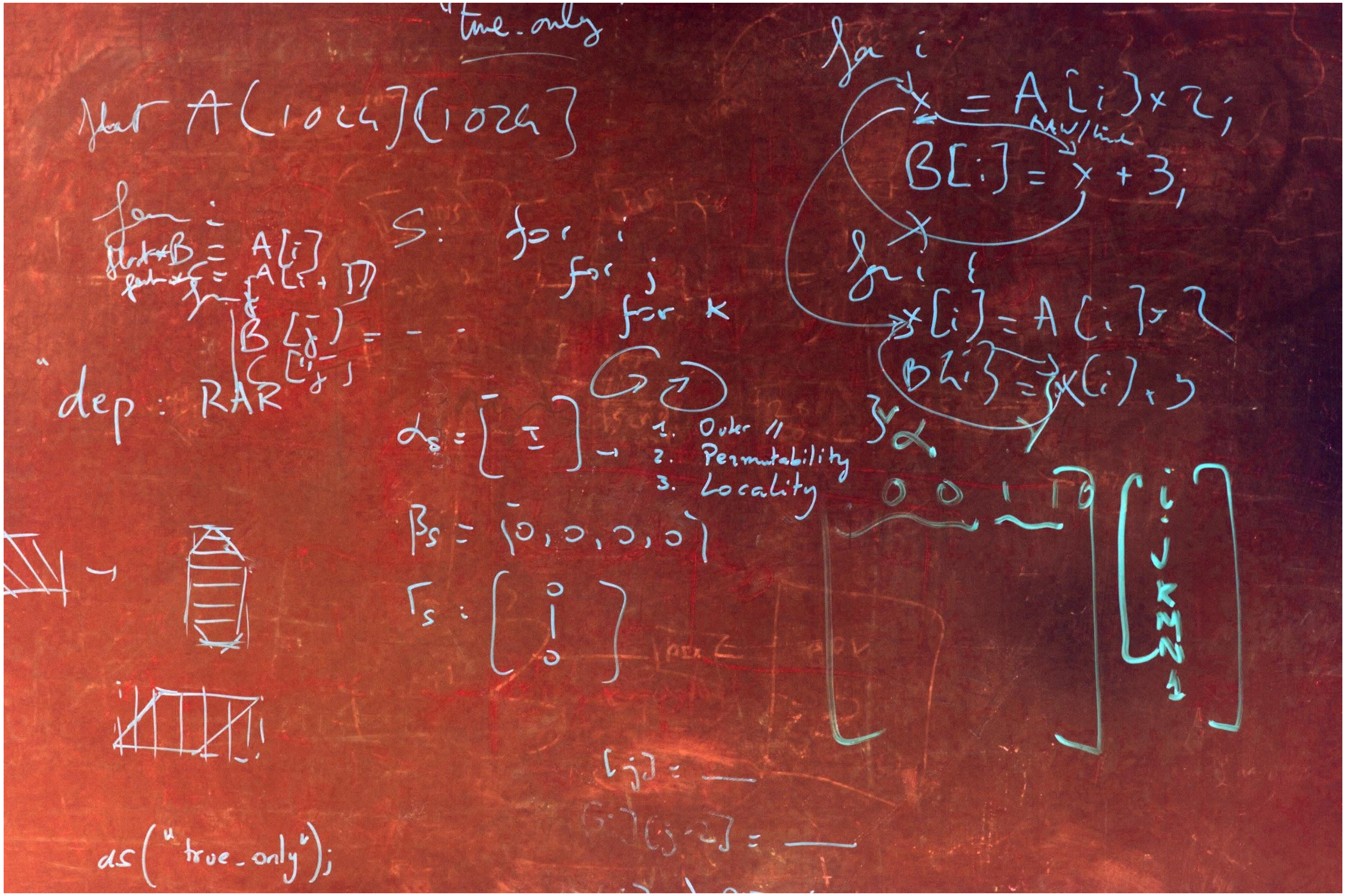
# Explanation and Tensor Decomposition

## Reservoir Labs



## Credits to Reservoir Team

Aditya Gudibanda ***
Muthu Baskaran
Jim Ezick
Tom Henretty
Harper Langston
(Richard Lethin)

## Outline

- Context and objectives
- Tensor decompositions as explanations
- A discussion on sparsity
    - Why
    - Methods
        - L2 norm regularization
        - Second-order optimization
        - Loss function
        - Orthogonality
- Conclusion

## Outline

- Context and objectives
- Tensor decompositions as explanations
- A discussion on sparsity
  - Why
  - Methods
    - L2 norm regularization
    - Second-order optimization
    - Loss function
    - Orthogonality
- Conclusion

## CLSAC 2018 Theme

## High Impact Large-Scale Analytics

“Consequence analytics” “Taking Action” “Credible, trustworthy”

Implies we would benefit from **EXPLANATIONS** for the results of our analytics.
Or that our analytics be more **EXPLAINABLE.**

Philosophy of Explanations can be as deep and long as you have time

- Reduction and emergence: “why does water boil at 212℉?”
- Ref beliefs, epistemology: “Why is the atom of copper at the tip of the nose of the statue of Churchill in London’s Parliament square, there?”
- Intent: “Explain why you think that packet ping is malicious.”

See David Deutsch, The Fabric of Reality 1997

**David Hume** (1711-1776) - Scottish Empiricist

- Our ideas are based on our impressions, which come from our senses.

**Immanuel Kant** (1724-1804) - Prussian

- Our knowledge is formed taking experience in reference to “a-priori.”

**Judea Pearl** (1936-Present) - American

- “No machine can derive explanations from raw data. It needs a push.”

**William of Occam** (1285 - 1347) - English Franciscan

- “When you have two competing theories that make exactly the same predictions, the simpler one is better.”

This drives us to specify **MODELS** that are compact and have **SPARSITY**

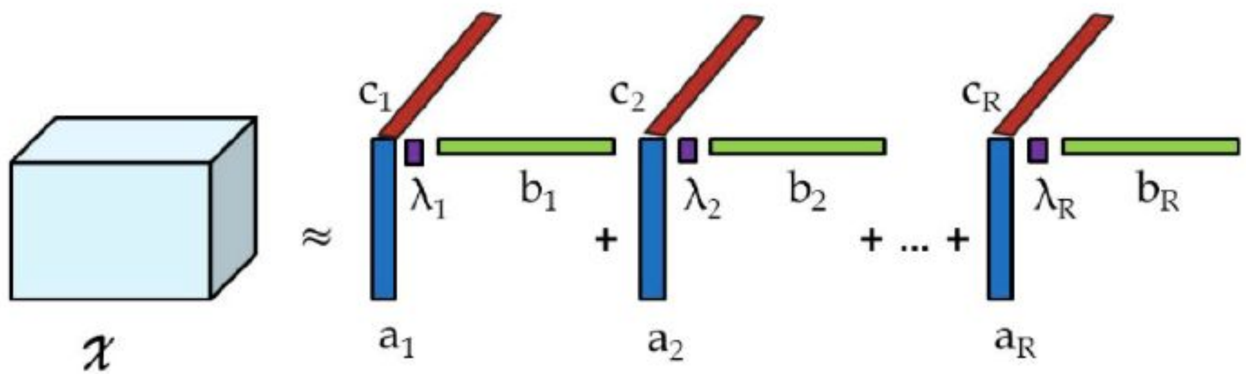
## Outline

- Context and objectives
- Tensor decompositions as explanations
- A discussion on sparsity
  - Why
  - Methods
    - L2 norm regularization
    - Second-order optimization
    - Loss function
    - Orthogonality
- Conclusion

# Tensor Decompositions as Explanations

## Introduction to Tensor Decompositions

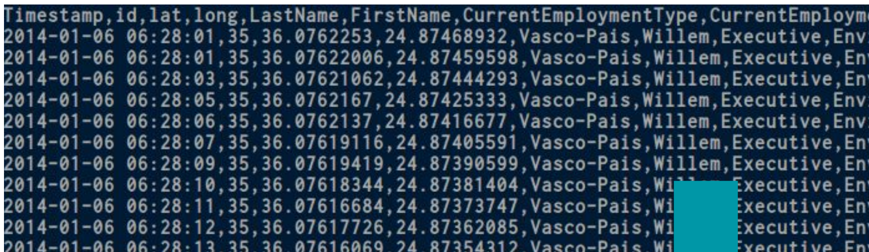
- Tensor analog of low-rank matrix factorizations
- Unsupervised learning
- Tensor is decomposed into a non-unique weighted sum of a predefined number R of **LOW RANK** components
  - Component = "**EXPLANATORY FACTOR**" [Hong, Kolda, Deursch 2018]
  - Tensor decomposition compresses = shorter -> **better**



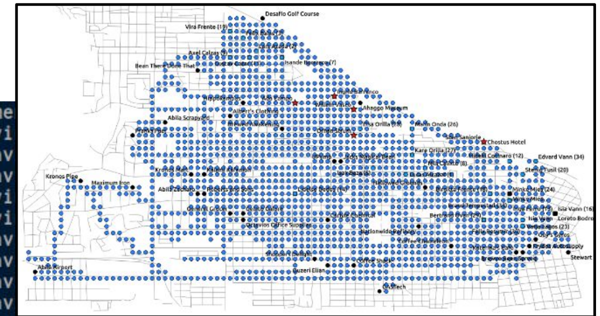

$O\left(\prod_{i=1}^{N} m_i\right)$ $O\left(R(\sum_{i=1}^{N} m_i)\right)$

# Introduction to Tensor Decompositions

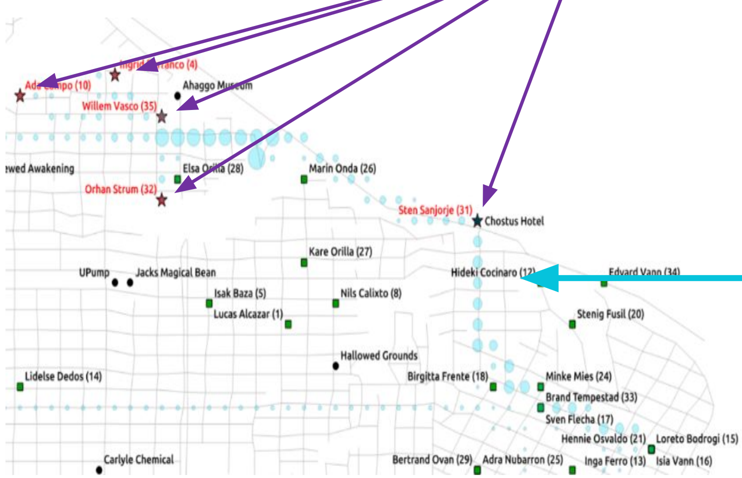
## Example: Making Sense of Geospatial Data

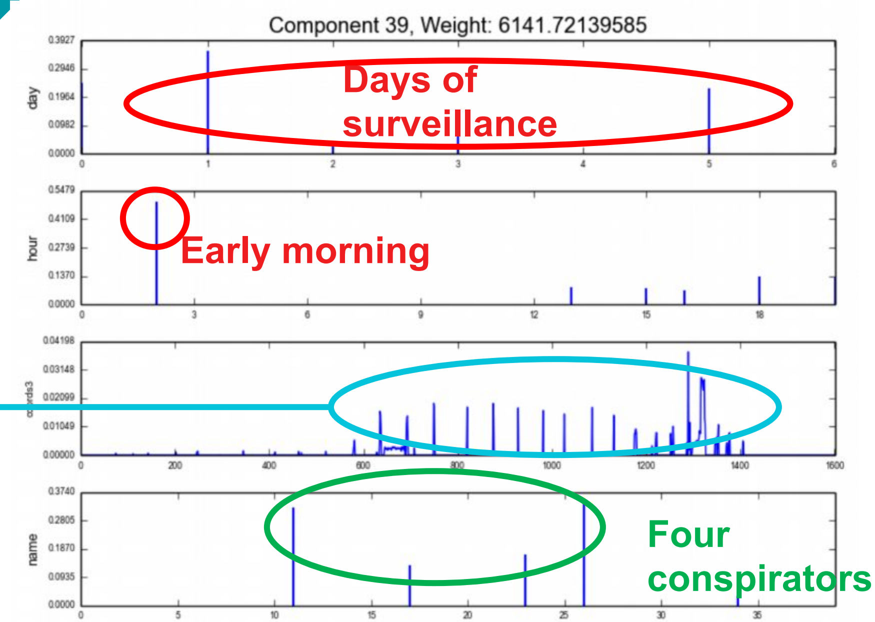


```
Timestamp,id,lat,long,LastName,FirstName,CurrentEmploymentType,CurrentEmployme
2014-01-06 06:28:01,35,36.0762253,24.87468932,Vasco-Pais,Willem,Executive,Envi
2014-01-06 06:28:01,35,36.07622006,24.87459598,Vasco-Pais,Willem,Executive,Env
2014-01-06 06:28:03,35,36.07621062,24.87444293,Vasco-Pais,Willem,Executive,Env
2014-01-06 06:28:05,35,36.0762167,24.87425333,Vasco-Pais,Willem,Executive,Envi
2014-01-06 06:28:06,35,36.0762137,24.87416677,Vasco-Pais,Willem,Executive,Envi
2014-01-06 06:28:07,35,36.07619116,24.87405591,Vasco-Pais,Willem,Executive,Env
2014-01-06 06:28:09,35,36.07619419,24.87390599,Vasco-Pais,Willem,Executive,Env
2014-01-06 06:28:10,35,36.07618344,24.87381404,Vasco-Pais,Wi    Executive,Env
2014-01-06 06:28:11,35,36.07616684,24.87373747,Vasco-Pais,Wi    Executive,Env
2014-01-06 06:28:12,35,36.07617726,24.87362085,Vasco-Pais,Wi    Executive,Environmental Safety Advisor,36.076,24.874,36.08,
2014-01-06 06:28:13,35,36.07616069,24.87354312,Vasco-Pais,Wi    Executive,Environmental Safety Advisor,36.076,24.874,36.08,
```




**Targets**



# Application Areas for Tensor Decomposition


# Exascale NonStationary Graph Notation (ENSIGN)

## Unique Insight from Large-Scale Multi-Attribute Data

| Class | Differentiating Specifics | Benefit to Analyst |
|---|---|---|
| **Modeling** | First Order Decompositions<br>Second Order Decompositions<br>Joint Decompositions<br>Multiple Probability Models<br>Customer Special Decompositions<br>… more coming | Breadth of models enabled<br>Framework for graph fusion<br>Platform for anomaly detection<br>Sparsity-maximizing approaches |
| **Performance** | Optimized Sparse Tensor Data Structures<br>Mixed static/dynamic optimization<br>Memory-efficiency optimizations<br>Algorithmic improvements<br>Shared memory parallelism<br>Distributed memory parallelism | Extend the range, scale, and scope of analysis<br>Analyze tensors with $10^9$ non-zeroes and beyond<br>Enable large rank R decompositions<br>Enable large number of mode decompositions<br>Leverage High Performance Computing (HPC) Systems |
| **Streaming** | Streaming CP, Tucker | Efficient update with arrival of new data<br>Discovery of new behaviors through new component formation |
| **Usability** | GUI & CLI<br>Python Bindings<br>C Bindings<br>QGIS Support<br>Virtual Machine Distributions<br>Documented, Tested, Supported | Interactive large scale exploration<br>In standard environments (e.g., Jupyter notebooks)<br>Integration with existing corporate data lakes/pipelines<br>Visualization<br>Reliable install and operation<br>Training, Someone to Call |

See https://www.reservoir.com/product/ensign-cyber/ and https://www.reservoir.com/research/tech/tensor-analysis/ for product info.

Pat. US 9,471,377, Pats. Pending

# Outline

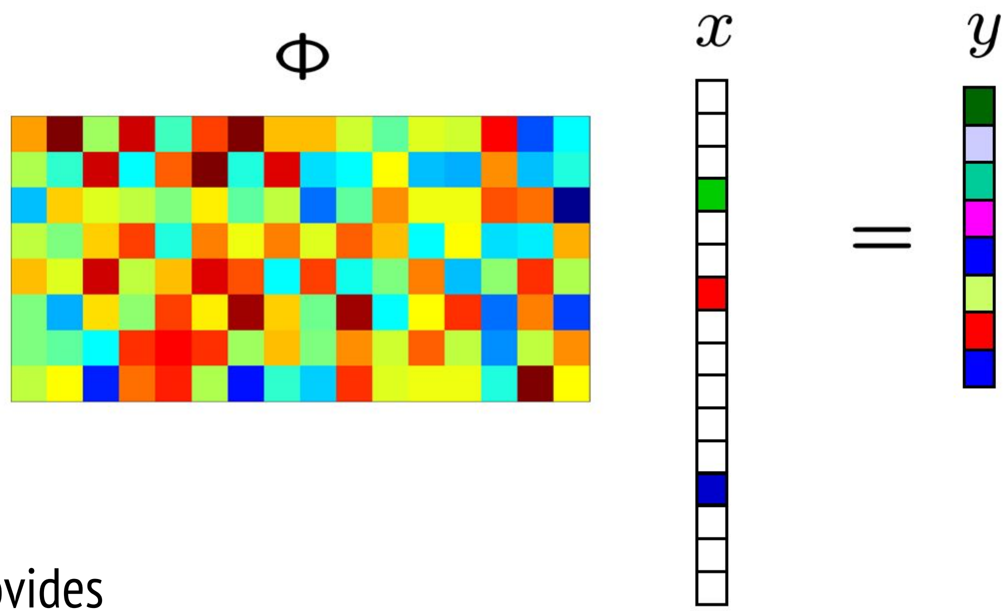
- Context and objectives
- Tensor decompositions as explanations
- A discussion on sparsity
  - Why
  - Methods
    - L2 norm regularization
    - Second-order optimization
    - Loss function
    - Orthogonality
- Conclusion

# Compression and Sparsity

## Compressive Sensing

David Donoho 2004, Terry Tao, Emmanuel Candes 2006

- Observe $y$, choose $x$ that maximizes **SPARSITY**
- Intractable, but, L1 norm minimization is equivalent




Sparsity provides

- Performance benefit - need fewer measurements
- Interpretation/explanatory benefit

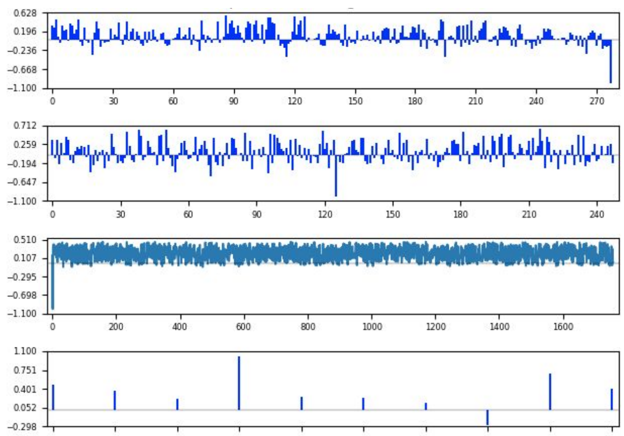
(Image from http://informationtransfereconomics.blogspot.com/2017/10/compressed-sensing-and-information.html)

## Outline

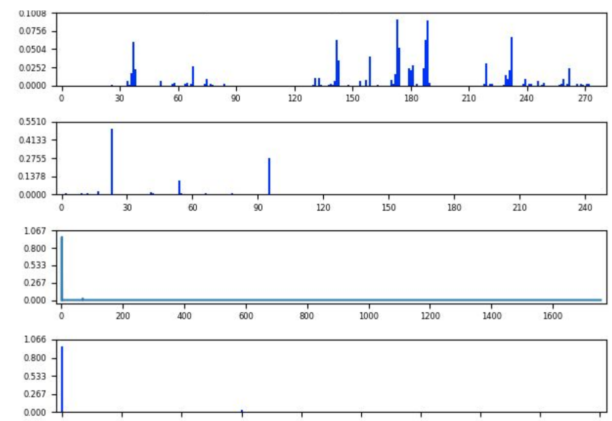
- Context and objectives
- Tensor decompositions as explanations
- A discussion on sparsity
  - Why
  - Methods
    - L2 norm regularization
    - Second-order optimization
    - Loss function
    - Orthogonality
- Conclusion

# Sparsity

- Sparsity is a form of *model selection*
- Enforcing sparsity forces the model to throw away less important information
- The information the model keeps is forced to capture more useful patterns or features present in the data
- Leads to crisper, more interpretable results




Less sparse, more difficult to interpret




More sparse, more easy to interpret

# Not all Sparse Solutions are Created Equal

- There is a tradeoff between sparsity/interpretability and model accuracy
  - True model may be above human understanding
  - Should be careful not to sacrifice too much accuracy at the cost of interpretability
- Therefore, the method we use to induce sparsity is important
  - Changes what sacrifices (if any) are made for the sake of sparsity
  - Different methods strike different balances

# Outline

- Context and objectives
- Tensor decompositions as explanations
- A discussion on sparsity
  - Why
  - Methods
    - L2 norm regularization
    - Second-order optimization
    - Loss function
    - Orthogonality
- Conclusion

## Outline

- Context and objectives
- Tensor decompositions as explanations
- A discussion on sparsity
  - Why
  - Methods
    - L2 norm regularization
    - Second-order optimization
    - Loss function
    - Orthogonality
- Conclusion

## L2 Regularization

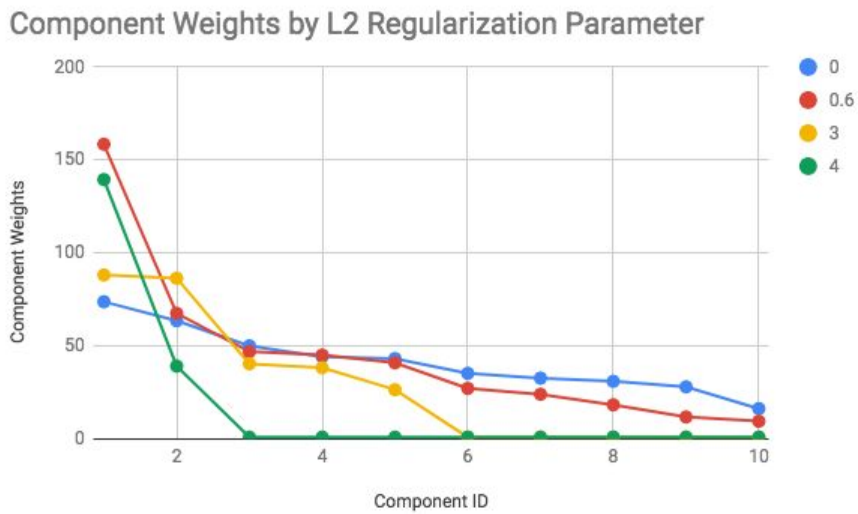
- L0 norm directly counts the number of nonzeros, but is infeasible to optimize
- L2 norm is a relaxation of the L0 norm which is easier to optimize
- Corresponds to a Gaussian prior
- Closed form analytic updates available [Royer, Comon, Thirion-Moreau 2011]

$$\mathcal{G}(\mathbf{A},\mathbf{B},\mathbf{C}) = \mathcal{F}(\mathbf{A},\mathbf{B},\mathbf{C}) + \underbrace{\alpha\|\mathbf{A}\|_F^2 + \beta\|\mathbf{B}\|_F^2 + \gamma\|\mathbf{C}\|_F^2}_{\text{L2 regularization term}}.$$

$$\left.\begin{aligned}
\widehat{\mathbf{A}} &= \mathbf{T}_{(1)}^{I,JK}(\mathbf{C}\odot\mathbf{B})\left[(\mathbf{C}\odot\mathbf{B})^T)(\mathbf{C}\odot\mathbf{B}) + \alpha\mathbf{I}_F\right]^{\dagger},\\
\widehat{\mathbf{B}} &= \mathbf{T}_{(2)}^{J,KI}(\mathbf{C}\odot\mathbf{A})\left[(\mathbf{C}\odot\mathbf{A})^T)(\mathbf{C}\odot\mathbf{A}) + \beta\mathbf{I}_F\right]^{\dagger},\\
\widehat{\mathbf{C}} &= \mathbf{T}_{(3)}^{K,JI}(\mathbf{B}\odot\mathbf{A})\left[(\mathbf{B}\odot\mathbf{A})^T)(\mathbf{B}\odot\mathbf{A}) + \gamma\mathbf{I}_F\right]^{\dagger}.
\end{aligned}\right\}\text{Closed form analytic updates}$$

# L2 Regularization Results

- Performed rank 10 decompositions of Cyber data from SCinet with different L2 hyperparameters
- Measured how L2 regularization affected component weights
- Shows how the decomposition compresses the data into lower rank
- Increasing L2 regularization parameter led to more compression
- Less components to analyze, more interpretable



## Outline

- Context and objectives
- Tensor decompositions as explanations
- A discussion on sparsity
  - Why
  - Methods
    - L2 norm regularization
    - Second-order optimization
    - Loss function
    - Orthogonality
- Conclusion

## CP-APR

- Loss function is Kullback-Leibler divergence from Poisson likelihood function [Chi and Kolda 2012]
    - Specialized to count data
- Workhorse algorithm for cyber and geospatial datasets
- Includes nonnegativity constraints
- Alternatively updates one factor matrix at a time
    - Performs first-order gradient descent

$$f(\mathcal{M}) = \sum_{\mathbf{i}} m_{\mathbf{i}} - x_{\mathbf{i}} \log m_{\mathbf{i}},$$

CP-APR Loss Function

**Algorithm 2** CP-APR Algorithm

initialize $\mathbf{A}^{(1)} \ldots \mathbf{A}^{(N)}$
**repeat**
  **for** $n = 1 \ldots N$ **do**
    $\mathbf{\Pi} = (\odot_{m \neq n} \mathbf{A}^{(m)})^T$
    **repeat**
      $\mathbf{\Phi} = (\mathbf{X}_{(n)} \oslash (\mathbf{A}^{(n)} \mathbf{\Pi})) \mathbf{\Pi}^T$
      $\mathbf{A}^{(n)} = \mathbf{A}^{(n)} * \mathbf{\Phi}$
    **until** convergence
  **end for**
**until** convergence

[Chi and Kolda 2012]

# CP-APR-PQNR

- Uses same loss function as CP-APR
- Optimizes each row of a factor matrix one at a time
- Uses L-BFGS to optimize each row (second-order optimization method)

**Algorithm 1** Alternating Block Framework

Given data tensor $\mathfrak{X}$ of size $I_1 \times I_2 \times \cdots \times I_N$, and the number of components $R$

Return a model $\mathcal{M} = [\boldsymbol{\lambda}; \mathbf{A}^{(1)} \ldots \mathbf{A}^{(N)}]$

Initialize $\mathbf{A}^{(n)} \in \mathbf{R}^{I_n \times R}$ for $n = 1, \ldots, N$
**repeat**
  **for** $n = 1, \ldots, N$ **do**
    Let $\mathbf{\Pi}^{(n)} = (\mathbf{A}^{(N)} \odot \ldots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \ldots \odot \mathbf{A}^{(1)})^T$
    Use Algorithm 2 to compute $\mathbf{B}^*$ that minimizes $f(\mathbf{B}^{(n)})$ s.t. $\mathbf{B}^{(n)} \geq 0$
    $\boldsymbol{\lambda} \leftarrow \mathbf{e}^T \mathbf{B}^*$
    $\mathbf{A}^{(n)} \leftarrow \mathbf{B}^* \mathbf{\Lambda}^{-1}$, where $\mathbf{\Lambda} = \text{diag}(\boldsymbol{\lambda})$
  **end for**
**until** all mode subproblems have converged

**Algorithm 2** Row Subproblem Framework for Solving (2.6)

Given $\mathbf{X}_{(n)}$ of size $I_n \times J_n$, and $\mathbf{\Pi}^{(n)}$ of size $R \times J_n$

Return a solution $\mathbf{B}^*$ consisting of row vectors $\hat{\mathbf{b}}_1^*, \ldots, \hat{\mathbf{b}}_{I_N}^*$

**for** $i = 1, \ldots, I_n$ **do**
  Select row $\hat{\mathbf{x}}_i$ of $\mathbf{X}_{(n)}$
  Generate one column of $\mathbf{\Pi}^{(n)}$ for each nonzero in $\hat{\mathbf{x}}_i$
  Use Algorithm 3 or 4 to compute $\hat{\mathbf{b}}_i^*$ that solves

$$\min f_{\text{row}}(\hat{\mathbf{b}}_i^{(n)}, \hat{\mathbf{x}}_i^{(n)}, \mathbf{\Pi}^{(n)}) \text{ subject to } \hat{\mathbf{b}}_i \geq 0$$

**end for**

[Hansen, Plantenga, and Kolda 2015]

## APR vs PQNR Results

- APR and PQNR both have same number of max iterations, so must compare for a given number of iterations or runtime
- PQNR results are significantly more sparse than APR
- PQNR iterates become sparser faster during the optimization process
- PQNR is second-order method, can escape saddle points more easily

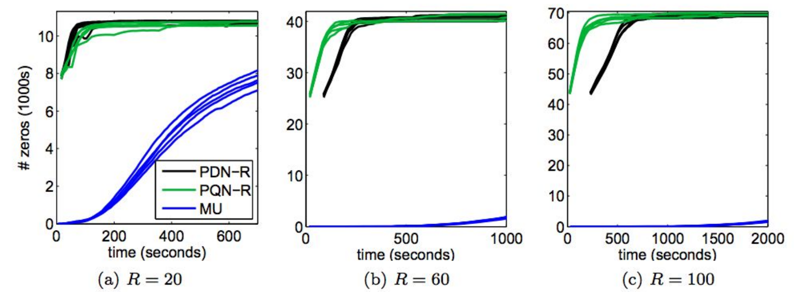
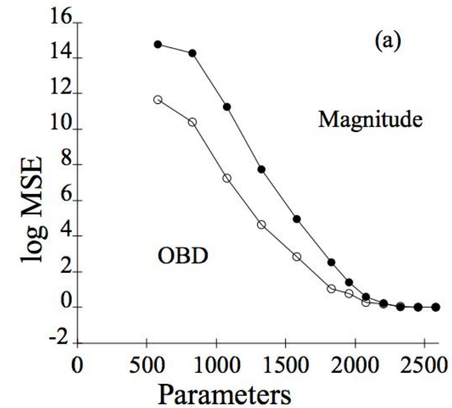



Fig. 4.8: Effectiveness of the algorithms in finding a sparse solution for a full three-way solution. In each case the total number of elements in $\mathbf{A}^{(1)}$, $\mathbf{A}^{(2)}$, and $\mathbf{A}^{(3)}$ equal to zero is plotted against execution time. The PDN-R (black lines) and PQN-R (green) algorithms are much faster than MU (blue). Each algorithm was run on ten different tensors, so the final number of zero elements has ten different values.

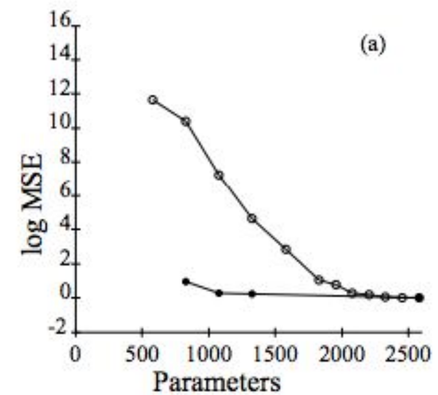
[Hansen, Plantenga, Kolda 2015]

## Advantages of Second-order Methods

- "Optimal Brain Damage" [Le Cun, Denker, Solla 1990]
  - Second derivative yields "**saliency**" or "importance" of a weight to the loss function
  - Iteratively pruned weights with lowest saliency and retrained
  - Able to prune 60% of the parameters and retain accuracy
  - Gradient is more "volatile" for more important parameters
- Second-order methods can take advantage of the saliency information of model parameters, leading to sparser solutions




Pruning by second derivative leads to less loss in accuracy than pruning by magnitude




Bottom curve: With retraining after pruning
Top curve: No retraining after pruning

Able to prune 60% of parameters and retain accuracy

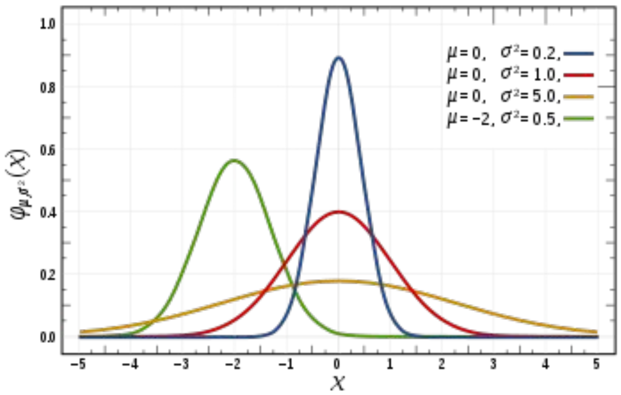
[Le Cun1990]

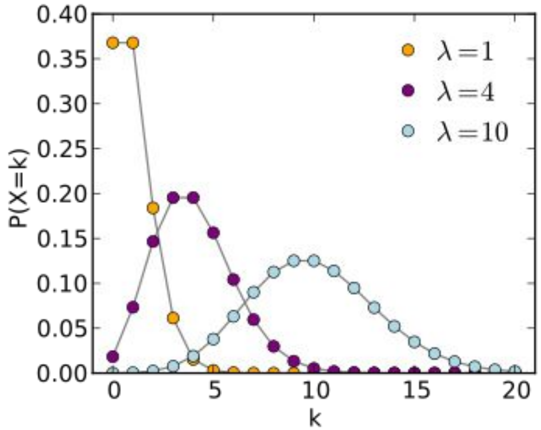
# Outline

- Context and objectives
- Tensor decompositions as explanations
- A discussion on sparsity
  - Why
  - Methods
    - L2 norm regularization
    - Second-order optimization
    - Loss function
    - Orthogonality
- Conclusion

## Gaussian vs Poisson

- Gaussian
  - Continuous-valued, includes negative support
- Poisson
  - Discrete, non-negative support




$$f(\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(N)}) \equiv \frac{1}{2}\left\| \mathcal{Z} - [\![\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(N)}]\!] \right\|^2.$$

CP-ALS Loss Function

$$f(\mathcal{M}) = \sum_{\mathbf{i}} m_{\mathbf{i}} - x_{\mathbf{i}} \log m_{\mathbf{i}},$$

CP-APR Loss Function

[Chi and Kolda 2012]

# CP-ALS vs CP-APR on Cyber Data

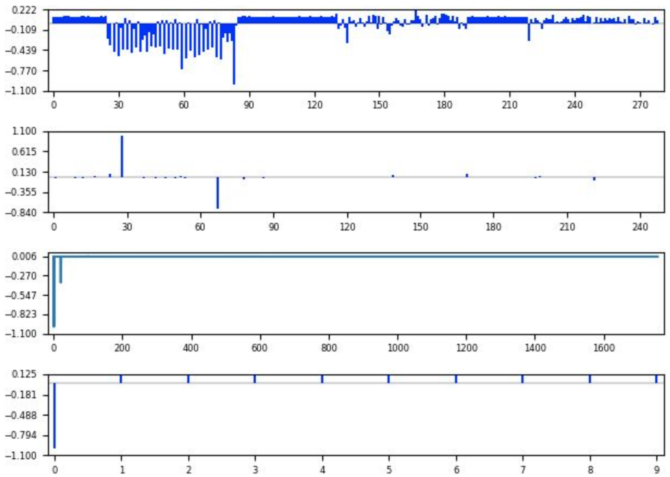
- Cyber tensor: four modes
  - *sender x receiver x sender port x hour*
- APR components are sparser and crisper
- ALS components contain more noise
  - More specialized to continuous data

$$f(\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}) \equiv \frac{1}{2} \left\| \mathcal{Z} - [\![ \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)} ]\!] \right\|^2.$$

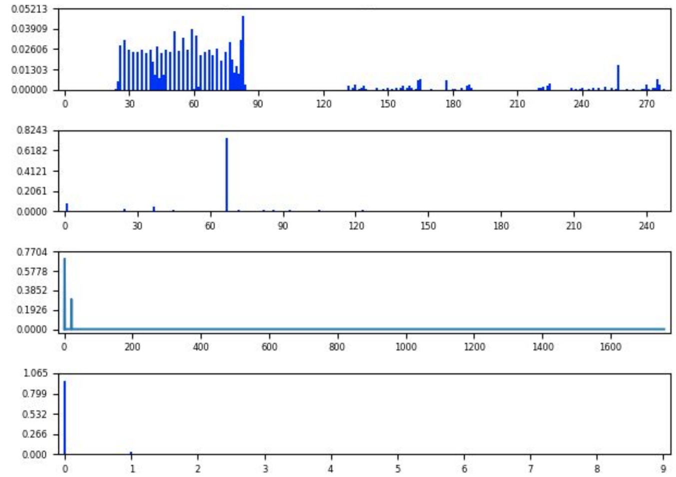
CP-ALS Loss Function

$$f(\mathcal{M}) = \sum_{\mathbf{i}} m_{\mathbf{i}} - x_{\mathbf{i}} \log m_{\mathbf{i}},$$

CP-APR Loss Function



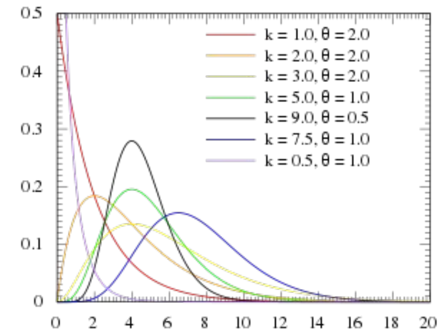
Example ALS Component



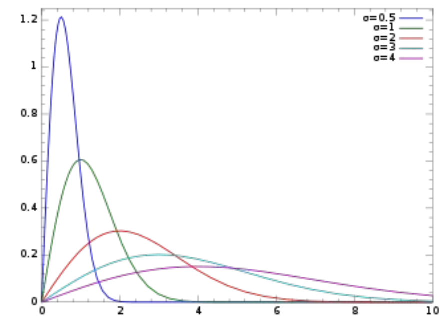
Example APR Component

## Loss Functions

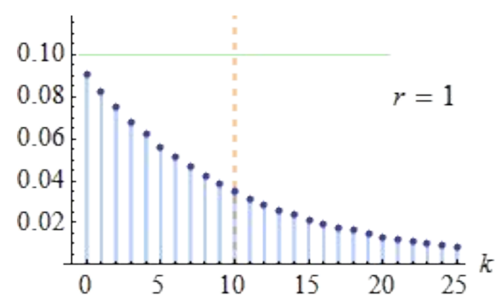
- One reason behind the success of CP-APR is that the likelihood function is a Poisson distribution
  - It is a better fit to count data
- Main idea: Use the best likelihood function for your data
  - More flexibility in model choices leads to better fit to the data
  - Different model choices lead to different levels and *types* of sparsity
  - This is the idea behind Generalized CP Decompositions [Hong and Kolda 2018]




Gamma




Rayleigh




Negative Binomial

## Outline

- Context and objectives
- Tensor decompositions as explanations
- A discussion on sparsity
  - Why
  - Methods
    - L2 norm regularization
    - Second-order optimization
    - Loss function
    - Orthogonality
- Conclusion

## Orthogonality

- Imposed by adding a regularization term to the loss function
- Hadamard product of all component matrix Gramians is shifted towards the identity matrix
- Two components are considered "orthogonal" if they have dot product zero in at least one mode
- Encourages all components to be pairwise orthogonal

$$\mathcal{F} = \underbrace{\frac{1}{2}\left\|\mathcal{X}_{(n)} - A^{(n)}\Phi^{(n)^T}\right\|_F^2 + \frac{\psi_b}{2}\sum_{n=1}^{N}\sum_{r=1}^{R}\left(\left\|A_r^{(n)}\right\| - 1\right)^2}_{\mathcal{F}_1}$$
$$+ \underbrace{\frac{\psi_a}{2}\left\|\left(\left(A^{(n)^T}A^{(n)}\right) * C_n\right) - I\right\|_F^2}_{\mathcal{F}_2}$$

[Afshar, Perros, Ho, Khalil, et. al. 2017]

## Orthogonality

- "Local" vs "Global" sparsity
  - Local: L2 regularization penalizes each term *individually*
  - Global: Orthogonality penalizes a term *with respect to other entries*
    - Penalizes more if other components already contain nonzero in that index
    - The idea is to make the components *non-overlapping*
- Increases sparsity as a *by-product*

# Decorrelating Representations

- Non-overlapping components is the same as the idea of Decorrelating Representations
- Reduces overfitting and improves interpretability
- Visual cortex decorrelates representations when adapting to new shapes

nature COMMUNICATIONS

Article | OPEN | Published: 19 September 2018

Adaptation decorrelates shape representations

Marcelo G. Mattar, Maria Olkkonen, Russell A. Epstein & Geoffrey K. Aguirre

*Nature Communications* **9**, Article number: 3812 (2018) | Download Citation

Under review as a conference paper at ICLR 2018

LEARNING LESS-OVERLAPPING REPRESENTATIONS

**Anonymous authors**
Paper under double-blind review

Published as a conference paper at ICLR 2016

REDUCING OVERFITTING IN DEEP NETWORKS BY DECORRELATING REPRESENTATIONS

**Michael Cogswell**
Virginia Tech
Blacksburg, VA
cogswell@vt.edu

**Faruk Ahmed**
Université de Montréal
Montréal, Quebec, Canada
faruk.ahmed@umontreal.ca

**Ross Girshick**
Facebook AI Research (FAIR)
Seattle, WA
rbg@fb.com

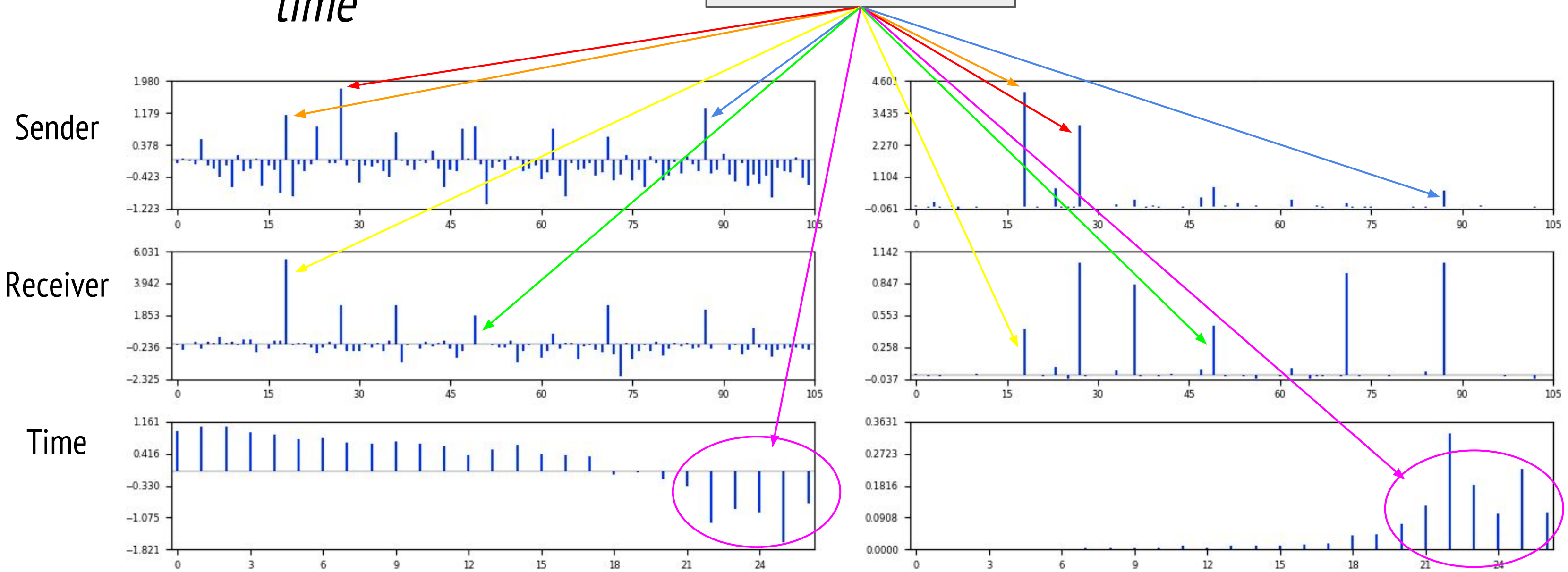
**Larry Zitnick**
Microsoft Research
Seattle, WA
larryz@microsoft.com

**Dhruv Batra**
Virginia Tech
Blacksburg, VA
dbatra@vt.edu

# Orthogonality Results

- Enron emails dataset: three mode tensor
  - *sender x receiver x time*
- Orthogonality cleans up noise, sparsifies and isolates dominant spikes

Corresponding spikes




Without orthogonality

With orthogonality

## Summary of Methods

| Sparsification Approach | Benefit | Drawback |
|---|---|---|
| **L2 norm regularization** | Directly penalize each individual parameter | Doesn't take into account "global" information |
| **Second-order optimization** | Use optimization procedure that utilizes "saliency" information | More computationally expensive |
| **Loss function** | Change underlying model to fit data generating process better | Is a subjective decision |
| **Orthogonality** | Decorrelate representations, reduce overfitting | True patterns of activity might not be completely decorrelated |

No free lunch!

## Outline

- Context and objectives
- Tensor decompositions as explanations
- A discussion on sparsity
  - Why
  - Methods
    - L2 norm regularization
    - Second-order optimization
    - Loss function
    - Orthogonality
- Conclusion

## Conclusion

- Methods induce sparsity directly and indirectly
- Reasoning for each method is based on making the model/algorithm more “cognitive,” leads to interpretability/sparsity as a byproduct
    - Focus is on algorithm’s process, not results
- Conclusion: to have more interpretable models, think from the model’s perspective - give it the right framework to work with (loss function), the right information (“saliency”), and slightly nudge it in the right direction (decorrelate representations)
    - Important to have a tool which has these multiple capabilities

# References

Afshar, A., Perros, I., Ho, J. C., Khalil, E. B., Sunderam, V., Dilkina, B., & Xiong, L. (2017). CP-ORTHO: An Orthogonal Tensor Factorization Framework for Spatio-Temporal Data. In *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems* (Vol. 2017–Novem).

Chi, Eric & G. Kolda, Tamara. (2011). On Tensors, Sparsity, and Nonnegative Factorizations. SIAM Journal on Matrix Analysis and Applications. 33. 1272–1299. 10.1137/110859063.

Candes, E. J., Romberg, J. K., & Tao, T. (2006). Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, *59*(8), 1207-1223.

Cun, Y. Le, Denker, J. S., & Solla, S. A. (1990). Optimal Brain Damage. In *Advances in Neural Information Processing Systems 2* (Vol. 2, pp. 598–605). https://doi.org/10.1.1.32.7223

Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on information theory*, *52*(4), 1289-1306.

Hansen, S., Plantenga, T., & Kolda, T. G. (2015). Newton-based optimization for Kullback-Leibler nonnegative tensor factorizations. *Optimization Methods and Software*, *30*(5), 1002–1029.

Hong, D., Kolda, T. G., & Duersch, J. A. (2018). Generalized Canonical Polyadic Tensor Decomposition. Retrieved from http://arxiv.org/abs/1808.07452

Jean-Philip Royer, Pierre Comon, Nadège Thirion-Moreau. Computing the nonnegative 3-way tensor factorization using Tikhonov regularization. IEEE. ICASSP, May 2011, Prague, Czech Republic. IEEE, pp.2732-2735, 2011.

# References (Reservoir Labs, Part 1/2)

Letourneau, P.D., Baskaran, M., Henretty, T., Ezick, J. and Lethin, R. (2018) Computationally Efficient CP Tensor Decomposition Update Framework for Emerging Component Discovery in Streaming Data. In High Performance Extreme Computing (HPEC) Conference. IEEE. [Best Paper Award]. (https://www.reservoir.com/publication/cp-tensor-decomposition-update-framework/)

Gudibanda, A., Henretty, T., Baskaran, M., Ezick, J. and Lethin, R. (2018), All-at-once Decomposition of Coupled Billion-scale Tensors in Apache Spark. In *High Performance Extreme Computing (HPEC) Conference*. IEEE. (https://www.reservoir.com/publication/coupled-tensor-decomposition-apache-spark/)

Henretty, T. S., Langston, M. H., Baskaran, M., Ezick, J., & Lethin, R. (2018). Topic modeling for analysis of big data tensor decompositions. In *Disruptive Technologies in Information Sciences*. International Society for Optics and Photonics. (https://www.reservoir.com/publication/topic-modeling-for-analysis-of-big-data-tensor-decompositions/)

Baskaran, M. M., Henretty, T., Ezick, J., Lethin, R., & Bruns-Smith, D. (2017). Enhancing Network Visibility and Security through Tensor Analysis. In *4th International Workshop on Innovating the Network for Data Intensive Science (INDIS) held in conjunction with SC17*. (https://www.reservoir.com/publication/enhancing-network-visibility-security-tensor-analysis/)

Baskaran, M., Henretty, T., Pradelle, B., Langston, M. H., Bruns-Smith, D., Ezick, J., & Lethin, R. (2017). Memory-efficient parallel tensor decompositions. In *High Performance Extreme Computing Conference (HPEC)*. IEEE. [Best paper award] (https://www.reservoir.com/publication/memory-efficient-parallel-tensor-decompositions/)

Henretty, T., Baskaran, M., Ezick, J., Bruns-Smith, D., & Simon, T. A. (2017). A quantitative and qualitative analysis of tensor decompositions on spatiotemporal data. In *High Performance Extreme Computing Conference (HPEC)*. IEEE. (https://www.reservoir.com/publication/quantitative-qualitative-analysis-tensor-decompositions-spatiotemporal-data/)

# References (Reservoir Labs, Part 2/2)

Baskaran, M., Langston, M. H., Ramananandro, T., Bruns-Smith, D., Henretty, T., Ezick, J., & Lethin, R. (2016). Accelerated low-rank updates to tensor decompositions. In *High Performance Extreme Computing Conference (HPEC)*. IEEE. (https://www.reservoir.com/publication/accelerated-low-rank-updates-tensor-decompositions/)

Bruns-Smith, D., Baskaran, M. M., Ezick, J., Henretty, T., & Lethin, R. (2016). Cyber security through multidimensional data decompositions. In *2016 Cybersecurity Symposium (CYBERSEC)*. IEEE. (https://www.reservoir.com/publication/cyber-security-multidimensional-data-decompositions/)

Cai, J., Baskaran, M., Meister, B., & Lethin, R. (2015). Optimization of symmetric tensor computations. In *High Performance Extreme Computing Conference (HPEC)*.. IEEE. (https://www.reservoir.com/publication/optimization-symmetric-tensor-computations/)

Baskaran, M., Meister, B., & Lethin, R. (2014). Low-overhead load-balanced scheduling for sparse tensor computations. In *High Performance Extreme Computing Conference (HPEC)*. IEEE. (https://www.reservoir.com/publication/low-overhead-load-balanced-scheduling-sparse-tensor-computations/)

Baskaran, M. M., Meister, B., & Lethin, R. (2014). Parallelizing and optimizing sparse tensor computations. In *Proceedings of the 28th ACM international conference on Supercomputing*. ACM. (https://www.reservoir.com/publication/parallelizing-optimizing-sparse-tensor-computations/)

Baskaran, M., Meister, B., Vasilache, N., & Lethin, R. (2012). Efficient and scalable computations with sparse tensors. In *High Performance Extreme Computing (HPEC)*. IEEE. (https://www.reservoir.com/publication/efficient-scalable-computations-sparse-tensors/)