Tensor Methods for large-scale Machine Learning

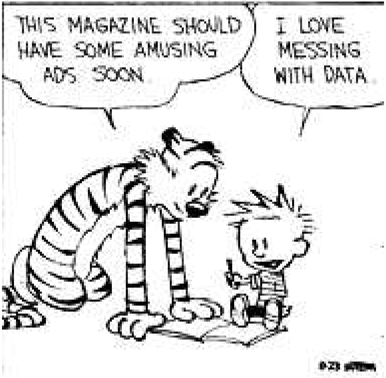
Anima Anandkumar

U.C. Irvine

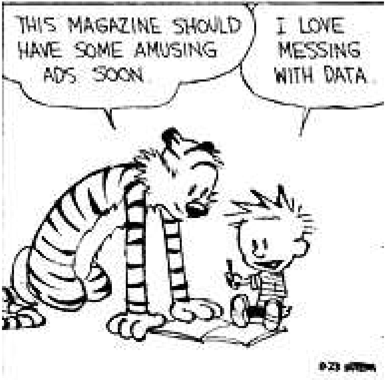
Learning with Big Data

Data vs. Information

Data vs. Information

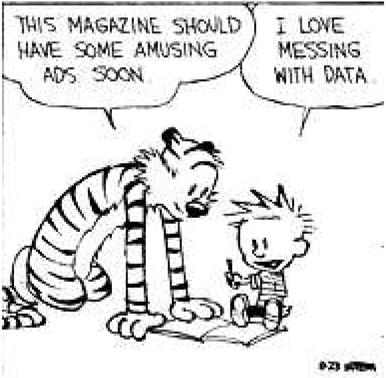


Data vs. Information



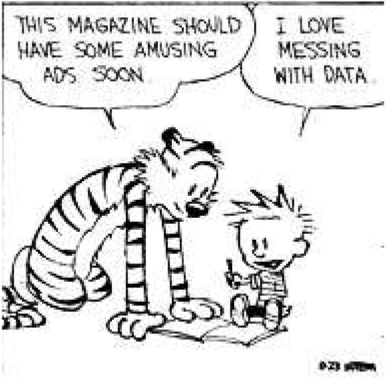
- Missing observations, gross corruptions, outliers.

Data vs. Information



- Missing observations, gross corruptions, outliers.
- High dimensional regime: as data grows, more variables !

Data vs. Information



- Missing observations, gross corruptions, outliers.
- High dimensional regime: as data grows, more variables !

Data deluge an information desert!

Learning in High Dimensional Regime

- Useful information: low-dimensional structures.
- Learning with big data: ill-posed problem.

Learning in High Dimensional Regime

- Useful information: low-dimensional structures.
- Learning with big data: ill-posed problem.

Learning is finding needle in a haystack

Learning in High Dimensional Regime

- Useful information: low-dimensional structures.
- Learning with big data: ill-posed problem.

Learning is finding needle in a haystack

- Learning with big data: computationally challenging!

Principled approaches for finding low dimensional structures?

How to model information structures?

Latent variable models

- Incorporate hidden or latent variables.
- Information structures: Relationships between latent variables and observed data.

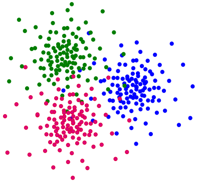
How to model information structures?

Latent variable models

- Incorporate hidden or latent variables.
- Information structures: Relationships between latent variables and observed data.

Basic Approach: mixtures/clusters

- Hidden variable is categorical.



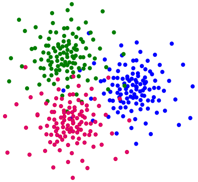
How to model information structures?

Latent variable models

- Incorporate hidden or latent variables.
- Information structures: Relationships between latent variables and observed data.

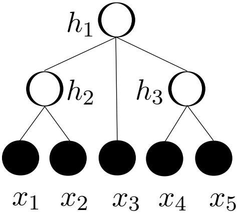
Basic Approach: mixtures/clusters

- Hidden variable is categorical.



Advanced: Probabilistic models

- Hidden variables have more general distributions.
- Can model mixed membership/hierarchical groups.



Latent Variable Models (LVMs)

Document modeling

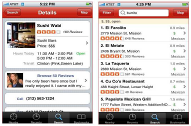
- Observed: words.
- Hidden: topics.

Social Network Modeling

- Observed: social interactions.
- Hidden: communities, relationships.

Recommendation Systems

- Observed: recommendations (e.g., reviews).
- Hidden: User and business attributes



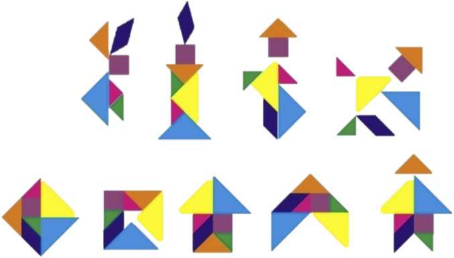
Unsupervised Learning: Learn LVM without labeled examples.

LVM for Feature Engineering

- Learn good features/representations for classification tasks, e.g., computer vision and NLP.

Sparse Coding/Dictionary Learning

- Sparse representations, low dimensional hidden structures.
- A few dictionary elements make complicated shapes.



Challenges in Learning LVMs

Computational Challenges

- Maximum likelihood: non-convex optimization. NP-hard.
- Practice: Local search approaches such as gradient descent, EM, Variational Bayes have no consistency guarantees.
- Can get stuck in bad local optima. Poor convergence rates.
- Hard to paralle

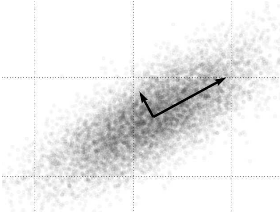
Alternatives? Guaranteed and efficient learning?

Outline

Classical Spectral Methods: Matrix PCA

For centered samples $\{x_i\}$, find projection P with $\text{Rank}(P) = k$ s.t.

$$\boxed{\min_P \frac{1}{n} \sum_{i \in [n]} \|x_i - Px_i\|^2.}$$



Result: Eigen-decomposition of $\text{Cov}(X)$.

Beyond PCA: Spectral Methods on Tensors?

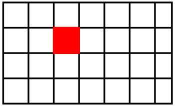
Moment Matrices and Tensors

Multivariate Moments

$$M_1 := \mathbb{E}[x], \quad M_2 := \mathbb{E}[x \otimes x], \quad M_3 := \mathbb{E}[x \otimes x \otimes x].$$

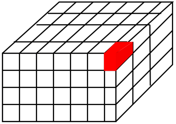
Matrix

- $\mathbb{E}[x \otimes x] \in \mathbb{R}^{d \times d}$ is a second order tensor.
- $\mathbb{E}[x \otimes x]_{i_1,i_2} = \mathbb{E}[x_{i_1} x_{i_2}]$.
- For matrices: $\mathbb{E}[x \otimes x] = \mathbb{E}[xx^\top]$.



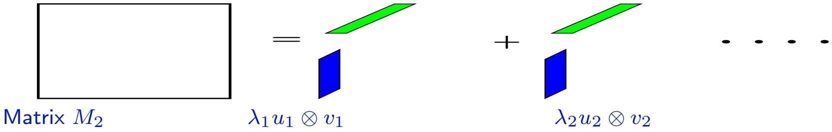
Tensor

- $\mathbb{E}[x \otimes x \otimes x] \in \mathbb{R}^{d \times d \times d}$ is a third order tensor.
- $\mathbb{E}[x \otimes x \otimes x]_{i_1,i_2,i_3} = \mathbb{E}[x_{i_1} x_{i_2} x_{i_3}]$.



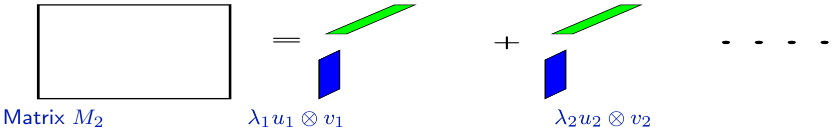
Spectral Decomposition of Tensors

$$\boxed{M_2 = \sum_i \lambda_i u_i \otimes v_i}$$

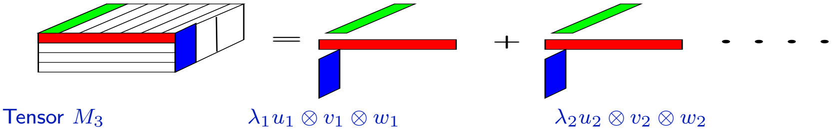


Spectral Decomposition of Tensors

$$\boxed{M_2 = \sum_i \lambda_i u_i \otimes v_i}$$



$$\boxed{M_3 = \sum_i \lambda_i u_i \otimes v_i \otimes w_i}$$



- $u \otimes v \otimes w$ is a rank-1 tensor since its $(i_1, i_2, i_3)^{\text{th}}$ entry is $u_{i_1} v_{i_2} w_{i_3}$.

How to solve this non-convex problem?

Orthogonal Tensor Power Method

Symmetric orthogonal tensor $T \in \mathbb{R}^{d\times d\times d}$:

$$T = \sum_{i\in[k]} \lambda_i v_i \otimes v_i \otimes v_i.$$

Orthogonal Tensor Power Method

Symmetric orthogonal tensor $T \in \mathbb{R}^{d \times d \times d}$:

$$T = \sum_{i \in [k]} \lambda_i v_i \otimes v_i \otimes v_i.$$

Recall matrix power method: $v \mapsto \dfrac{M(I,v)}{\|M(I,v)\|}$.

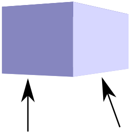
Orthogonal Tensor Power Method

Symmetric orthogonal tensor $T \in \mathbb{R}^{d \times d \times d}$:

$$T = \sum_{i \in [k]} \lambda_i v_i \otimes v_i \otimes v_i.$$

Recall matrix power method: $v \mapsto \dfrac{M(I, v)}{\|M(I, v)\|}$.

Algorithm: tensor power method: $\boxed{v \mapsto \dfrac{T(I, v, v)}{\|T(I, v, v)\|}}$.



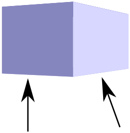
Orthogonal Tensor Power Method

Symmetric orthogonal tensor $T \in \mathbb{R}^{d \times d \times d}$:

$$T = \sum_{i \in [k]} \lambda_i v_i \otimes v_i \otimes v_i.$$

Recall matrix power method: $v \mapsto \dfrac{M(I,v)}{\|M(I,v)\|}$.

Algorithm: tensor power method: $\boxed{v \mapsto \dfrac{T(I,v,v)}{\|T(I,v,v)\|}}$.



How do we avoid spurious solutions (not part of decomposition)?

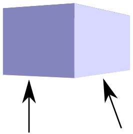
Orthogonal Tensor Power Method

Symmetric orthogonal tensor $T \in \mathbb{R}^{d\times d\times d}$:

$$T = \sum_{i\in[k]} \lambda_i v_i \otimes v_i \otimes v_i.$$

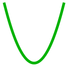
Recall matrix power method: $v \mapsto \dfrac{M(I,v)}{\|M(I,v)\|}$.

Algorithm: tensor power method: $\boxed{v \mapsto \dfrac{T(I,v,v)}{\|T(I,v,v)\|}}$.



How do we avoid spurious solutions (not part of decomposition)?

- $\{v_i\}$'s are the only robust fixed points.



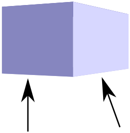
Orthogonal Tensor Power Method

Symmetric orthogonal tensor $T \in \mathbb{R}^{d\times d\times d}$:

$$T = \sum_{i\in[k]} \lambda_i v_i \otimes v_i \otimes v_i.$$

Recall matrix power method: $v \mapsto \dfrac{M(I,v)}{\|M(I,v)\|}$.

Algorithm: tensor power method: $\boxed{v \mapsto \dfrac{T(I,v,v)}{\|T(I,v,v)\|}}$.



How do we avoid spurious solutions (not part of decomposition)?

- $\{v_i\}$'s are the only robust fixed points.

- All other eigenvectors are saddle points.

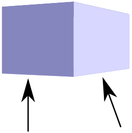
Orthogonal Tensor Power Method

Symmetric orthogonal tensor $T \in \mathbb{R}^{d\times d\times d}$:

$$T = \sum_{i\in[k]} \lambda_i v_i \otimes v_i \otimes v_i.$$

Recall matrix power method: $v \mapsto \dfrac{M(I,v)}{\|M(I,v)\|}$.

Algorithm: tensor power method: $\boxed{v \mapsto \dfrac{T(I,v,v)}{\|T(I,v,v)\|}}$.



How do we avoid spurious solutions (not part of decomposition)?

- $\{v_i\}$'s are the only robust fixed points.

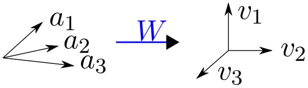
- All other eigenvectors are saddle points.

For an orthogonal tensor, no spurious local optima!

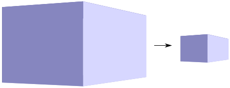
Putting it together

Non-orthogonal tensor $M_3 = \sum_i w_i a_i \otimes a_i \otimes a_i$, $M_2 = \sum_i w_i a_i \otimes a_i$.

- Whitening matrix W:



- Multilinear transform: $T = M_3(W, W, W)$

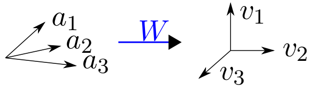


Tensor M_3 Tensor T

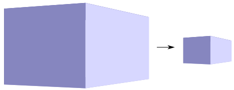
Putting it together

Non-orthogonal tensor $M_3 = \sum_i w_i a_i \otimes a_i \otimes a_i$, $M_2 = \sum_i w_i a_i \otimes a_i$.

- Whitening matrix W:



- Multilinear transform: $T = M_3(W, W, W)$



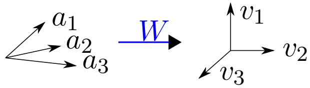
Tensor M_3 Tensor T

Tensor Decomposition: Guaranteed Non-Convex Optimization!

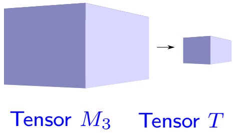
Putting it together

Non-orthogonal tensor $M_3 = \sum_i w_i a_i \otimes a_i \otimes a_i$, $M_2 = \sum_i w_i a_i \otimes a_i$.

- Whitening matrix W:



- Multilinear transform: $T = M_3(W, W, W)$

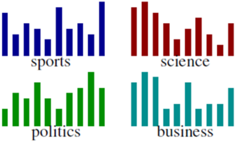


Tensor Decomposition: Guaranteed Non-Convex Optimization!

For what latent variable models can we obtain M_2 and M_3 forms?

Outline

Topic Modeling

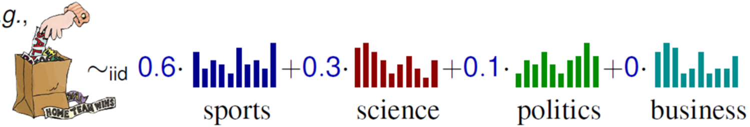


k topics (distributions over vocab words).

Each document $\leftrightarrow$ mixture of topics.

Words in document $\sim_{\text{iid}}$ mixture dist.

E.g.,



aardvark	0
athlete	3
	⋮
zygote	1

$\Pr_\theta[\text{“play”} \mid \text{sports}] = 0.0002$

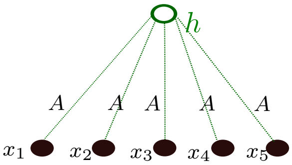
$\Pr_\theta[\text{“game”} \mid \text{sports}] = 0.0003$

$\Pr_\theta[\text{“season”} \mid \text{sports}] = 0.0001$

⋮

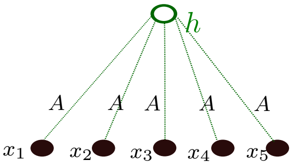
Moments for Single Topic Models

- $\boxed{\mathbb{E}[x_i|h] = Ah.}$
- $\boxed{w := \mathbb{E}[h].}$
- Learn topic-word matrix A, vector w



Moments for Single Topic Models

- $\boxed{\mathbb{E}[x_i|h] = Ah.}$
- $\boxed{w := \mathbb{E}[h].}$
- Learn topic-word matrix A, vector w



Pairwise Co-occurence Matrix M_2

$$M_2 := \mathbb{E}[x_1 \otimes x_2] = \mathbb{E}[\mathbb{E}[x_1 \otimes x_2|h]] = \sum_{i=1}^{k} w_i a_i \otimes a_i$$

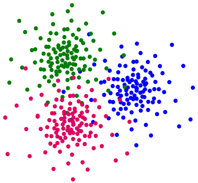
Triples Tensor M_3

$$M_3 := \mathbb{E}[x_1 \otimes x_2 \otimes x_3] = \mathbb{E}[\mathbb{E}[x_1 \otimes x_2 \otimes x_3|h]] = \sum_{i=1}^{k} w_i a_i \otimes a_i \otimes a_i$$

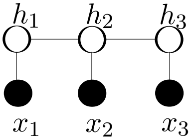
- Can be extended to learning LDA: mutiple topics in a document.

Tractable Learning for LVMs

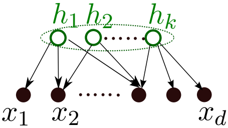
GMM



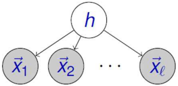
HMM



ICA



Multiview and Topic Models



$h \in [k]$,

$\vec{x}_1 \in \mathbb{R}^{d_1}, \vec{x}_2 \in \mathbb{R}^{d_2}, \ldots, \vec{x}_\ell \in \mathbb{R}^{d_\ell}$.

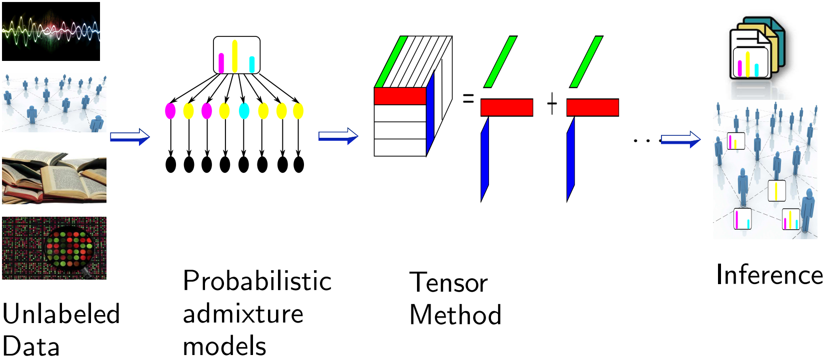
k = # components, ℓ = # views (*e.g.*, audio, video, text).

View 1: $\vec{x}_1 \in \mathbb{R}^{d_1}$

View 2: $\vec{x}_2 \in \mathbb{R}^{d_2}$

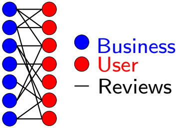
View 3: $\vec{x}_3 \in \mathbb{R}^{d_3}$

Overall Framework



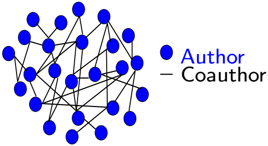
Outline

Learning Communities through Tensor Methods



Yelp

$n \sim 40k$



DBLP(sub)

$n \sim 1$ million($\sim 100k$)

Error ($\mathcal{E}$) and Recovery ratio ($\mathcal{R}$)

Dataset	$\hat{k}$	Method	**Running Time**	$\mathcal{E}$	$\mathcal{R}$
DBLP sub(k=250)	500	ours	10,157	0.139	89%
DBLP sub(k=250)	500	variational	558,723	16.38	99%
DBLP(k=6000)	100	ours	5407	0.105	95%

Thanks to Prem Gopalan and David Mimno for providing variational code.

Experimental Results on Yelp

Lowest error business categories & largest weight businesses

Rank	Category	Business	Stars	Review Counts
1	Latin American	Salvadoreno Restaurant	4.0	36
2	Gluten Free	P.F. Chang's China Bistro	3.5	55
3	Hobby Shops	Make Meaning	4.5	14
4	Mass Media	KJZZ 91.5FM	4.0	13
5	Yoga	Sutra Midtown	4.5	31

Experimental Results on Yelp

Lowest error business categories & largest weight businesses

Rank	Category	Business	Stars	Review Counts
1	Latin American	Salvadoreno Restaurant	4.0	36
2	Gluten Free	P.F. Chang's China Bistro	3.5	55
3	Hobby Shops	Make Meaning	4.5	14
4	Mass Media	KJZZ 91.5FM	4.0	13
5	Yoga	Sutra Midtown	4.5	31

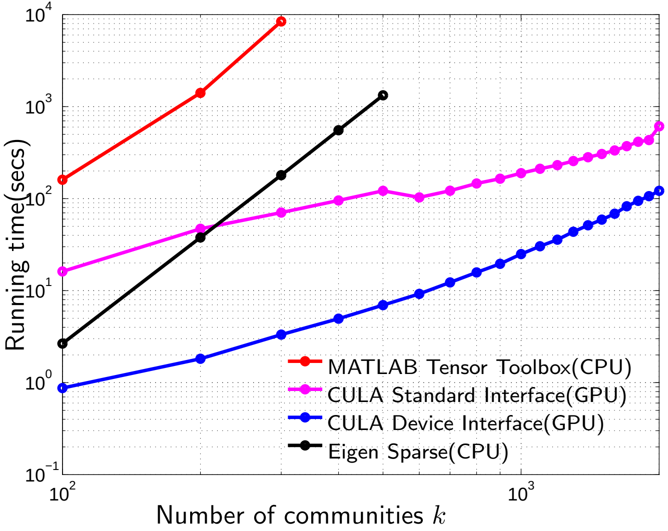
Bridgeness: Distance from vector $[1/\hat{k}, \ldots, 1/\hat{k}]^\top$

Top-5 bridging nodes (businesses)

Business	Categories
Four Peaks Brewing	Restaurants, Bars, American, Nightlife, Food, Pubs, Tempe
Pizzeria Bianco	Restaurants, Pizza, Phoenix
FEZ	Restaurants, Bars, American, Nightlife, Mediterranean, Lounges, Phoenix
Matt's Big Breakfast	Restaurants, Phoenix, Breakfast& Brunch
Cornish Pasty Co	Restaurants, Bars, Nightlife, Pubs, Tempe

Tensor Decomposition on GPUs

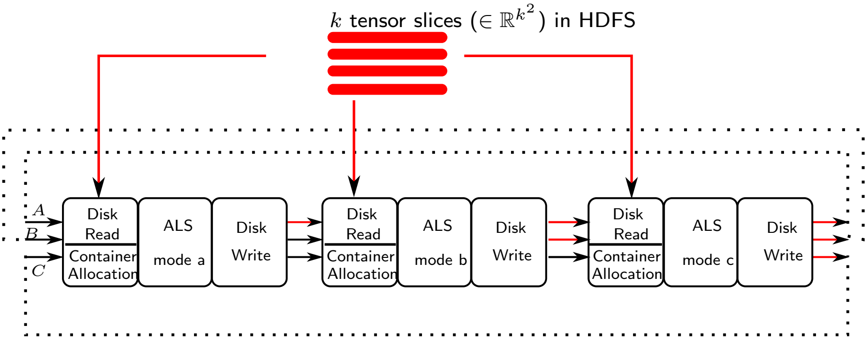
Embarrassingly Parallel and fast!



Tensor Methods on the Cloud

Communication and System Architecture Overhead

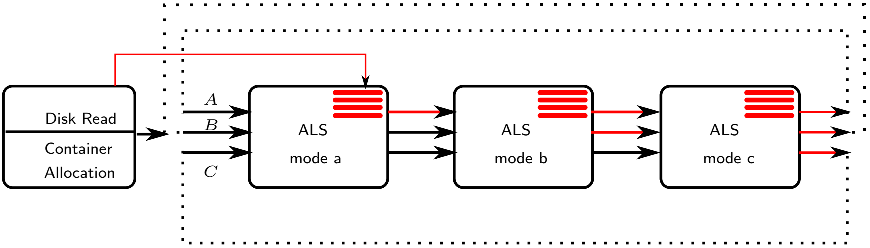
- Map-Reduce Framework



- Overhead: Disk reading, Container Allocation, Intense Key/Value Design

Tensor Methods on Cloud

Solution: Retainable Evaluator Execution Framework (REEF)



- Open source distributed system
- One time container allocation
- keep the tensor in memory

www.reef-project.org

Initial Results from Cloud Implementation

New York Times Corpus

	Stochastic Variational Inference (SVI)	Tensor Decomposition
Perplexity	4000	3400

	SVI	1 node Map Red	1 node REEF	4 node REEF
overall	2 hours	4 hours 31 mins	68 mins	36 mins
Whiten		16 mins	16 mins	16 mins
Matricize		15 mins	15 mins	4 mins
ALS		4 hours	37 mins	16 mins

Outline

Conclusion: Tensor Methods for Learning

Tensor Decomposition

- Efficient sample and computational complexities
- Better performance compared to EM, Variational Bayes etc.

In practice

- Scalable and embarrassingly parallel: handle large datasets.
- Efficient performance: perplexity or ground truth validation.

Related Topics

- Tensor Methods for Discriminative Learning: Learning neural networks, mixtures of classifiers, etc.
- Overcomplete Tensor Decomposition: Neural networks, sparse coding and ICA models tend to be overcomplete (more neurons than input dimensions).

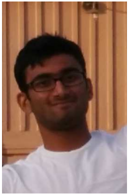
My Research Group and Resources

Furong Huang

Majid Janzamin

Hanie Sedghi

Niranjan UN



Forough Arabshahi

- ML summer school lectures available at
  http://newport.eecs.uci.edu/anandkumar/MLSS.html