Guaranteed Machine Learning using Tensor Methods

Anima Anandkumar

U.C. Irvine

Tutorial at MIT 2016

Regime of Modern Machine Learning

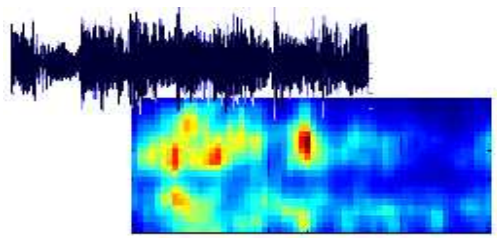
Massive datasets, growth in computation power, challenging tasks

Success of Supervised Learning

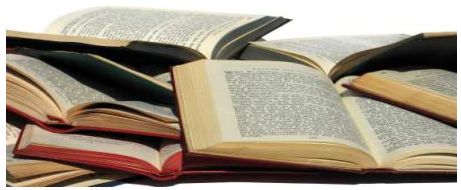
- Learn $p(y|x)$ from labeled samples $\{(x_i, y_i)\}$.
- Extract relevant features from large amounts of labeled data.



Image classification



Speech recognition



Text processing

Regime of Modern Machine Learning

Massive datasets, growth in computation power, challenging tasks

Missing Link in AI: Unsupervised Learning

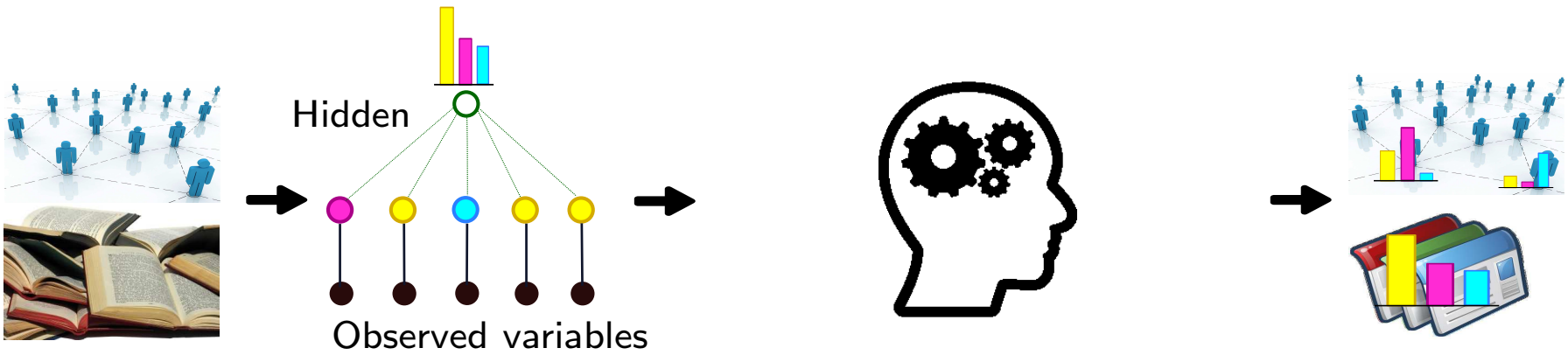
- Learn $p(x)$ from unlabeled samples $\{x_i\}$.
- Discover latent variables related to observed variable x.
- Human vs. Machine Learning: Make discoveries automatically.



Image Credit: Whizz Education

Unsupervised Learning via Probabilistic Models

Data $\rightarrow$ Model $\rightarrow$ Learning Algorithm $\rightarrow$ Predictions



Challenges in High dimensional Learning

- Dimension of $x \gg$ dim. of latent variable h.
- Learning is like finding needle in a haystack.
- Computationally & statistically challenging.



Overview of Unsupervised Learning Methods

Goal: learn model parameters θ from observations x.

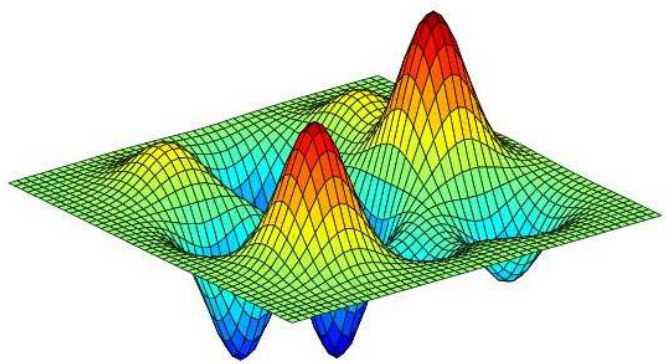
Bayesian: treat θ as random

- Compute the posterior distribution $p(\theta|x)$.
- Algorithm: Markov chain Monte Carlo (MCMC).
- Slow: Exponential mixing time.



Frequentist: treat θ as fixed

- Maximum likelihood: $\max_{\theta} p(x;\theta)$.
- Non-convex: stuck in local optima.
- Curse of dimensionality: Exponential no. of critical points.
- Heuristics: Expectation Maximization, Variational Inference ....

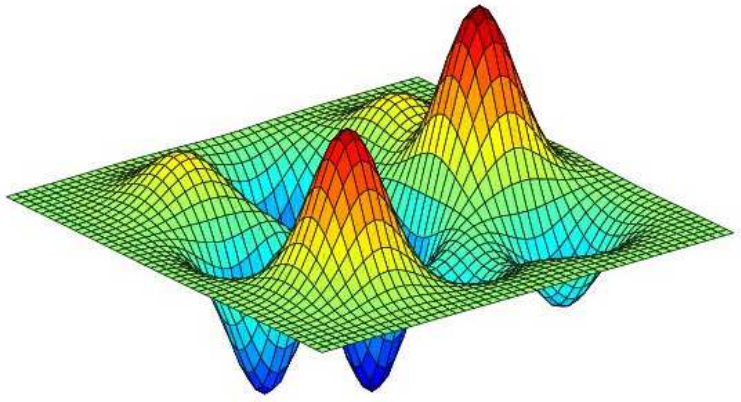


Guaranteed Learning through Tensor Methods



Replace the objective function

Max Likelihood vs. Best Tensor decomp.

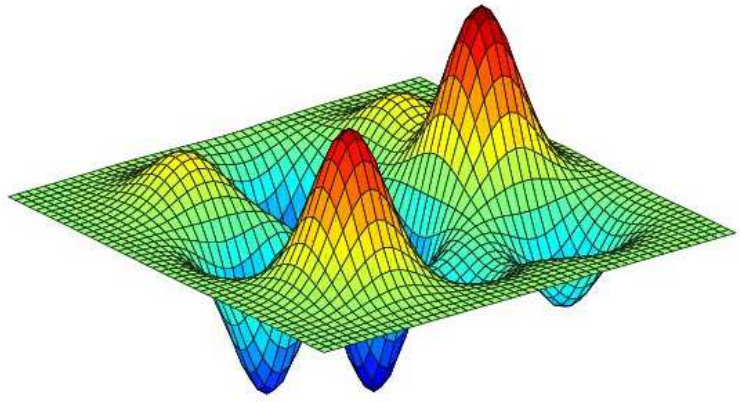


Preserves Global Optimum (infinite samples)

$$\arg\max_{\theta} p(x;\theta) = \arg\min_{\theta} \|\widehat{T}(x) - T(\theta)\|_{\mathbb{F}}^2$$

$\widehat{T}(x)$: empirical tensor, $T(\theta)$: low rank tensor based on θ.

Guaranteed Learning through Tensor Methods





Replace the objective function

Max Likelihood vs. Best Tensor decomp.

Preserves Global Optimum (infinite samples)

$$\arg\max_{\theta} p(x;\theta) = \arg\min_{\theta} \|\widehat{T}(x) - T(\theta)\|_{\mathbb{F}}^2$$

$\widehat{T}(x)$: empirical tensor, $T(\theta)$: low rank tensor based on θ.



Finding globally opt tensor decomposition

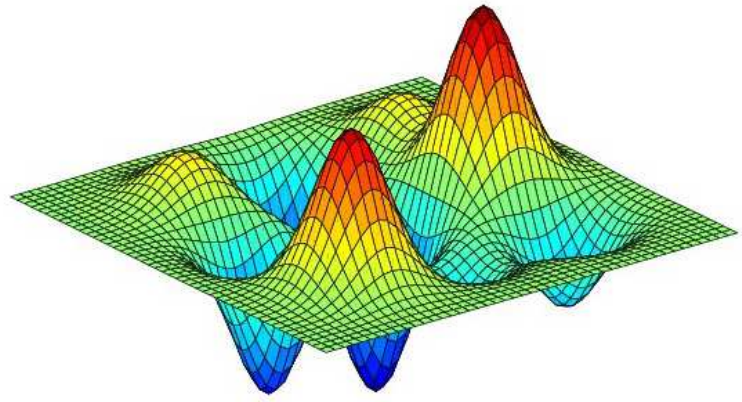
Simple algorithms succeed under mild and natural conditions for many learning problems.

Guaranteed Learning through Tensor Methods



Replace the objective function

Max Likelihood vs. Best Tensor decomp.



Preserves Global Optimum (infinite samples)

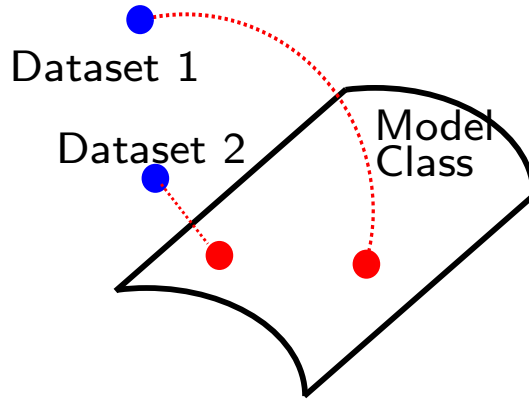
$$\arg\max_{\theta} p(x;\theta) = \arg\min_{\theta} \|\widehat{T}(x) - T(\theta)\|_{\mathbb{F}}^2$$

$\widehat{T}(x)$: empirical tensor, $T(\theta)$: low rank tensor based on θ.



Finding globally opt tensor decomposition

Simple algorithms succeed under mild and natural conditions for many learning problems.



Outline

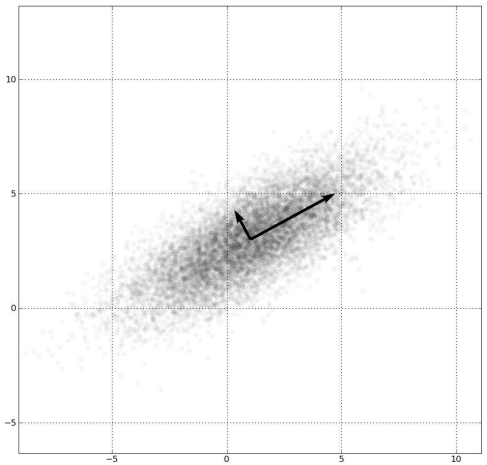
1. Introduction
2. Why Tensors?
3. Algorithms for Tensor Decomposition
4. Learning Probabilistic Models
5. Learning Representations
6. Conclusion

Recall PCA

Optimization problem

For (centered) points $x_i \in \mathbb{R}^d$, find projection P with Rank$(P) = k$ s.t.

$$\min_{P \in \mathbb{R}^{d \times d}} \frac{1}{n} \sum_{i \in [n]} \|x_i - P x_i\|^2.$$

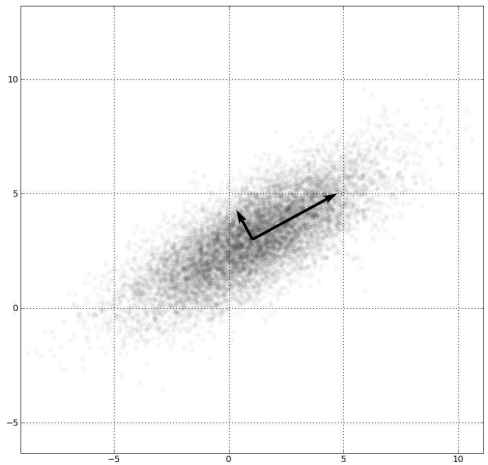


Recall PCA

Optimization problem

For (centered) points $x_i \in \mathbb{R}^d$, find projection P with Rank$(P) = k$ s.t.

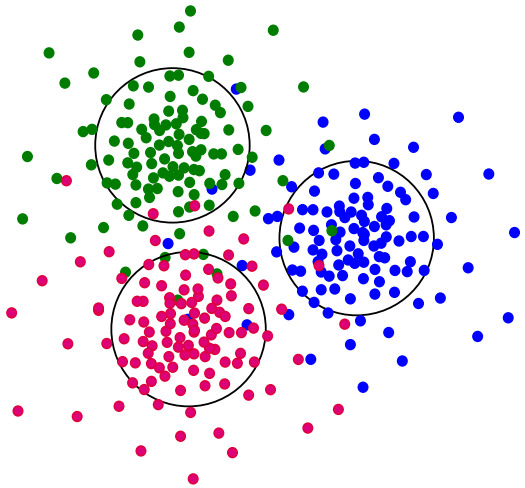
$$\min_{P \in \mathbb{R}^{d \times d}} \frac{1}{n} \sum_{i \in [n]} \|x_i - P x_i\|^2.$$



Result: If $X = [x_1 | x_2 \ldots]$ is data matrix, $S = \text{Cov}(X)$, and $S = U \Lambda U^\top$ is eigen decomposition, we have $P = U_{(k)} U_{(k)}^\top$, where $U_{(k)}$ are top-k eigen vectors.

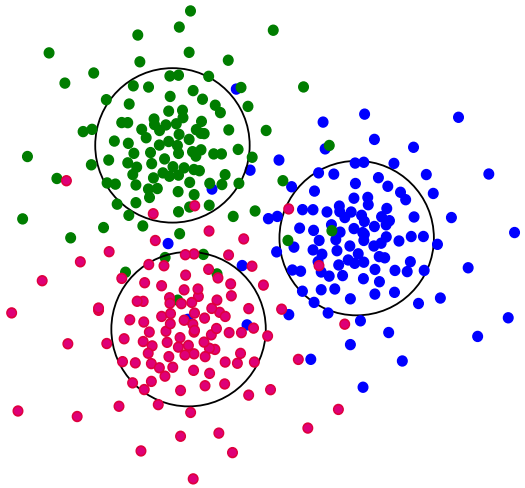
PCA on Gaussian Mixtures

- k Gaussians: each sample is $x = Ah + z$.
- $h \in [e_1, \ldots, e_k]$, the basis vectors. $\mathbb{E}[h] = w$.
- $A \in \mathbb{R}^{d \times k}$: columns are component means.
- Let $\mu := Aw$ be the mean.
- $z \sim \mathcal{N}(0, \sigma^2 I)$ is white Gaussian noise.



PCA on Gaussian Mixtures

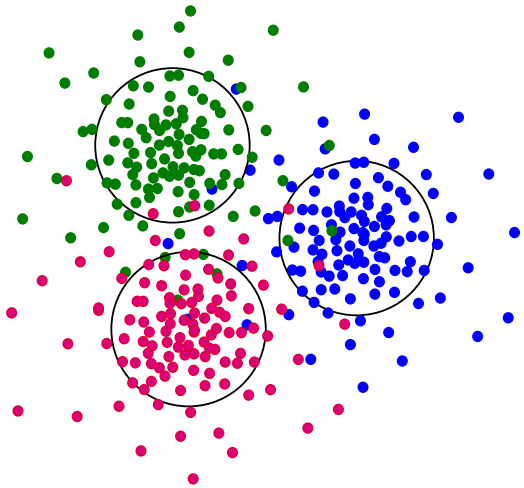
- k Gaussians: each sample is $x = Ah + z$.
- $h \in [e_1, \ldots, e_k]$, the basis vectors. $\mathbb{E}[h] = w$.
- $A \in \mathbb{R}^{d \times k}$: columns are component means.
- Let $\mu := Aw$ be the mean.
- $z \sim \mathcal{N}(0, \sigma^2 I)$ is white Gaussian noise.



$$\mathbb{E}[(x-\mu)(x-\mu)^\top] = \sum_{i \in [k]} w_i (a_i - \mu)(a_i - \mu)^\top + \sigma^2 I.$$

PCA on Gaussian Mixtures

- k Gaussians: each sample is $x = Ah + z$.
- $h \in [e_1, \ldots, e_k]$, the basis vectors. $\mathbb{E}[h] = w$.
- $A \in \mathbb{R}^{d \times k}$: columns are component means.
- Let $\mu := Aw$ be the mean.
- $z \sim \mathcal{N}(0, \sigma^2 I)$ is white Gaussian noise.



$$\mathbb{E}[(x-\mu)(x-\mu)^\top] = \sum_{i \in [k]} w_i (a_i - \mu)(a_i - \mu)^\top + \sigma^2 I.$$

How the above equation is obtained

$$\begin{aligned}\mathbb{E}[(x-\mu)(x-\mu)^\top] &= \mathbb{E}[(Ah-\mu)(Ah-\mu)^\top] + \mathbb{E}[zz^\top] \\ &= \sum_{i \in [k]} w_i (a_i - \mu)(a_i - \mu)^\top + \sigma^2 I.\end{aligned}$$

Example 1: PCA on Gaussian Mixtures

$$\mathbb{E}[(x-\mu)(x-\mu)^\top] = \sum_{i\in[k]} w_i(a_i-\mu)(a_i-\mu)^\top + \sigma^2 I.$$

- The vectors $\{a_i - \mu\}$ are linearly dependent: $\sum_i w_i(a_i-\mu) = 0$. The PSD matrix $\sum_{i\in[k]} w_i(a_i-\mu)(a_i-\mu)^\top$ has rank $\leq k-1$.

Example 1: PCA on Gaussian Mixtures

$$\mathbb{E}[(x-\mu)(x-\mu)^\top] = \sum_{i\in[k]} w_i(a_i-\mu)(a_i-\mu)^\top + \sigma^2 I.$$

- The vectors $\{a_i - \mu\}$ are linearly dependent: $\sum_i w_i(a_i-\mu) = 0$. The PSD matrix $\sum_{i\in[k]} w_i(a_i-\mu)(a_i-\mu)^\top$ has rank $\leq k-1$.
- $(k-1)$-PCA on covariance matrix $\cup\{\mu\}$ yields span(A).
- Lowest eigenvalue of covariance matrix yields σ^2.

Example 1: PCA on Gaussian Mixtures

$$\mathbb{E}[(x-\mu)(x-\mu)^\top] = \sum_{i\in[k]} w_i(a_i-\mu)(a_i-\mu)^\top + \sigma^2 I.$$

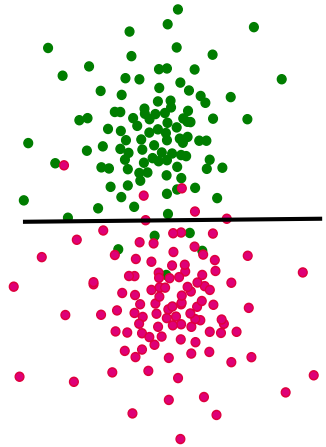
- The vectors $\{a_i - \mu\}$ are linearly dependent: $\sum_i w_i(a_i-\mu) = 0$. The PSD matrix $\sum_{i\in[k]} w_i(a_i-\mu)(a_i-\mu)^\top$ has rank $\leq k-1$.
- $(k-1)$-PCA on covariance matrix $\cup\{\mu\}$ yields span(A).
- Lowest eigenvalue of covariance matrix yields σ^2.

How to Learn A?

Learning through Spectral Clustering

Learning A through Spectral Clustering

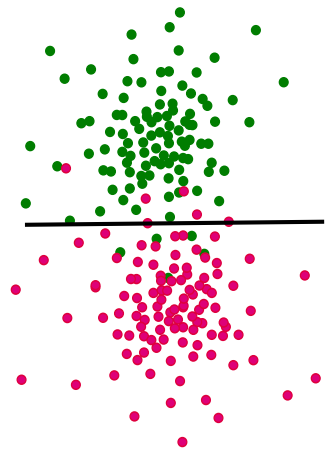
- Project samples x on to span(A).
- Distance-based clustering (e.g. k-means).
- A series of works, e.g. Vempala & Wang.



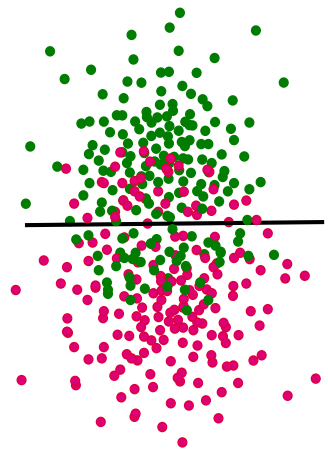
Learning through Spectral Clustering

Learning A through Spectral Clustering

- Project samples x on to span(A).
- Distance-based clustering (e.g. k-means).
- A series of works, e.g. Vempala & Wang.



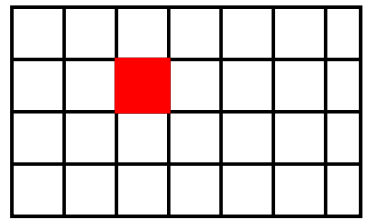
Failure to cluster under large variance.



Learning Gaussian Mixtures Without Separation Constraints?

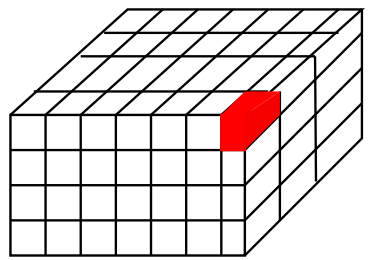
From Matrices to Tensors

Matrix: Pairwise Moments



- $\mathbb{E}[x \otimes x] \in \mathbb{R}^{d \times d}$ is a second order tensor.
- $\mathbb{E}[x \otimes x]_{i_1,i_2} = \mathbb{E}[x_{i_1} x_{i_2}]$.
- For matrices: $\mathbb{E}[x \otimes x] = \mathbb{E}[xx^\top]$.
- $M = uu^\top$ is rank-1 and $M_{i,j} = u_i u_j$.

Tensor: Higher order Moments



- $\mathbb{E}[x \otimes x \otimes x] \in \mathbb{R}^{d \times d \times d}$ is a third order tensor.
- $\mathbb{E}[x \otimes x \otimes x]_{i_1,i_2,i_3} = \mathbb{E}[x_{i_1} x_{i_2} x_{i_3}]$.
- $T = u \otimes u \otimes u$ is rank-1 and $T_{i,j,k} = u_i u_j u_k$.

Third order moment for Gaussian mixtures

$$\mathbb{E}[x \otimes x \otimes x] = \sum_i w_i a_i \otimes a_i \otimes a_i + \sigma^2 \sum_i (\mu \otimes e_i \otimes e_i + \ldots)$$

- σ^2 is obtained from $\sigma_{\min}(\mathsf{Cov}(x))$.
- $\mathbb{E}[x \otimes x] = \sum_i w_i a_i \otimes a_i + \sigma^2 I$.

Third order moment for Gaussian mixtures

$$\mathbb{E}[x \otimes x \otimes x] = \sum_i w_i a_i \otimes a_i \otimes a_i + \sigma^2 \sum_i (\mu \otimes e_i \otimes e_i + \ldots)$$

- σ^2 is obtained from $\sigma_{\min}(\mathrm{Cov}(x))$.
- $\mathbb{E}[x \otimes x] = \sum_i w_i a_i \otimes a_i + \sigma^2 I$.

Can obtain

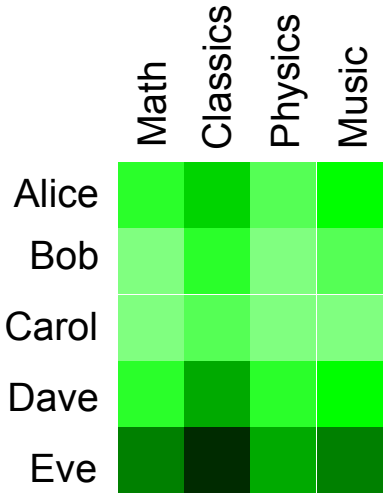
$$M_3 = \sum_i w_i a_i \otimes a_i \otimes a_i$$

$$M_2 = \sum_i w_i a_i \otimes a_i.$$

Obtain parameters A and w from M_3.

Visualization

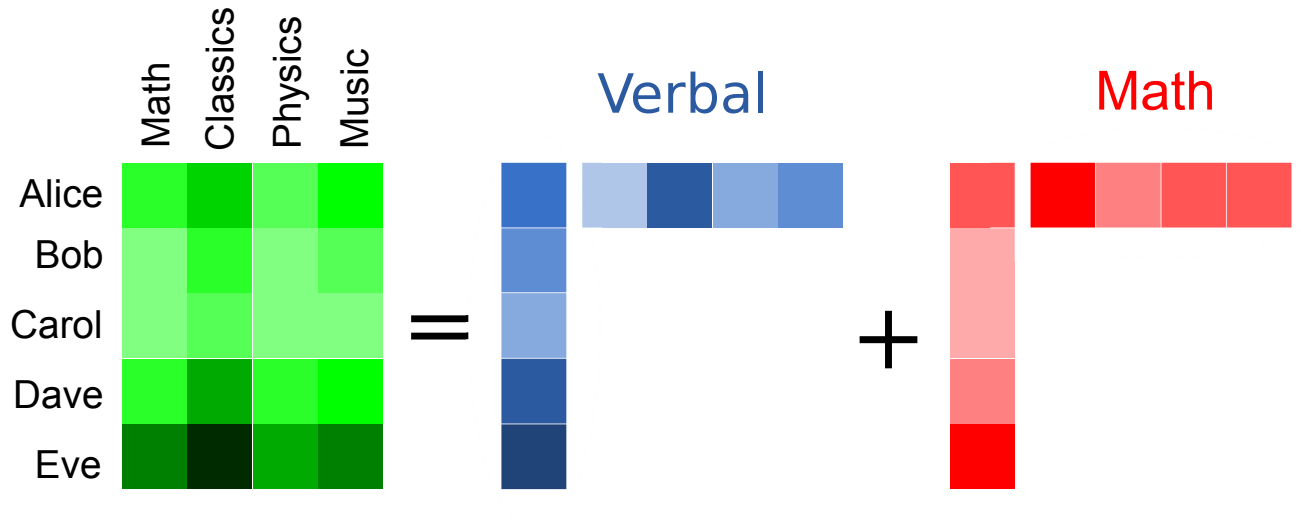
Example 2: Discovering Latent Factors



- List of scores for students in different tests
- Learn hidden factors for Verbal and Mathematical Intelligence [C. Spearman 1904]

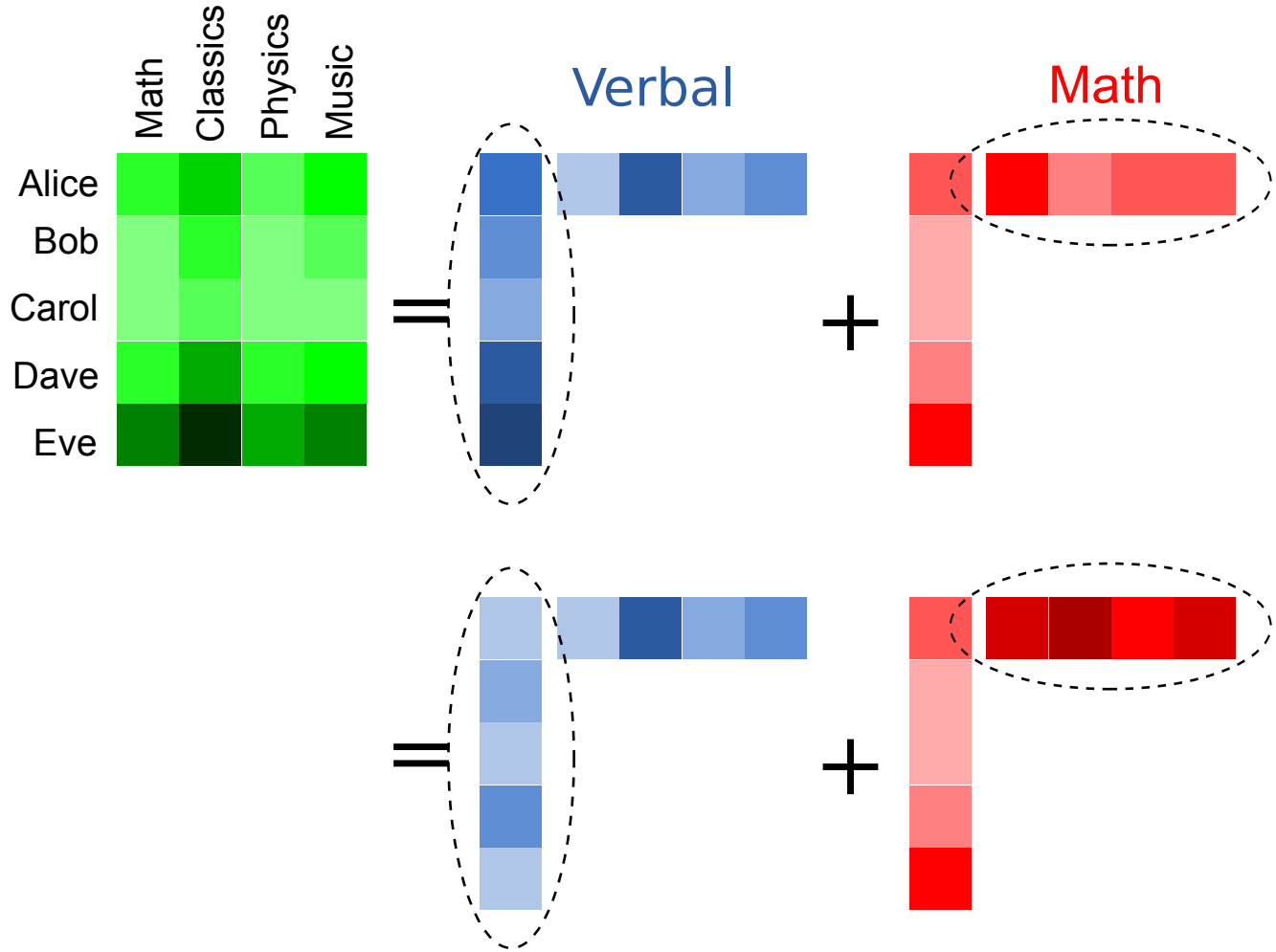
$$\text{Score (student,test)} = \text{student}_{\text{verbal-intlg}} \times \text{test}_{\text{verbal}} + \text{student}_{\text{math-intlg}} \times \text{test}_{\text{math}}$$

Matrix Decomposition: Discovering Latent Factors



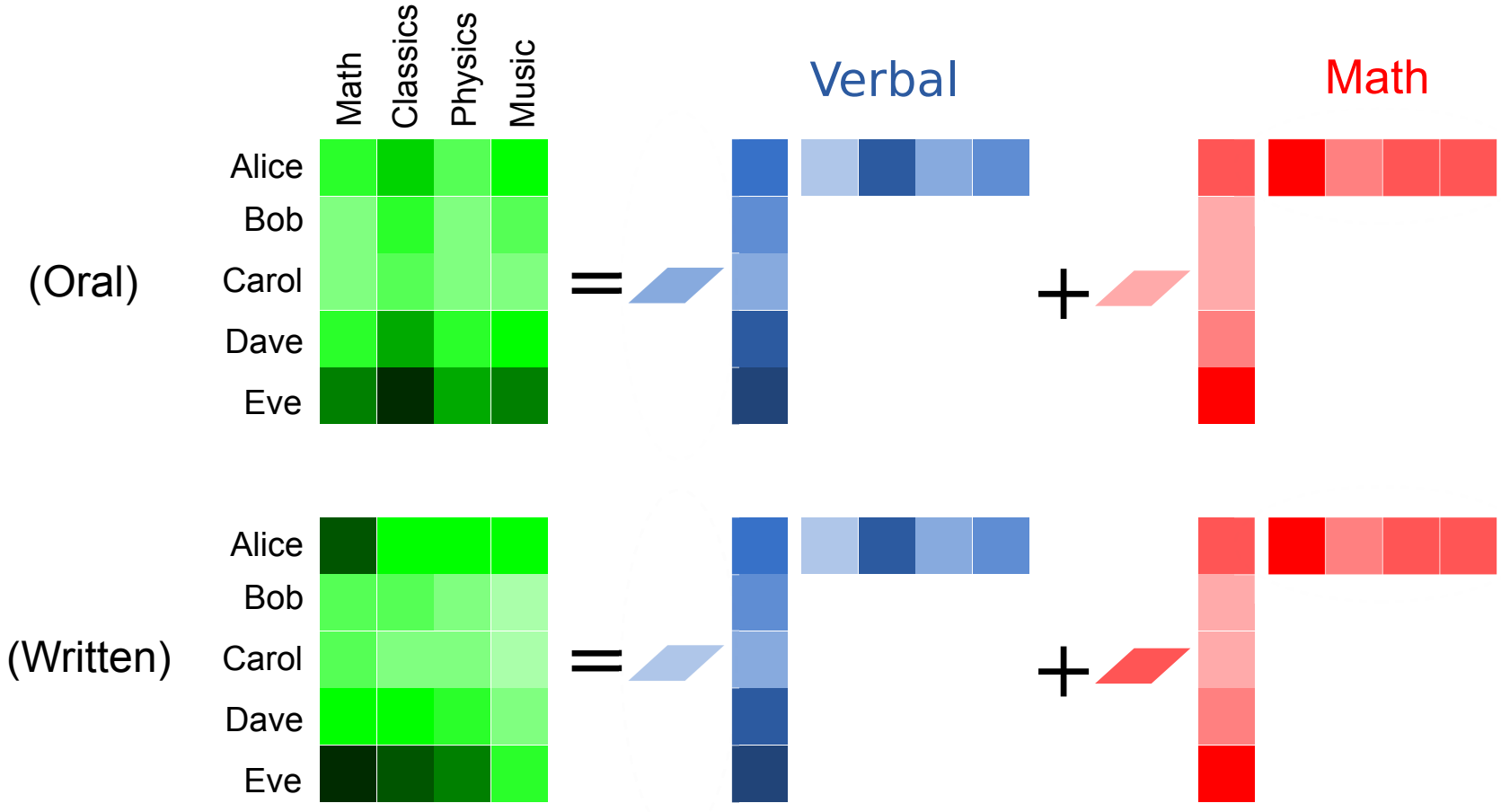
- Identifying hidden factors influencing the observations
- Characterized as matrix decomposition

Matrix Decomposition: Discovering Latent Factors



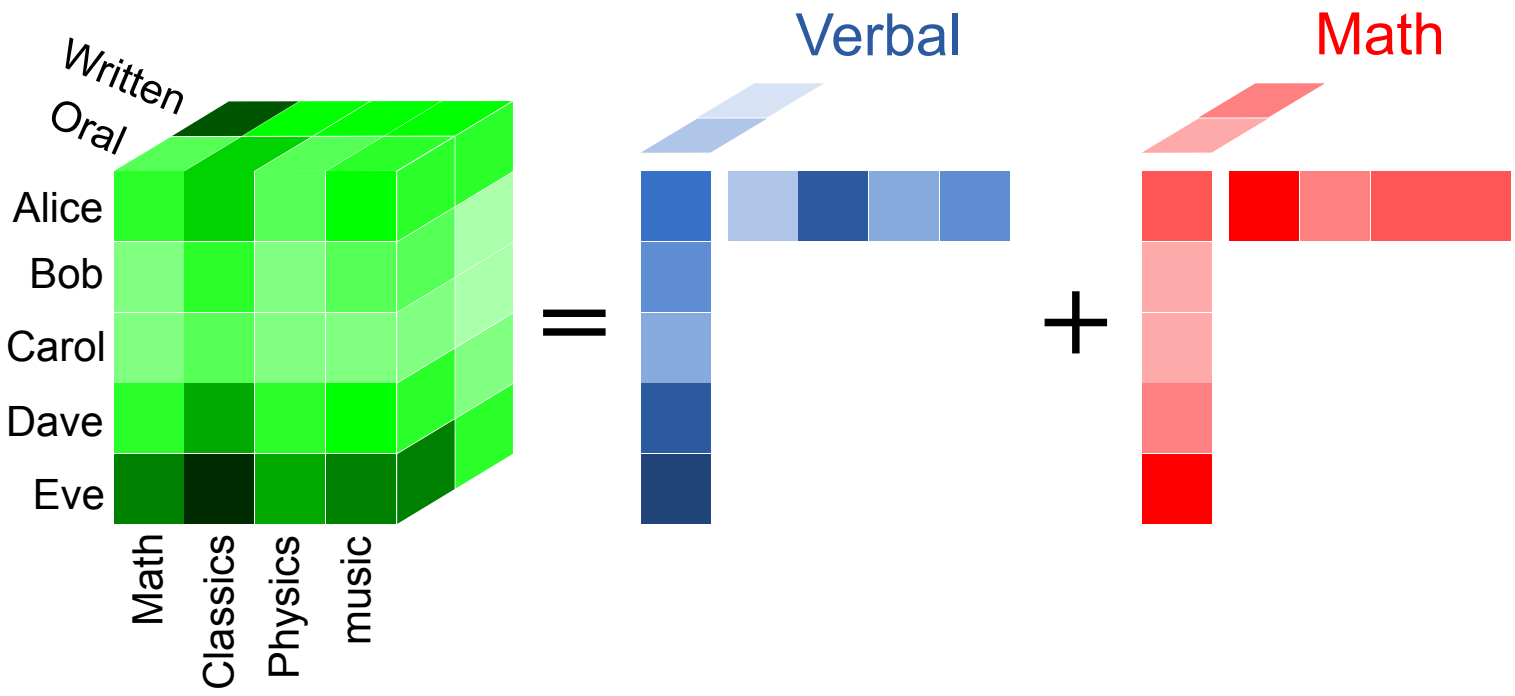
- Decomposition is not necessarily unique.
- Decomposition cannot be overcomplete.

Tensor: Shared Matrix Decomposition



- Shared decomposition with different scaling factors
- Combine matrix slices as a tensor

Tensor Decomposition



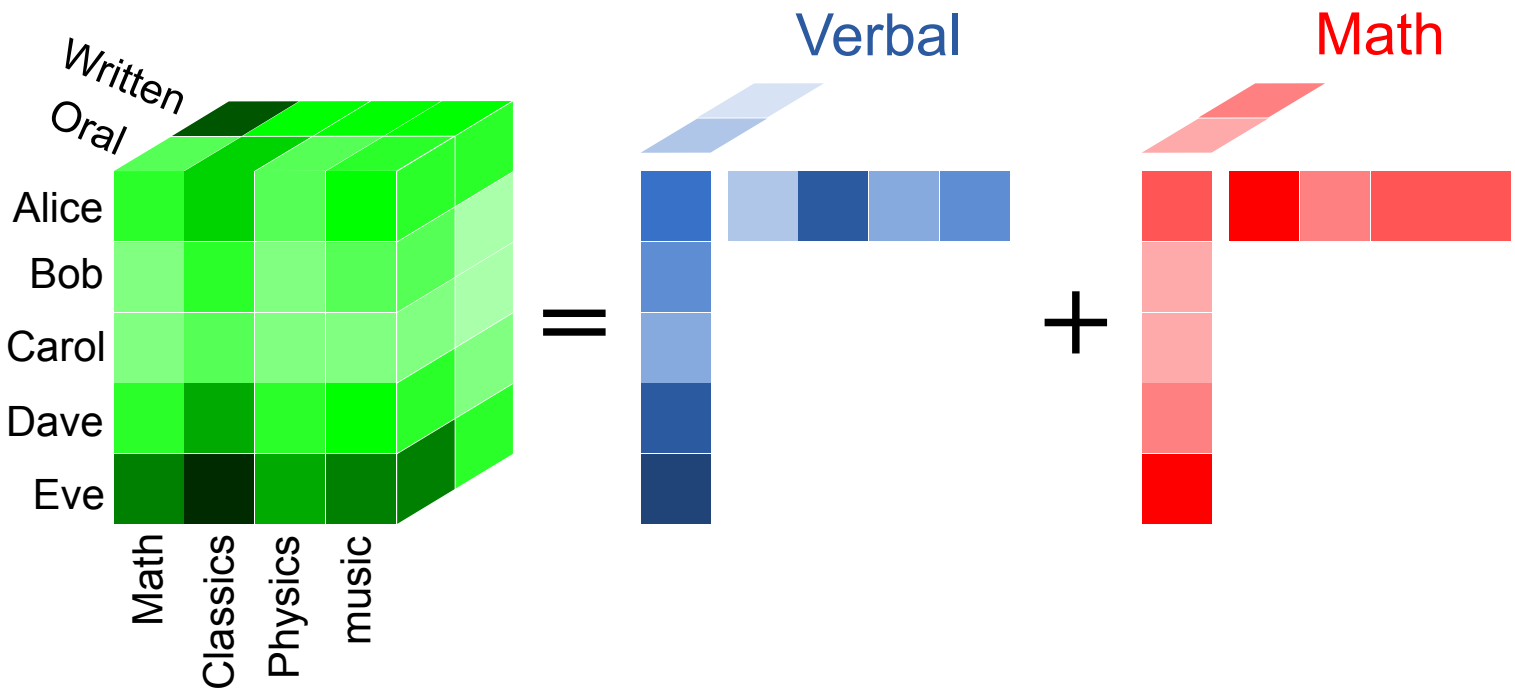
- Outer product notation:

$$T = u \otimes v \otimes w + \tilde{u} \otimes \tilde{v} \otimes \tilde{w}$$

$$\Updownarrow$$

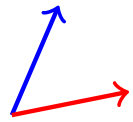
$$T_{i_1,i_2,i_3} = u_{i_1} \cdot v_{i_2} \cdot w_{i_3} + \tilde{u}_{i_1} \cdot \tilde{v}_{i_2} \cdot \tilde{w}_{i_3}$$

Tensor Decomposition

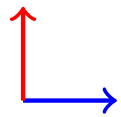


Uniqueness of Tensor Decomposition [J. Kruskal 1977]

- Above tensor decomposition: unique when rank one pairs are linearly independent

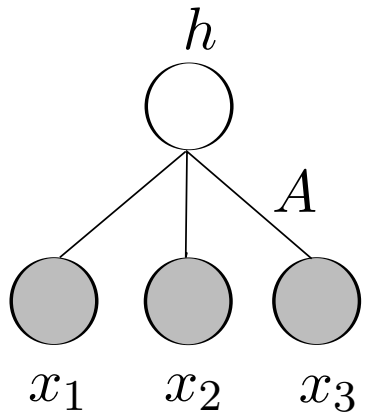


- Matrix case: when rank one pairs are orthogonal



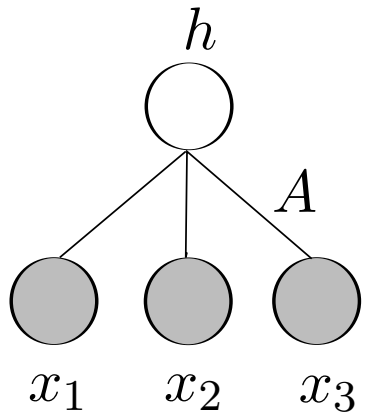
Example 3: Learning Naive Bayes Model

- Naive Bayes Model: $x_1 \perp\!\!\!\perp x_2 \perp\!\!\!\perp x_3 | h$,
- $h = e_i$ for state i, $\mathbb{E}[x_i|h] = Ah$, $\mathbb{E}[h] = \lambda$.
- Learn matrix $A = [a_1|a_2 \dots a_k]$.



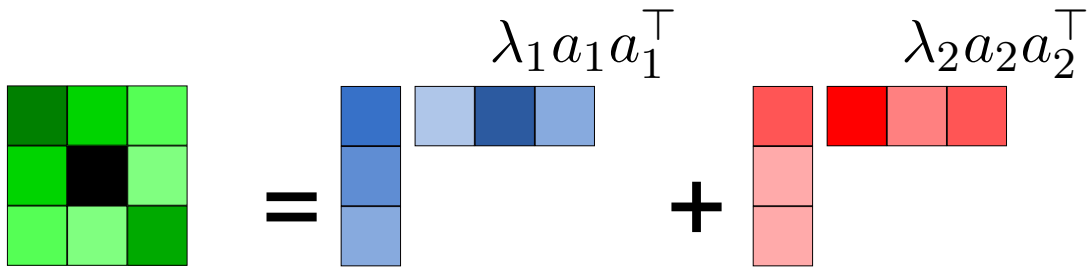
Example 3: Learning Naive Bayes Model

- Naive Bayes Model: $x_1 \perp\!\!\!\perp x_2 \perp\!\!\!\perp x_3 | h$,
- $h = e_i$ for state i, $\mathbb{E}[x_i|h] = Ah$, $\mathbb{E}[h] = \lambda$.
- Learn matrix $A = [a_1|a_2 \dots a_k]$.



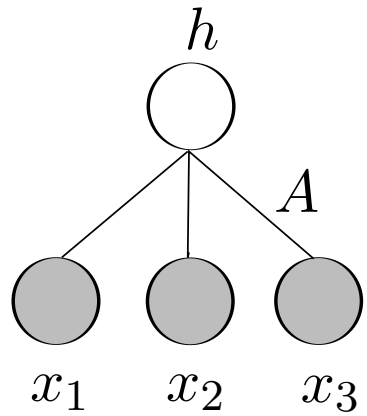
Learning using matrix methods

- Pairwise moment $M_2 := \mathbb{E}[x_1 x_2^\top] = \mathbb{E}[\mathbb{E}[x_1 x_2^\top | h]] = \sum_i \lambda_i a_i a_i^\top$
- Given matrix M_2, recover A?



Example 3: Learning Naive Bayes Model

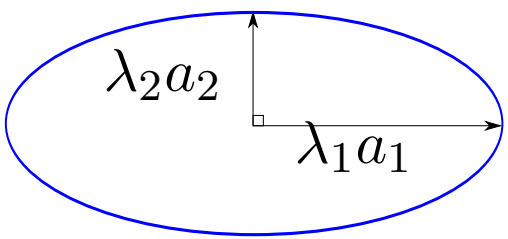
- Naive Bayes Model: $x_1 \perp\!\!\!\perp x_2 \perp\!\!\!\perp x_3 | h$,
- $h = e_i$ for state i, $\mathbb{E}[x_i|h] = Ah$, $\mathbb{E}[h] = \lambda$.
- Learn matrix $A = [a_1|a_2 \dots a_k]$.

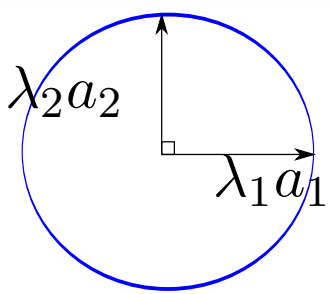


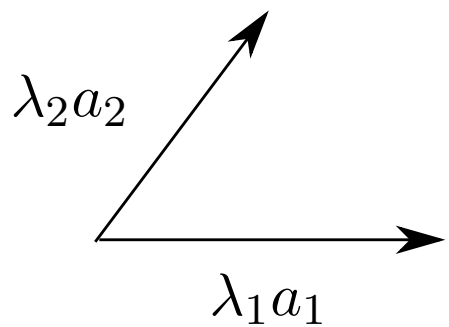
Learning using matrix methods

- Pairwise moment $M_2 := \mathbb{E}[x_1 x_2^\top] = \mathbb{E}[\mathbb{E}[x_1 x_2^\top | h]] = \sum_i \lambda_i a_i a_i^\top$
- Given matrix M_2, recover A?

Matrix decomposition unique only for orthogonal factors.

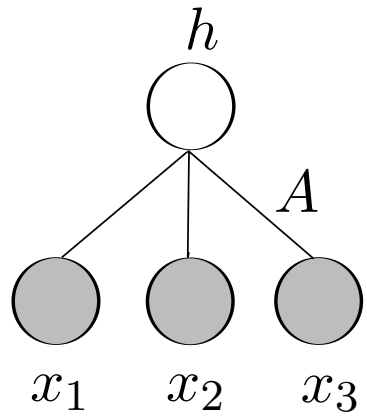






Example 3: Learning Naive Bayes Model

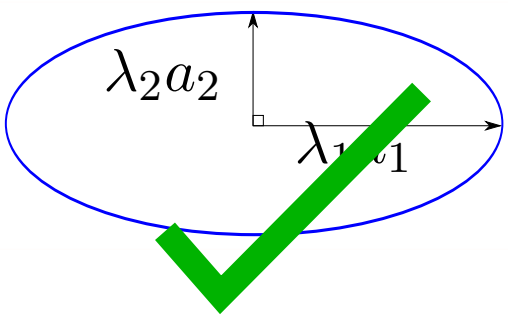
- Naive Bayes Model: $x_1 \perp\!\!\!\perp x_2 \perp\!\!\!\perp x_3 | h$,
- $h = e_i$ for state i, $\mathbb{E}[x_i|h] = Ah$, $\mathbb{E}[h] = \lambda$.
- Learn matrix $A = [a_1|a_2 \dots a_k]$.

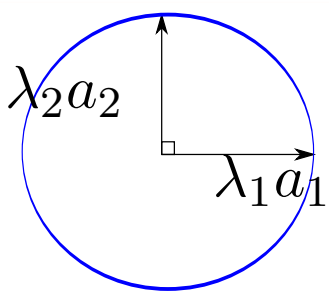


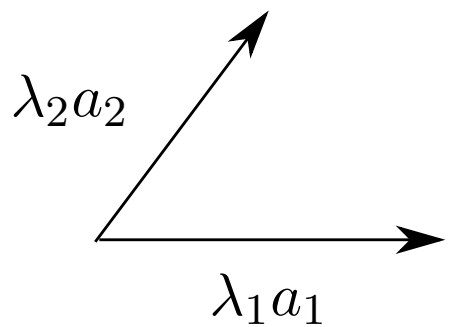
Learning using matrix methods

- Pairwise moment $M_2 := \mathbb{E}[x_1 x_2^\top] = \mathbb{E}[\mathbb{E}[x_1 x_2^\top | h]] = \sum_i \lambda_i a_i a_i^\top$
- Given matrix M_2, recover A?

Matrix decomposition unique only for orthogonal factors.

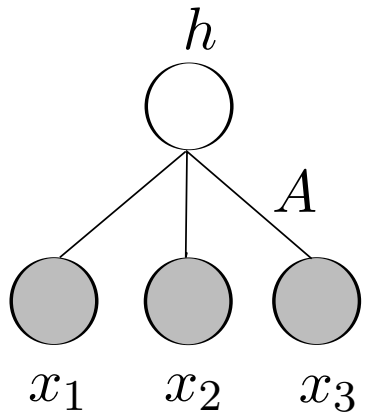






Example 3: Learning Naive Bayes Model

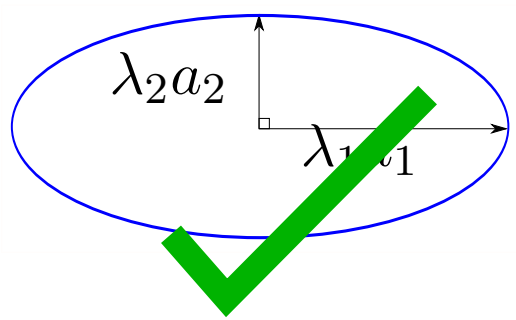
- Naive Bayes Model: $x_1 \perp\!\!\!\perp x_2 \perp\!\!\!\perp x_3 | h$,
- $h = e_i$ for state i, $\mathbb{E}[x_i|h] = Ah$, $\mathbb{E}[h] = \lambda$.
- Learn matrix $A = [a_1|a_2 \dots a_k]$.

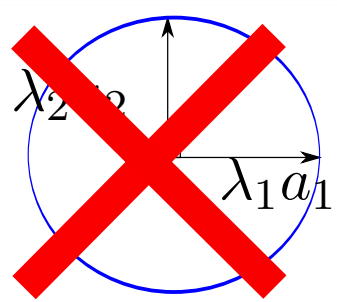


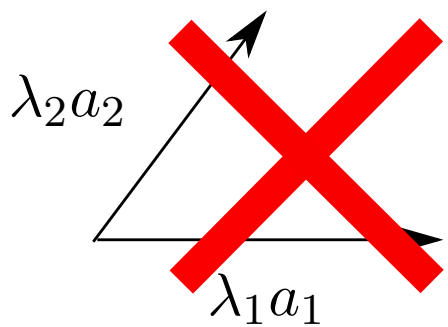
Learning using matrix methods

- Pairwise moment $M_2 := \mathbb{E}[x_1 x_2^\top] = \mathbb{E}[\mathbb{E}[x_1 x_2^\top | h]] = \sum_i \lambda_i a_i a_i^\top$
- Given matrix M_2, recover A?

Matrix decomposition unique only for orthogonal factors.



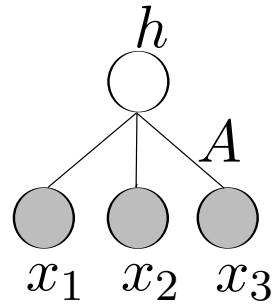




From Matrix to Tensor

Moments of Naive Bayes Model

- $M_2 = \mathbb{E}[x_1 \otimes x_2] = \sum_i \lambda_i a_i \otimes a_i$.
- $M_3 = \mathbb{E}[x_1 \otimes x_2 \otimes x_3] = \sum_i \lambda_i a_i \otimes a_i \otimes a_i$.



Recover $A = [a_1|a_2 \dots a_k]$ from decomposing M_3

Outline

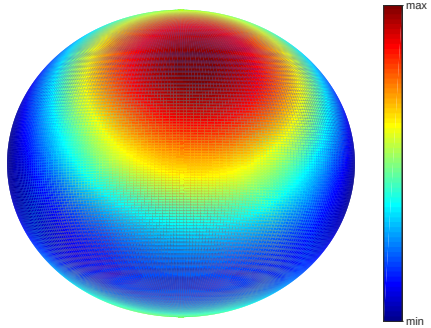
1. Introduction
2. Why Tensors?
3. Algorithms for Tensor Decomposition
4. Learning Probabilistic Models
5. Learning Representations
6. Conclusion

Matrix Eigen-analysis

Find decomposition of matrix $M = \sum_i \lambda_i v_i v_i^\top$.

- Optimization: find top eigenvector.

$$\max_v \langle v, Mv \rangle \text{ s.t. } \|v\| = 1, v \in \mathbb{R}^d.$$

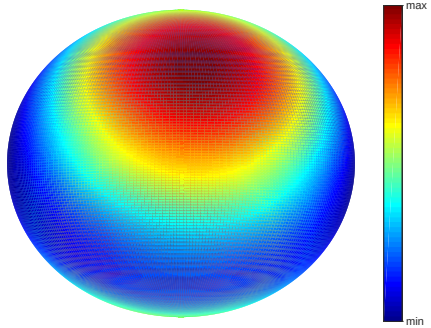


Matrix Eigen-analysis

Find decomposition of matrix $M = \sum_i \lambda_i v_i v_i^\top$.

- Optimization: find top eigenvector.

$$\max_v \langle v, Mv \rangle \text{ s.t. } \|v\| = 1, v \in \mathbb{R}^d.$$



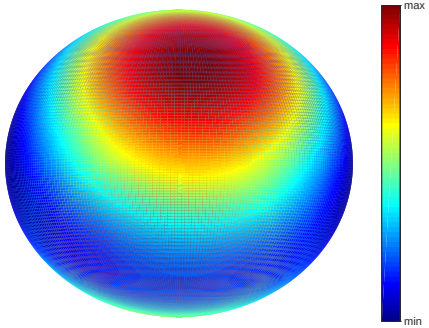
- Lagrangian: $\langle v, Mv \rangle - \lambda(\|v\|^2 - 1).$

Matrix Eigen-analysis

Find decomposition of matrix $M = \sum_i \lambda_i v_i v_i^\top$.

- Optimization: find top eigenvector.

$$\max_v \langle v, Mv \rangle \text{ s.t. } \|v\| = 1, v \in \mathbb{R}^d.$$



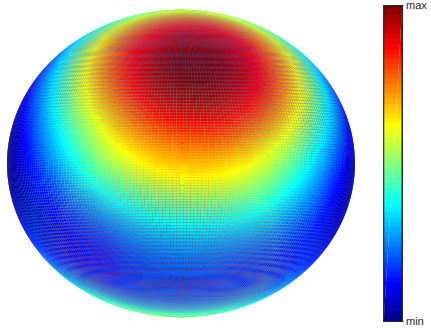
- Lagrangian: $\langle v, Mv \rangle - \lambda(\|v\|^2 - 1).$
- Critical points: $Mv = \lambda v$ (all eigenvectors).

Matrix Eigen-analysis

Find decomposition of matrix $M = \sum_i \lambda_i v_i v_i^\top$.

- Optimization: find top eigenvector.

$$\max_v \langle v, Mv \rangle \text{ s.t. } \|v\| = 1, v \in \mathbb{R}^d.$$



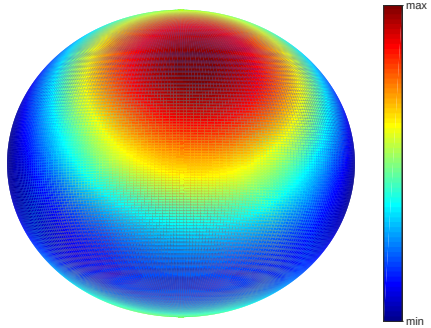
- Lagrangian: $\langle v, Mv \rangle - \lambda(\|v\|^2 - 1).$
- Critical points: $Mv = \lambda v$ (all eigenvectors).
- All saddle points (at most d) are non-degenerate.

Matrix Eigen-analysis

Find decomposition of matrix $M = \sum_i \lambda_i v_i v_i^\top$.

- Optimization: find top eigenvector.

$$\max_v \langle v, Mv \rangle \text{ s.t. } \|v\| = 1, v \in \mathbb{R}^d.$$



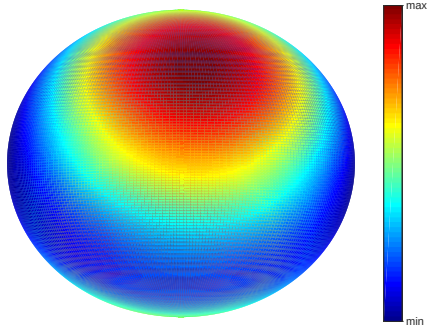
- Lagrangian: $\langle v, Mv \rangle - \lambda(\|v\|^2 - 1).$
- Critical points: $Mv = \lambda v$ (all eigenvectors).
- All saddle points (at most d) are non-degenerate.
- No. of local optima: 1

Matrix Eigen-analysis

Find decomposition of matrix $M = \sum_i \lambda_i v_i v_i^\top$.

- Optimization: find top eigenvector.

$$\max_v \langle v, Mv \rangle \text{ s.t. } \|v\| = 1, v \in \mathbb{R}^d.$$



- Lagrangian: $\langle v, Mv \rangle - \lambda(\|v\|^2 - 1).$
- Critical points: $Mv = \lambda v$ (all eigenvectors).
- All saddle points (at most d) are non-degenerate.
- No. of local optima: 1

Algorithmic implication

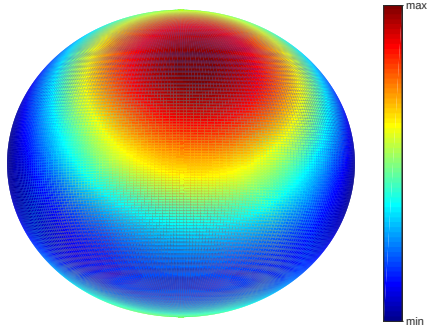
- Gradient ascent (power method) converges to global optimum!
- Saddle points avoided by random initialization!

Matrix Eigen-analysis

Find decomposition of matrix $M = \sum_i \lambda_i v_i v_i^\top$.

- Optimization: find top eigenvector.

$$\max_v \langle v, Mv \rangle \text{ s.t. } \|v\| = 1, v \in \mathbb{R}^d.$$



- Lagrangian: $\langle v, Mv \rangle - \lambda(\|v\|^2 - 1).$
- Critical points: $Mv = \lambda v$ (all eigenvectors).
- All saddle points (at most d) are non-degenerate.
- No. of local optima: 1

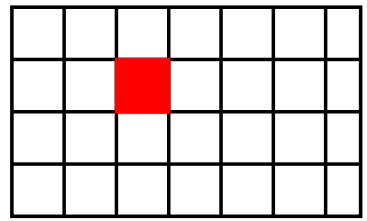
Local optimum $\equiv$ Global optimum!

Algorithmic implication

- Gradient ascent (power method) converges to global optimum!
- Saddle points avoided by random initialization!

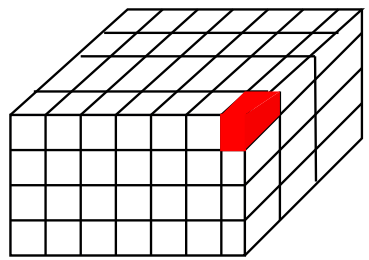
From Matrices to Tensors

Matrix: Pairwise Moments



- $\mathbb{E}[x \otimes x] \in \mathbb{R}^{d \times d}$ is a second order tensor.
- $\mathbb{E}[x \otimes x]_{i_1,i_2} = \mathbb{E}[x_{i_1} x_{i_2}]$.
- For matrices: $\mathbb{E}[x \otimes x] = \mathbb{E}[xx^\top]$.
- $M = uu^\top$ is rank-1 and $M_{i,j} = u_i u_j$.

Tensor: Higher order Moments



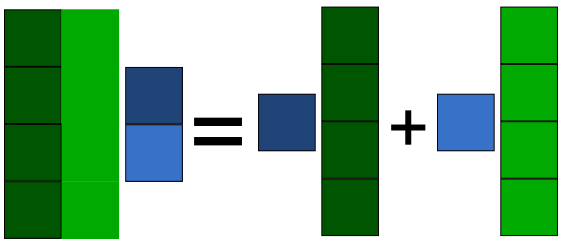
- $\mathbb{E}[x \otimes x \otimes x] \in \mathbb{R}^{d \times d \times d}$ is a third order tensor.
- $\mathbb{E}[x \otimes x \otimes x]_{i_1,i_2,i_3} = \mathbb{E}[x_{i_1} x_{i_2} x_{i_3}]$.
- $T = u \otimes u \otimes u$ is rank-1 and $T_{i,j,k} = u_i u_j u_k$.

Notion of Tensor Contraction

Extends the notion of matrix product

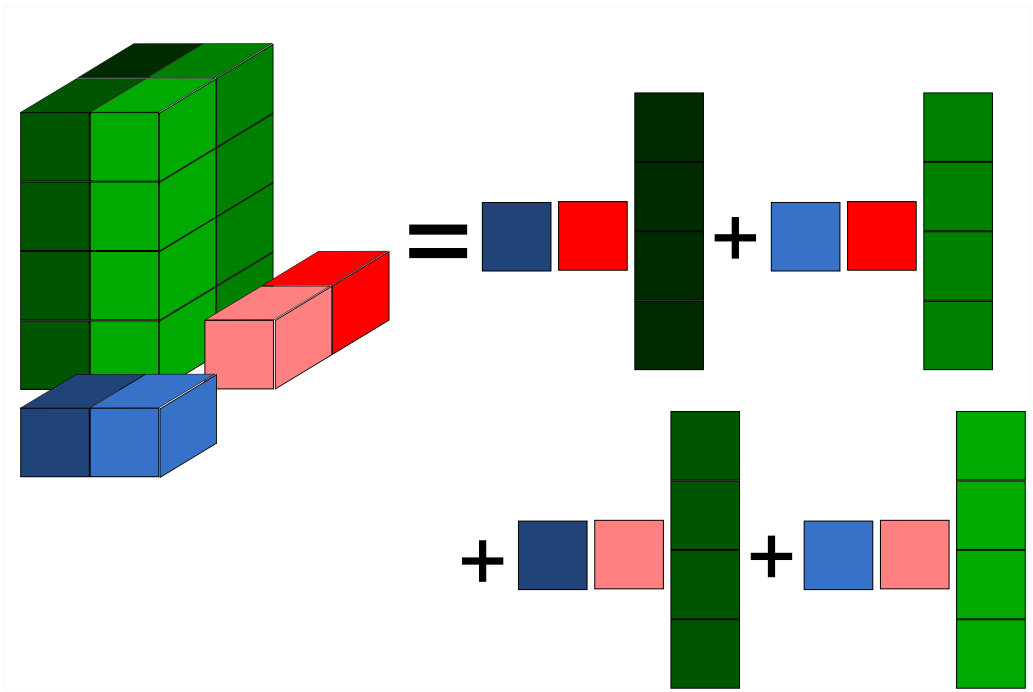
Matrix product

$$Mv = \sum_j v_j M_j$$



Tensor Contraction

$$T(u, v, \cdot) = \sum_{i,j} u_i v_j T_{i,j,:}$$



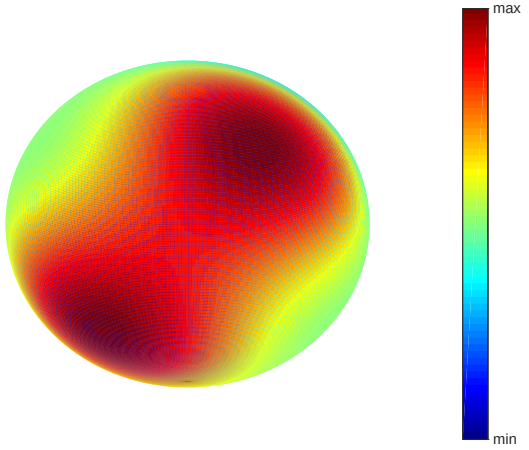
Problem of Tensor Decomposition

- Computationally hard for general tensors.
- Orthogonal tensors $\boxed{T = \sum_{i \in [k]} \lambda_i u_i \otimes u_i \otimes u_i}$: $u_i \perp u_j$ for $i \neq j$.

Decomposition through Tensor Norm Max.

$$\boxed{\max_{v} T(v, v, v),\ \|v\| = 1, v \in \mathbb{R}^d.}$$

- Lagrangian: $\boxed{T(v, v, v) - 1.5\lambda(\|v\|^2 - 1).}$

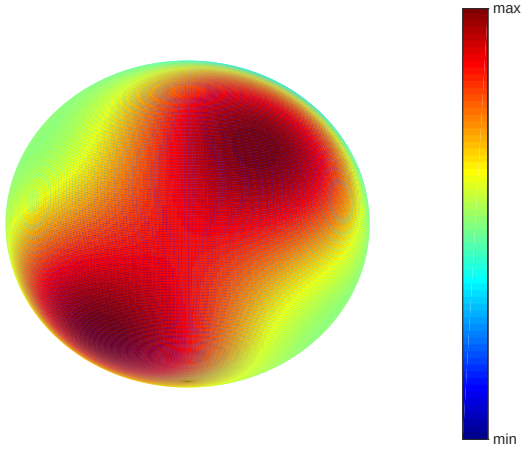


Problem of Tensor Decomposition

- Computationally hard for general tensors.
- Orthogonal tensors $\boxed{T = \sum_{i \in [k]} \lambda_i u_i \otimes u_i \otimes u_i}$: $u_i \perp u_j$ for $i \neq j$.

Decomposition through Tensor Norm Max.

$$\boxed{\max_{v} T(v, v, v),\ \|v\| = 1, v \in \mathbb{R}^d.}$$



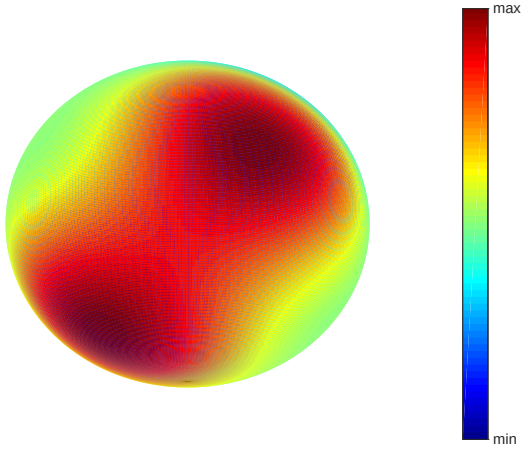
- Lagrangian: $\boxed{T(v, v, v) - 1.5\lambda(\|v\|^2 - 1).}$
- Critical points: $T(v, v, \cdot) = \lambda v$.

Problem of Tensor Decomposition

- Computationally hard for general tensors.
- Orthogonal tensors $\boxed{T = \sum_{i \in [k]} \lambda_i u_i \otimes u_i \otimes u_i}$: $u_i \perp u_j$ for $i \neq j$.

Decomposition through Tensor Norm Max.

$$\boxed{\max_{v} T(v, v, v),\ \|v\| = 1, v \in \mathbb{R}^d.}$$



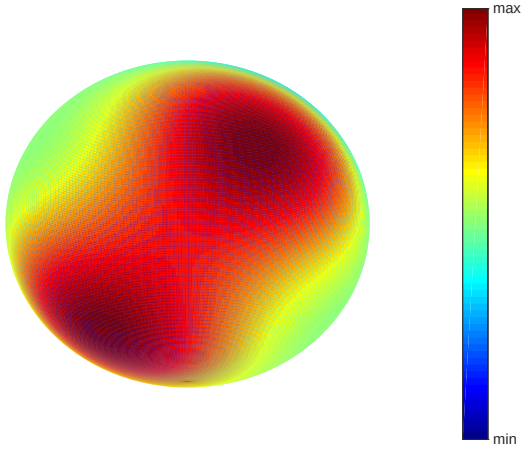
- Lagrangian: $\boxed{T(v, v, v) - 1.5\lambda(\|v\|^2 - 1).}$
- Critical points: $T(v, v, \cdot) = \lambda v$.
- No. of eigenvectors: $\exp(d)!$

Problem of Tensor Decomposition

- Computationally hard for general tensors.
- Orthogonal tensors $\boxed{T = \sum_{i \in [k]} \lambda_i u_i \otimes u_i \otimes u_i}$: $u_i \perp u_j$ for $i \neq j$.

Decomposition through Tensor Norm Max.

$$\boxed{\max_{v} T(v, v, v),\ \|v\| = 1, v \in \mathbb{R}^d.}$$



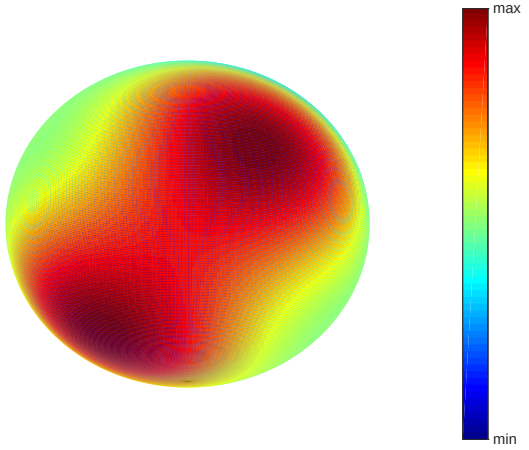
- Lagrangian: $\boxed{T(v, v, v) - 1.5\lambda(\|v\|^2 - 1).}$
- Critical points: $T(v, v, \cdot) = \lambda v$.
- No. of eigenvectors: $\exp(d)!$
- All saddle points are non-degenerate.

Problem of Tensor Decomposition

- Computationally hard for general tensors.
- Orthogonal tensors $\boxed{T = \sum_{i \in [k]} \lambda_i u_i \otimes u_i \otimes u_i}$: $u_i \perp u_j$ for $i \neq j$.

Decomposition through Tensor Norm Max.

$$\boxed{\max_{v} T(v, v, v),\ \|v\| = 1, v \in \mathbb{R}^d.}$$



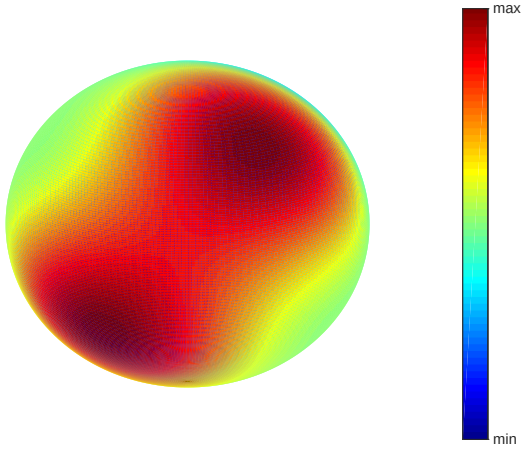
- Lagrangian: $\boxed{T(v, v, v) - 1.5\lambda(\|v\|^2 - 1).}$
- Critical points: $T(v, v, \cdot) = \lambda v$.
- No. of eigenvectors: $\exp(d)!$
- All saddle points are non-degenerate.
- Local optima: $\{u_i\}$ for $i = 1, \ldots, k$.

Problem of Tensor Decomposition

- Computationally hard for general tensors.
- Orthogonal tensors $\boxed{T = \sum_{i \in [k]} \lambda_i u_i \otimes u_i \otimes u_i}$: $u_i \perp u_j$ for $i \neq j$.

Decomposition through Tensor Norm Max.

$$\boxed{\max_{v} T(v, v, v),\ \|v\| = 1, v \in \mathbb{R}^d.}$$

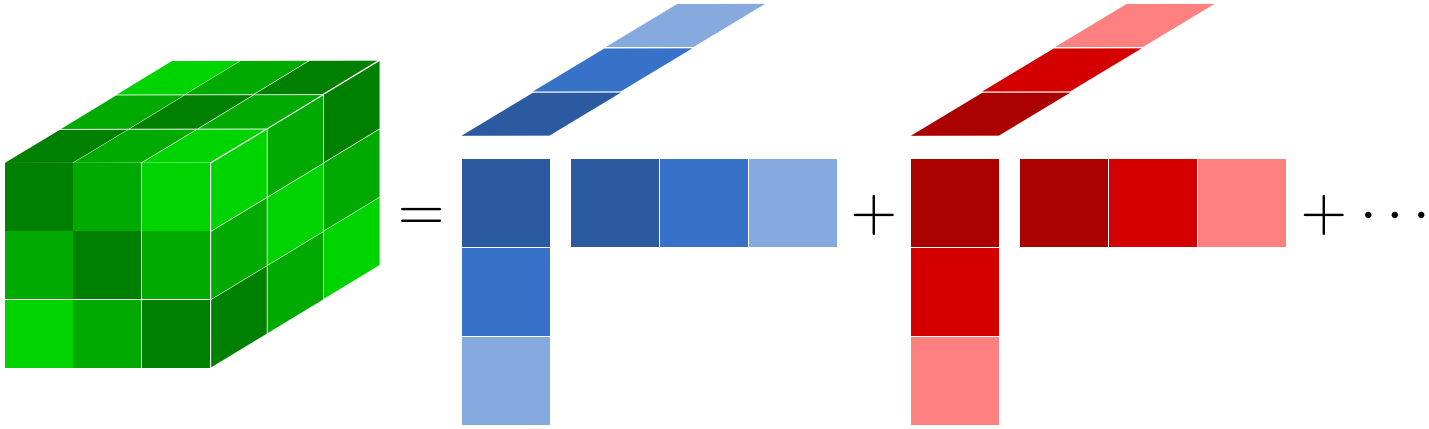


- Lagrangian: $\boxed{T(v, v, v) - 1.5\lambda(\|v\|^2 - 1).}$
- Critical points: $T(v, v, \cdot) = \lambda v$.
- No. of eigenvectors: $\exp(d)!$
- All saddle points are non-degenerate.
- Local optima: $\{u_i\}$ for $i = 1, \ldots, k$.

Multiple local optima, but they correspond to components!

Exponentially many saddle points!

Symmetric Tensor Decomposition



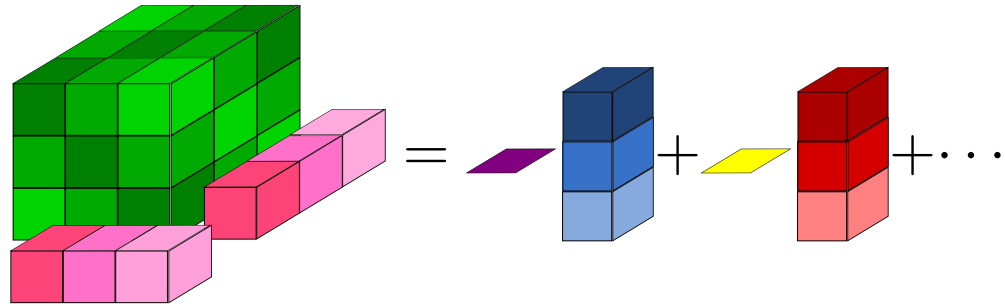
$$T = {v_1}^{\otimes 3} + {v_2}^{\otimes 3} + \cdots,$$

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Symmetric Tensor Decomposition

Tensor Power Method

$$v \mapsto \frac{T(v, v, \cdot)}{\|T(v, v, \cdot)\|}.$$



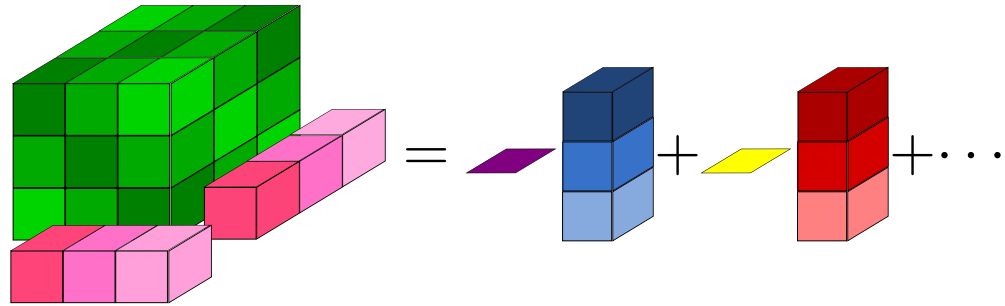
$$T(v, v, \cdot) = \langle v, v_1 \rangle^2 v_1 + \langle v, v_2 \rangle^2 v_2$$

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Symmetric Tensor Decomposition

Tensor Power Method

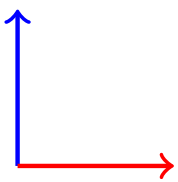
$$v \mapsto \frac{T(v, v, \cdot)}{\|T(v, v, \cdot)\|}.$$



$$T(v, v, \cdot) = \langle v, v_1 \rangle^2 v_1 + \langle v, v_2 \rangle^2 v_2$$

Orthogonal Tensors

- $\vec{v}_1 \perp \vec{v}_2$.
- $T(v_1, v_1, \cdot) = \lambda_1 v_1$.

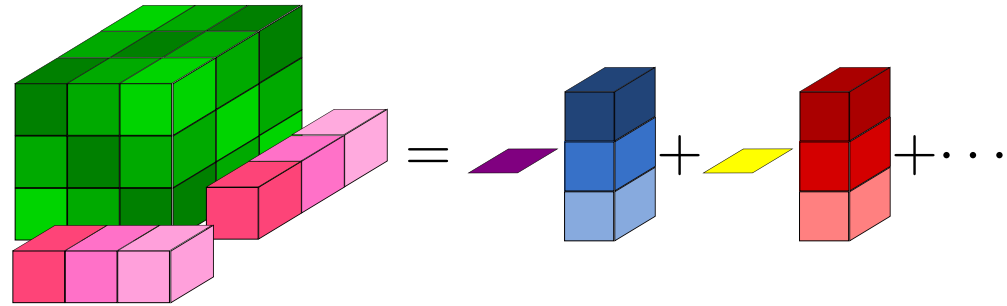


A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Symmetric Tensor Decomposition

Tensor Power Method

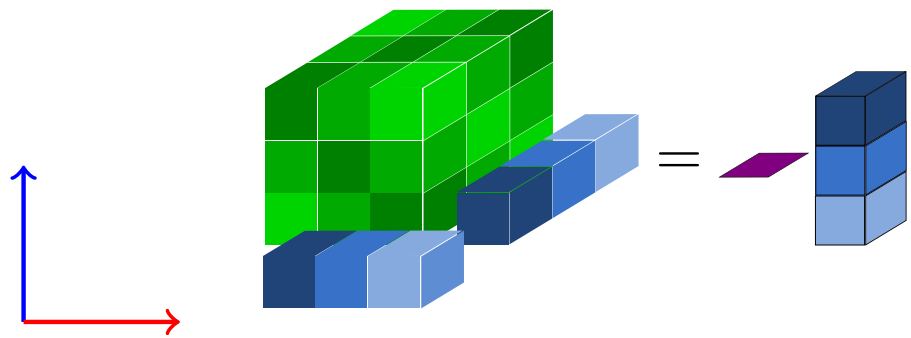
$$v \mapsto \frac{T(v, v, \cdot)}{\|T(v, v, \cdot)\|}.$$



$$T(v, v, \cdot) = \langle v, v_1 \rangle^2 v_1 + \langle v, v_2 \rangle^2 v_2$$

Orthogonal Tensors

- $\vec{v}_1 \perp \vec{v}_2$.
- $T(v_1, v_1, \cdot) = \lambda_1 v_1$.

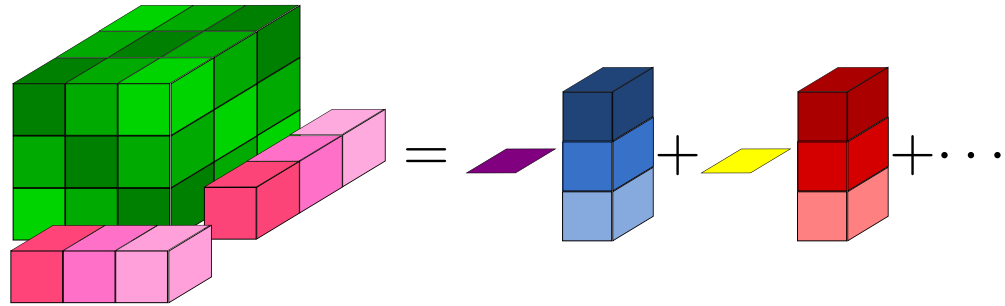


A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Symmetric Tensor Decomposition

Tensor Power Method

$$v \mapsto \frac{T(v, v, \cdot)}{\|T(v, v, \cdot)\|}.$$



$$T(v, v, \cdot) = \langle v, v_1 \rangle^2 v_1 + \langle v, v_2 \rangle^2 v_2$$

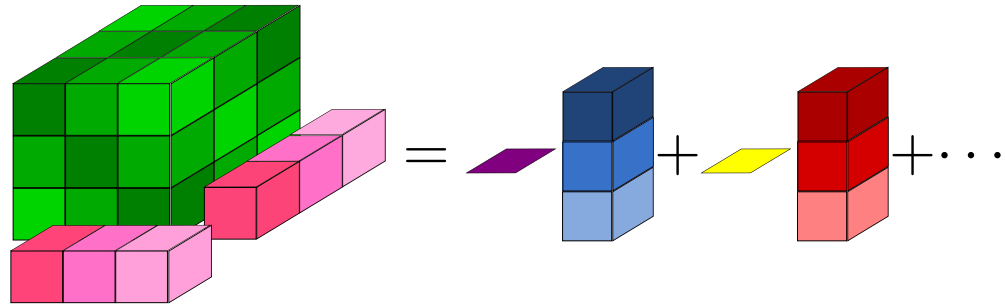
Exponential no. of stationary points for power method:
$T(v, v, \cdot) = \lambda v$

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Symmetric Tensor Decomposition

Tensor Power Method

$$v \mapsto \frac{T(v, v, \cdot)}{\|T(v, v, \cdot)\|}.$$

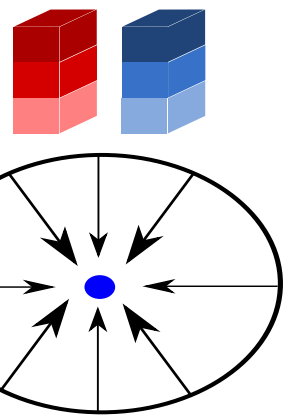


$$T(v, v, \cdot) = \langle v, v_1 \rangle^2 v_1 + \langle v, v_2 \rangle^2 v_2$$

Exponential no. of stationary points for power method:

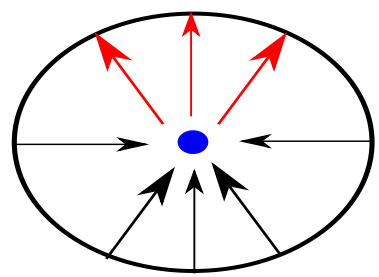
$T(v, v, \cdot) = \lambda v$

Stable



Unstable

Other statitionary points

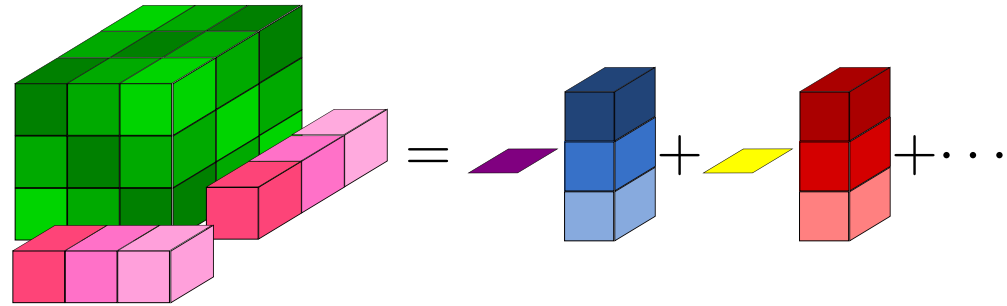


A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Symmetric Tensor Decomposition

Tensor Power Method

$$v \mapsto \frac{T(v, v, \cdot)}{\|T(v, v, \cdot)\|}.$$



$$T(v, v, \cdot) = \langle v, v_1 \rangle^2 v_1 + \langle v, v_2 \rangle^2 v_2$$

Exponential no. of stationary points for power method:
$T(v, v, \cdot) = \lambda v$

For power method on orthogonal tensor, no spurious stable points

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Implication: Guaranteed Tensor Decomposition

Given Orthogonal Tensor $T = \sum_{i \in [k]} \lambda_i u_i \otimes u_i \otimes u_i$.

Recover components one by one

- Run projected SGD on $\boxed{\max_{v:\|v\|=1} T(v,v,v).}$
- Guaranteed to recover a local optimum $\{u_i\}$ (upto scale).
- Find all components $\{u_i\}$ by deflation!

Implication: Guaranteed Tensor Decomposition

Given Orthogonal Tensor $T = \sum_{i \in [k]} \lambda_i u_i \otimes u_i \otimes u_i$.

Recover components one by one

- Run projected SGD on $\boxed{\max_{v:\|v\|=1} T(v,v,v).}$
- Guaranteed to recover a local optimum $\{u_i\}$ (upto scale).
- Find all components $\{u_i\}$ by deflation!

Alternative: simultaneous recovery of components (Ge et al '15)

- For fourth order tensor T $\boxed{\min_{\forall i, \|v_i\|=1} \sum_{i \neq j} T(v_i, v_i, v_j, v_j).}$
- All saddle points are non-degenerate.
- All local optima are global.

SGD recovers the orthogonal tensor components optimally

Perturbation Analysis for Tensor Decomposition

- Well understood for matrix decomposition vs. hard for polynomials.
- Subtle analysis for tensor decomposition.

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Perturbation Analysis for Tensor Decomposition

- Well understood for matrix decomposition vs. hard for polynomials.
- Subtle analysis for tensor decomposition.

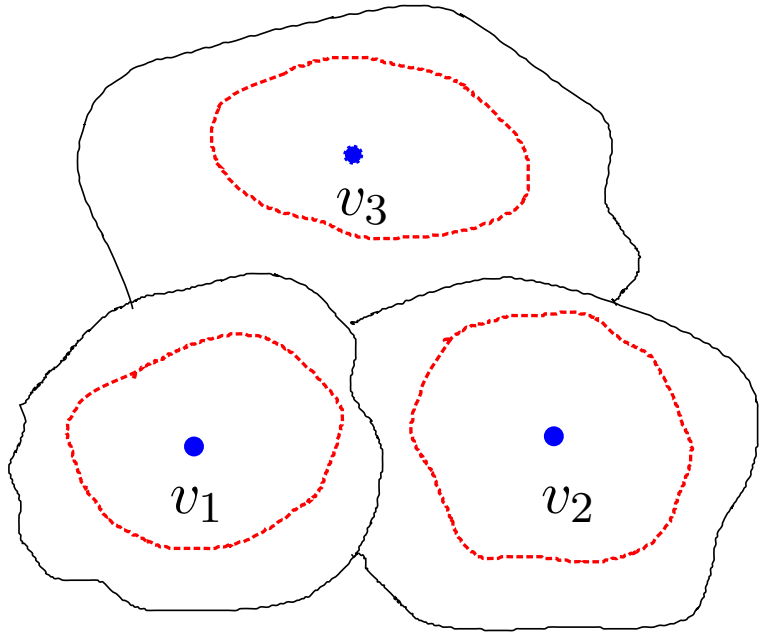
- $T \in \mathbb{R}^{d \times d \times d}$: Orthogonal tensor. E: noise tensor.

$$\hat{T} = T + E, \quad T = \sum_i \lambda_i v_i \otimes v_i \otimes v_i, \quad \|E\| := \max_{x:\|x\|=1} |E(x,x,x)|.$$

Theorem: When $\boxed{\|E\| < \frac{\lambda_{\min}}{\sqrt{d}}}$, power method recovers $\{v_i\}$ up to error $\|E\|$ with linear no. of restarts.

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

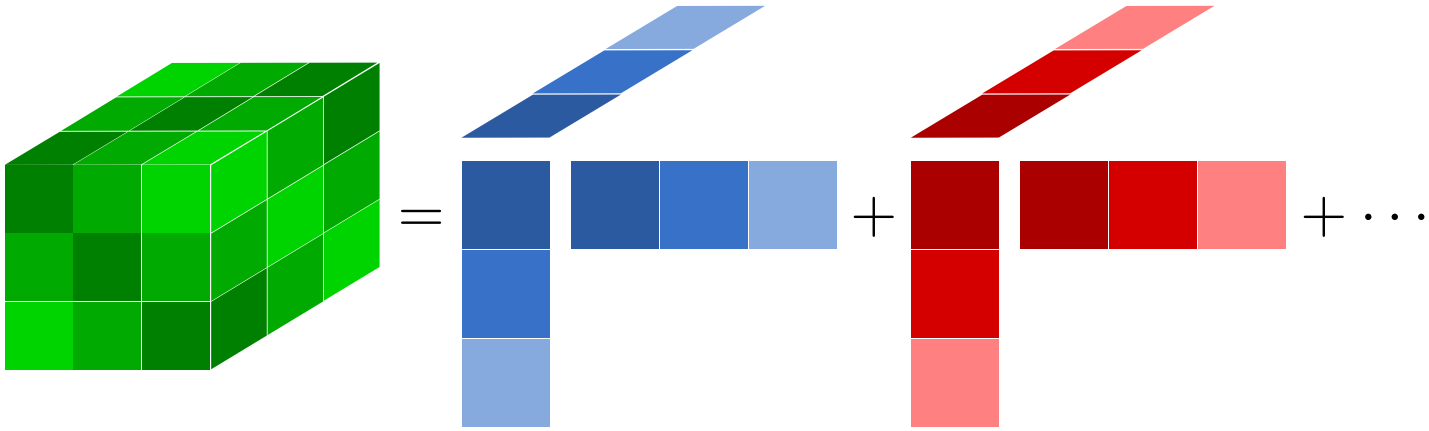
Perturbation Analysis for Tensor Decomposition



Theorem: When $\boxed{\|E\| < \frac{\lambda_{\min}}{\sqrt{d}}}$, power method recovers $\{v_i\}$ up to error $\|E\|$ with linear no. of restarts.

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Non-orthogonal Tensor Decomposition

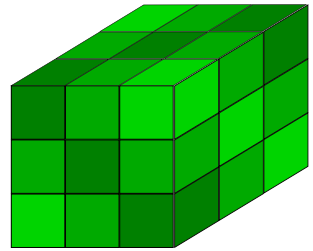


$$T = v_1{}^{\otimes 3} + v_2{}^{\otimes 3} + \cdots,$$

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Non-orthogonal Tensor Decomposition

Orthogonalization

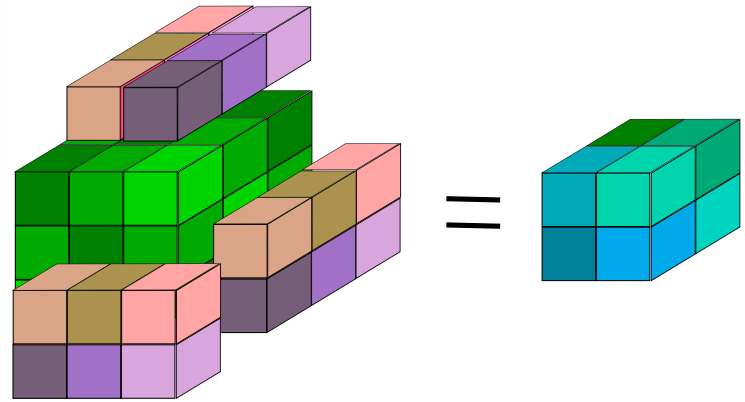


Input tensor T

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Non-orthogonal Tensor Decomposition

Orthogonalization

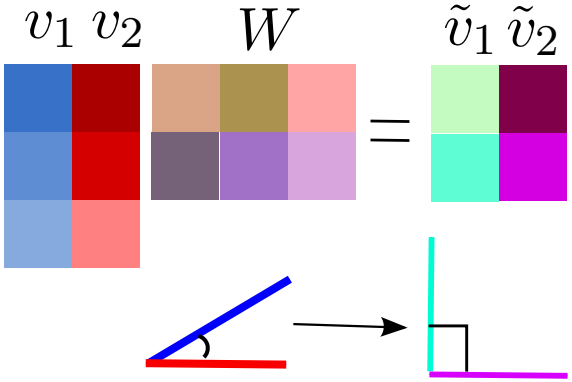


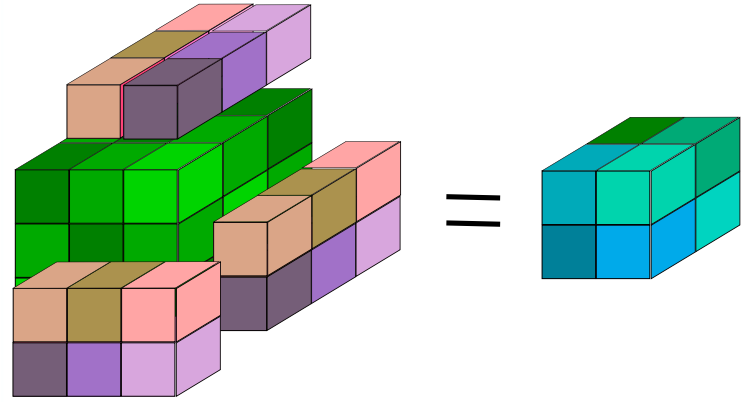
$$T(W, W, W) = \tilde{T}$$

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Non-orthogonal Tensor Decomposition

Orthogonalization



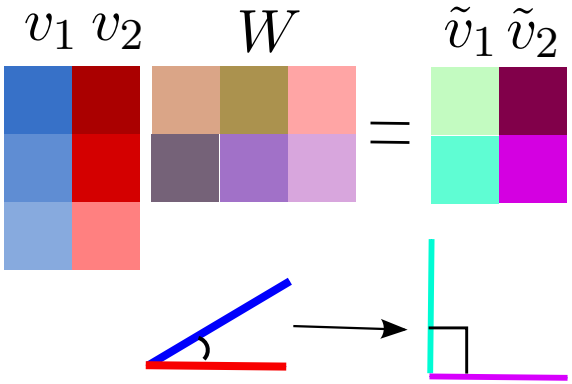


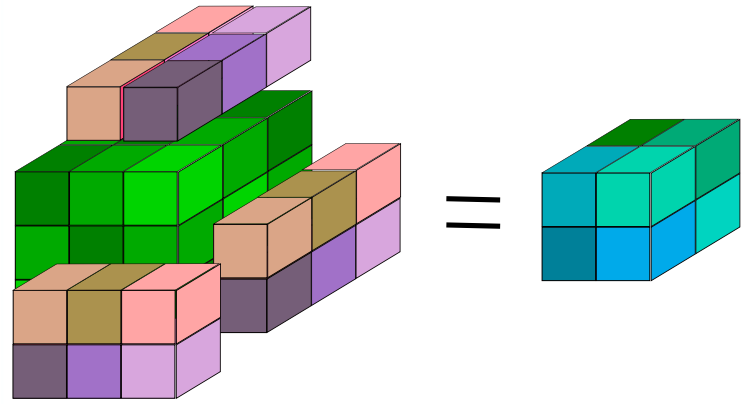
$$T(W, W, W) = \tilde{T}$$

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Non-orthogonal Tensor Decomposition

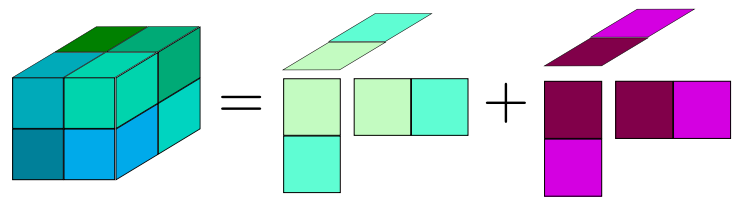
Orthogonalization





$$T(W, W, W) = \tilde{T}$$

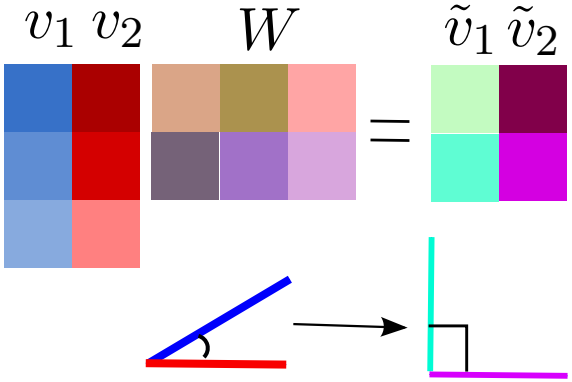
$$\tilde{T} = T(W, W, W) = \tilde{v_1}^{\otimes 3} + \tilde{v_2}^{\otimes 3} + \cdots,$$

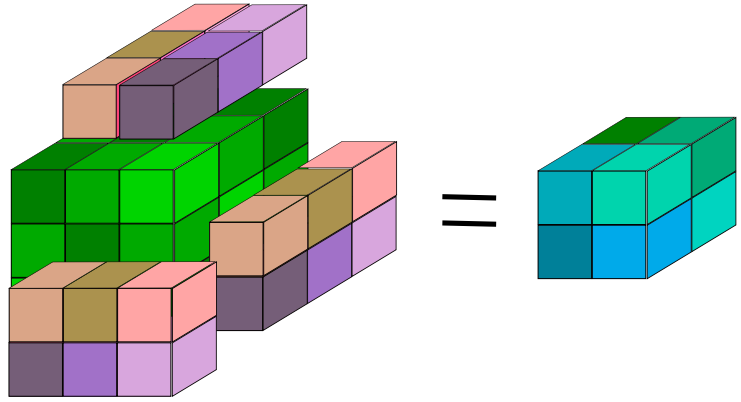


A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Non-orthogonal Tensor Decomposition

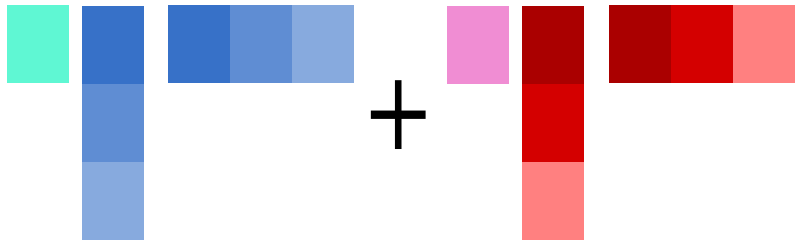
Orthogonalization



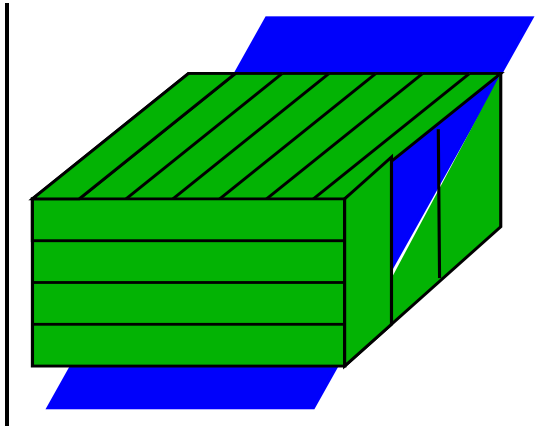


$$T(W, W, W) = \tilde{T}$$

Find W using SVD of Matrix Slice



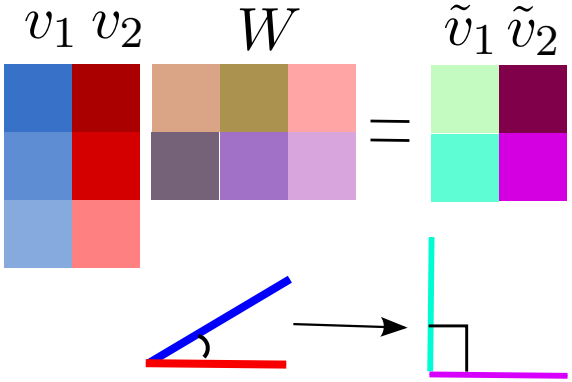
$$M = T(\cdot, \cdot, \theta) =$$

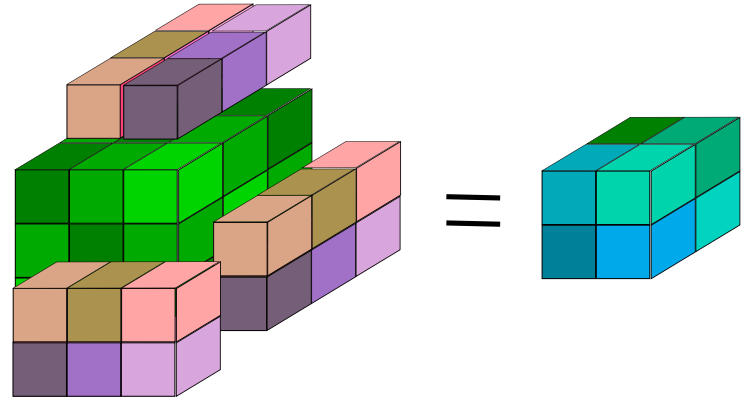


A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Non-orthogonal Tensor Decomposition

Orthogonalization





$$T(W, W, W) = \tilde{T}$$

Orthogonalization: invertible when v_i's linearly independent.

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Recap: Basic Tensor Decomposition Method

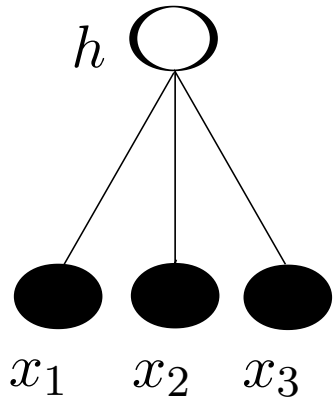
Toy Example in MATLAB

- Simulated Samples: Exchangeable Model
- Whiten The Samples
  - Second Order Moments
  - Matrix Decomposition
- Orthogonal Tensor Eigen Decomposition
  - Third Order Moments
  - Power Iteration

Simulated Samples: Exchangeable Model

Model Parameters

- Hidden State:
  $h \in$ basis $\{e_1, \ldots, e_k\}$
  $k = 2$
- Observed States:
  $x_i \in$ basis $\{e_1, \ldots, e_d\}$
  $d = 3$
- Conditional Independency:
  $x_1 \perp\!\!\!\perp x_2 \perp\!\!\!\perp x_3 | h$
  Transition Matrix: A
- Exchangeability:
  $\mathbb{E}[x_i|h] = Ah, \forall i \in 1, 2, 3$



Simulated Samples: Exchangeable Model

Model Parameters

- Hidden State:
  $h \in$ basis $\{e_1, \ldots, e_k\}$
  $k = 2$
- Observed States:
  $x_i \in$ basis $\{e_1, \ldots, e_d\}$
  $d = 3$
- Conditional Independency:
  $x_1 \perp\!\!\!\perp x_2 \perp\!\!\!\perp x_3 | h$
  Transition Matrix: A
- Exchangeability:
  $\mathbb{E}[x_i|h] = Ah, \forall i \in 1, 2, 3$

Generate Samples Snippet

```
for t = 1 : n
  % generate h for this sample
  h_category=(rand()>0.5) + 1;
  h(t,h_category)=1;
  transition_cum=cumsum(A_true(:,h_category));
  % generate x1 for this sample | h
  x_category=find(transition_cum> rand(),1);
  x1(t,x_category)=1;
  % generate x2 for this sample | h
  x_category=find(transition_cum >rand(),1);
  x2(t,x_category)=1;
  % generate x3 for this sample | h
  x_category=find(transition_cum > rand(),1);
  x3(t,x_category)=1;
  end
```

Whiten The Samples

Second Order Moments

- $M_2 = \frac{1}{n} \sum_t x_1^t \otimes x_2^t$

Whitening Matrix

- $W = U_w L_w^{-0.5}$,
  $[U_w, L_w] =$k-svd(M_2)

Whiten Data

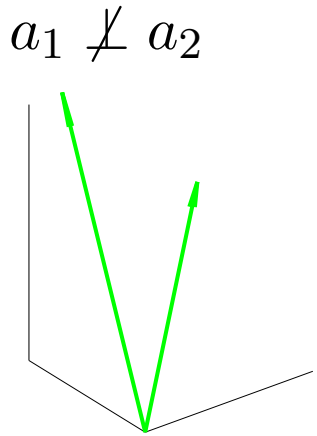
- $y_1^t = W^\top x_1^t$

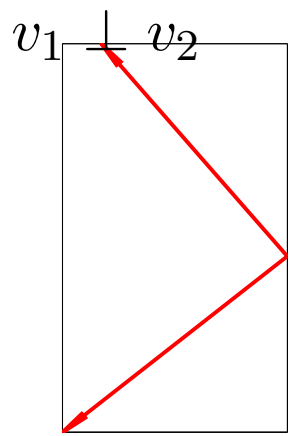
Orthogonal Basis

- $V = W^\top A \rightarrow V^\top V = I$

Whitening Snippet

```
fprintf('The second order moment M2:');
M2 = x1'*x2/n
[Uw, Lw, Vw]= svd(M2);
fprintf('M2 singular values:'); Lw
W = Uw(:,1:k)* sqrt(pinv(Lw(1:k,1:k)));
y1 = x1 * W; y2 = x2 * W; y3 = x3 * W;
```





Orthogonal Tensor Eigen Decomposition

Third Order Moments

$$T = \frac{1}{n} \sum_{t \in [n]} y_1^t \otimes y_2^t \otimes y_3^t \approx \sum_{i \in [k]} \lambda_i v_i \otimes v_i \otimes v_i, \quad V^\top V = I$$

Gradient Ascent

$$T(I, v_1, v_1) = \frac{1}{n} \sum_{t \in [n]} \langle v_1, y_2^t \rangle \langle v_1, y_3^t \rangle y_1^t \approx \sum_i \lambda_i \langle v_i, v_1 \rangle^2 v_i = \lambda_1 v_1.$$

- v_i are eigenvectors of tensor T.

Orthogonal Tensor Eigen Decomposition

$$T \leftarrow T - \sum_j \lambda_j v_j^{\otimes^3}, \quad v \leftarrow \frac{T(I,v,v)}{\|T(I,v,v)\|}$$

Power Iteration Snippet

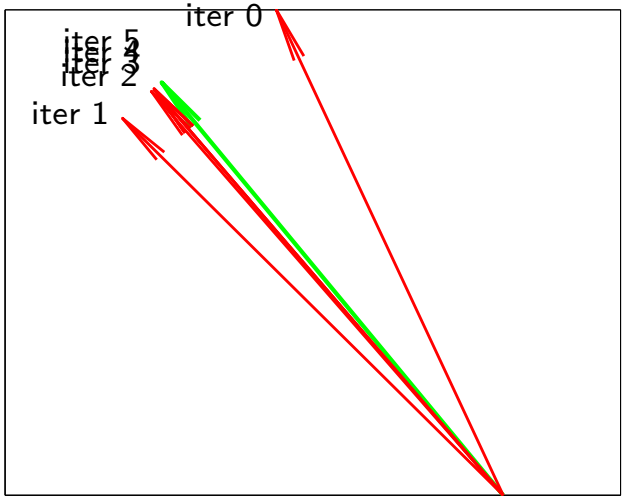
```
V = zeros(k,k); Lambda = zeros(k,1);
for i = 1:k
  v_old = rand(k,1); v_old = normc(v_old);
  for iter = 1 : Maxiter
    v_new = (y1'* ((y2*v_old).*(y3*v_old)))/n;
    if i >1
    % deflation
      for j = 1: i-1
        v_new=v_new-(V(:,j)*(v_old'*V(:,j))2)* Lambda(j);
      end
    end
    lambda = norm(v_new);v_new = normc(v_new);
    if norm(v_old - v_new) < TOL
      fprintf('Converged at iteration %d.', iter);
      V(:,i) = v_new; Lambda(i,1) = lambda;
      break;
    end
    v_old = v_new;
  end
end
```

Orthogonal Tensor Eigen Decomposition

$$T \leftarrow T - \sum_j \lambda_j v_j^{\otimes^3}, \quad v \leftarrow \frac{T(I,v,v)}{\|T(I,v,v)\|}$$

Power Iteration Snippet

```
V = zeros(k,k); Lambda = zeros(k,1);
for i = 1:k
  v_old = rand(k,1); v_old = normc(v_old);
  for iter = 1 : Maxiter
    v_new = (y1'* ((y2*v_old).*(y3*v_old)))/n;
    if i >1
    % deflation
      for j = 1: i-1
        v_new=v_new-(V(:,j)*(v_old'*V(:,j))2)* Lambda(j);
      end
    end
    lambda = norm(v_new);v_new = normc(v_new);
    if norm(v_old - v_new) < TOL
      fprintf('Converged at iteration %d.', iter);
      V(:,i) = v_new; Lambda(i,1) = lambda;
      break;
    end
    v_old = v_new;
  end
end
```



Green: Groundtruth

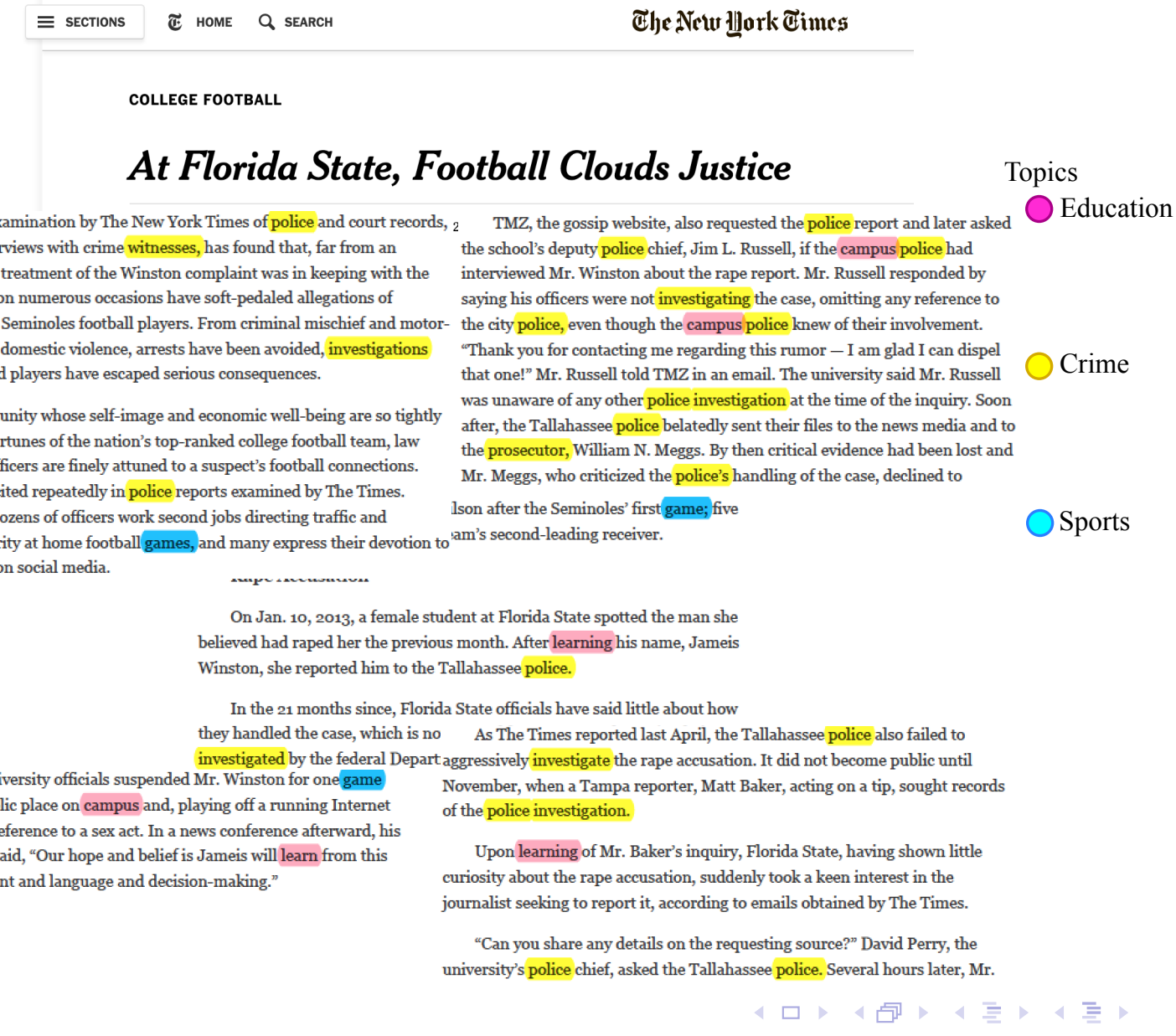
Red: Estimation at each iteration

Outline

1. Introduction
2. Why Tensors?
3. Algorithms for Tensor Decomposition
4. Learning Probabilistic Models
5. Learning Representations
6. Conclusion

Automated Categorization of Documents

Mixed topics



SECTIONS HOME SEARCH The New York Times

COLLEGE FOOTBALL

At Florida State, Football Clouds Justice

Now, an examination by The New York Times of police and court records, along with interviews with crime witnesses, has found that, far from an aberration, the treatment of the Winston complaint was in keeping with the way the police on numerous occasions have soft-pedaled allegations of wrongdoing by Seminoles football players. From criminal mischief and motor-vehicle theft to domestic violence, arrests have been avoided, investigations have stalled and players have escaped serious consequences.

In a community whose self-image and economic well-being are so tightly bound to the fortunes of the nation's top-ranked college football team, law enforcement officers are finely attuned to a suspect's football connections. Those ties are cited repeatedly in police reports examined by The Times. What's more, dozens of officers work second jobs directing traffic and providing security at home football games, and many express their devotion to the Seminoles on social media.

TMZ, the gossip website, also requested the police report and later asked the school's deputy police chief, Jim L. Russell, if the campus police had interviewed Mr. Winston about the rape report. Mr. Russell responded by saying his officers were not investigating the case, omitting any reference to the city police, even though the campus police knew of their involvement. "Thank you for contacting me regarding this rumor — I am glad I can dispel that one!" Mr. Russell told TMZ in an email. The university said Mr. Russell was unaware of any other police investigation at the time of the inquiry. Soon after, the Tallahassee police belatedly sent their files to the news media and to the prosecutor, William N. Meggs. By then critical evidence had been lost and Mr. Meggs, who criticized the police's handling of the case, declined to

lson after the Seminoles' first game; five

am's second-leading receiver.

On Jan. 10, 2013, a female student at Florida State spotted the man she believed had raped her the previous month. After learning his name, Jameis Winston, she reported him to the Tallahassee police.

In the 21 months since, Florida State officials have said little about how they handled the case, which is no

investigated by the federal Depart

Most recently, university officials suspended Mr. Winston for one game after he stood in a public place on campus and, playing off a running Internet gag, shouted a crude reference to a sex act. In a news conference afterward, his coach, Jimbo Fisher, said, "Our hope and belief is Jameis will learn from this and use better judgment and language and decision-making."

As The Times reported last April, the Tallahassee police also failed to aggressively investigate the rape accusation. It did not become public until November, when a Tampa reporter, Matt Baker, acting on a tip, sought records of the police investigation.

Upon learning of Mr. Baker's inquiry, Florida State, having shown little curiosity about the rape accusation, suddenly took a keen interest in the journalist seeking to report it, according to emails obtained by The Times.

"Can you share any details on the requesting source?" David Perry, the university's police chief, asked the Tallahassee police. Several hours later, Mr.

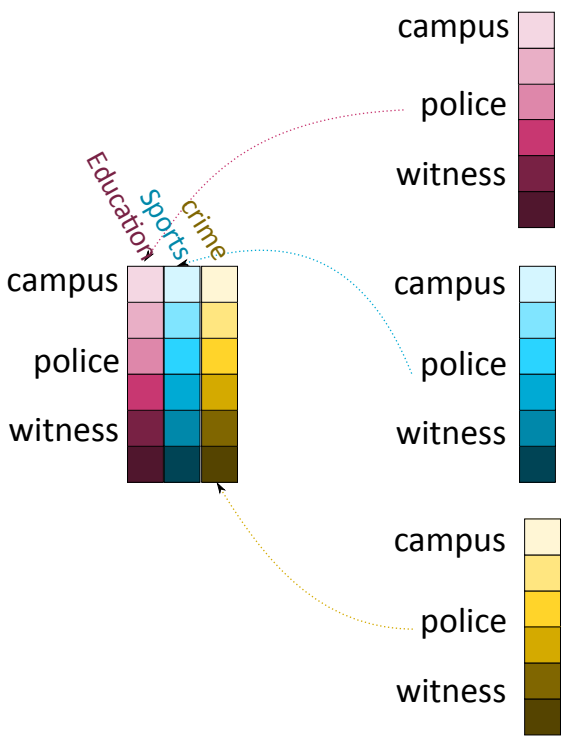
Topics

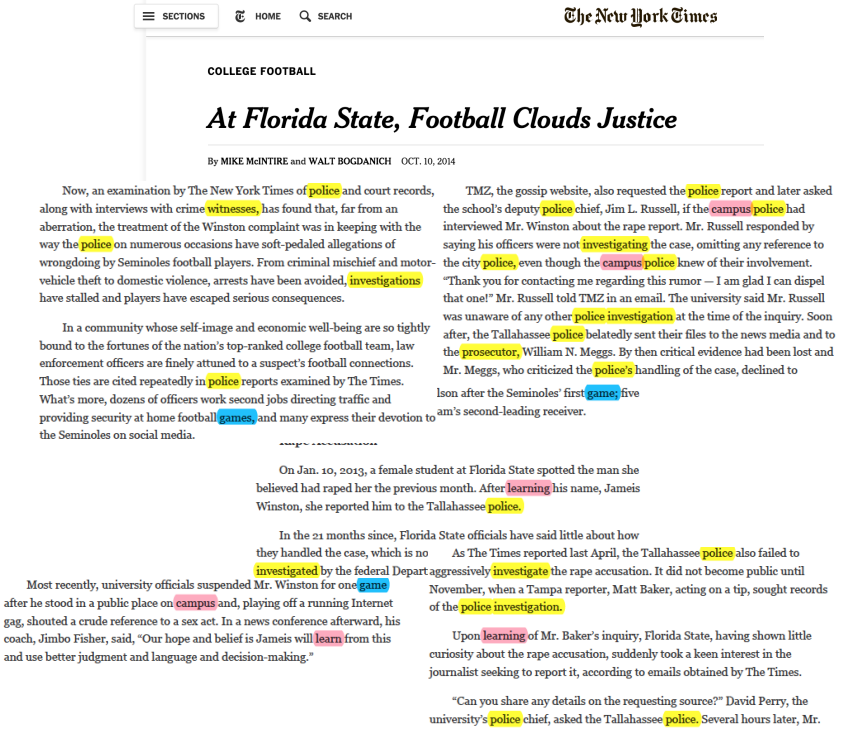
Education

Crime

Sports

Extracting Topics from Documents




SECTIONS HOME SEARCH The New York Times

COLLEGE FOOTBALL

At Florida State, Football Clouds Justice

By MIKE McINTIRE and WALT BOGDANICH OCT. 10, 2014

Now, an examination by The New York Times of police and court records, along with interviews with crime witnesses, has found that, far from an aberration, the treatment of the Winston complaint was in keeping with the way the police on numerous occasions have soft-pedaled allegations of wrongdoing by Seminoles football players. From criminal mischief and motor-vehicle theft to domestic violence, arrests have been avoided, investigations have stalled and players have escaped serious consequences.

In a community whose self-image and economic well-being are so tightly bound to the fortunes of the nation's top-ranked college football team, law enforcement officers are finely attuned to a suspect's football connections. Those ties are cited repeatedly in police reports examined by The Times. What's more, dozens of officers work second jobs directing traffic and providing security at home football games, and many express their devotion to the Seminoles on social media.

TMZ, the gossip website, also requested the police report and later asked the school's deputy police chief, Jim L. Russell, if the campus police had interviewed Mr. Winston about the rape report. Mr. Russell responded by saying his officers were not investigating the case, omitting any reference to the city police, even though the campus police knew of their involvement. "Thank you for contacting me regarding this rumor — I am glad I can dispel that one!" Mr. Russell told TMZ in an email. The university said Mr. Russell was unaware of any other police investigation at the time of the inquiry. Soon after, the Tallahassee police belatedly sent their files to the news media and to the prosecutor, William N. Meggs. By then critical evidence had been lost and Mr. Meggs, who criticized the police's handling of the case, declined to

lson after the Seminoles' first game; five
am's second-leading receiver.

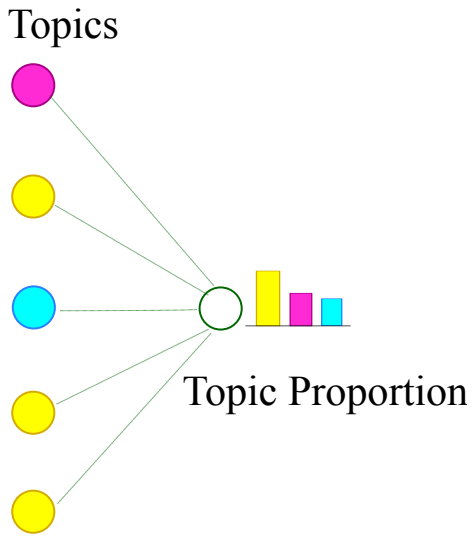
On Jan. 10, 2013, a female student at Florida State spotted the man she believed had raped her the previous month. After learning his name, Jameis Winston, she reported him to the Tallahassee police.

In the 21 months since, Florida State officials have said little about how they handled the case, which is no As The Times reported last April, the Tallahassee police also failed to investigated by the federal Depart aggressively investigate the rape accusation. It did not become public until November, when a Tampa reporter, Matt Baker, acting on a tip, sought records of the police investigation.

Most recently, university officials suspended Mr. Winston for one game after he stood in a public place on campus and, playing off a running Internet gag, shouted a crude reference to a sex act. In a news conference afterward, his coach, Jimbo Fisher, said, "Our hope and belief is Jameis will learn from this and use better judgment and language and decision-making."

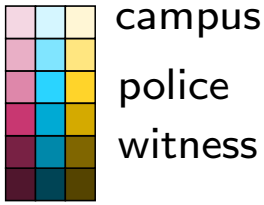
Upon learning of Mr. Baker's inquiry, Florida State, having shown little curiosity about the rape accusation, suddenly took a keen interest in the journalist seeking to report it, according to emails obtained by The Times.

"Can you share any details on the requesting source?" David Perry, the university's police chief, asked the Tallahassee police. Several hours later, Mr.



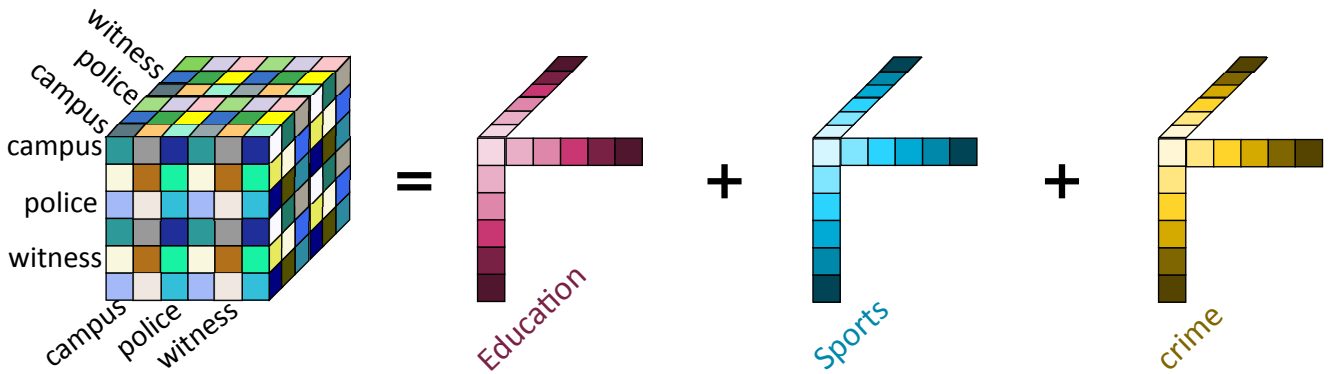
A., D. P. Foster, D. Hsu, S.M. Kakade, Y.K. Liu. "Two SVDs Suffice: Spectral decompositions for probabilistic topic modeling and latent Dirichlet allocation," NIPS 2012.

Tensor Methods for Topic Modeling



- Topic-word matrix $\mathbb{P}[\text{word} = i | \text{topic} = j]$
- Linearly independent columns

Moment Tensor: Co-occurrence of Word Triplets



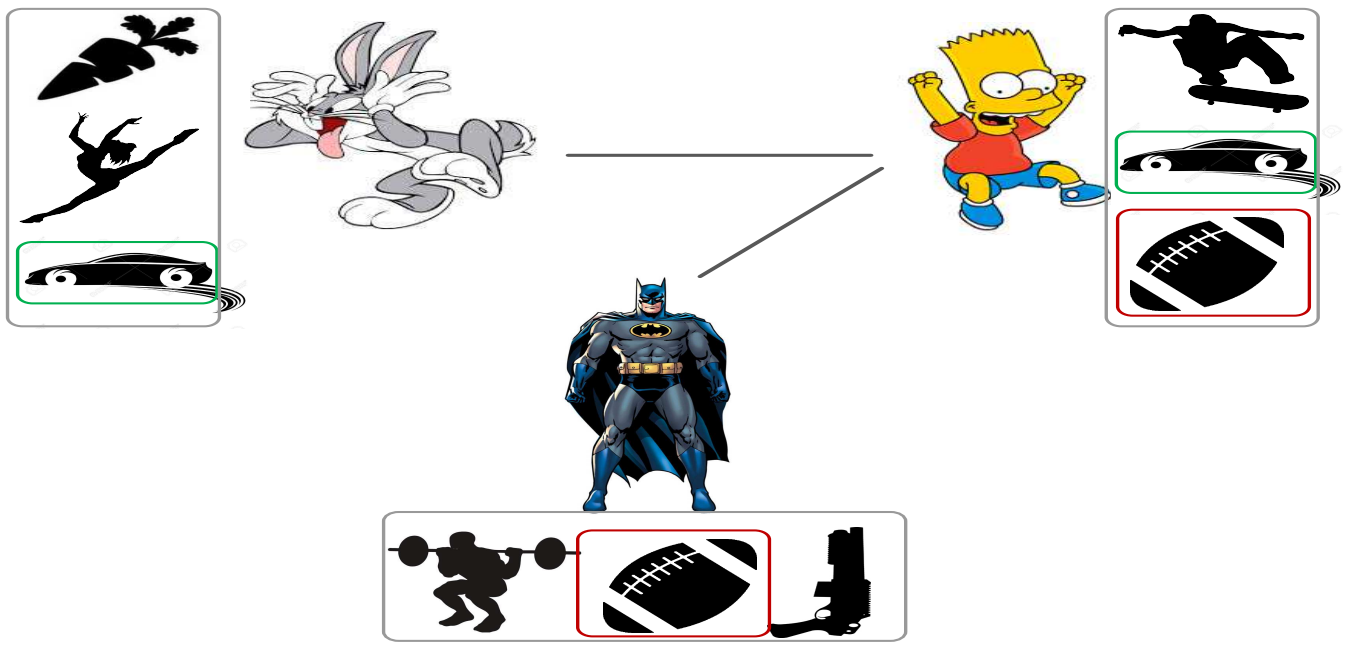
Visualization

http://newport.eecs.uci.edu/anandkumar/Lab/Lab_sub/NewYorkTime

Tensor Methods for Community Modeling

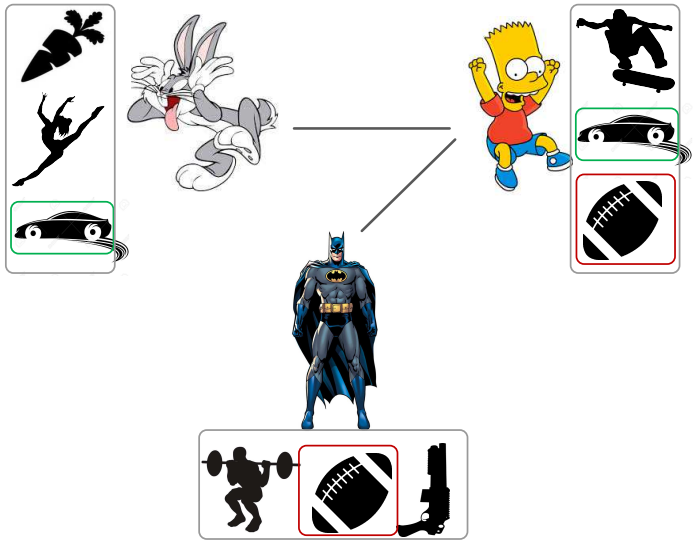
Mixed Membership Stochastic Block Model

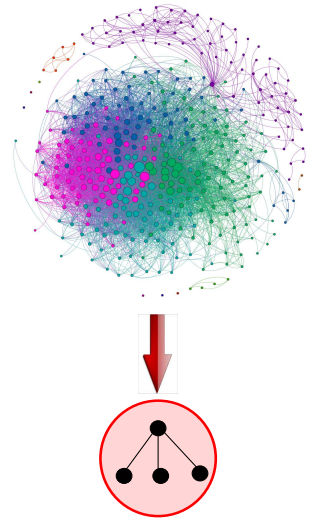
- Edges conditionally independent given community memberships.
- Community membership proportions $\sim \mathrm{Dir}(\alpha)$.



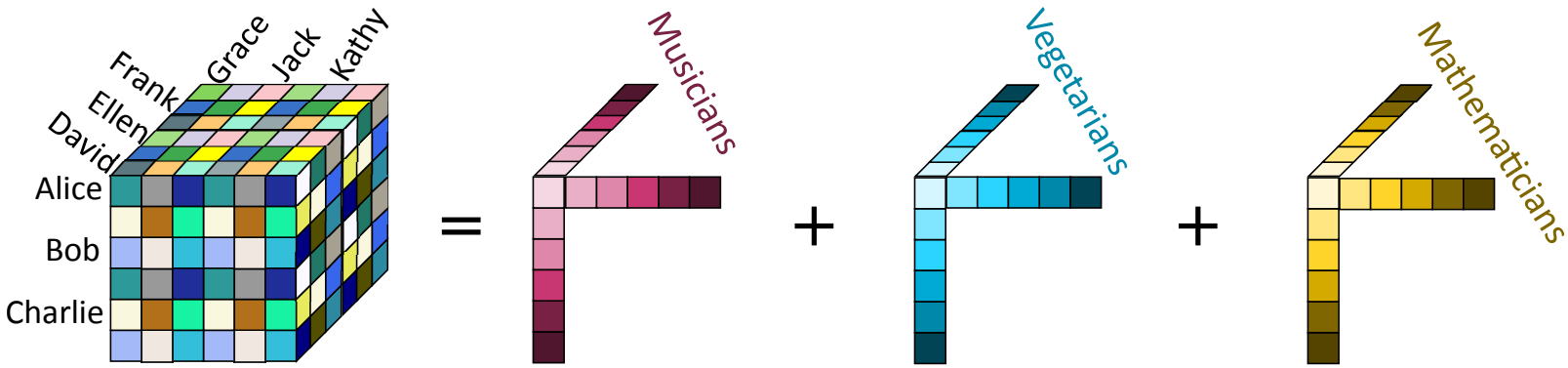
Airoldi, Blei, Fienberg, and Xing. Mixed membership stochastic blockmodels. J. of Machine Learning Research, June 2008.

Extracting Communities in Social Networks





Moment Tensor: Common Friends among Node Triplets

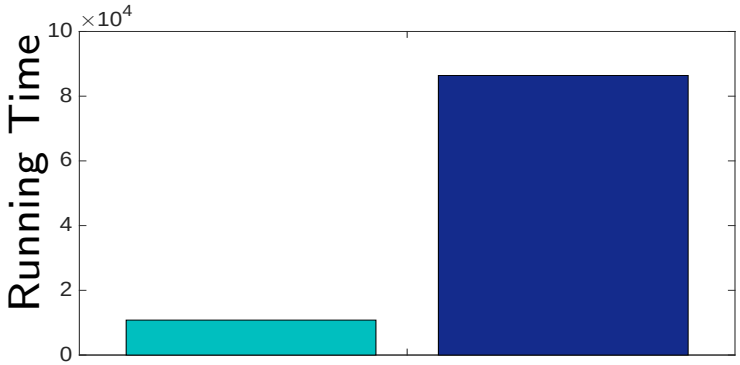


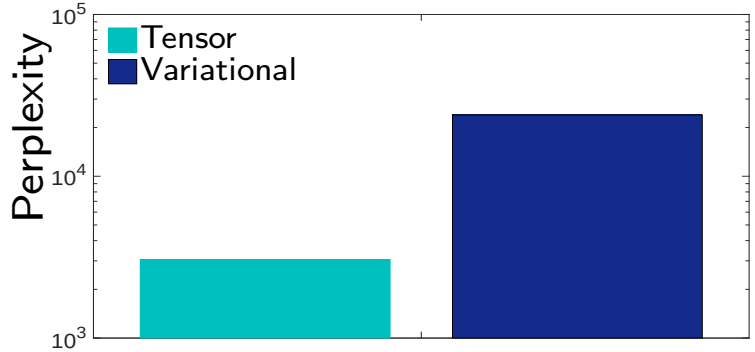
A., R. Ge, D. Hsu, S.M. Kakade. "A Tensor Spectral Approach to Learning Mixed Membership Community Models" COLT 2013.

Tensors vs. Variational Inference

Criterion: Perplexity $= \exp[-\text{likelihood}]$.

Learning Topics from PubMed on Spark, 8mil articles



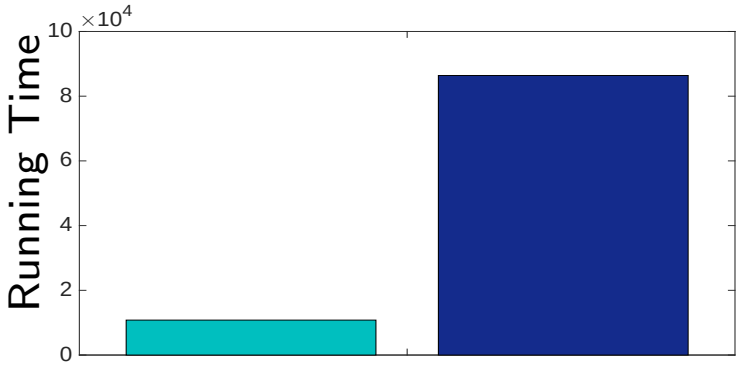


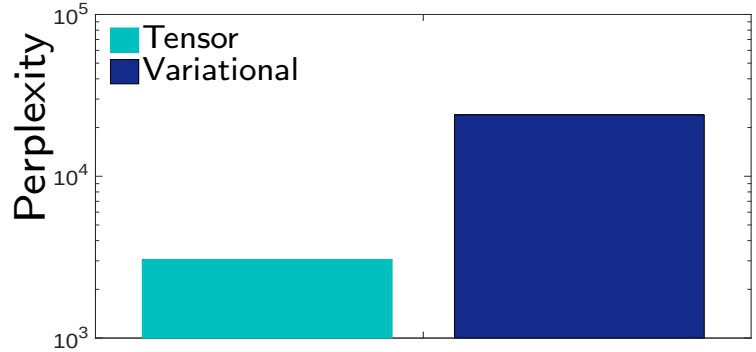
F. Huang, U.N. Niranjan, M. Hakeem, A, "Online tensor methods for training latent variable models," JMLR 2014.

Tensors vs. Variational Inference

Criterion: Perplexity $= \exp[-\text{likelihood}]$.

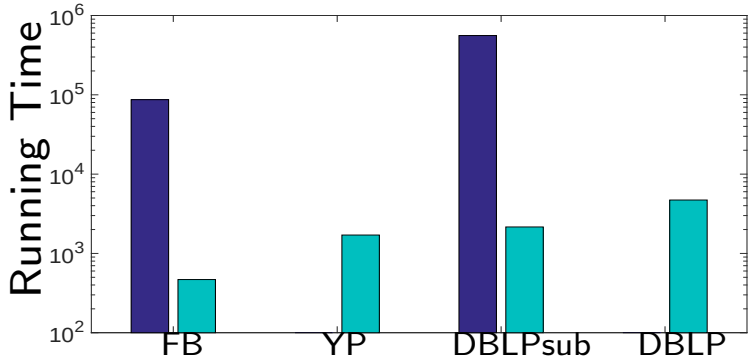
Learning Topics from PubMed on Spark, 8mil articles

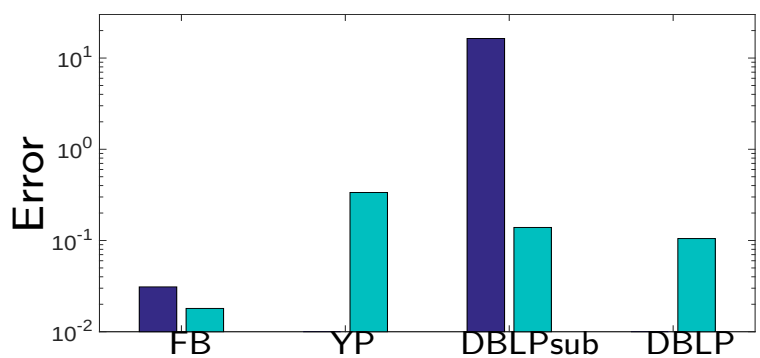




Learning network communities from social network data

Facebook $n \sim 20k$, Yelp $n \sim 40k$, DBLP-sub $n \sim 1e5$, DBLP $n \sim 1e6$.



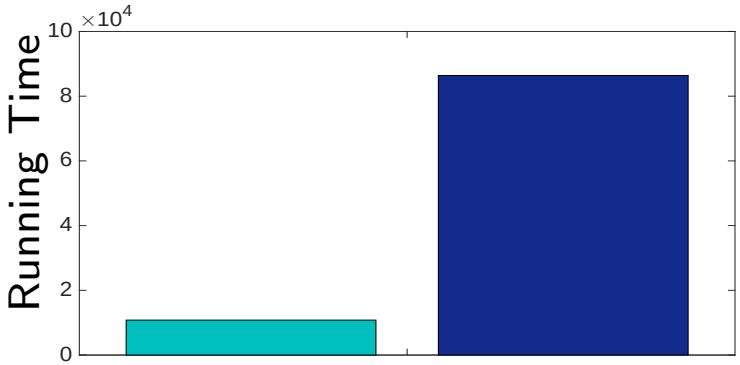


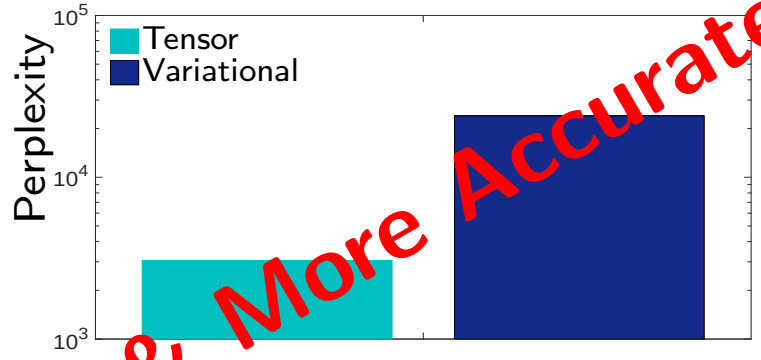
F. Huang, U.N. Niranjan, M. Hakeem, A, "Online tensor methods for training latent variable models," JMLR 2014.

Tensors vs. Variational Inference

Criterion: Perplexity $= \exp[-\text{likelihood}]$.

Learning Topics from PubMed on Spark, 8mil articles

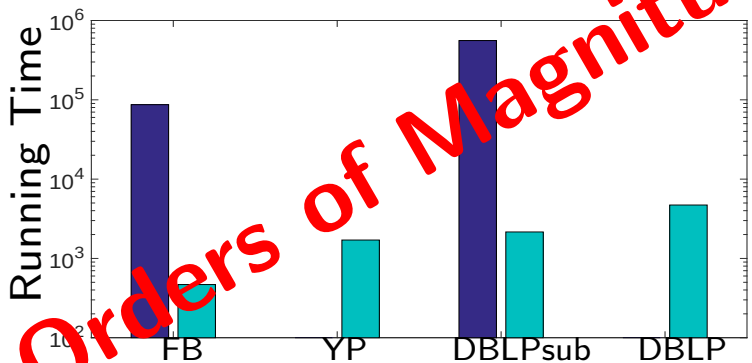


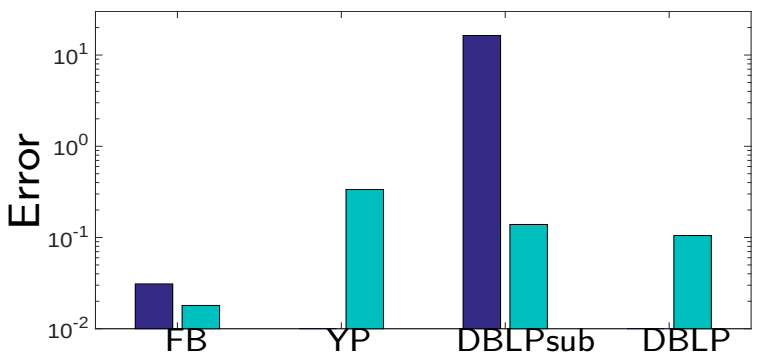


Orders of Magnitude Faster & More Accurate

Learning network communities from social network data

Facebook $n \sim 20k$, Yelp $n \sim 40k$, DBLP-sub $n \sim 1e5$, DBLP $n \sim 1e6$.

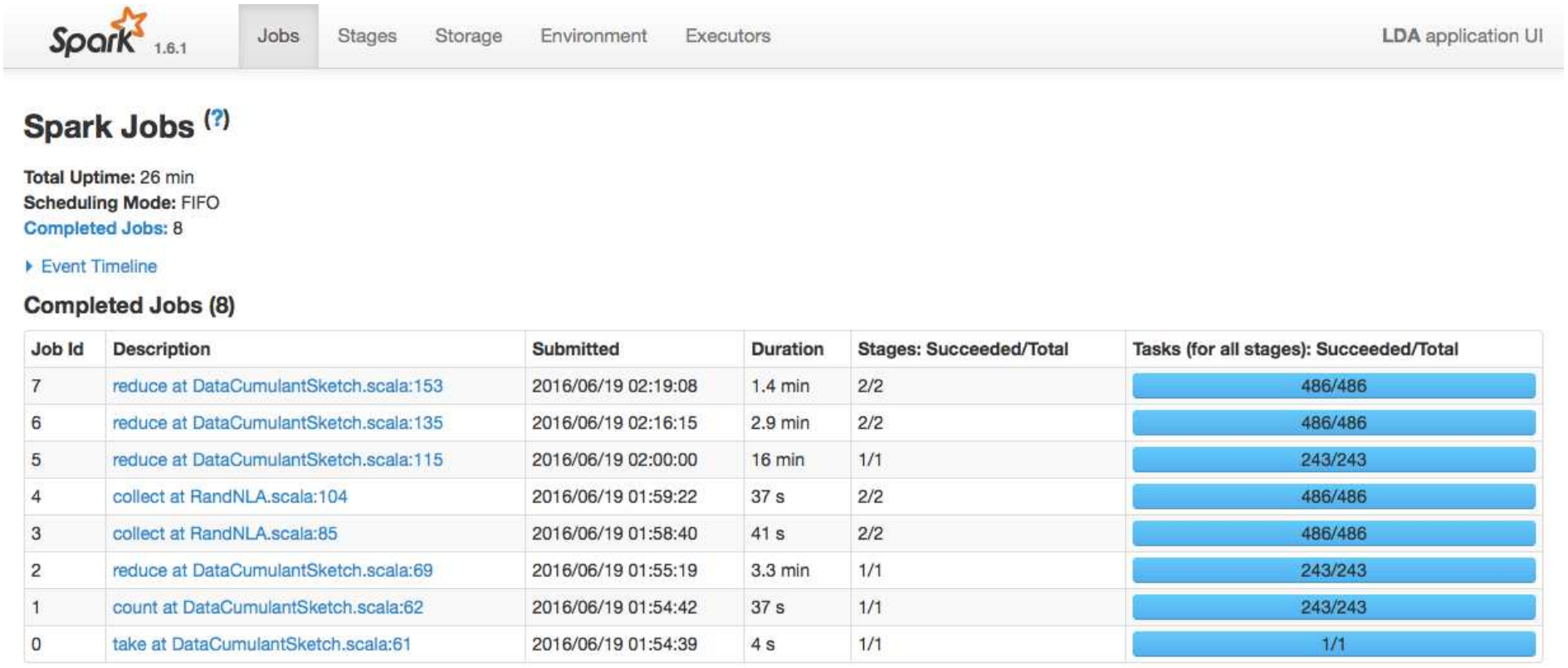




F. Huang, U.N. Niranjan, M. Hakeem, A, "Online tensor methods for training latent variable models," JMLR 2014.

Sample Screenshot

- Infrastructure: AWS 6 nodes with 36 cores each.
- Data: PubMed 8.1 million articles, 140k vocabulary. 50 topics.
- Total runtime: 27 mins.
- https://github.com/FurongHuang/SpectralLDA-TensorSpark

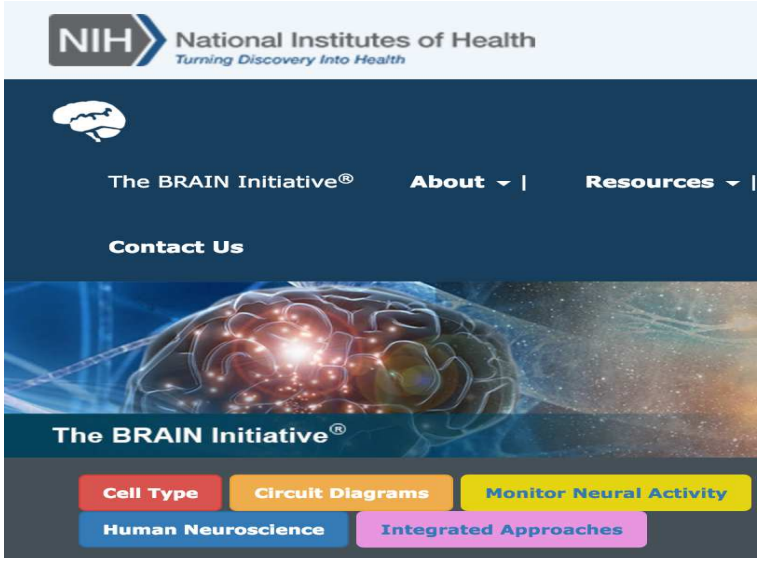


Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
7	reduce at DataCumulantSketch.scala:153	2016/06/19 02:19:08	1.4 min	2/2	486/486
6	reduce at DataCumulantSketch.scala:135	2016/06/19 02:16:15	2.9 min	2/2	486/486
5	reduce at DataCumulantSketch.scala:115	2016/06/19 02:00:00	16 min	1/1	243/243
4	collect at RandNLA.scala:104	2016/06/19 01:59:22	37 s	2/2	486/486
3	collect at RandNLA.scala:85	2016/06/19 01:58:40	41 s	2/2	486/486
2	reduce at DataCumulantSketch.scala:69	2016/06/19 01:55:19	3.3 min	1/1	243/243
1	count at DataCumulantSketch.scala:62	2016/06/19 01:54:42	37 s	1/1	243/243
0	take at DataCumulantSketch.scala:61	2016/06/19 01:54:39	4 s	1/1	1/1

Cataloging Neuronal Cell Types In the Brain

Goal

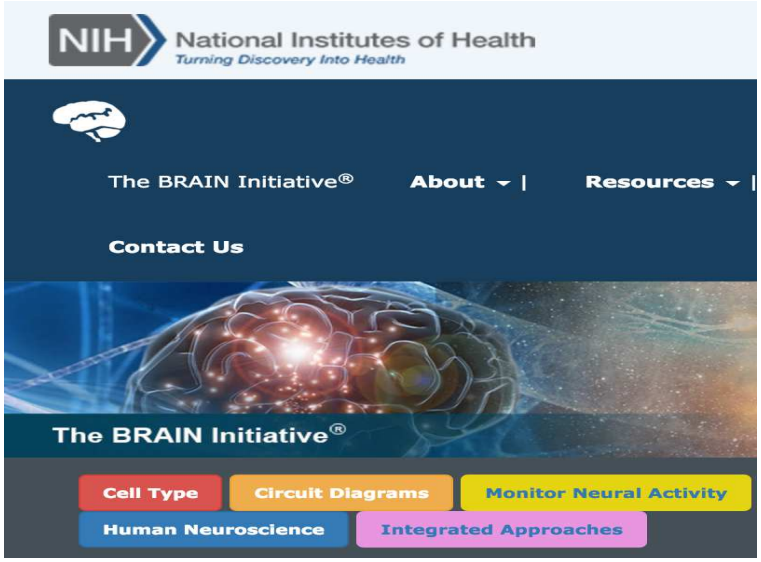
- Cataloging the neuronal cell types
- Inferring gene expression profiles of cell types



Cataloging Neuronal Cell Types In the Brain

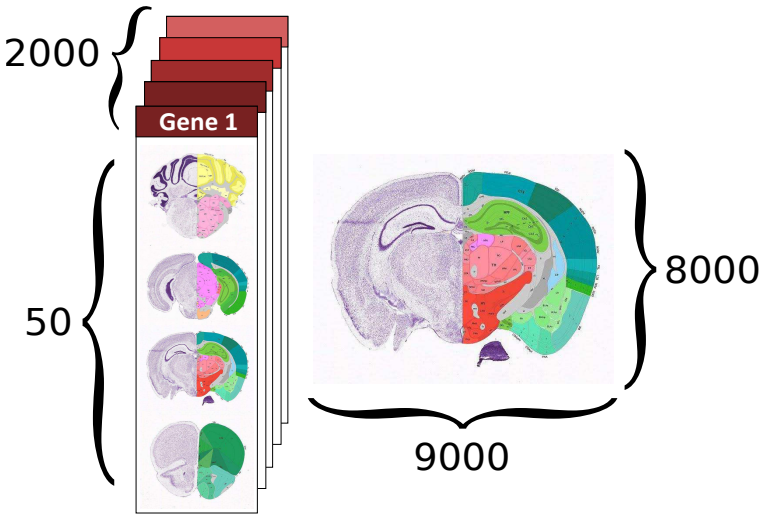
Goal

- Cataloging the neuronal cell types
- Inferring gene expression profiles of cell types



Data

- Allen Brain Atlas: Mouse Brain
- Cellular resolution *in-situ* hybridization image
- Mis-alignment



Big data: **64 TB** scale data

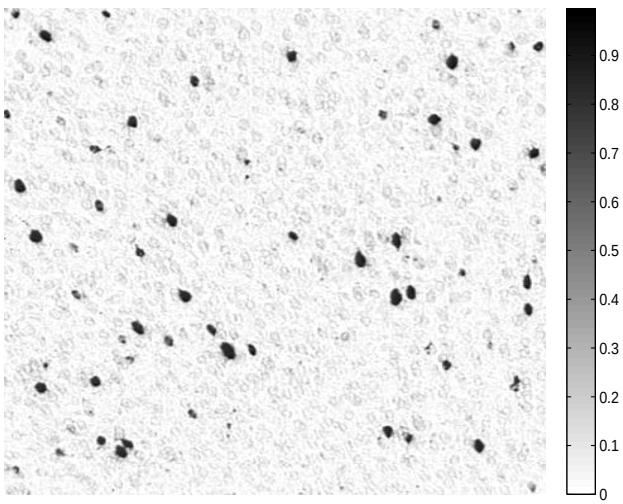
Feature Extraction - From Image to Cell

- Previous: Average Expression Level / Volume
- Ours: Measure distribution of neuron features
  - size, orientation, counts in a unit radius, gene expression level

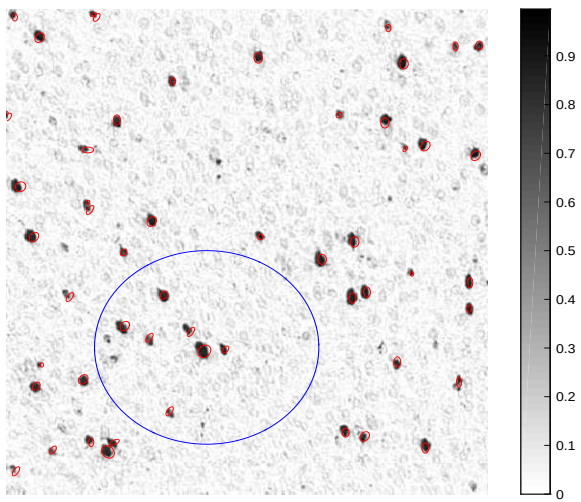
Feature Extraction - From Image to Cell

- Previous: Average Expression Level / Volume
- Ours: Measure distribution of neuron features
  - size, orientation, counts in a unit radius, gene expression level

Detect Cells

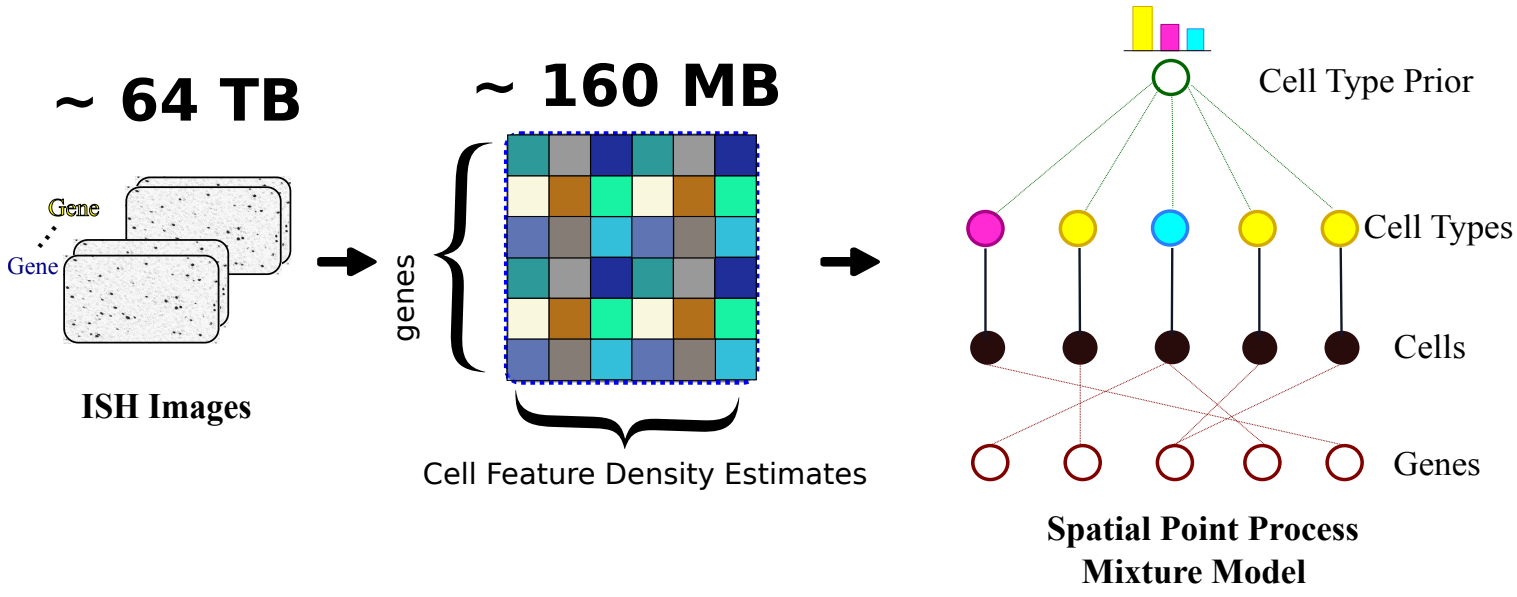


(a) Patch from gene Pvalb slice



(b) Cell detection and extraction

Spatial Point Process Mixture Model



- Discover Cell Types via **Un-mixing** of Spatial Point Process Mixtures
  - Image/Gene is a mixture of cell types
  - Cell is a mixture of genes

Discoveries via Unsupervised Learning

Type	Top Genes			
Cell-type 1	**Gfap**	*Itga3*	*Asb4*	*Actr2*
Cell-type 2	**Sst**	**Gad2**	**Gad1**	*Ifnar1*
Cell-type 3	*Bhlhe22*	*Ptgds*	*Srgap3*	*Cry2*

1. *astrocytes*
2. *interneurons*
3. *oligodendrocytes*

Discoveries via Unsupervised Learning

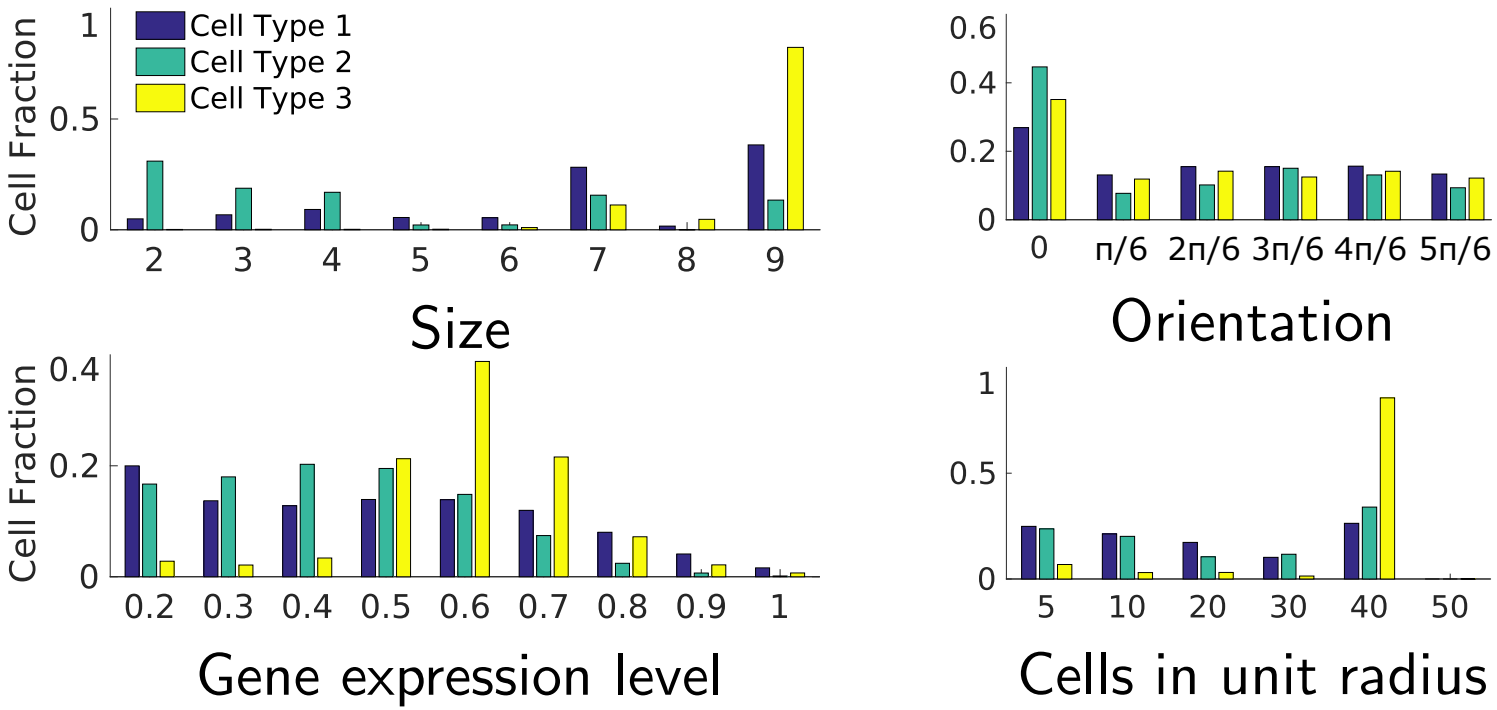
Type	Top Genes			
Cell-type 1	**Gfap**	*Itga3*	*Asb4*	*Actr2*
Cell-type 2	**Sst**	**Gad2**	**Gad1**	*Ifnar1*
Cell-type 3	*Bhlhe22*	*Ptgds*	*Srgap3*	*Cry2*

1 *astrocytes*

2 *interneurons*

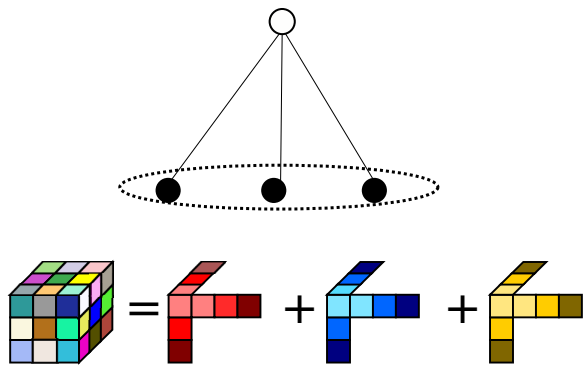
3 *oligodendrocytes*

Discovered cell feature distribution

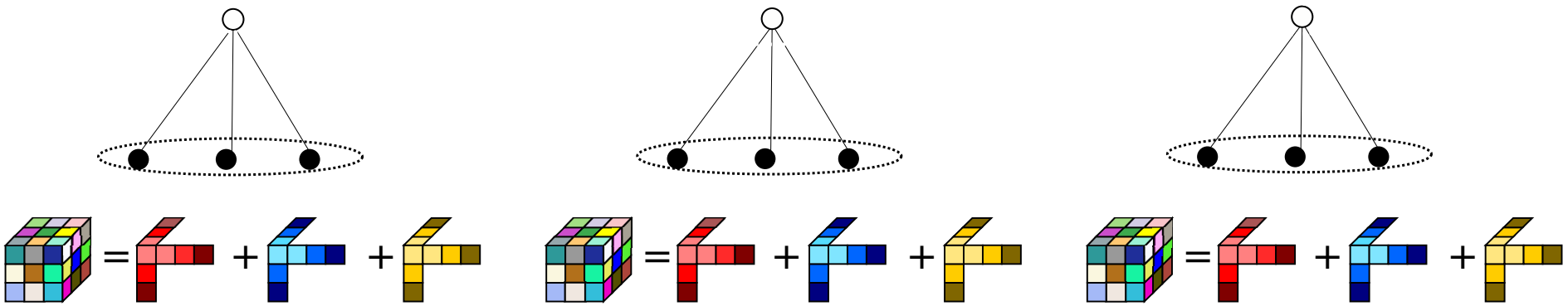


" Discovering Neuronal Cell Types and Their Gene Expression Profiles Using a Spatial Point Process Mixture Model " by F. Huang, A. Anandkumar, C. Borgs, J. Chayes, E. Fraenkel, M. Hawrylycz, E. Lein, A. Ingrosso, S. Turaga, NIPS 2015 BigNeuro workshop.

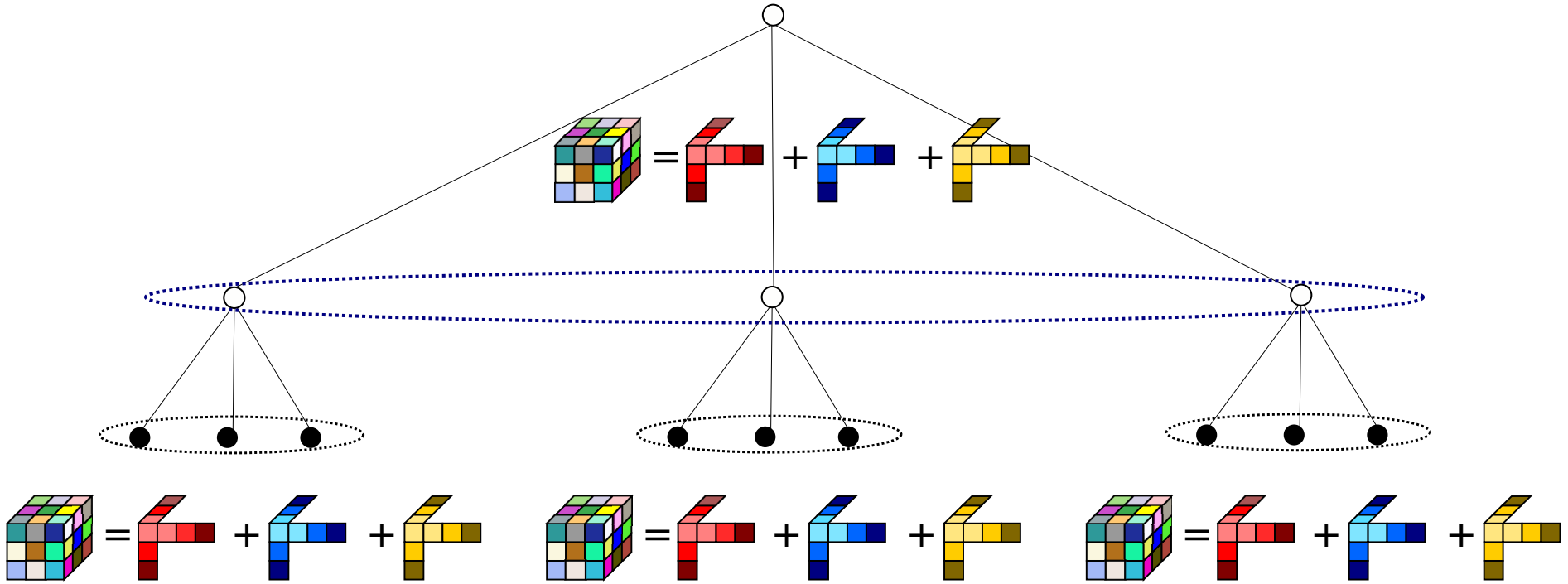
Hierarchical Tensor Decomposition



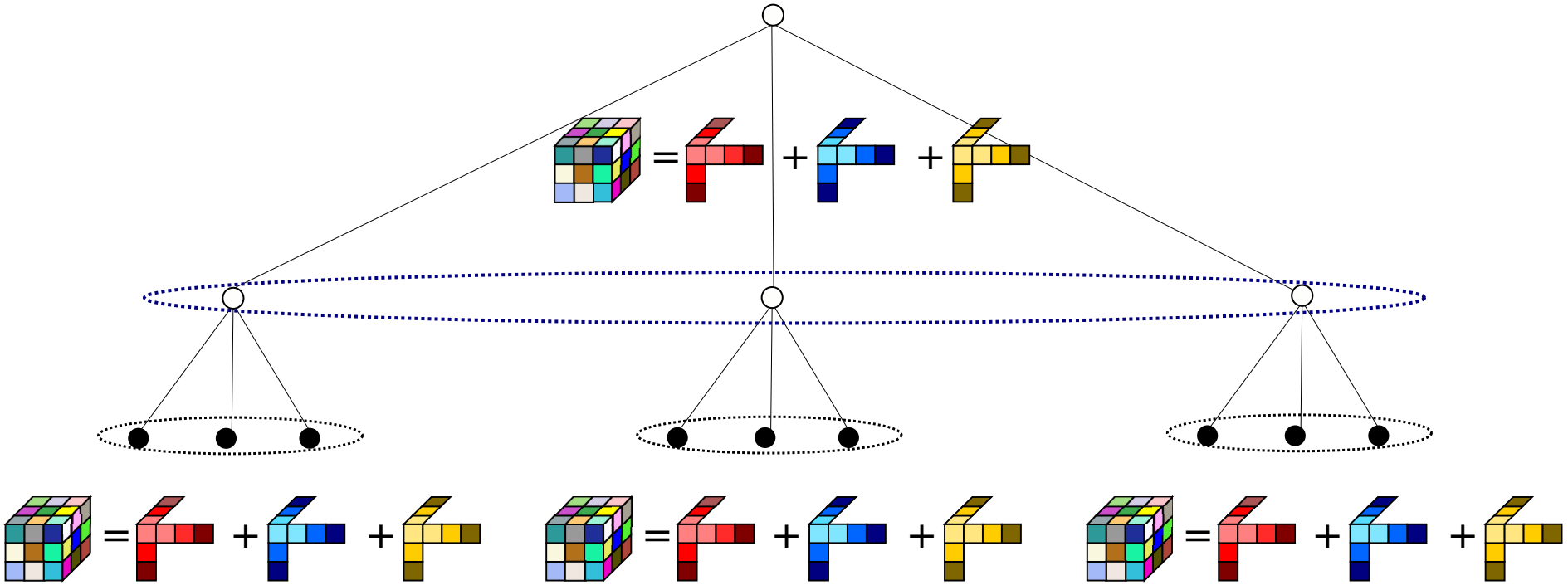
Hierarchical Tensor Decomposition



Hierarchical Tensor Decomposition



Hierarchical Tensor Decomposition

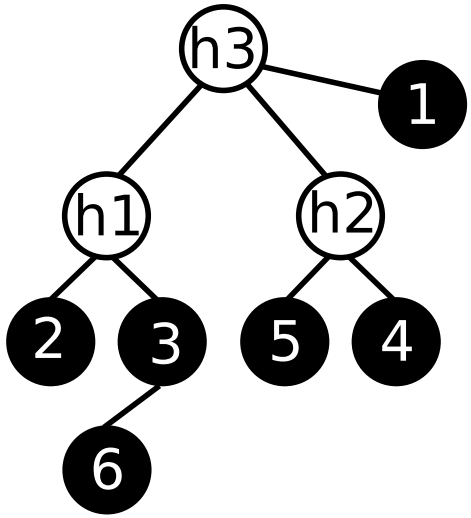


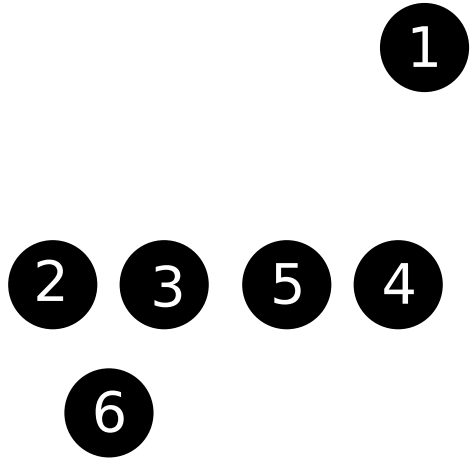
Latent tree graphical model: less restrictive

- Versatile in modeling hierarchical relations
- Number and location of hidden variables unknown

Latent Tree Model Learning - Structure

Structure & Parameter Learning: Divide and Conquer

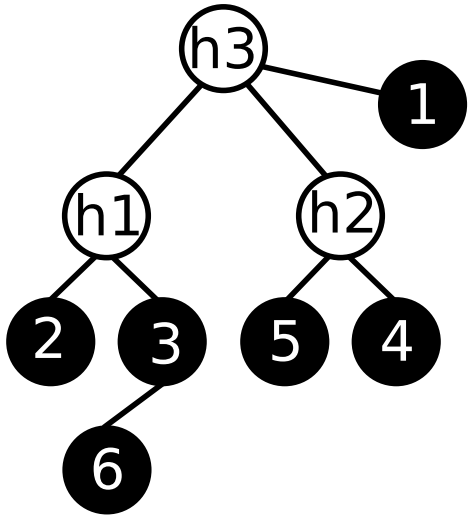


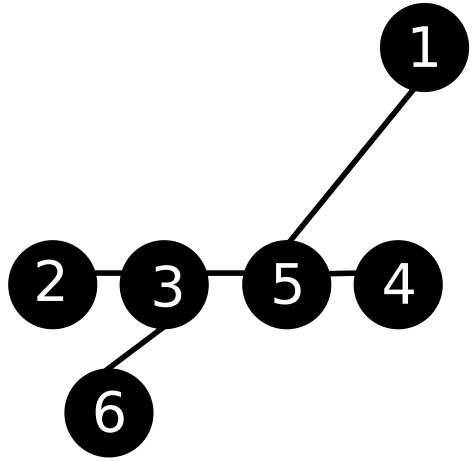


- "Divide-and-conquer" strategy

Latent Tree Model Learning - Structure

Structure & Parameter Learning: Divide and Conquer

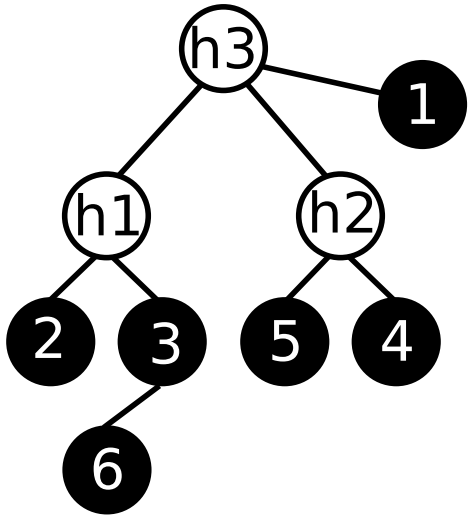


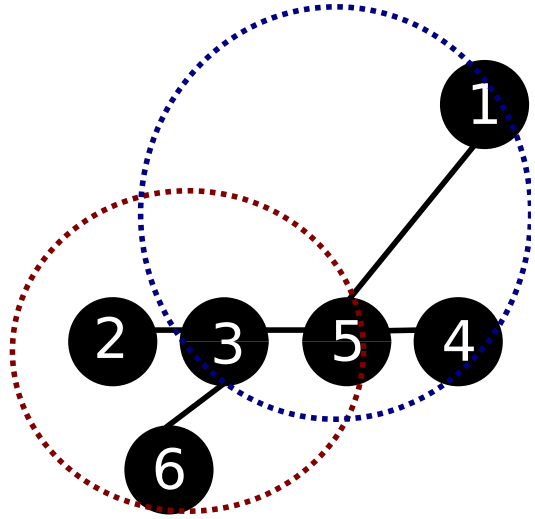


- "Divide-and-conquer" strategy
  - Find local groups: MST

Latent Tree Model Learning - Structure

Structure & Parameter Learning: Divide and Conquer

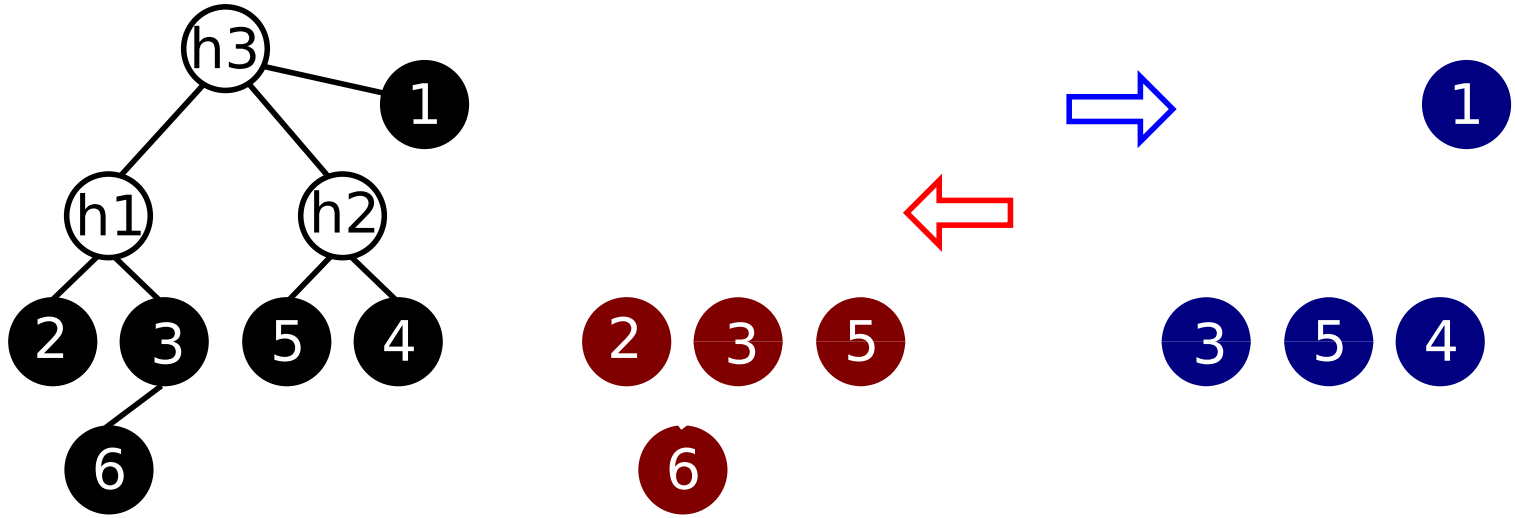




- "Divide-and-conquer" strategy
  - Find local groups: MST
  - Learn over small groups

Latent Tree Model Learning - Structure

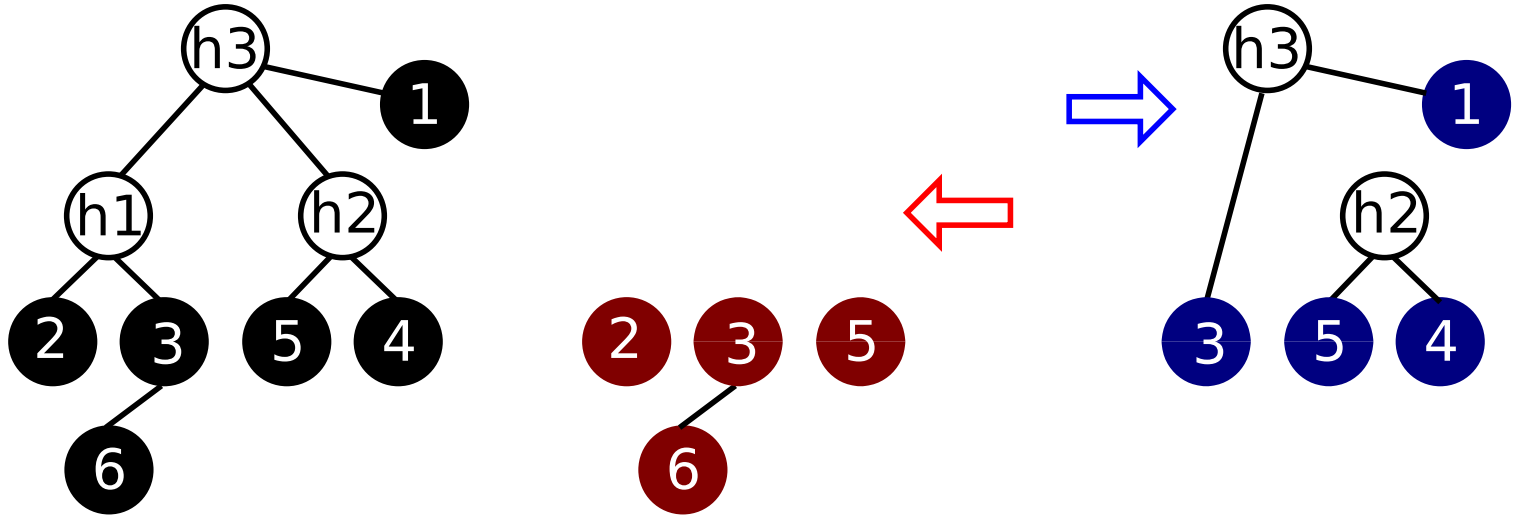
Structure & Parameter Learning: Divide and Conquer



- "Divide-and-conquer" strategy
  - Find local groups: MST
  - Learn over small groups

Latent Tree Model Learning - Structure

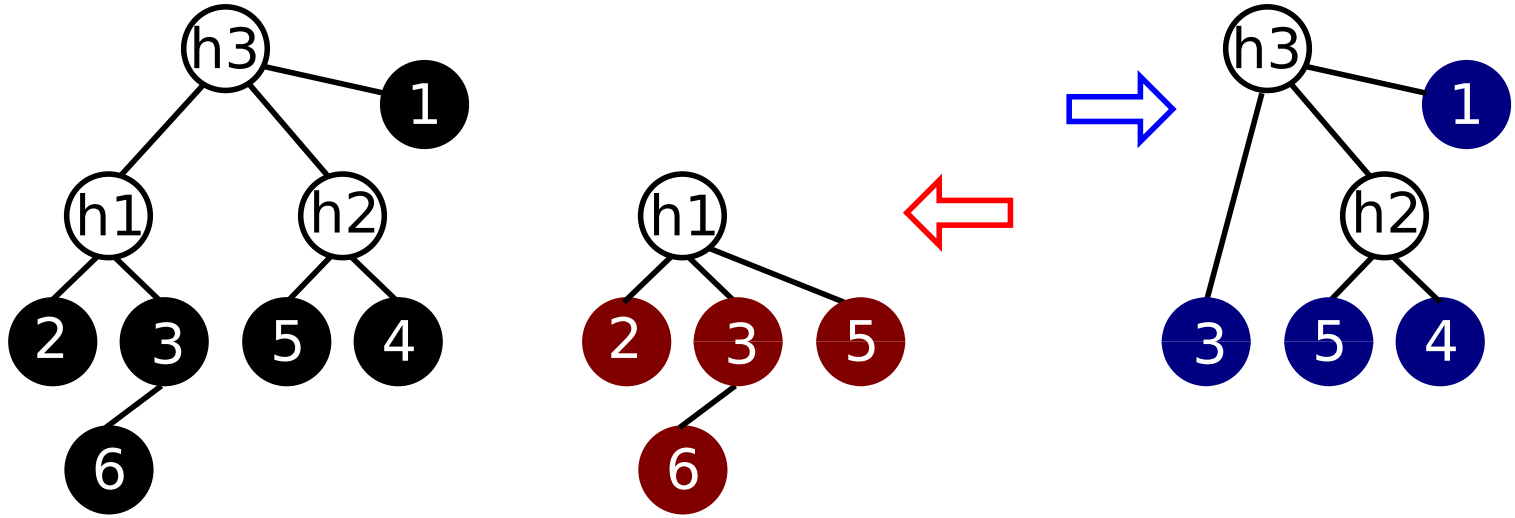
Structure & Parameter Learning: Divide and Conquer



- "Divide-and-conquer" strategy
  - Find local groups: MST
  - Learn over small groups

Latent Tree Model Learning - Structure

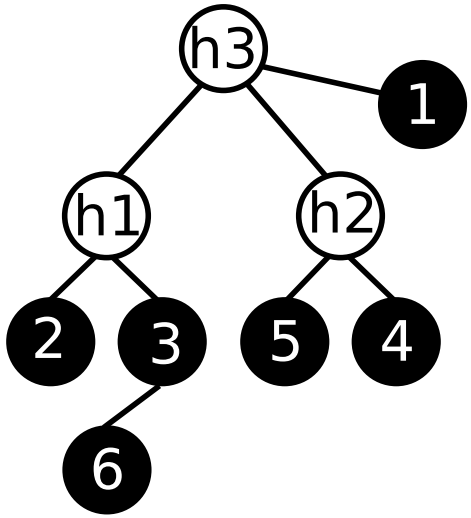
Structure & Parameter Learning: Divide and Conquer

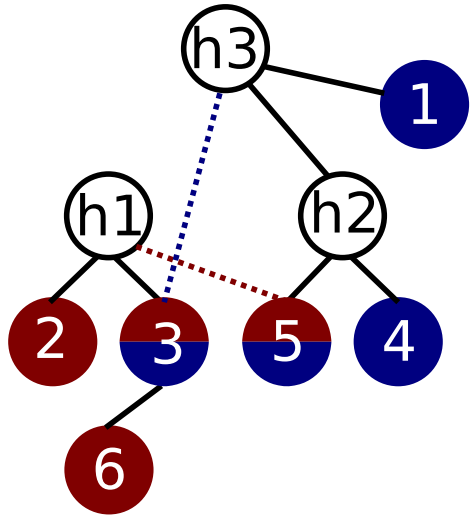


- "Divide-and-conquer" strategy
  - Find local groups: MST
  - Learn over small groups

Latent Tree Model Learning - Structure

Structure & Parameter Learning: Divide and Conquer

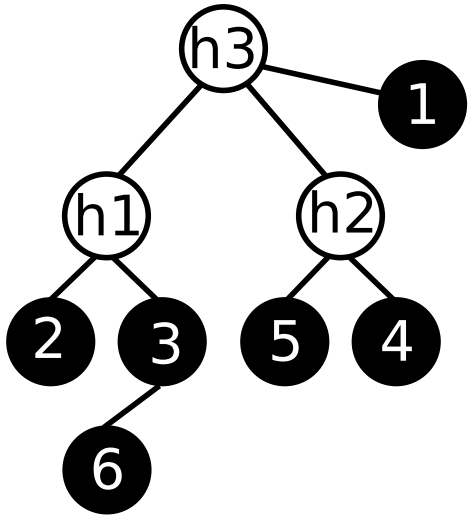


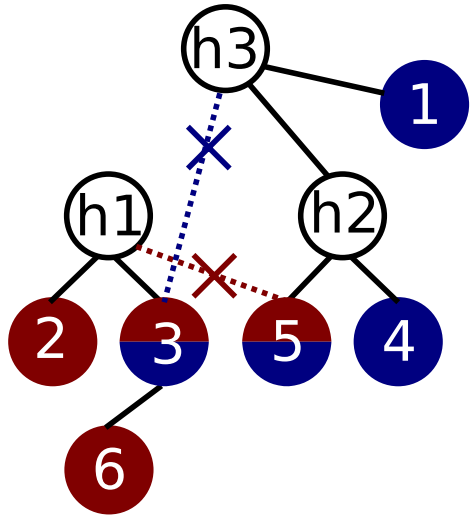


- "Divide-and-conquer" strategy
  - Find local groups: MST
  - Learn over small groups
  - Merge into global solution

Latent Tree Model Learning - Structure

Structure & Parameter Learning: Divide and Conquer

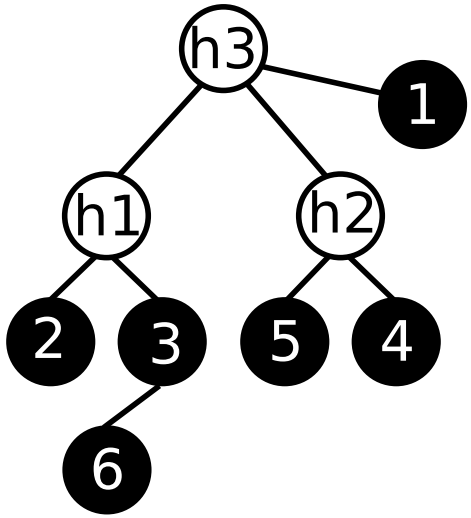


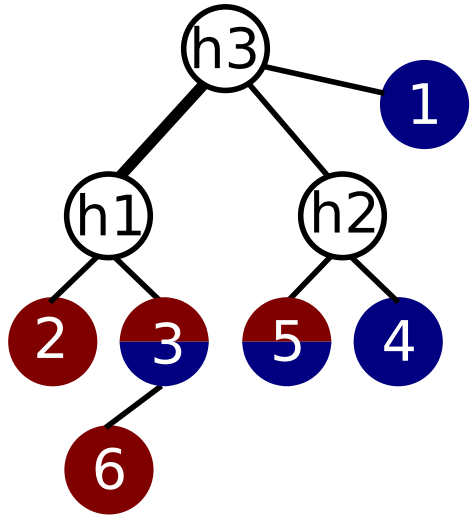


- "Divide-and-conquer" strategy
  - Find local groups: MST
  - Learn over small groups
  - Merge into global solution

Latent Tree Model Learning - Structure

Structure & Parameter Learning: Divide and Conquer

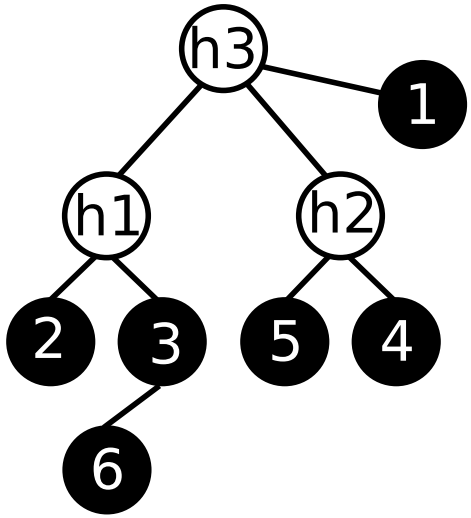


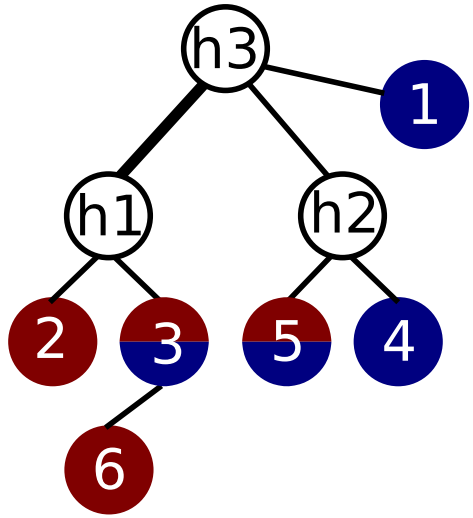


- "Divide-and-conquer" strategy
  - Find local groups: MST
  - Learn over small groups
  - Merge into global solution
- Local method w. global consistency guarantee

Latent Tree Model Learning - Structure

Structure & Parameter Learning: Divide and Conquer

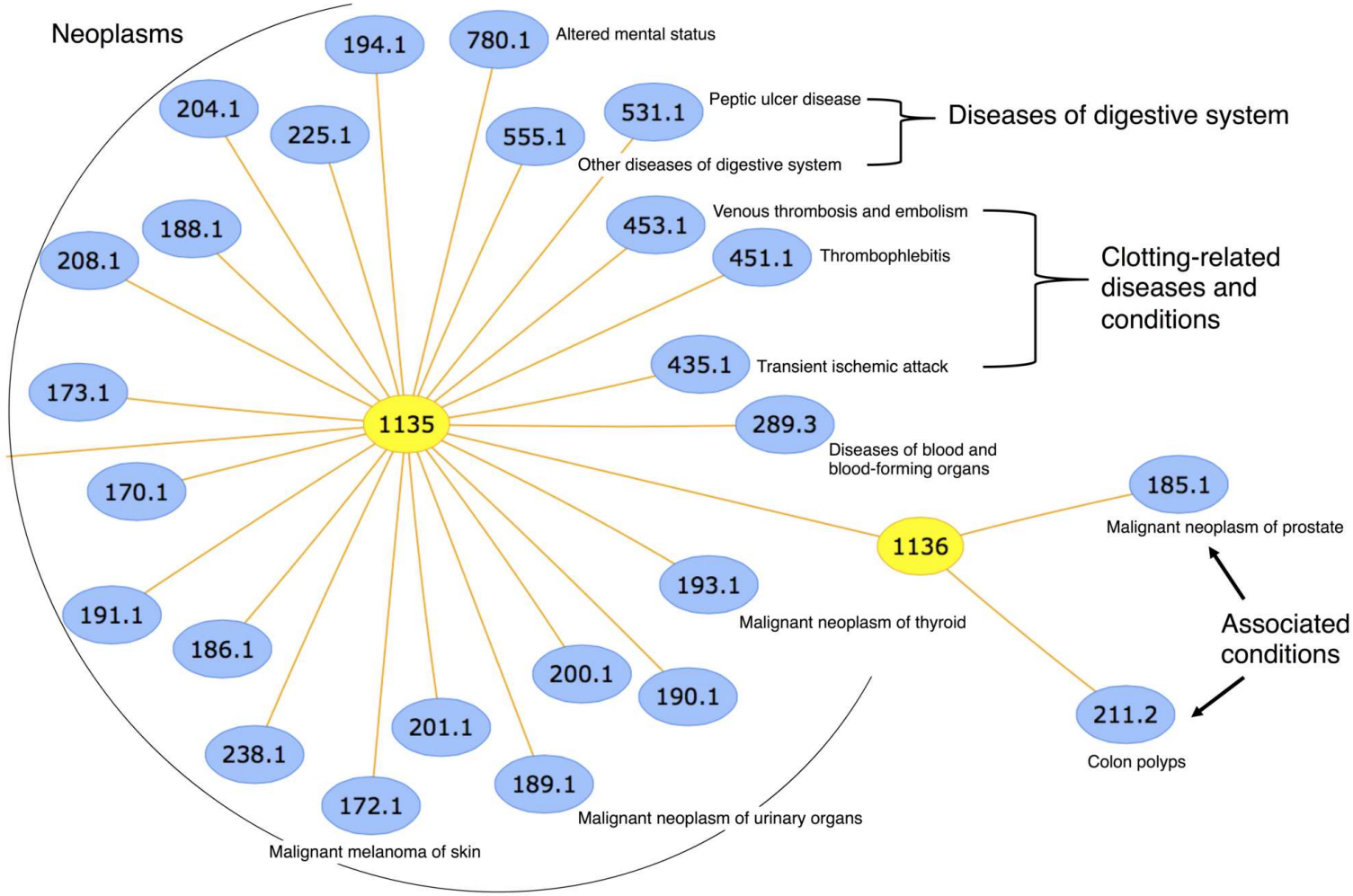




- "Divide-and-conquer" strategy
  - Find local groups: MST
  - Learn over small groups
  - Merge into global solution
- Local method w. global consistency guarantee
- Bulk asynchronous parallel
  - $\log$(# variables); linear(dimension)

Latent Tree for Discovering Human Disease Hierarchy

CMS: 1.6 million patients, 168 million diagnostic events, 11 k diseases.



" Scalable Latent TreeModel and its Application to Health Analytics " by H, N. U.Niranjan, I. Perros, R. Chen, J. Sun, A. Anandkumar, NIPS 2015 MLHC workshop.

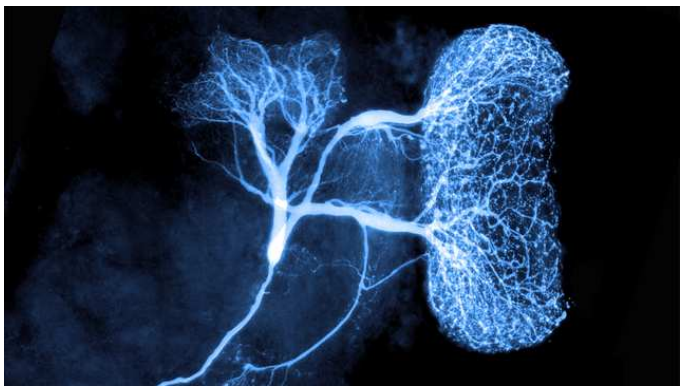
Outline

1. Introduction
2. Why Tensors?
3. Algorithms for Tensor Decomposition
4. Learning Probabilistic Models
5. Learning Representations
6. Conclusion

Overcomplete Dictionary learning

Sparse coding prevalent in neural signaling.

Neural sparse coding [Papadopoulou11]



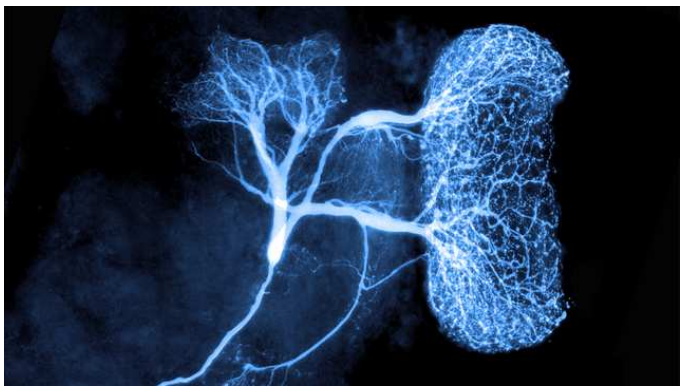
A. Agarwal, A, P. Jain, P. Netrapalli, "Learning Sparsely Used Overcomplete Dictionaries," COLT 2014.
A, M. Janzamin, R. Ge, "Overcomplete Tensor Decomposition, " COLT 2015.

Huang, A., "Convolutional Dictionary Learning through Tensor Factorization", Proc. of JMLR 2015.

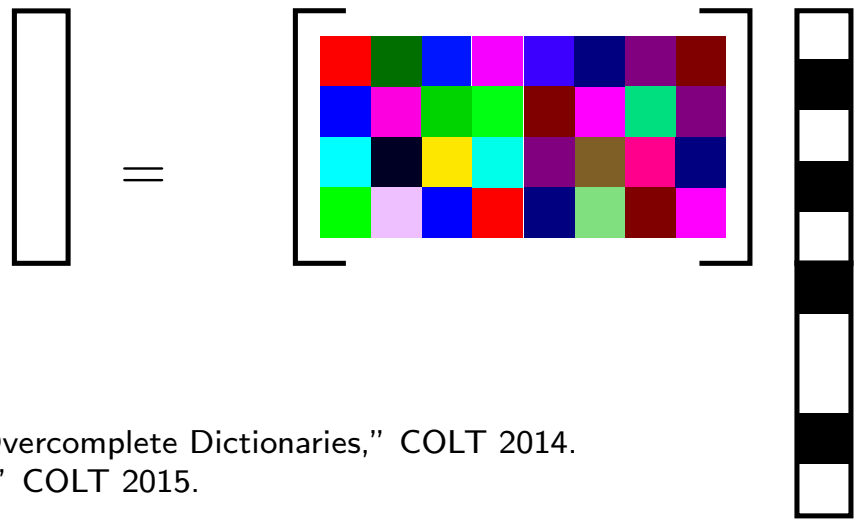
Overcomplete Dictionary learning

Sparse coding prevalent in neural signaling.

Neural sparse coding [Papadopoulou11]



Linear Model with Overcomplete Dictionary



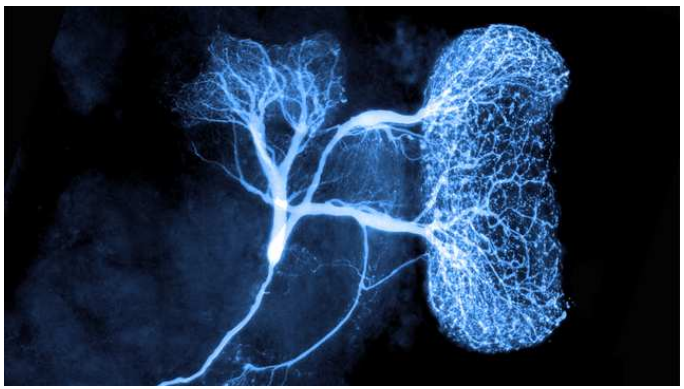
A. Agarwal, A, P. Jain, P. Netrapalli, "Learning Sparsely Used Overcomplete Dictionaries," COLT 2014.
A, M. Janzamin, R. Ge, "Overcomplete Tensor Decomposition, " COLT 2015.

Huang, A., "Convolutional Dictionary Learning through Tensor Factorization", Proc. of JMLR 2015.

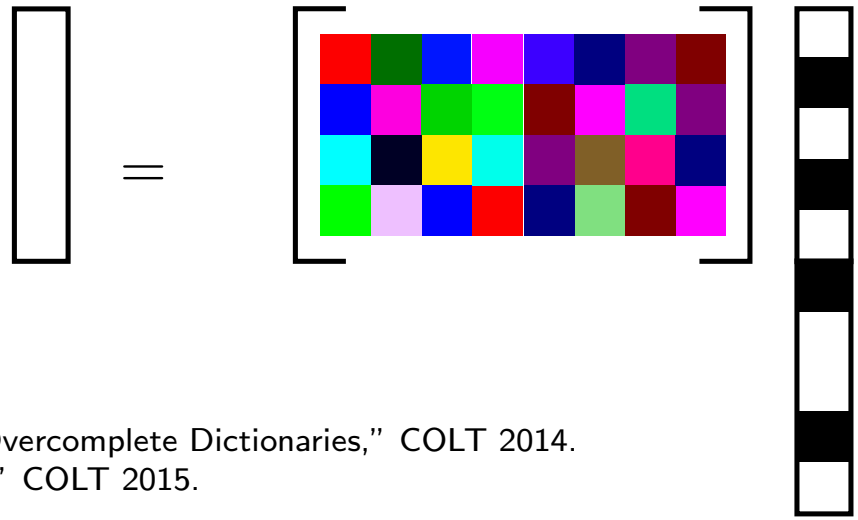
Overcomplete Dictionary learning

Contribution: learn overcomplete incoherent dictionaries

Neural sparse coding [Papadopoulou11]



Linear Model with Overcomplete Dictionary



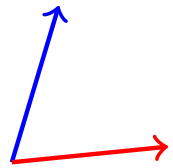
A. Agarwal, A, P. Jain, P. Netrapalli, "Learning Sparsely Used Overcomplete Dictionaries," COLT 2014.
A, M. Janzamin, R. Ge, "Overcomplete Tensor Decomposition, " COLT 2015.

Huang, A., "Convolutional Dictionary Learning through Tensor Factorization", Proc. of JMLR 2015.

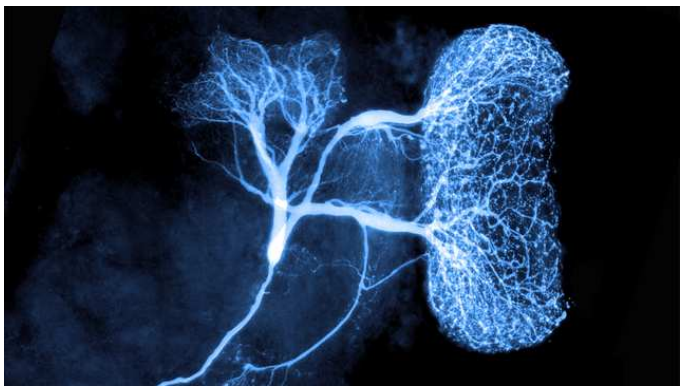
Overcomplete Dictionary learning

Contribution: learn overcomplete incoherent dictionaries

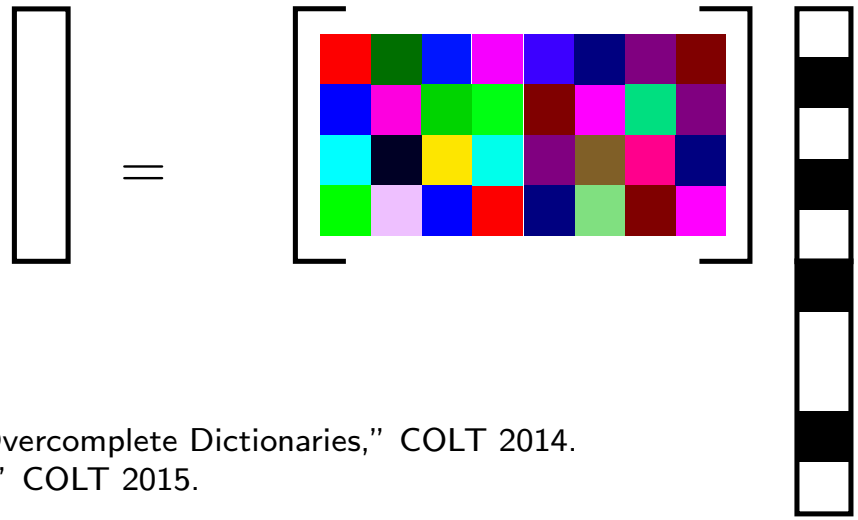
- Incoherent dictionary: rules out redundancy.
- Blessing of high dimension: exponential no. of vectors!



Neural sparse coding [Papadopoulou11]



Linear Model with Overcomplete Dictionary



A. Agarwal, A, P. Jain, P. Netrapalli, "Learning Sparsely Used Overcomplete Dictionaries," COLT 2014.
A, M. Janzamin, R. Ge, "Overcomplete Tensor Decomposition, " COLT 2015.

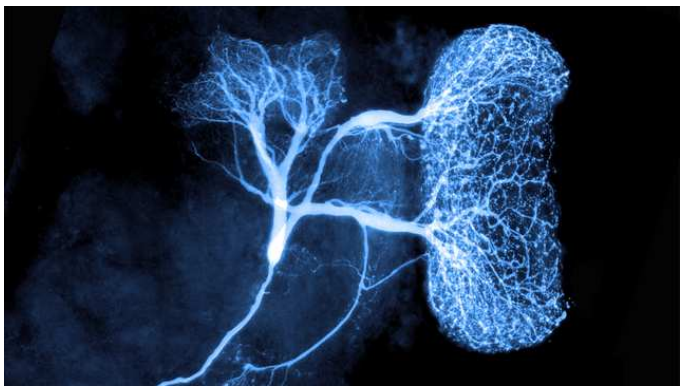
Huang, A., "Convolutional Dictionary Learning through Tensor Factorization", Proc. of JMLR 2015.

Overcomplete Dictionary learning

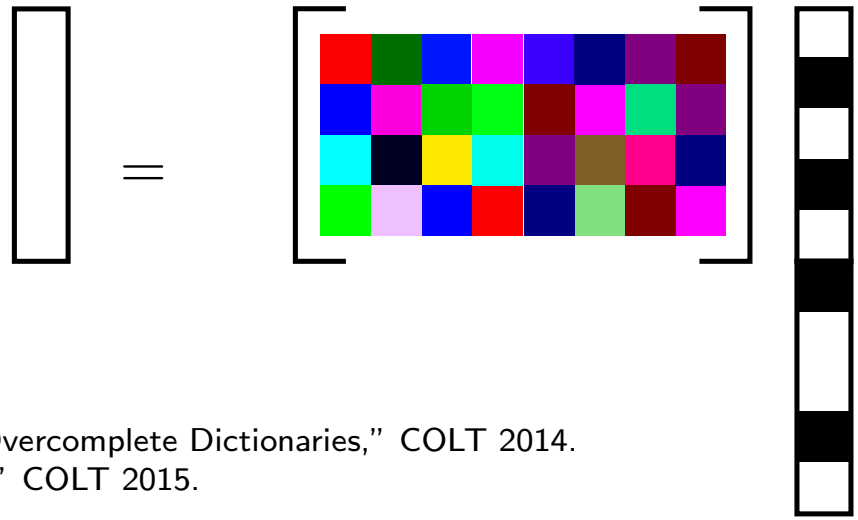
Contribution: learn overcomplete incoherent dictionaries

- Power method for overcomplete tensor decomposition.
- Initialization algorithms for alternating minimization.

Neural sparse coding [Papadopoulou11]



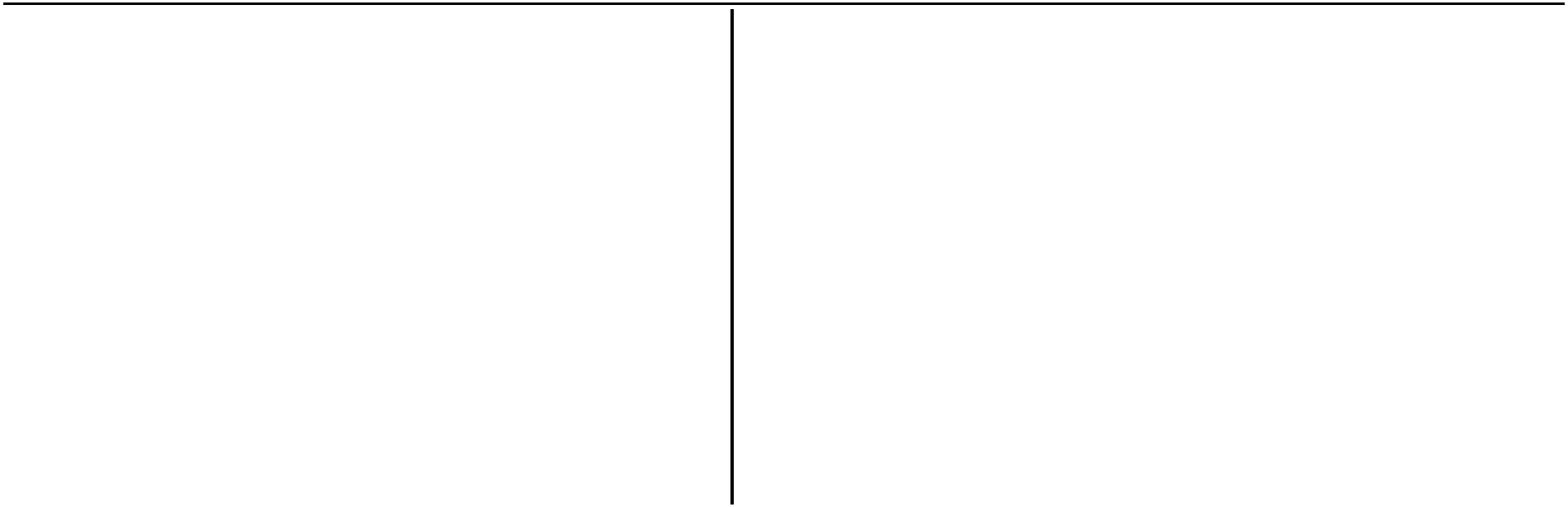
Linear Model with Overcomplete Dictionary



A. Agarwal, A, P. Jain, P. Netrapalli, "Learning Sparsely Used Overcomplete Dictionaries," COLT 2014.
A, M. Janzamin, R. Ge, "Overcomplete Tensor Decomposition, " COLT 2015.

Huang, A., "Convolutional Dictionary Learning through Tensor Factorization", Proc. of JMLR 2015.

Convolutional Dictionary learning

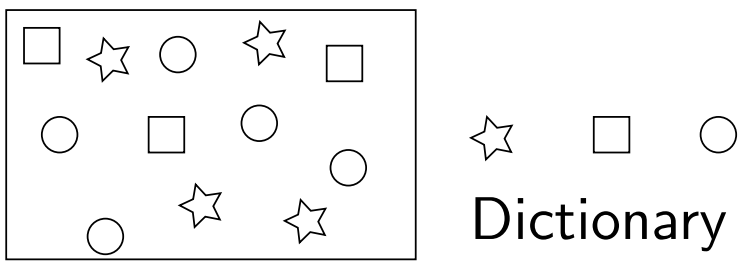


A. Agarwal, A, P. Jain, P. Netrapalli, "Learning Sparsely Used Overcomplete Dictionaries," COLT 2014.
A, M. Janzamin, R. Ge, "Overcomplete Tensor Decomposition, " COLT 2015.

Huang, A., "Convolutional Dictionary Learning through Tensor Factorization", Proc. of JMLR 2015.

Convolutional Dictionary learning

Shift-invariant Dictionary



Image

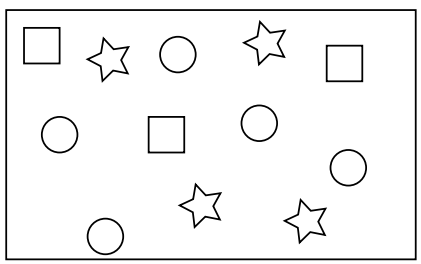
Dictionary

A. Agarwal, A, P. Jain, P. Netrapalli, "Learning Sparsely Used Overcomplete Dictionaries," COLT 2014.
A, M. Janzamin, R. Ge, "Overcomplete Tensor Decomposition, " COLT 2015.

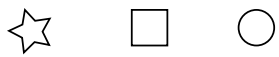
Huang, A., "Convolutional Dictionary Learning through Tensor Factorization", Proc. of JMLR 2015.

Convolutional Dictionary learning

Shift-invariant Dictionary

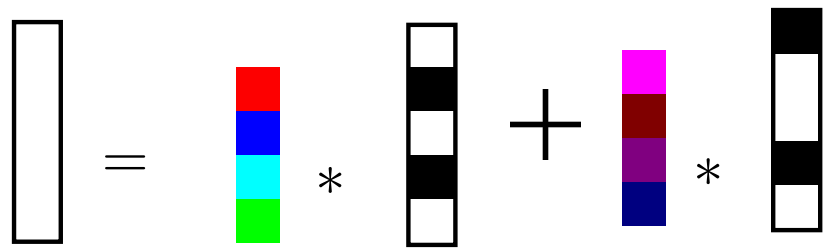


Image



Dictionary

Convolutional Model

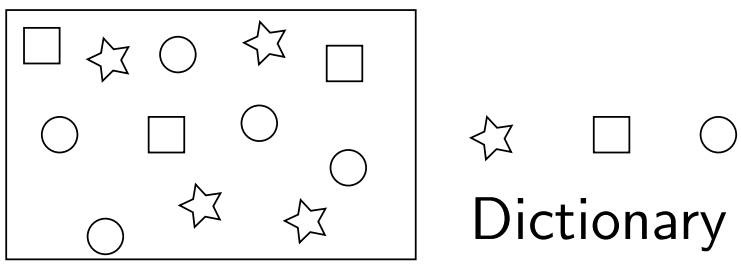


A. Agarwal, A, P. Jain, P. Netrapalli, "Learning Sparsely Used Overcomplete Dictionaries," COLT 2014.
A, M. Janzamin, R. Ge, "Overcomplete Tensor Decomposition, " COLT 2015.

Huang, A., "Convolutional Dictionary Learning through Tensor Factorization", Proc. of JMLR 2015.

Convolutional Dictionary learning

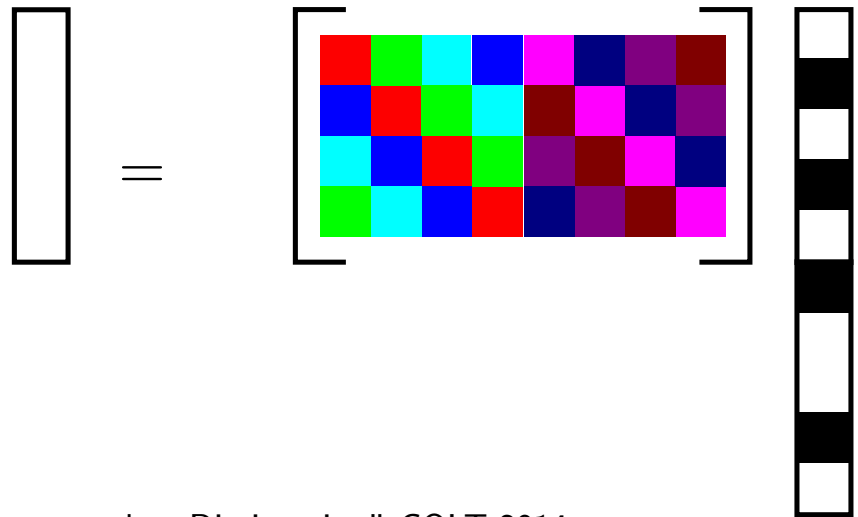
Shift-invariant Dictionary



Image

Dictionary

Linear Model with Constraints

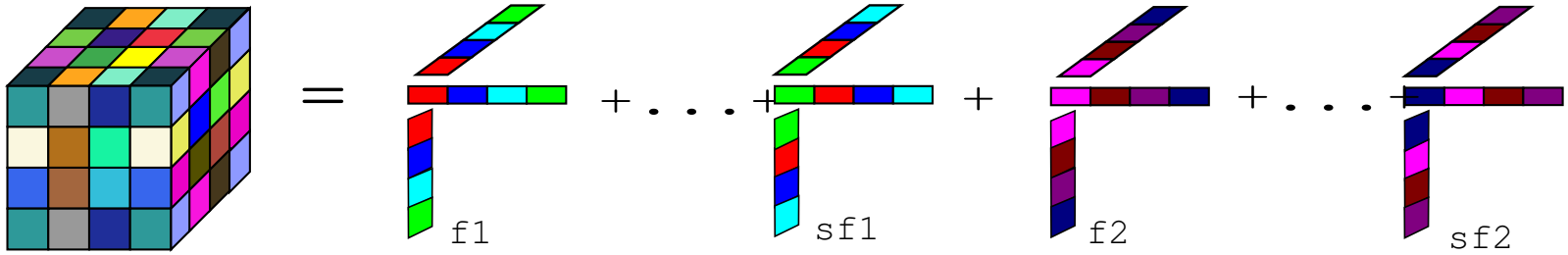


A. Agarwal, A, P. Jain, P. Netrapalli, "Learning Sparsely Used Overcomplete Dictionaries," COLT 2014.
A, M. Janzamin, R. Ge, "Overcomplete Tensor Decomposition, " COLT 2015.

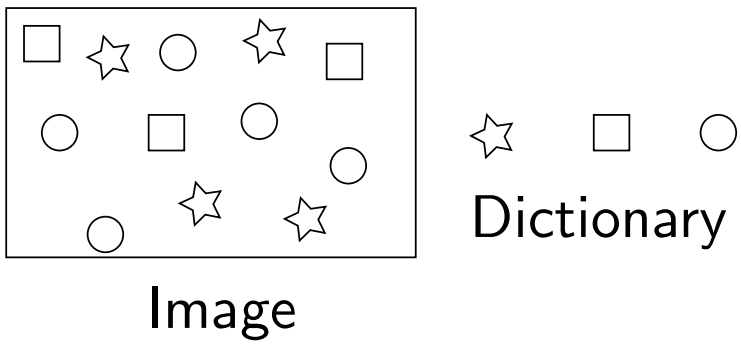
Huang, A., "Convolutional Dictionary Learning through Tensor Factorization", Proc. of JMLR 2015.

Convolutional Dictionary learning

Efficient Tensor Decomposition with Shifted Components



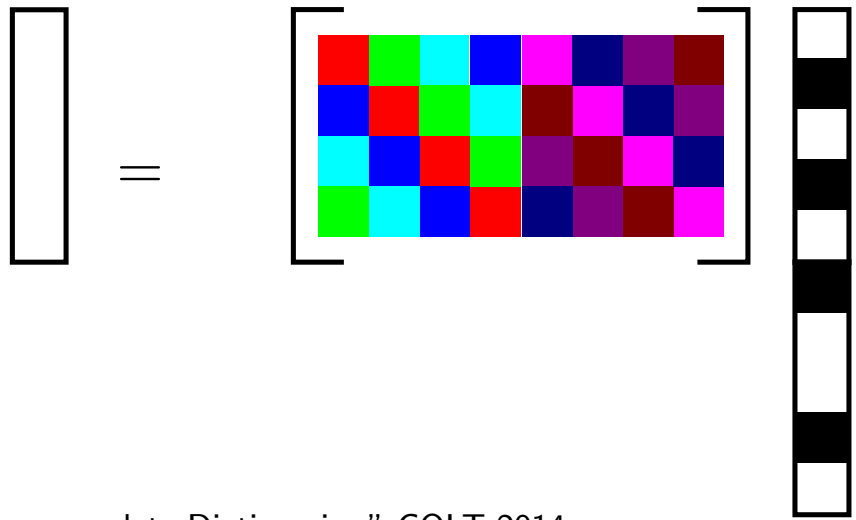
Shift-invariant Dictionary



Image

Dictionary

Linear Model with Constraints



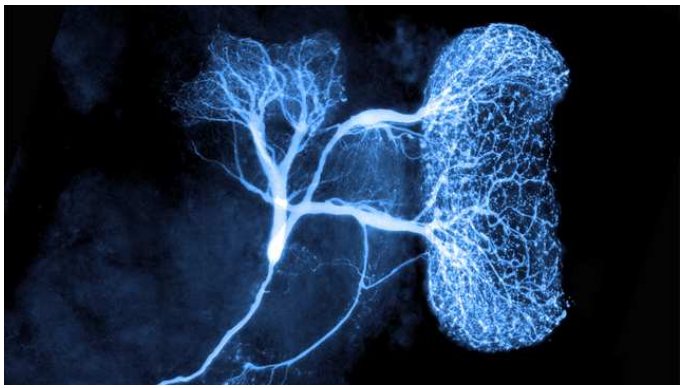
A. Agarwal, A, P. Jain, P. Netrapalli, "Learning Sparsely Used Overcomplete Dictionaries," COLT 2014.
A, M. Janzamin, R. Ge, "Overcomplete Tensor Decomposition, " COLT 2015.

Huang, A., "Convolutional Dictionary Learning through Tensor Factorization", Proc. of JMLR 2015.

Learning Representations

Sparse coding prevalent in neural signaling.

Neural sparse coding [Papadopoulou11]



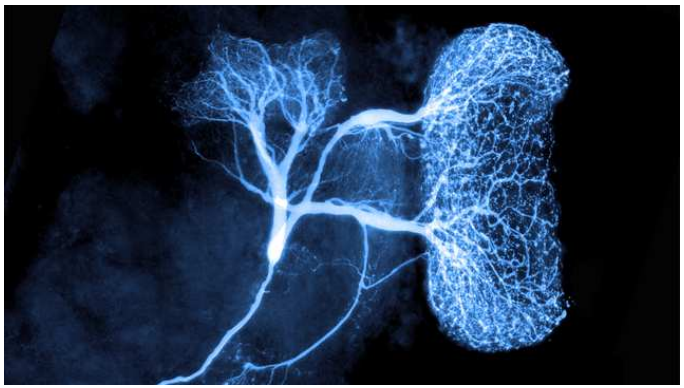
A. Agarwal, A, P. Jain, P. Netrapalli, "Learning Sparsely Used Overcomplete Dictionaries," COLT 2014.

A, M. Janzamin, R. Ge, "Overcomplete Tensor Decomposition, " COLT 2015.

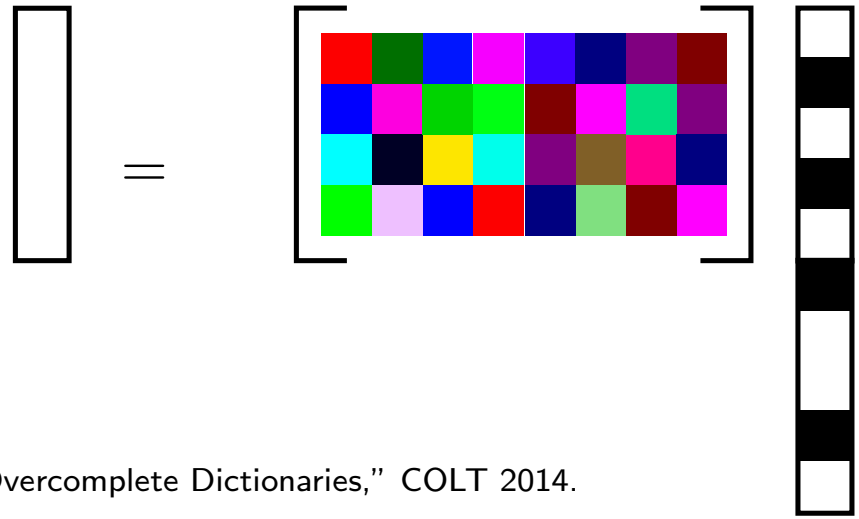
Learning Representations

Sparse coding prevalent in neural signaling.

Neural sparse coding [Papadopoulou11]



Linear Model with Overcomplete Dictionary



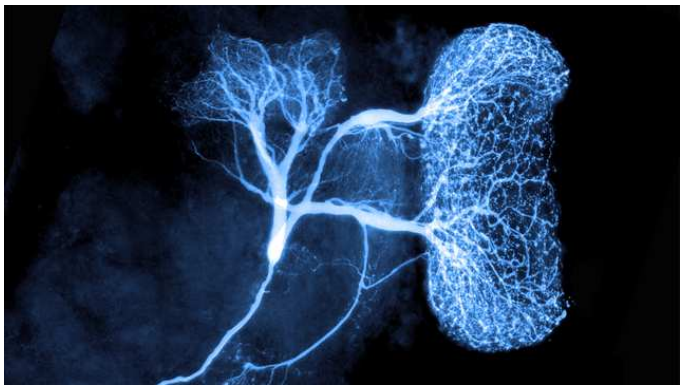
A. Agarwal, A, P. Jain, P. Netrapalli, "Learning Sparsely Used Overcomplete Dictionaries," COLT 2014.

A, M. Janzamin, R. Ge, "Overcomplete Tensor Decomposition, " COLT 2015.

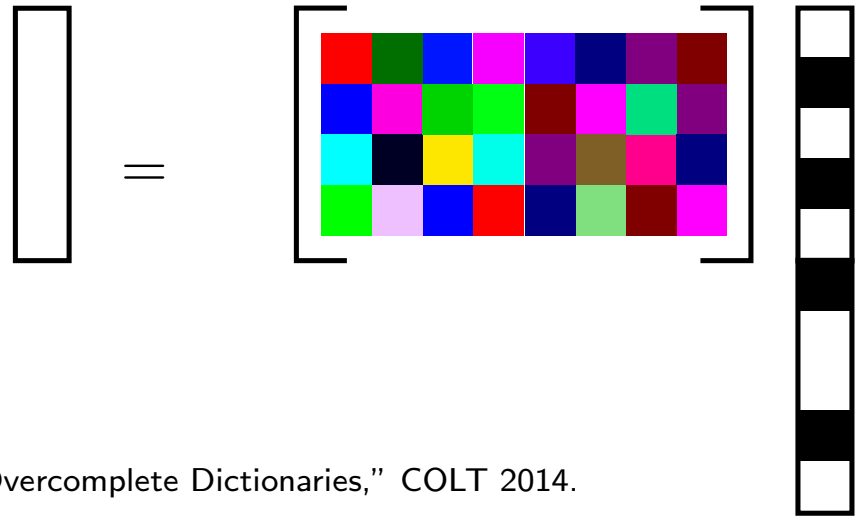
Learning Representations

Contribution: learn overcomplete incoherent dictionaries

Neural sparse coding
[Papadopoulou11]



Linear Model with
Overcomplete Dictionary



A. Agarwal, A, P. Jain, P. Netrapalli, "Learning Sparsely Used Overcomplete Dictionaries," COLT 2014.

A, M. Janzamin, R. Ge, "Overcomplete Tensor Decomposition, " COLT 2015.

Moment forms for Dictionary Models

$$x_i = Ah_i, \quad i \in [n].$$

Independent components analysis (ICA)

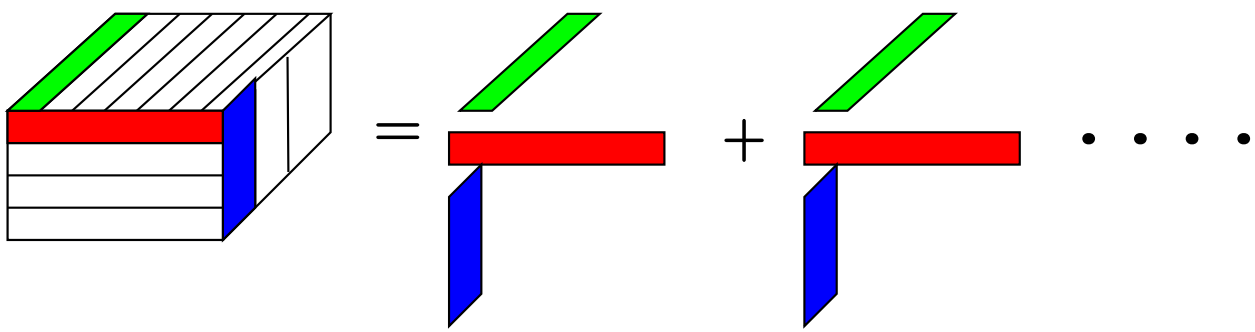
- h_i are independent, e.g. Bernoulli Gaussian

$$M_4 := \mathbb{E}[x \otimes x \otimes x \otimes x] - T, \text{ where}$$

$$T_{i_1,i_2,i_3,i_4} := \mathbb{E}[x_{i_1}x_{i_2}]\mathbb{E}[x_{i_3}x_{i_4}] + \mathbb{E}[x_{i_1}x_{i_3}]\mathbb{E}[x_{i_2}x_{i_4}] + \mathbb{E}[x_{i_1}x_{i_4}]\mathbb{E}[x_{i_2}x_{i_3}],$$

Let $\kappa_j := \mathbb{E}[h_j^4] - 3\mathbb{E}^2[h_j^2]$, $j \in [k]$. Then, we have

$$M_4 = \sum_{j \in [k]} \kappa_j a_j \otimes a_j \otimes a_j \otimes a_j.$$



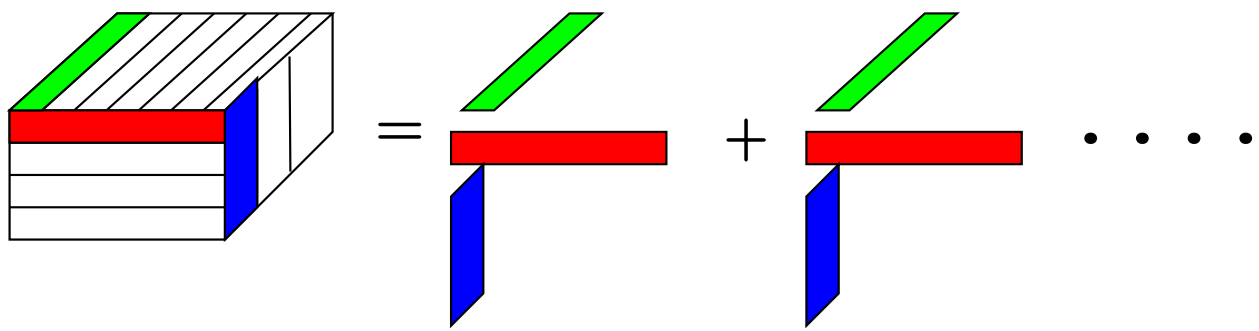
Moment forms for Dictionary Models

General (sparse) coefficients

$$x_i = Ah_i, \quad i \in [n], \quad \mathbb{E}[h_i] = s.$$

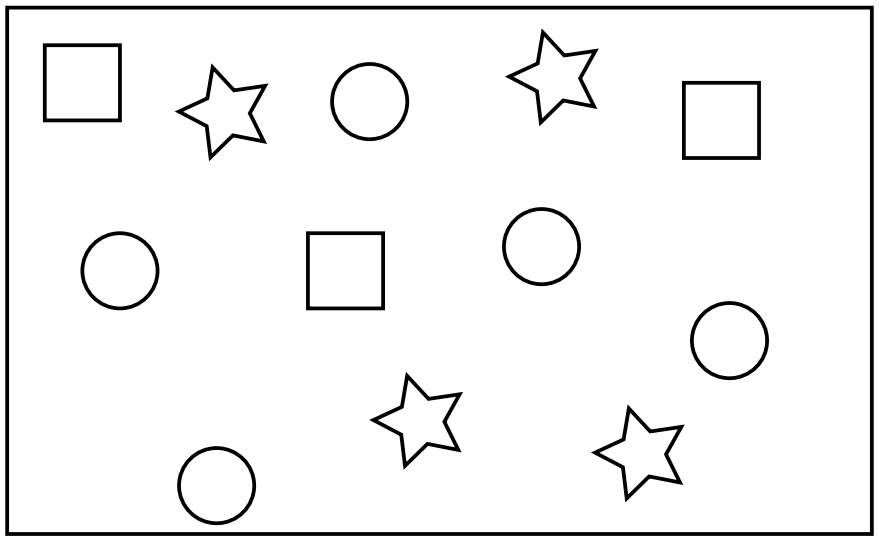
$$\begin{aligned} \mathbb{E}\left[h_i^4\right] = \mathbb{E}\left[h_i^2\right] &= \beta s/k, \\ \mathbb{E}\left[h_i^2 h_j^2\right] &\leq \tau, \quad i \neq j, \\ \mathbb{E}\left[h_i^3 h_j\right] &= 0, \quad i \neq j, \end{aligned}$$

$$\mathbb{E}[x \otimes x \otimes x \otimes x] = \sum_{j \in [k]} \kappa_j a_j \otimes a_j \otimes a_j \otimes a_j + E, \text{ where } \|E\| \leq \tau \|A\|^4.$$

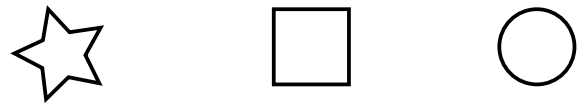


Convolutional Dictionary Model

- So far, invariances in dictionary are not incorporated.
- Convolutional models: incorporate invariances such as shift invariance.

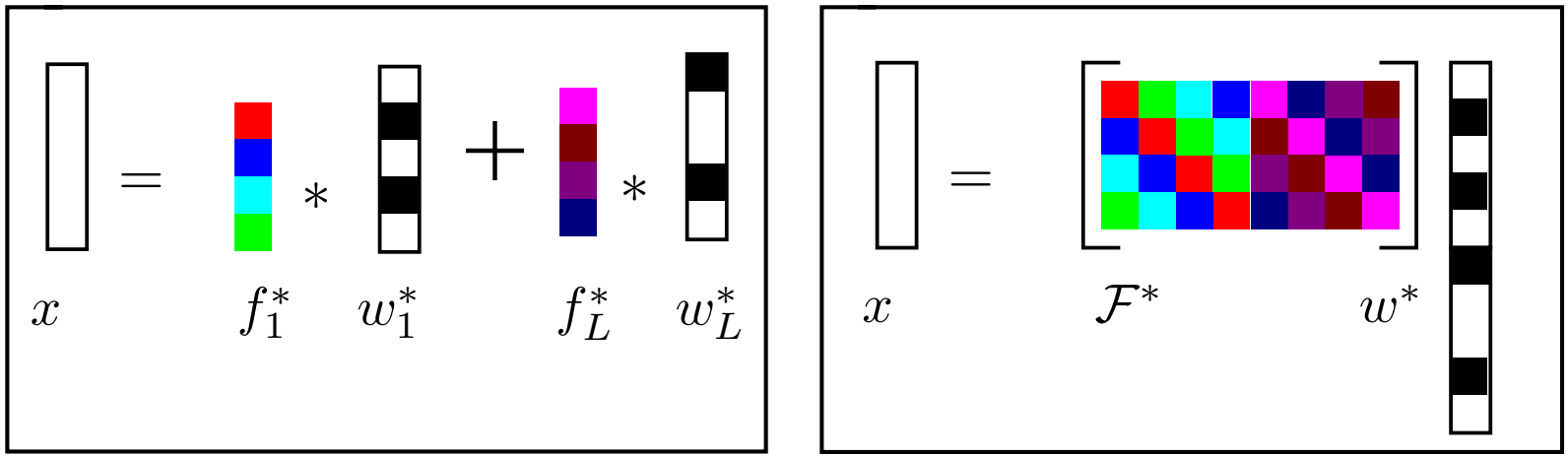


Image



Dictionary elements

Rewriting as a standard dictionary model



(a) Convolutional model **(b)** Reformulated model

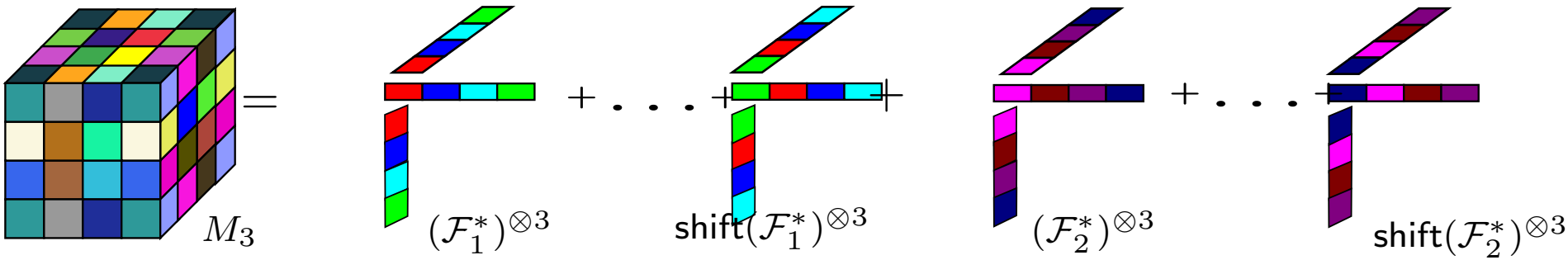
$$x = \sum_i f_i * w_i = \sum_i \mathsf{Cir}(f_i) w_i = \mathcal{F}^* w^*$$

- Circulant matrix has eigen decomposition
  $$\mathsf{Cir}(f) = U\mathsf{Diag}(\mathsf{DFT}_{\text{1-d}}(f))U^{\mathsf{H}} = U\mathsf{Diag}(\sqrt{n}U^{\mathsf{H}} \cdot f)U^{\mathsf{H}}$$
- U is the Discrete Fourier Transform Matrix.

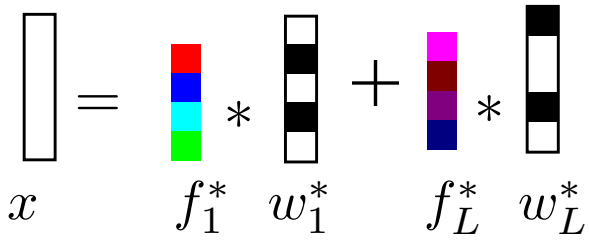
Moment forms and optimization

$$x = \sum_i f_i * w_i = \sum_i \mathsf{Cir}(f_i) w_i = \mathcal{F}^* w^*$$

- Assume coefficients w_i are independent (convolutional ICA model)
- Cumulant tensor has decomposition with components $\mathcal{F}_i^*$.



Analysis



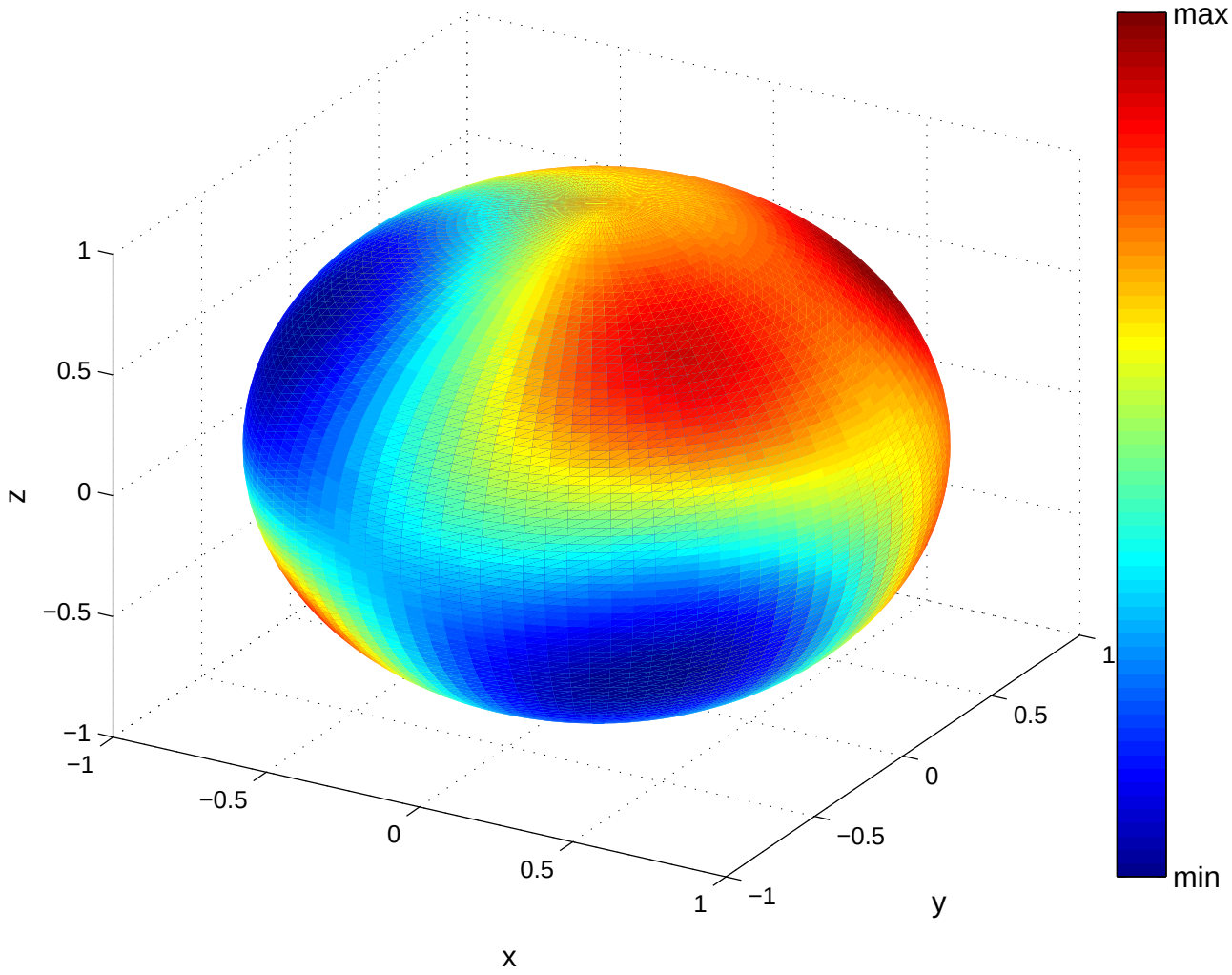
Comparison with Alternating Minimization(AM) method:

Methods	**Running Time**	**Processors**
Tensor Factorization	$O(\log(n)+\log(L))$	$O(L^2\ n^3)$
AM	$O(\max(\log(n)\log(L),\ \log(n)\log(N))$	$O(\max(\ \frac{nNL}{logN}, \frac{nNL}{logL}))$

Table: Computation complexity (L is the number of filters, n is the dimension of filters. N is the number of samples)

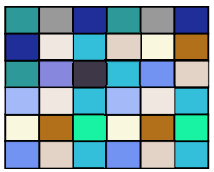
Analysis

- Non-convex optimization: guaranteed convergence to local optimum
- Local optima are shifted filters



Application: Paraphrase Detection

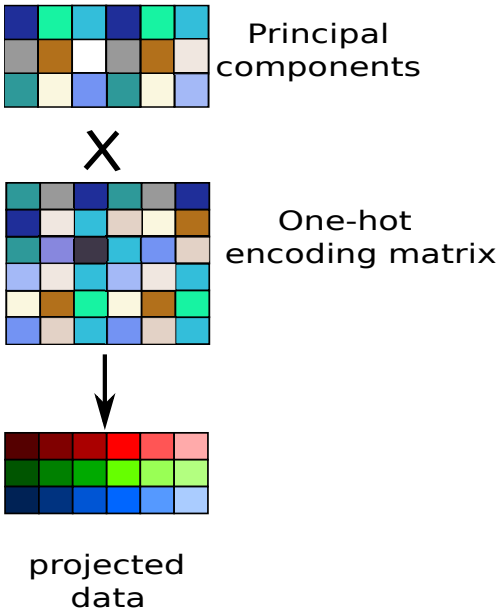
Microsoft paraphrase data: 5800 pairs of sentences



One-hot
encoding matrix

Application: Paraphrase Detection

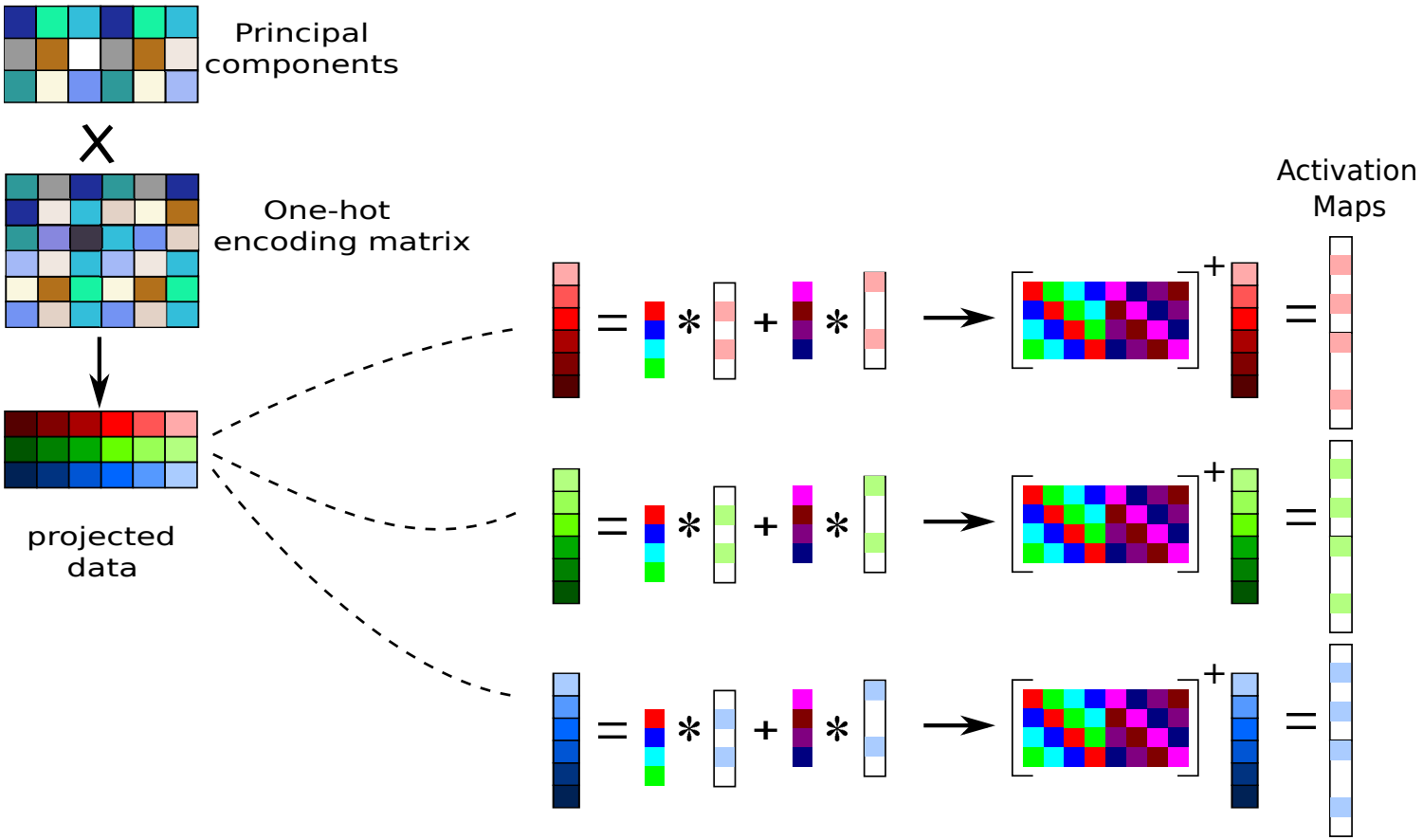
Microsoft paraphrase data: 5800 pairs of sentences



- PCA on One-hot Encoding Matrix $\rightarrow$ Subspace and Projected data

Application: Paraphrase Detection

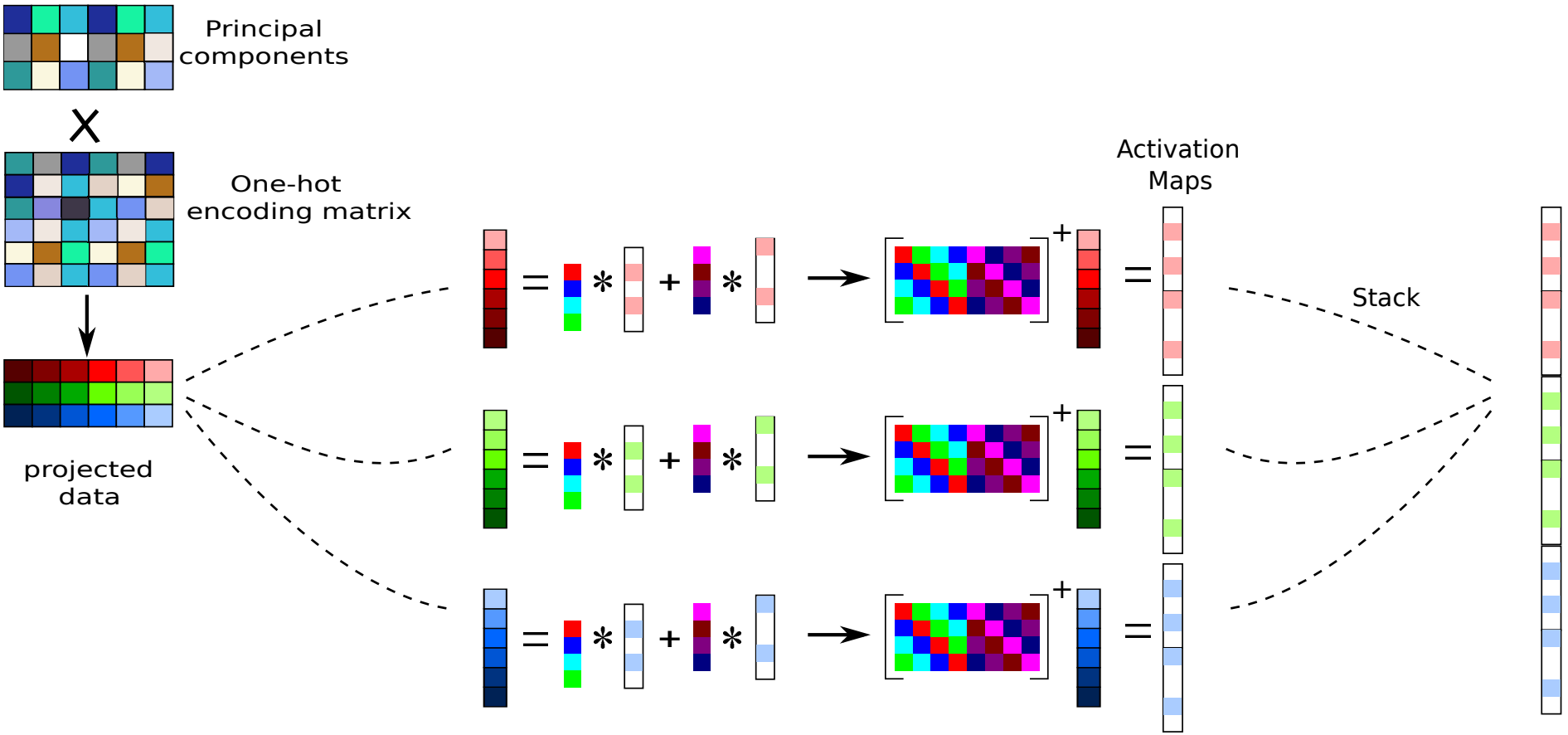
Microsoft paraphrase data: 5800 pairs of sentences



- CT on each coordinate $\rightarrow$ activation map for each coordinate

Application: Paraphrase Detection

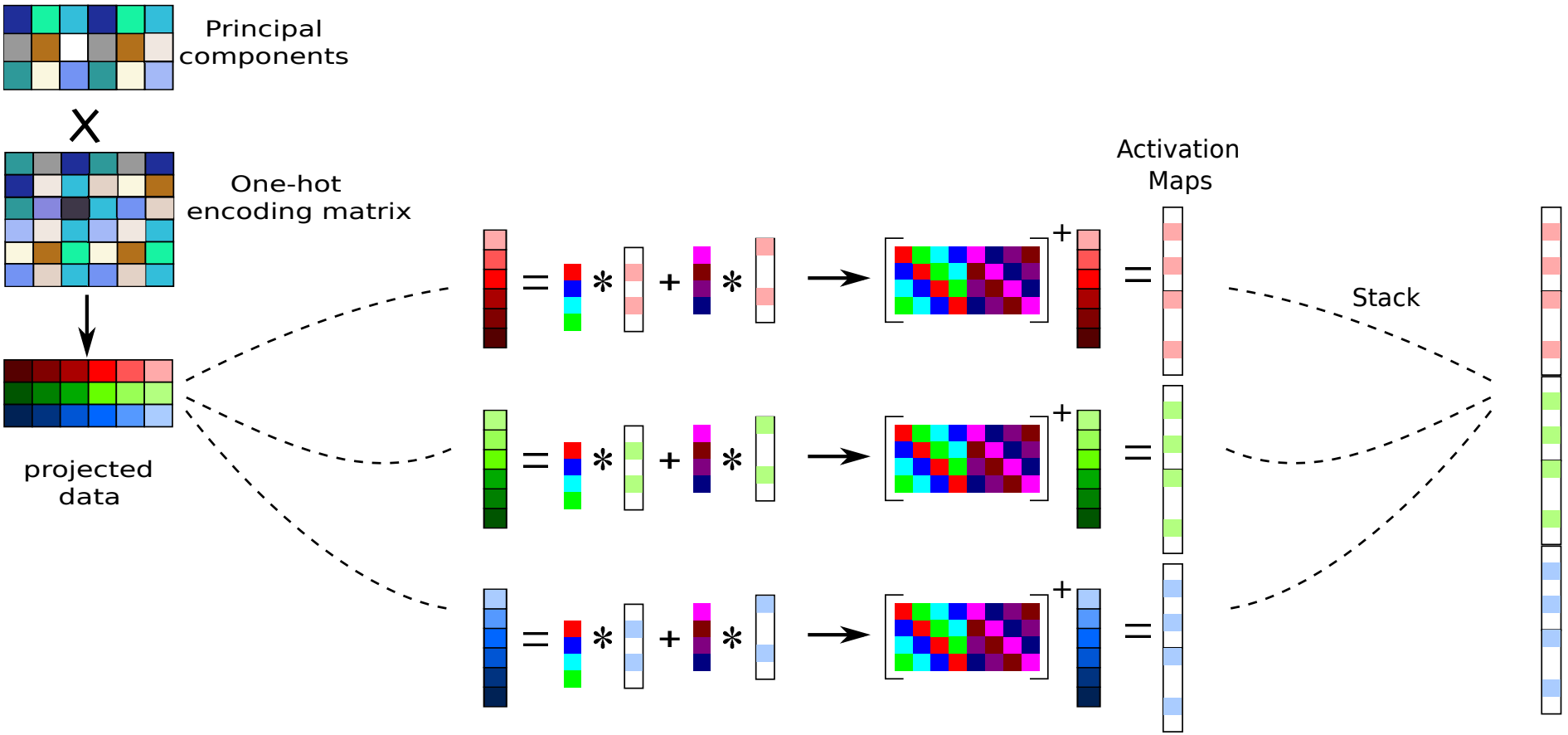
Microsoft paraphrase data: 5800 pairs of sentences



- Stack all activation maps → Sentence Embedding

Application: Paraphrase Detection

Microsoft paraphrase data: 5800 pairs of sentences



- Detects from scratch (unsupervised).
- Incorporates context.

Results using Sentence Embeddings

Sentiment Analysis

Method	MR	SUBJ
MNB	79.0	93.6
Paragraph-vector	74.8	90.5
Skip-thought	75.5	92.1
ConvDic+DeconvDec	**78.9**	**92.4**

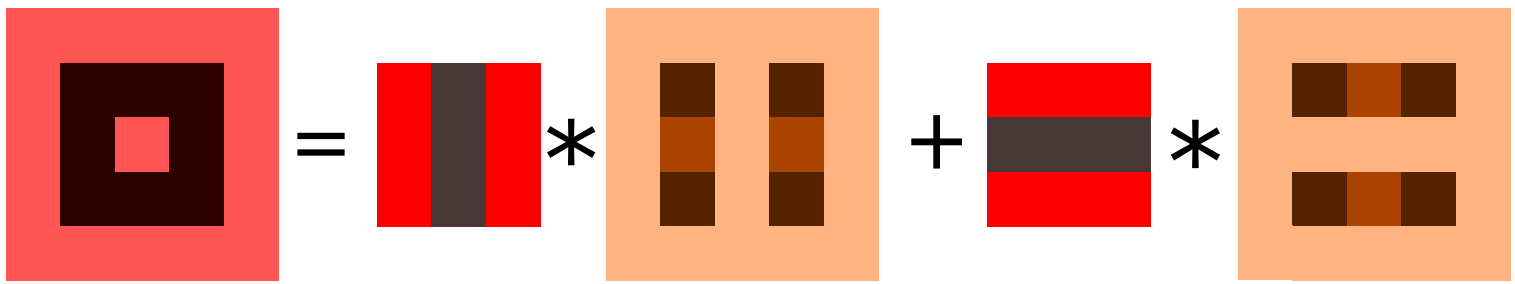
Paraphrase Detection

Method	Outside Information	F score
Vector Similarity	word similarity	75.3%
RMLMG	syntacticinfo	80.5%
ConvDic+DeconvDec	none	**80.7%**
Skip-thought	book corpus	81.9%

Image Pattern Learning through Tensor Factorization

- 2-D Convolutional Model

$$X = \sum_{j=1}^{L} V_j^* * W_j^*$$



Slides in this section prepared by Y. Shi

Image Pattern Learning through Tensor Factorization

Key points:

- Recall: 1-D circulant matrix eigen decomposition corresponds to 1-D Discrete Fourier Transform

Image Pattern Learning through Tensor Factorization

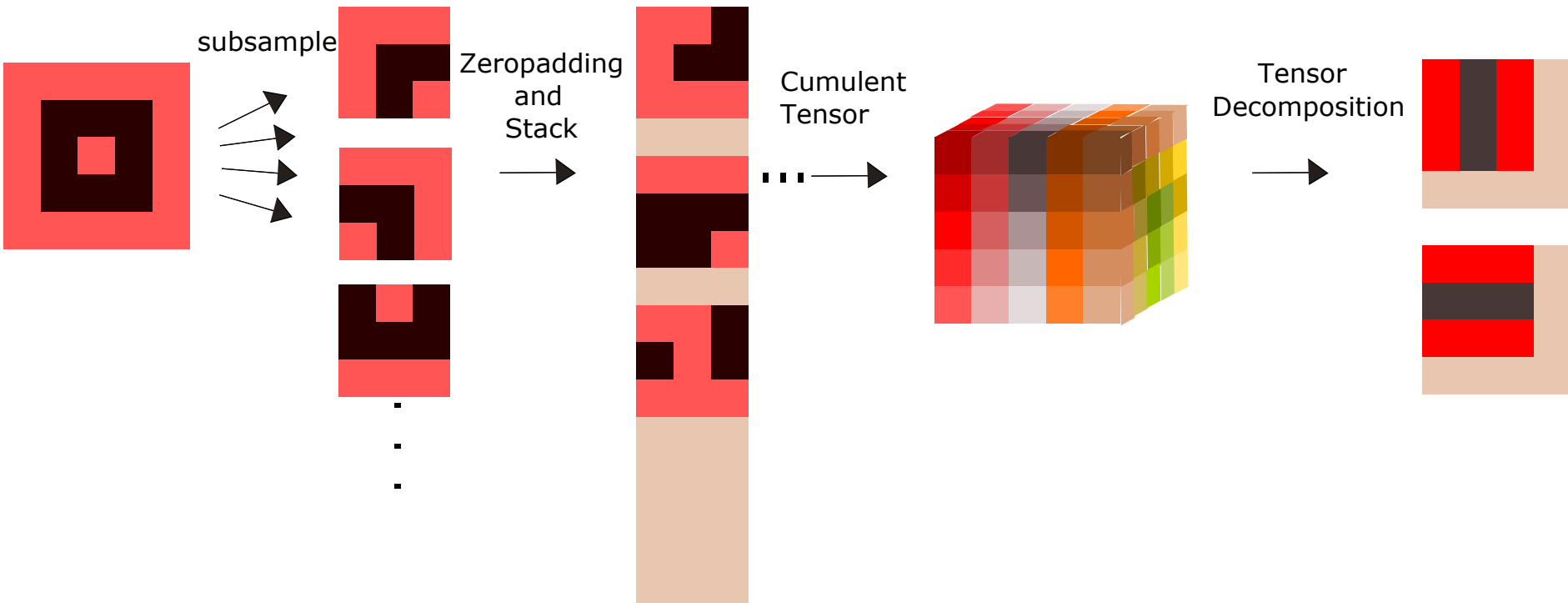
Key points:

- Recall: 1-D circulant matrix eigen decomposition corresponds to 1-D Discrete Fourier Transform
- 2-D circulant matrix eigen decomposition:
  $\mathrm{Cir}_{2\text{-d}}(V) = (U \otimes U)\mathrm{Diag}(\mathrm{DFT}_{2\text{-d}}(V))(U \otimes U)^{\mathrm{H}}$
- The eigenvector matrix for 2-D circulant matrix is $U \otimes U$.

Image Pattern Learning through Tensor Factorization

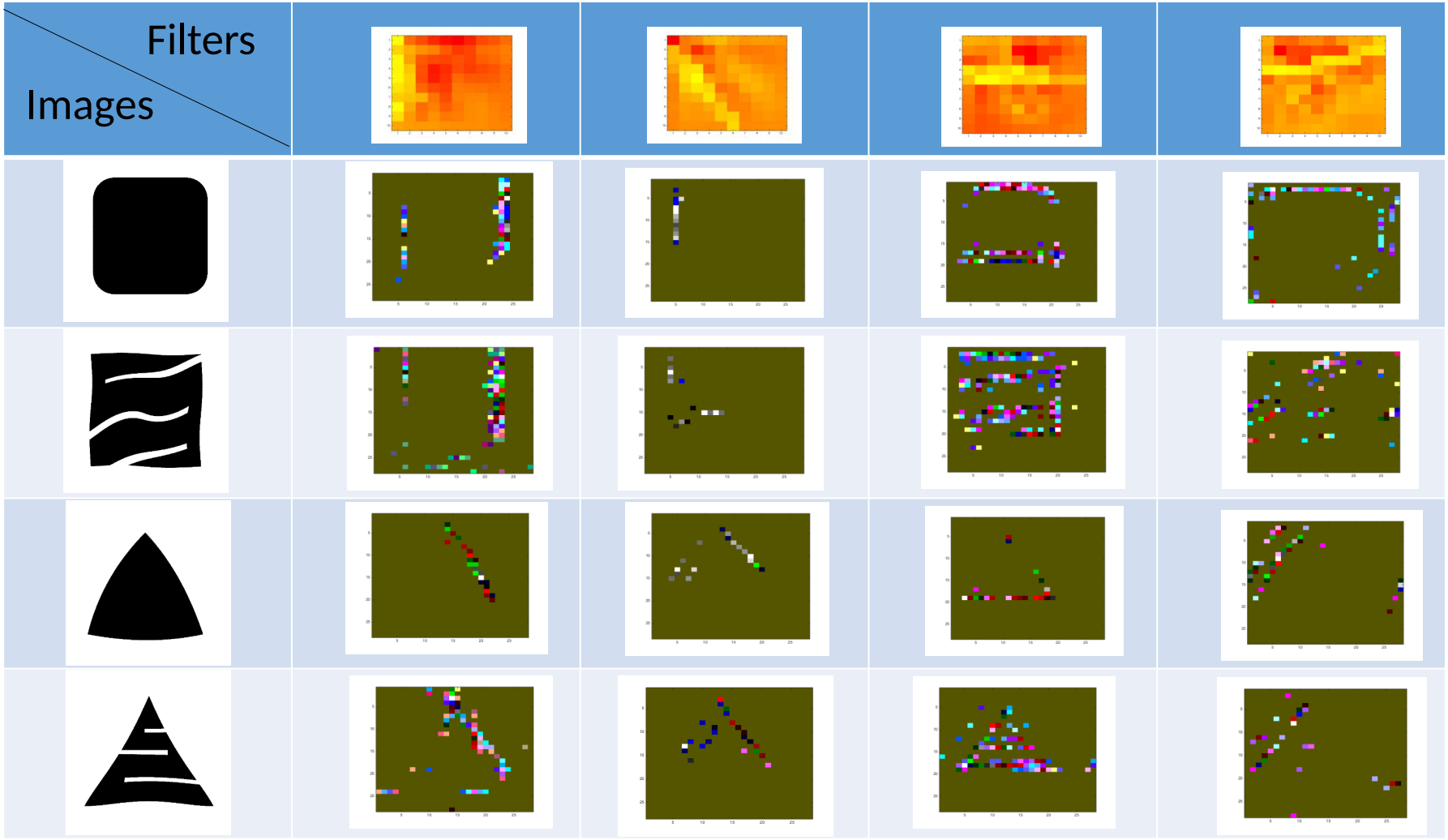
Main algorithm:

1. Subsample and batch
2. Form third order cumulant tensor
3. Tensor factorization



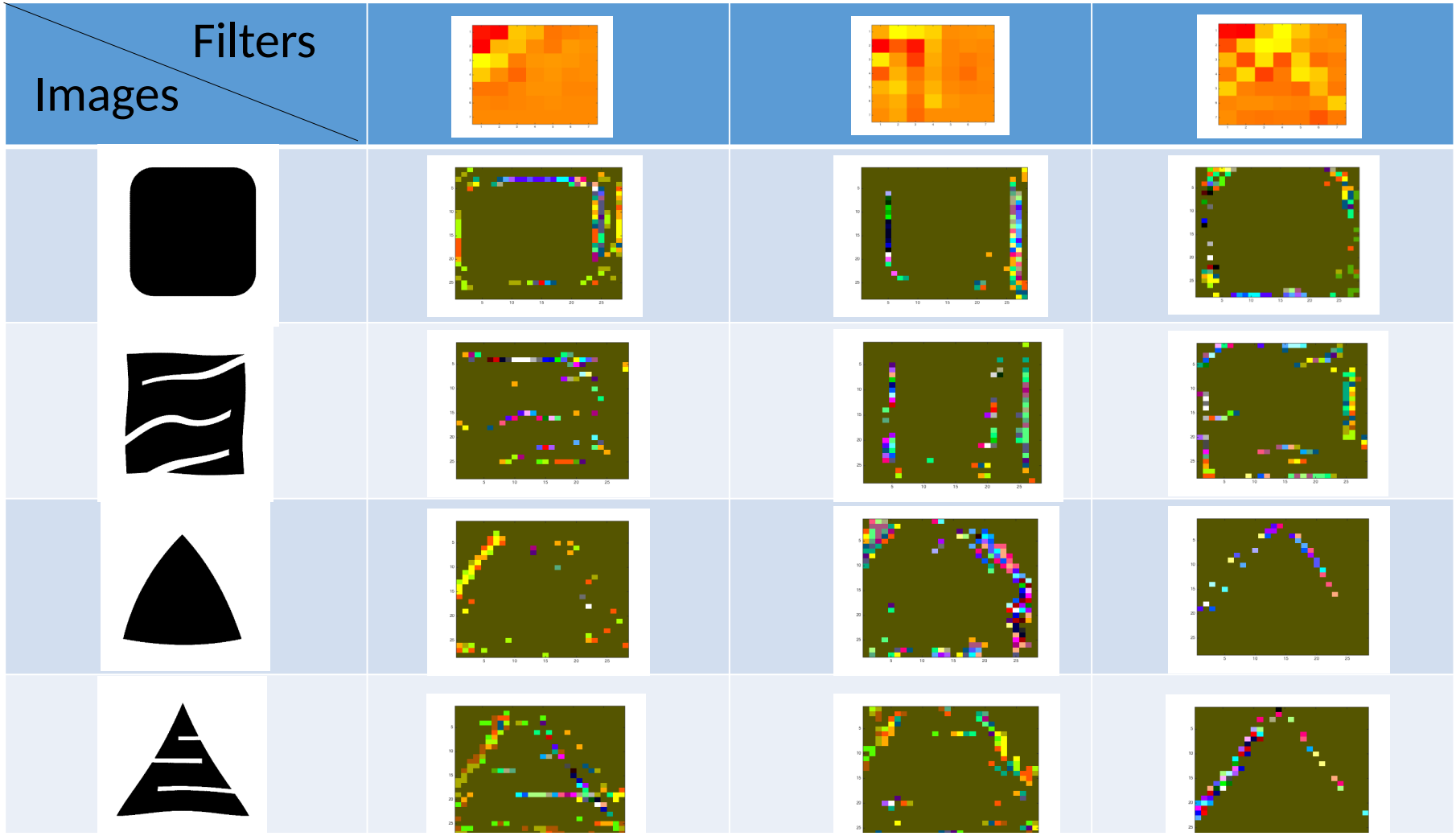
MPEG7 Dataset

- Image/activation map size: 28 X 28, filter size: 10 X 10
- First layer filters.

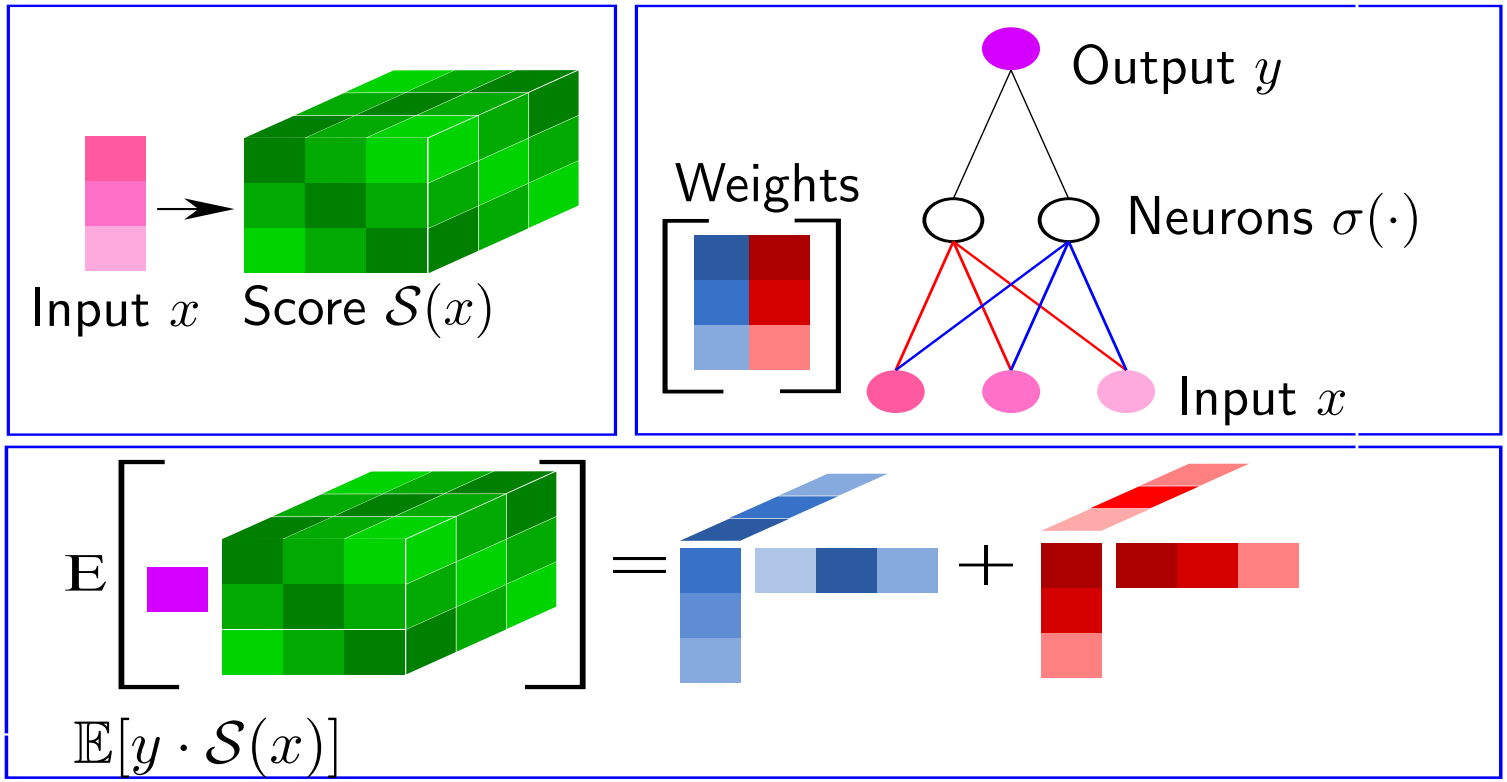


MPEG7 Dataset

- Image/activation map size: 28 X 28, filter size: 10 X 10
- Second layer filters after max-pooling.

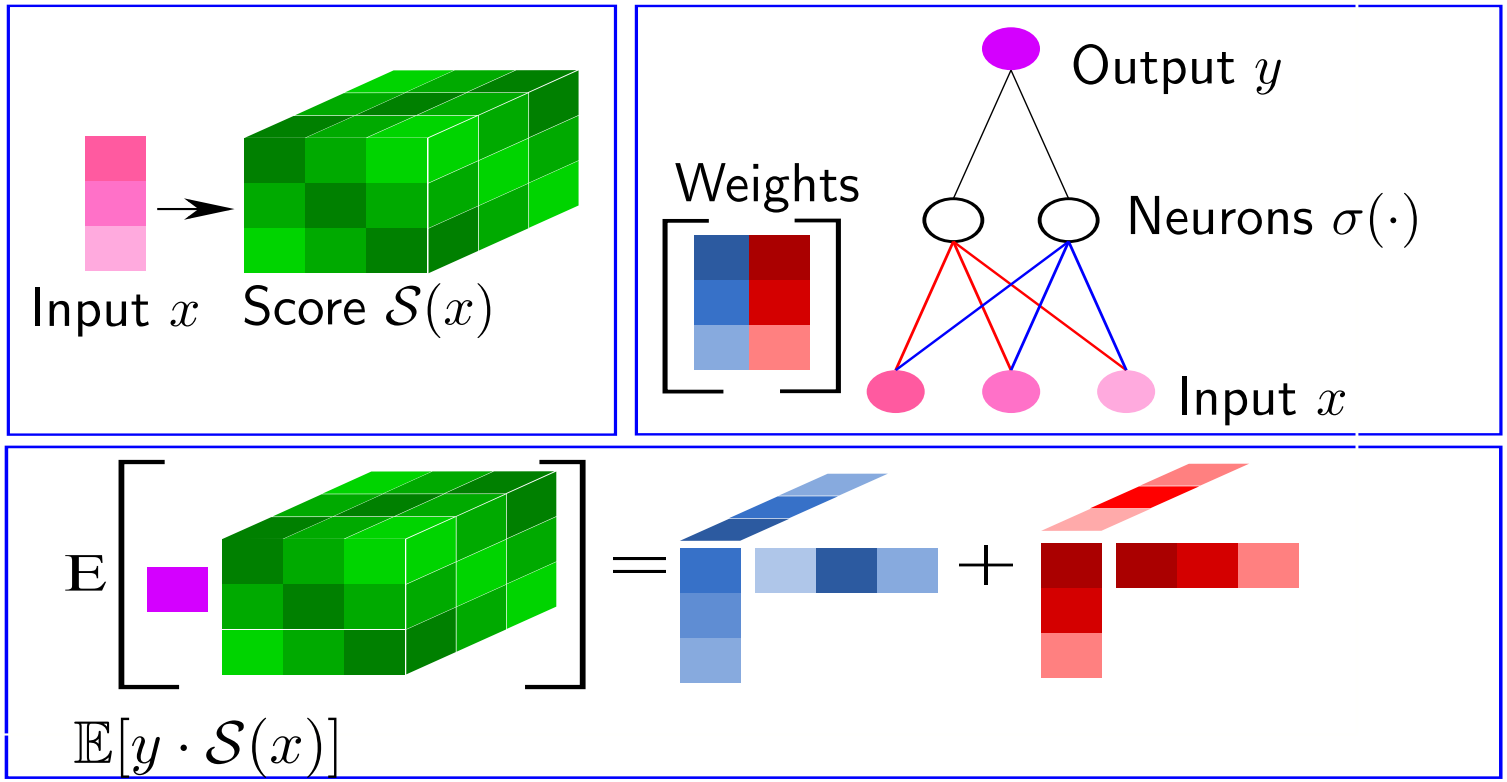


Training Neural Networks with Tensors

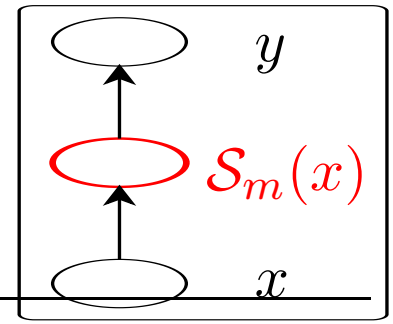


M. Janzamin, H. Sedghi, and A., "Beating the Perils of Non-Convexity: Guaranteed Training of Neural Networks using Tensor Methods," June. 2015.

Training Neural Networks with Tensors



Given input pdf $p(\cdot)$, $\mathcal{S}_m(x) := (-1)^m \frac{\nabla^{(m)} p(x)}{p(x)}$.



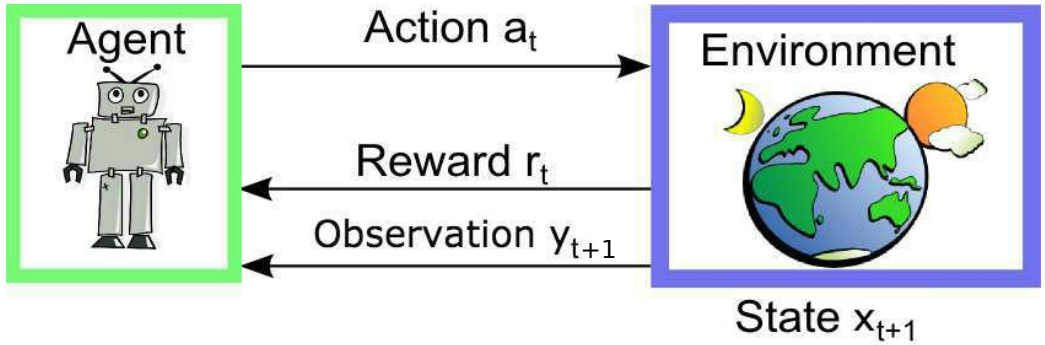
Gaussian $x \Rightarrow$ Hermite polynomials.

M. Janzamin, H. Sedghi, and A., "Beating the Perils of Non-Convexity: Guaranteed Training of Neural Networks using Tensor Methods," June. 2015.

Reinforcement Learning of POMDPs

Learning in Adaptive Environments

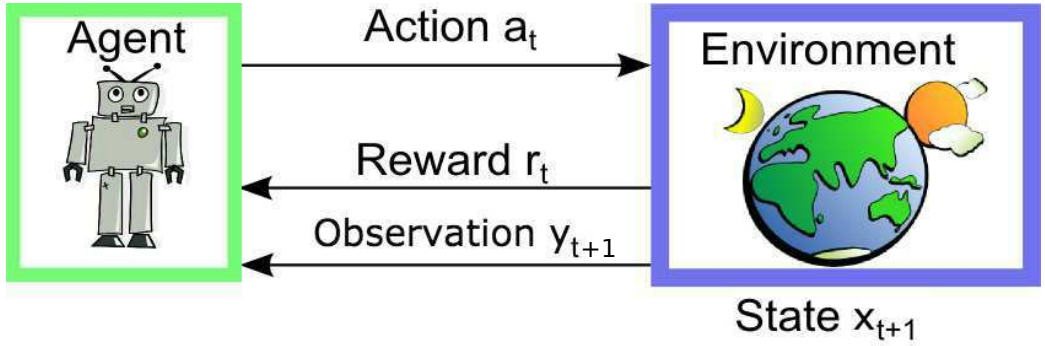
- Rewards from hidden state.
- Actions drive hidden state evolution.



Reinforcement Learning of POMDPs

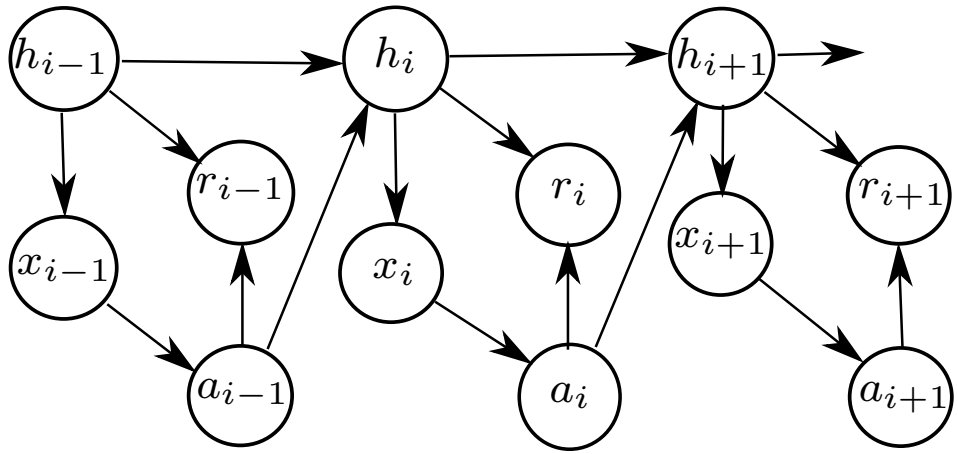
Learning in Adaptive Environments

- Rewards from hidden state.
- Actions drive hidden state evolution.



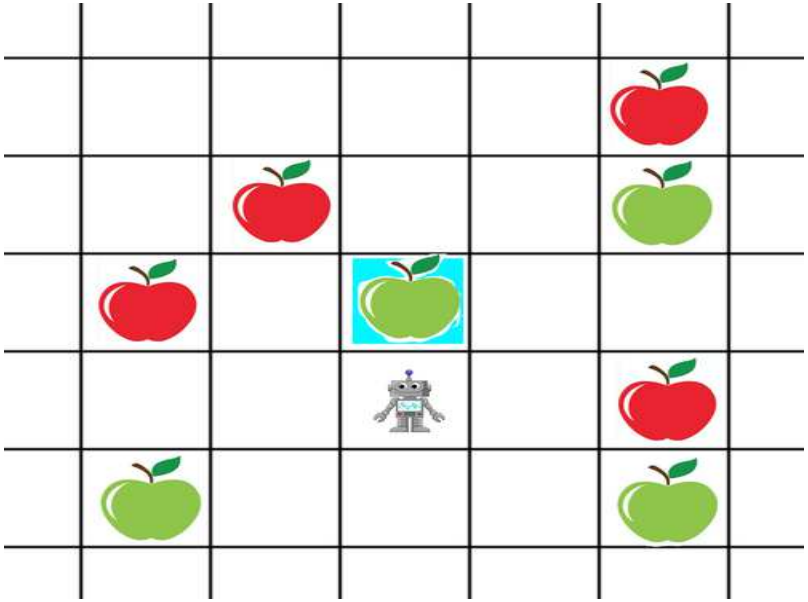
Partially Observable Markov Decision Process

Learning using tensor methods under memoryless policies

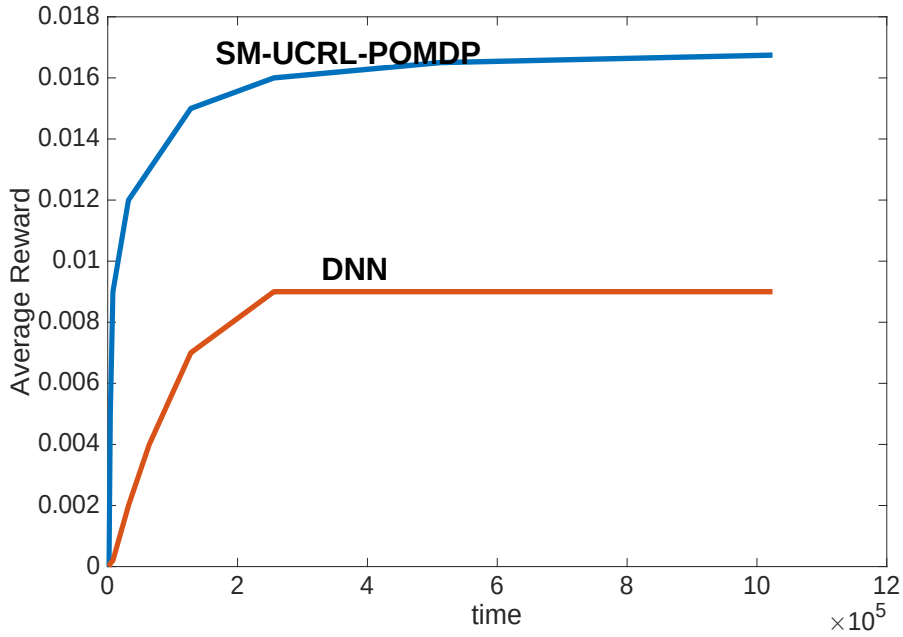


Playing Atari Game

Observation Window



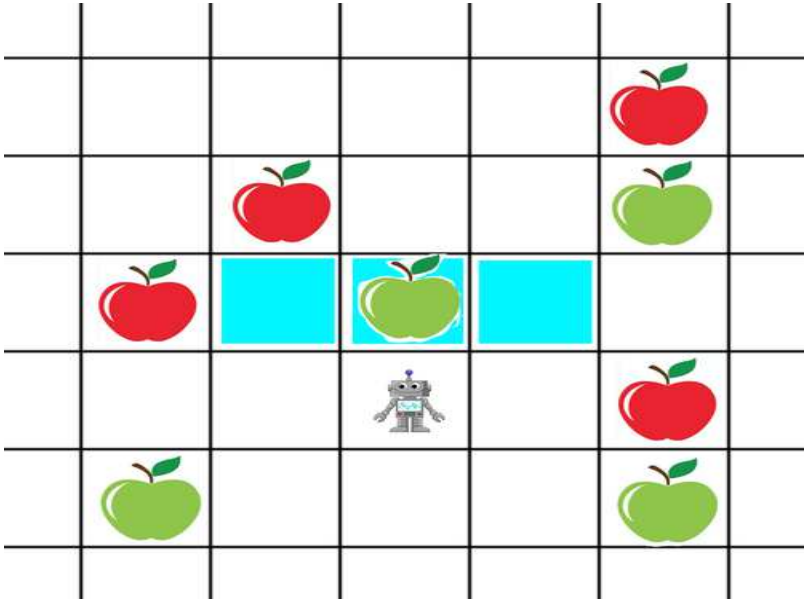
Average Reward vs. Time.



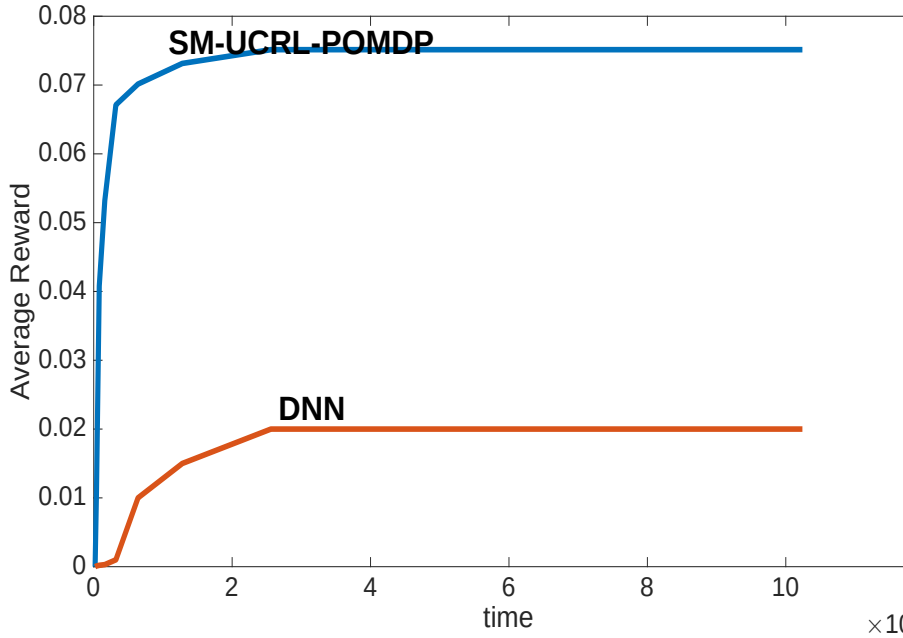
- POMDP model with 3 hidden states (trained using tensor methods) vs. NN with 3 hidden layers 10 neurons each (trained using RmsProp).

Playing Atari Game

Observation Window



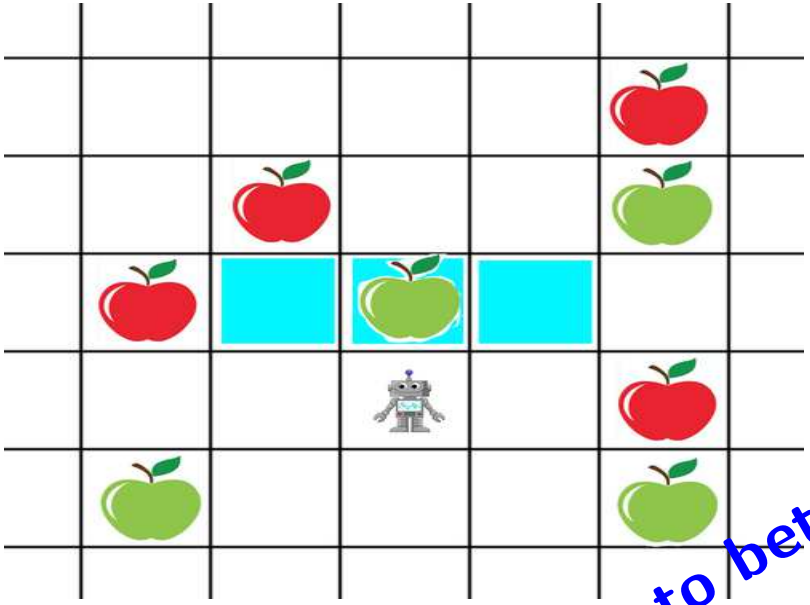
Average Reward vs. Time.



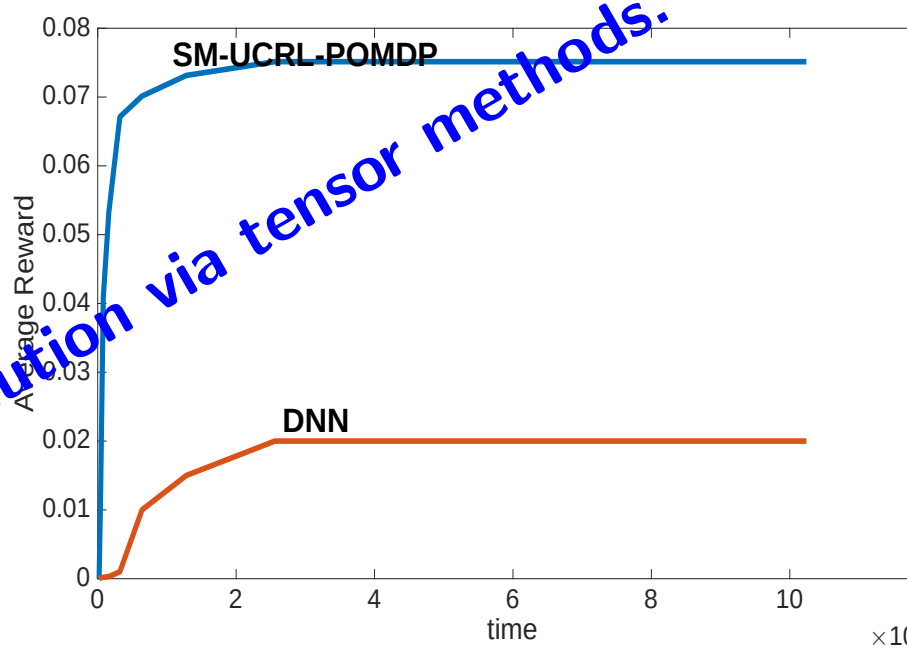
- POMDP model with 8 hidden states (trained using tensor methods) vs. NN with 3 hidden layers 30 neurons each (trained using RmsProp).

Playing Atari Game

Observation Window



Average Reward vs. Time.



Faster convergence to better solution via tensor methods.

- POMDP model with 8 hidden states (trained using tensor methods) vs. NN with 3 hidden layers 30 neurons each (trained using RmsProp).

Outline

1. Introduction
2. Why Tensors?
3. Algorithms for Tensor Decomposition
4. Learning Probabilistic Models
5. Learning Representations
6. Conclusion

Conclusion

Guaranteed Unsupervised Learning

- Matrix and tensor methods have desirable guarantees on reaching global optimum.
- Method of moment estimators using tensor decomposition.
- Polynomial computational and sample complexity.
- Faster and better performance in practice.

Steps Forward

- Scaling up tensor methods: sketching algorithms, extended BLAS, ...
- Unified conditions on when unsupervised learning is tractable?

Resources for Today

- Tensor introduction
  `http://newport.eecs.uci.edu/anandkumar/tensor.html`
- Fast tensor decomposition algorithms.
  `https://bitbucket.org/megaDataLab/tensormethodsforml/overv`
- Spectral methods for training latent Dirichlet allocation topic models on Apache Spark.
  `https://github.com/FurongHuang/SpectralLDA-TensorSpark`