Dictionary Learning Using Tensor Methods

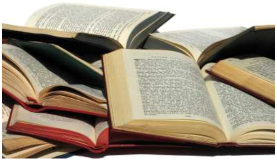
Anima Anandkumar

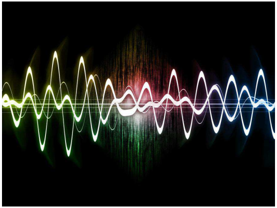
U.C. Irvine

Joint work with Rong Ge, Majid Janzamin and Furong Huang.

Feature learning as cornerstone of ML

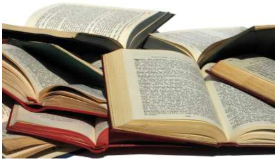
ML Practice

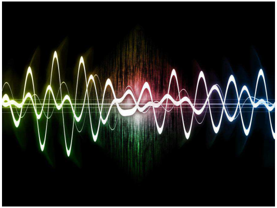




Feature learning as cornerstone of ML

ML Practice





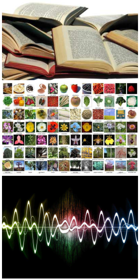
ML Papers

$$
\begin{array}{c|cccc}
\text{Label} & & \text{Features} & & \\
\hline
0 & 2.1 & 5.2 & 0 & 0 \\
1 & 0 & 0 & 2 & 1 \\
1 & 1.1 & 0 & 0 & 0 \\
0 & 0 & 0 & 7 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots
\end{array}
$$

Feature learning as cornerstone of ML

- Find efficient representation of data, e.g. based on sparsity, Invariances, low dimensional structures etc.

ML Practice



ML Papers

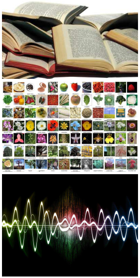
$$
\begin{array}{cc}
\text{Label} & \text{Features} \\
\left[\begin{array}{c|ccccc}
0 & 2.1 & 5.2 & 0 & 0 & \cdots \\
1 & 0 & 0 & 2 & 1 & \cdots \\
1 & 1.1 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 7 & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots &
\end{array}\right] &
\end{array}
$$

- Feature engineering typically critical for good performance
- Deep learning has shown considerable promise for feature learning

Feature learning as cornerstone of ML

- Find efficient representation of data, e.g. based on sparsity, Invariances, low dimensional structures etc.

ML Practice



ML Papers

$$
\begin{array}{c}
\text{Label} \qquad\qquad \text{Features} \\
\left[\begin{array}{c|ccccc}
0 & 2.1 & 5.2 & 0 & 0 & \cdots \\
1 & 0 & 0 & 2 & 1 & \cdots \\
1 & 1.1 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 7 & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots &
\end{array}\right]
\end{array}
$$

- Feature engineering typically critical for good performance
- Deep learning has shown considerable promise for feature learning
- **Can we provide principled approaches which are guaranteed to learn good features?**

Applications of Representation Learning

Compressed sensing

- Extensive literature on compressed sensing
- Few linear measurements to recover sparse signals
- What if the signal is not sparse in input representation?
- What if the dictionary has invariances, e.g. shift, rotation.

Applications of Representation Learning

Compressed sensing

- Extensive literature on compressed sensing
- Few linear measurements to recover sparse signals
- What if the signal is not sparse in input representation?
- What if the dictionary has invariances, e.g. shift, rotation.
- **Can we learn a representation where the signal is sparse?**

Applications of Representation Learning

Compressed sensing

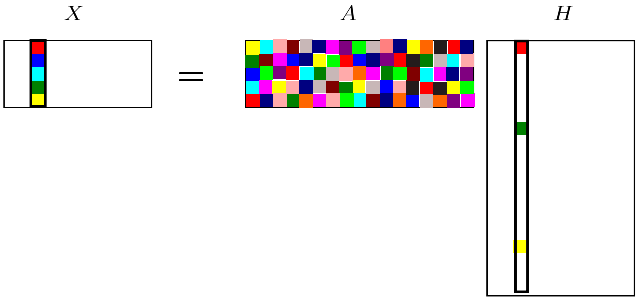
- Extensive literature on compressed sensing
- Few linear measurements to recover sparse signals
- What if the signal is not sparse in input representation?
- What if the dictionary has invariances, e.g. shift, rotation.
- **Can we learn a representation where the signal is sparse?**

Topic Modeling

- Unsupervised learning of admixtures.
- In text documents, social networks (community modeling), biological models, ...

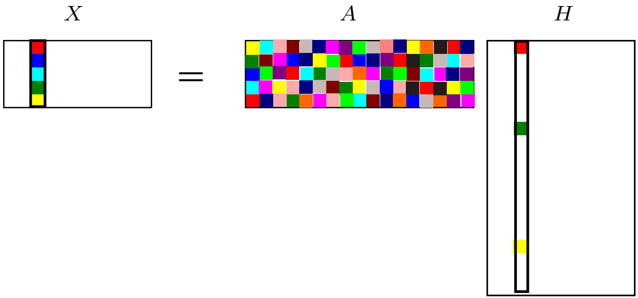
Dictionary Learning Model

Goal: Find dictionary A with k elements such that each data point is a linear combination of sparse combination of dictionary elements.



Dictionary Learning Model

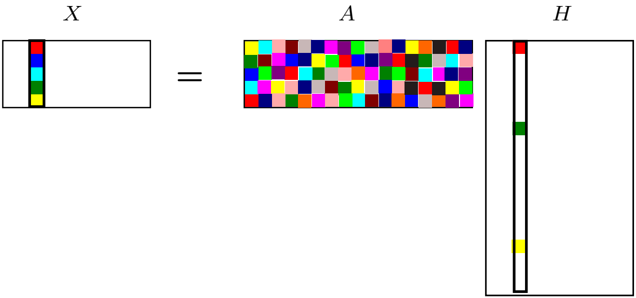
Goal: Find dictionary A with k elements such that each data point is a linear combination of sparse combination of dictionary elements.



- **Topic models:** x_i is a document, A contains *topics*, h_i gives topics in document i

Dictionary Learning Model

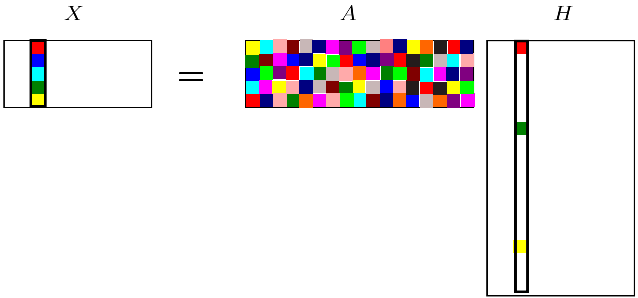
Goal: Find dictionary A with k elements such that each data point is a linear combination of sparse combination of dictionary elements.



- **Topic models:** x_i is a document, A contains *topics*, h_i gives topics in document i
- **Compressed sensing:** x_i are the signals, A is a basis with sparse representation

Dictionary Learning Model

Goal: Find dictionary A with k elements such that each data point is a linear combination of sparse combination of dictionary elements.



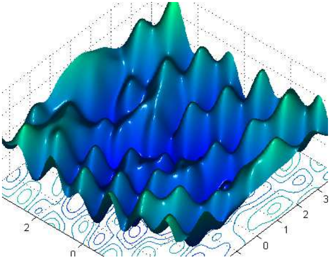
- **Topic models:** x_i is a document, A contains *topics*, h_i gives topics in document i
- **Compressed sensing:** x_i are the signals, A is a basis with sparse representation
- **Images:** x_i is an image, A contains *filters*, h_i gives filters present in image i (also need to incorporate invariances)

Outline

Learning Dictionary Models

Computational Challenges

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



Tensor methods can yield guaranteed learning

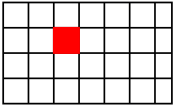
Moment Matrices and Tensors

Multivariate Moments

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

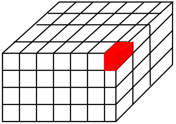
Matrix

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



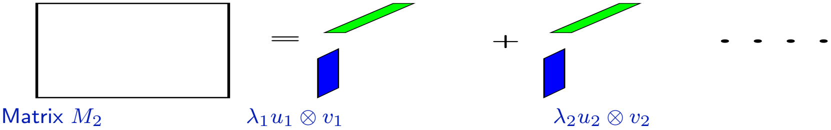
Tensor

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



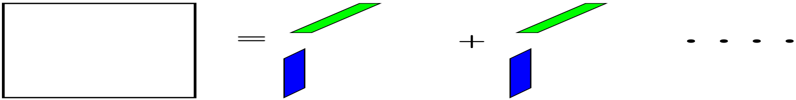
Spectral Decomposition of Tensors

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



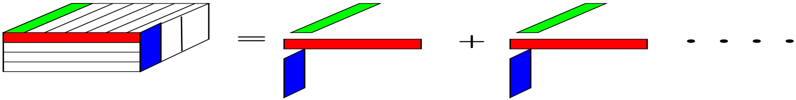
Spectral Decomposition of Tensors

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



Matrix M_2 $\qquad$ $\lambda_1 u_1 \otimes v_1$ $\qquad$ $\lambda_2 u_2 \otimes v_2$

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



Tensor M_3 $\qquad$ $\lambda_1 u_1 \otimes v_1 \otimes w_1$ $\qquad$ $\lambda_2 u_2 \otimes v_2 \otimes w_2$

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

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

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

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

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

Tensor Rank and Tensor Decomposition

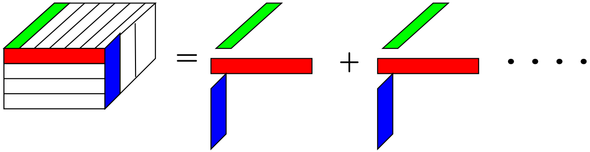
Rank-1 tensor: $T = w \cdot a \otimes b \otimes c \Leftrightarrow T(i,j,l) = w \cdot a(i) \cdot b(j) \cdot c(l)$.

Tensor Rank and Tensor Decomposition

Rank-1 tensor: $T = w \cdot a \otimes b \otimes c \Leftrightarrow T(i,j,l) = w \cdot a(i) \cdot b(j) \cdot c(l)$.

CANDECOMP/PARAFAC (CP) Decomposition

$$\boxed{T = \sum_{j \in [k]} w_j a_j \otimes b_j \otimes c_j \in \mathbb{R}^{d \times d \times d}, \quad a_j, b_j, c_j \in \mathcal{S}^{d-1}.}$$



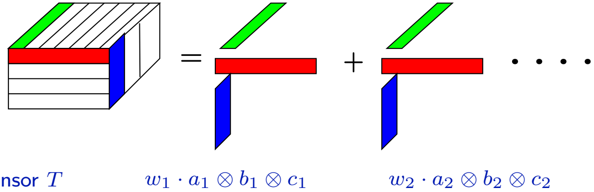
Tensor T $\qquad$ $w_1 \cdot a_1 \otimes b_1 \otimes c_1$ $\qquad$ $w_2 \cdot a_2 \otimes b_2 \otimes c_2$

Tensor Rank and Tensor Decomposition

Rank-1 tensor: $T = w \cdot a \otimes b \otimes c \Leftrightarrow T(i,j,l) = w \cdot a(i) \cdot b(j) \cdot c(l)$.

CANDECOMP/PARAFAC (CP) Decomposition

$$\boxed{T = \sum_{j \in [k]} w_j a_j \otimes b_j \otimes c_j \in \mathbb{R}^{d \times d \times d}, \quad a_j, b_j, c_j \in \mathcal{S}^{d-1}.}$$



- k: tensor rank, d: ambient dimension.
- $k \leq d$: undercomplete and $k > d$: overcomplete.

Orthogonal Tensor Power Method

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

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

Orthogonal Tensor Power Method

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

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

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

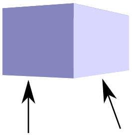
Orthogonal Tensor Power Method

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

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

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

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



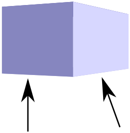
Orthogonal Tensor Power Method

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

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

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

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



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

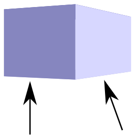
Orthogonal Tensor Power Method

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

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

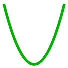
Recall matrix power method: $v \mapsto \dfrac{M(I,v)}{\|M(I,v)\|}$.

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



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

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



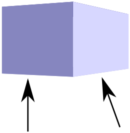
Orthogonal Tensor Power Method

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

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

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

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



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

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

- All other eigenvectors are saddle points.

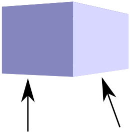
Orthogonal Tensor Power Method

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

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

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

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



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

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

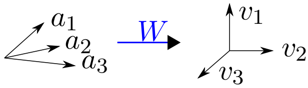
- All other eigenvectors are saddle points.

For an orthogonal tensor, no spurious local optima!

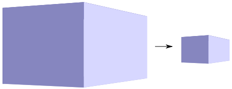
Putting it together

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

- Whitening matrix W:



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

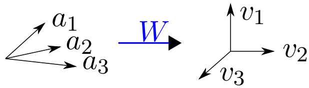


Tensor M_3 Tensor T

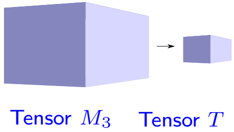
Putting it together

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

- Whitening matrix W:



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



Tensor Decomposition in Undercomplete Case: Solved!

Overcomplete Setting

- In general, tensor decomposition NP-hard.
- Tractable when A is incoherence, i.e. $\langle a_i, a_j \rangle \approx \frac{1}{\sqrt{d}}$ for $i \neq j$.

Overcomplete Setting

- In general, tensor decomposition NP-hard.
- Tractable when A is incoherence, i.e. $\langle a_i, a_j \rangle \approx \frac{1}{\sqrt{d}}$ for $i \neq j$.

SVD Initialization

- Find the top singular vectors of $T(I, I, \theta)$ for $\theta \sim \mathcal{N}(0, I)$.
- Use them for initialization of power method. L trials.

Overcomplete Setting

- In general, tensor decomposition NP-hard.
- Tractable when A is incoherence, i.e. $\langle a_i, a_j \rangle \approx \frac{1}{\sqrt{d}}$ for $i \neq j$.

SVD Initialization

- Find the top singular vectors of $T(I, I, \theta)$ for $\theta \sim \mathcal{N}(0, I)$.
- Use them for initialization of power method. L trials.

Assumptions

- Number of initializations: $L \geq k^{\Omega(k/d)^2}$, Tensor Rank: $k = O(d)$
- No. of Iterations: $N = \Theta\left(\log(1/\|E\|)\right)$. Recall $\|E\|$: recovery error.

Theorem (Global Convergence)[AGJ-COLT2015]:

$$\boxed{\|a_1 - \hat{a}^{(N)}\| \leq O(\|E\|).}$$

Improved Sample Complexity Analysis

- Dictionary $A \in \mathbb{R}^{d \times k}$ satisfying RIP, sparse-ICA model with sub-Gaussian variables.
- Sparsity level s. Number of samples n.

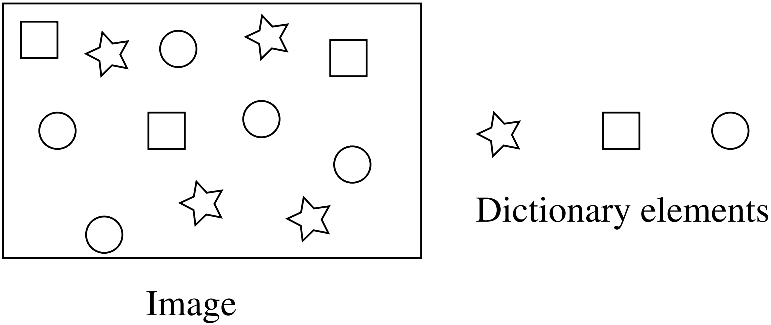
$$\|\widehat{M_4} - M_4\| = \tilde{O}\left(\frac{s^2}{n} + \sqrt{\frac{s^4}{d^3 n}}\right)$$

- Careful ϵ-net covering and bucketing.

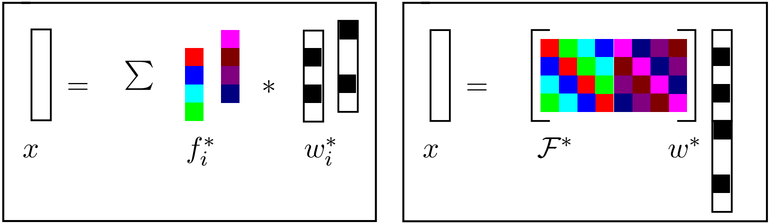
Outline

Convolutional Dictionary Model

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



Rewriting as a standard dictionary model



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

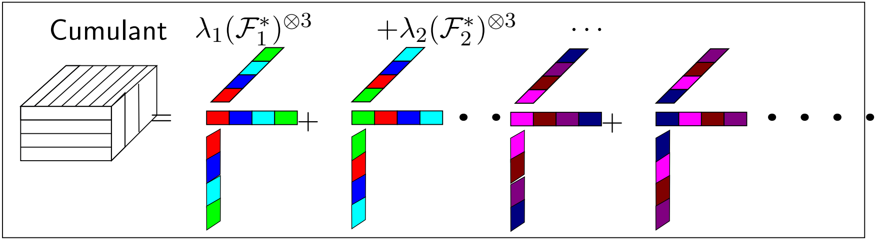
$$x = \sum_i f_i * w_i = \sum_i \mathrm{Cir}(f_i) w_i = \mathcal{F}^* w^*$$

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

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 $\mathcal{F}_i^*$.



Efficient Optimization Techniques

cumulant $= \sum_j \lambda_j \mathcal{F}_j^{\otimes 3}$ or matricization: cumulant $= \mathcal{F}^* \Lambda^* (\mathcal{F}^* \odot \mathcal{F}^*)^\top$

Efficient Optimization Techniques

cumulant $= \sum_j \lambda_j \mathcal{F}_j^{\otimes 3}$ or matricization: cumulant $= \mathcal{F}^* \Lambda^* (\mathcal{F}^* \odot \mathcal{F}^*)^\top$

Objective function: $\min_{\mathcal{F}} \quad \|\text{Cumulant} - \mathcal{F}\Lambda\left(\mathcal{F} \odot \mathcal{F}\right)^\top\|_{\mathbb{F}}^2$

s.t. $\text{blk}_l(\mathcal{F}) = U\text{Diag}(\text{FFT}(f_l))U^{\mathsf{H}}, \ \|f_l\|_2 = 1.$

Efficient Optimization Techniques

$$\text{cumulant} = \sum_j \lambda_j \mathcal{F}_j^{\otimes 3} \text{ or matricization: cumulant} = \mathcal{F}^* \Lambda^* (\mathcal{F}^* \odot \mathcal{F}^*)^\top$$

Objective function: $\min_{\mathcal{F}} \quad \|\text{Cumulant} - \mathcal{F}\Lambda\left(\mathcal{F} \odot \mathcal{F}\right)^\top\|_{\mathbb{F}}^2$

s.t. $\text{blk}_l(\mathcal{F}) = U\text{Diag}(\text{FFT}(f_l))U^{\text{H}}, \; \|f_l\|_2 = 1.$

Alternating minimization: Relax $\mathcal{F}\Lambda\left(\mathcal{F} \odot \mathcal{F}\right)^\top$ to $\mathcal{F}\Lambda\left(\mathcal{H} \odot \mathcal{G}\right)^\top$

Efficient Optimization Techniques

cumulant $= \sum_j \lambda_j \mathcal{F}_j^{\otimes 3}$ or matricization: cumulant $= \mathcal{F}^* \Lambda^* (\mathcal{F}^* \odot \mathcal{F}^*)^\top$

Objective function: $\min_{\mathcal{F}} \quad \|\text{Cumulant} - \mathcal{F}\Lambda\left(\mathcal{F} \odot \mathcal{F}\right)^\top\|_{\mathbb{F}}^2$
s.t. $\text{blk}_l(\mathcal{F}) = U\text{Diag}(\text{FFT}(f_l))U^{\mathsf{H}}, \ \|f_l\|_2 = 1.$

Alternating minimization: Relax $\mathcal{F}\Lambda\left(\mathcal{F} \odot \mathcal{F}\right)^\top$ to $\mathcal{F}\Lambda\left(\mathcal{H} \odot \mathcal{G}\right)^\top$

Under full column rank $\mathcal{H} \odot \mathcal{G}$, form: $T := \text{Cumulant}\left((\mathcal{H} \odot \mathcal{G})^\top\right)^\dagger.$

Efficient Optimization Techniques

cumulant $= \sum_j \lambda_j \mathcal{F}_j^{\otimes 3}$ or matricization: cumulant $= \mathcal{F}^* \Lambda^* (\mathcal{F}^* \odot \mathcal{F}^*)^\top$

Objective function: $\min_{\mathcal{F}} \quad \|\text{Cumulant} - \mathcal{F}\Lambda\left(\mathcal{F} \odot \mathcal{F}\right)^\top\|_{\mathbb{F}}^2$

s.t. $\mathsf{blk}_l(\mathcal{F}) = U\mathsf{Diag}(\mathsf{FFT}(f_l))U^{\mathsf{H}}, \ \|f_l\|_2 = 1.$

Alternating minimization: Relax $\mathcal{F}\Lambda\left(\mathcal{F} \odot \mathcal{F}\right)^\top$ to $\mathcal{F}\Lambda\left(\mathcal{H} \odot \mathcal{G}\right)^\top$

Under full column rank $\mathcal{H} \odot \mathcal{G}$, form: $T :=$ Cumulant $\left(\left(\mathcal{H} \odot \mathcal{G}\right)^\top\right)^\dagger$.

Main Result: Optimal solution f_l^{opt}, $\forall p \in [n], q := (i-j) \mod n$,

$$f_l^{\text{opt}}(p) = \frac{\sum\limits_{i,j\in[n]} \|\mathsf{blk}_l(T)_j\|^{-1} \cdot \mathsf{blk}_l(T)_j^i \cdot I_{p-1}^q}{\sum\limits_{i,j\in[n]} I_{p-1}^q},$$

Efficient Optimization Techniques

Under full column rank $\mathcal{H} \odot \mathcal{G}$, form: $T :=$ Cumulant $\left((\mathcal{H} \odot \mathcal{G})^\top\right)^\dagger$.

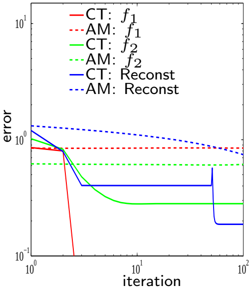
- Optimal solution is then computed in closed form.
- Bottleneck computation: $\left((\mathcal{H} \odot \mathcal{G})^\top\right)^\dagger$. Naive implementation: $O(n^6)$ time, where n is the length of signal.

Running time of our method: For length-n signals and L number of filters, $O(\log n + \log L)$ time with $O(L^2 n^3)$ processors.

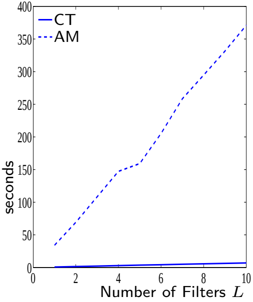
- Involves $2L$ FFT's, some matrix multiplications, inverse of diagonal matrices.

Experiments (synthetic)

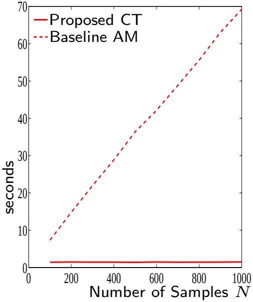
- Convolutional tensor (CT). Alternating minimization (AM).



(a) Reconstruction Error



(b) Running Times Scale with L



(c) Running Times Scale with N

Experiments (NLP)

- Microsoft paraphrase dataset. 4096 sentence pairs. Unsupervised convolutional tensor method: no outside information. F score.

Method	Description	Outside Information	F score
Vector Similarity	cosine similarity with tf-idf weights	word similarity	75.3%
ESA	explicit semantic space	word semantic profiles	79.3%
LSA	latent semantic space	word semantic profiles	79.9%
RMLMG	graph subsumption	lexical&syntactic&synonymy info	80.5%
CT (proposed)	**convolutional dictionary learning**	**none**	**80.7%**
MCS	combine word similarity measures	word similarity	81.3%
STS	combine semantic&string similarity	semantic similarity	81.3%
SSA	salient semantic space	word semantic profiles	81.4%
matrixJcn	JCN WordNet similarity with matrix	word similarity	82.4%

Paraphrase detected: *(1) Amrozi accused his brother, whom he called "the witness", of deliberately distorting his evidence. (2) Referring to him as only "the witness", Amrozi accused his brother of deliberately distorting his evidence.*

Non-paraphrase detected : *(1) I never organised a youth camp for the diocese of Bendigo. (2) I never attended a youth camp organised by that diocese."*

Outline

Summary and Outlook

Summary

- Method of moments for learning dictionary elements.
- Invariances in convolutional models can be handled efficiently.

Summary and Outlook

Summary

- Method of moments for learning dictionary elements.
- Invariances in convolutional models can be handled efficiently.

Outlook

- Analyze optimization landscape for convolutional models for tensor methods.
- Extend to other kinds of invariances (e.g. rotation).

Summary and Outlook

Summary

- Method of moments for learning dictionary elements.
- Invariances in convolutional models can be handled efficiently.

Outlook

- Analyze optimization landscape for convolutional models for tensor methods.
- Extend to other kinds of invariances (e.g. rotation).

How is feature learning useful for classification?

- Precise characterization for training neural networks: first polynomial time methods!
- "Beating the Perils of Non-Convexity: Guaranteed Training of Neural Networks using Tensor Methods" by Majid Janzamin, Hanie Sedghi and A.