Convolutional Dictionary Learning through Tensor Factorization

*Furong Huang and Animashree Anandkumar

University of California, Irvine

Feature learning as cornerstone of ML

- Find efficient representation of data based on
  - sparsity / group invariance / low dimensional structures
- Principled approaches guaranteed to learn good features?

Summary

- Goal: Feature Learning or Representation Learning
- Methods: Tensor Decomposition
  - powerful paradigm and consistent learning
  - proven success for a wide class of latent variable
    - topic model/ ICA/ Mixture of Gaussian/ HMM
- Contribution: models with invariances
  - Shift invariance: convolutional dictionary models
  - ALS method with shift invariance constraints
- Validation: convolutional tensor vs alternating minimization
  - CT: converges faster, better reconstruction error
  - AM: pass through data in every iteration

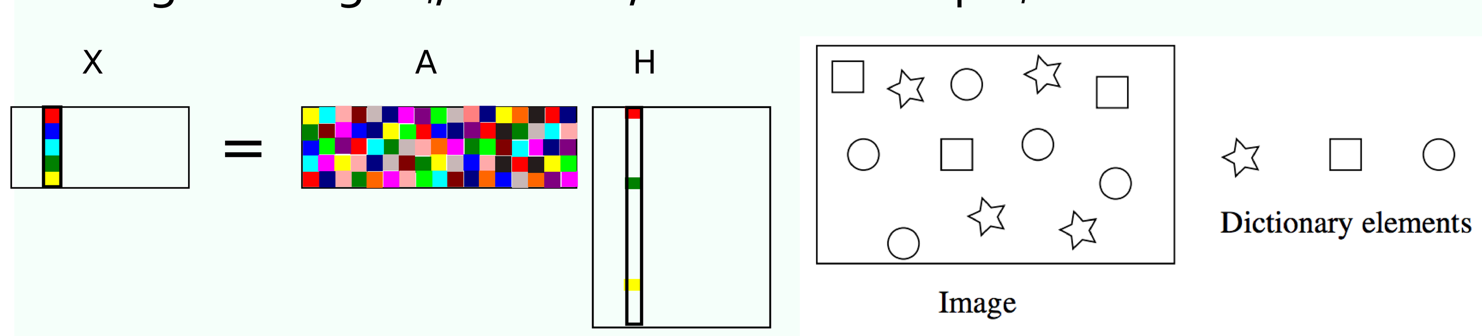
Standard **Linear** vs **Convolutional** Dictionary Models

Dictionary learning: Find dictionary A

- k dictionary components/elements
- Signal $=$ **linear**/sparse combination of dictionary elements

1 Topic model: doc x_i, topic-word matrix A, topics in doc h_i

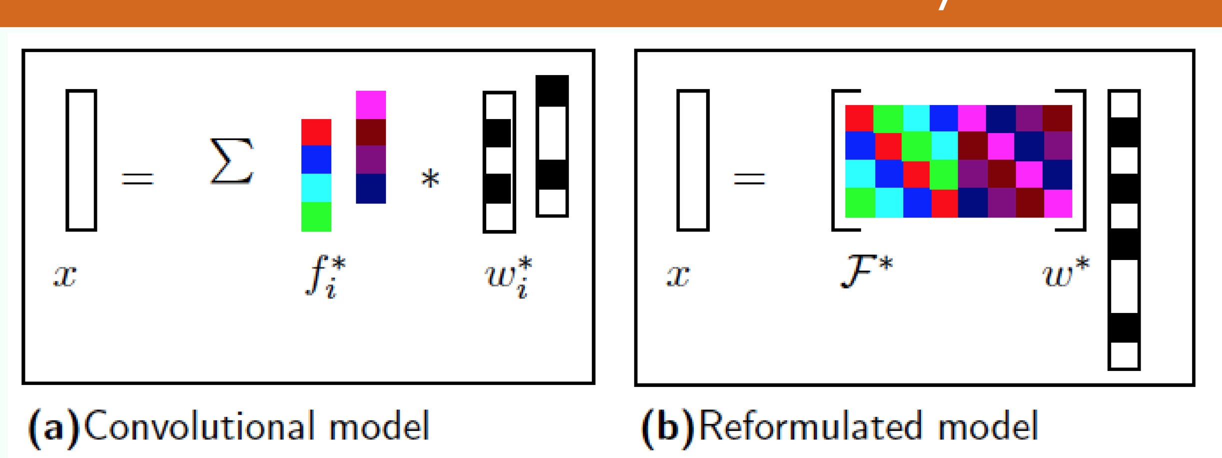
2 Images: image x_i, filters A, activation map h_i



Problem in standard dictionary model: NO invariances.

Convolutional models incorporate **shift invariance**

From convolutional to standard dictionary model



$$x = \sum_l f_l \star w_l = \sum_l \mathrm{Cir}(f_l) w_l = \mathcal{F}^\star w^\star$$

1 Assume coefficients w_i are independent (convolutional ICA model)

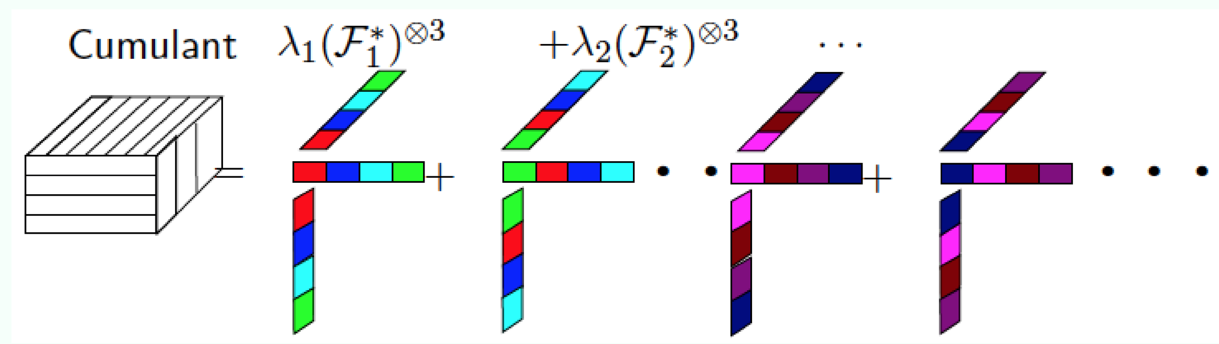
2 Cumulant tensor has decomposition with components $\mathcal{F}_i^*$.

Data Moments Relating Model Parameters

$$C_3 := \mathbb{E}[x(x \odot x)^\top] - \mathrm{unfold}(Z)$$

- **Shift term** Z: composed of 1^{st} and 2^{nd} empirical moments.

Decomposition form of C_3



$$C_3 = \mathrm{unfold}\left(\sum_{j\in[nL]} \lambda_j^* \mathcal{F}_j^{*\otimes 3}\right) = \mathcal{F}^* \Lambda^* (\mathcal{F}^* \odot \mathcal{F}^*)^\top$$

where λ_j^* is the third order cumulant corresponding to the (univariate) distribution of $w^*(j)$.

Optimization with Closed Form Result

Estimate Filter Dictionary

- Goal: estimate filters f_l
- minimize Frobenius norm $\|\cdot\|_{\mathbb{F}}$ of reconstruction
  - on cumulant tensor C_3

$$\min_{\mathcal{F}} \quad \|C_3 - \mathcal{F}\Lambda(\mathcal{F}\odot\mathcal{F})^\top\|_F^2$$

s.t. $\mathrm{blk}_l(\mathcal{F}) = U\mathrm{diag}(\mathrm{FFT}(f_l))U^H,\ \|f_l\|_2 = 1,\ \Lambda = \mathrm{diag}(\lambda)$.

ALS Relaxation: $\mathcal{F}\Lambda(\mathcal{F}\odot\mathcal{F})^\top \to \mathcal{F}\Lambda(\mathcal{H}\odot\mathcal{G})^\top$

Closed Form Main Result: The optimal solution f_l^{opt} is

$$f_l^{\mathrm{opt}}(p) = \frac{\sum_{i,j\in[n]} \|\mathrm{blk}_l(M)_j\|^{-1} \cdot \mathrm{blk}_l(M)_j^i \cdot f_{p-1}^q}{\sum_{i,j\in[n]} f_{p-1}^q},\ \forall p \in [n], q := (i-j) \mod n.$$

Note that $M := C_3\left((\mathcal{H}\odot\mathcal{G})^\top\right)^\dagger$.

Efficient Optimization

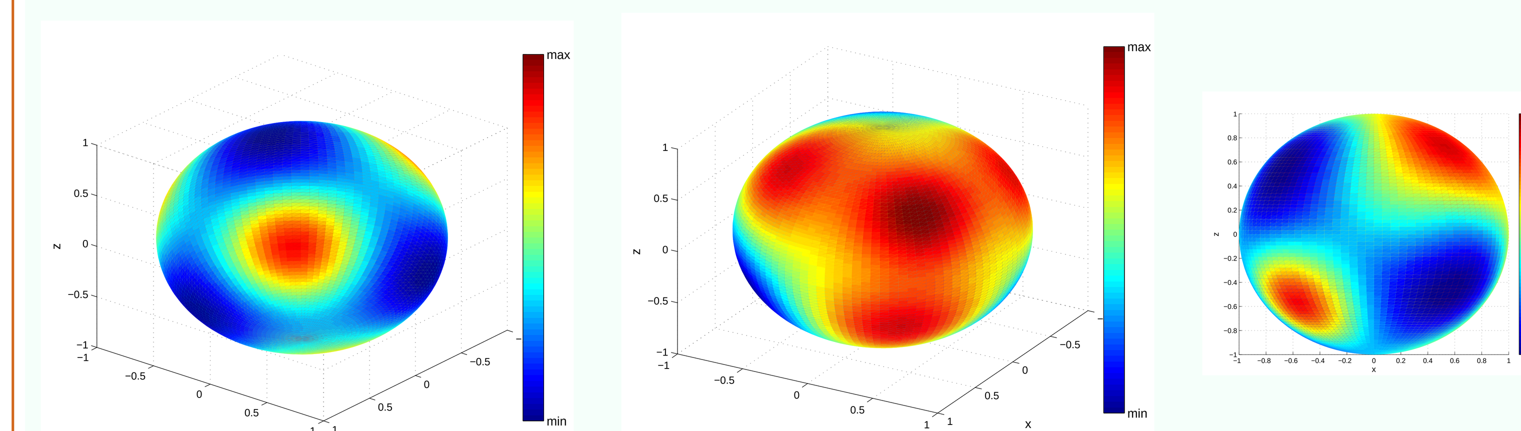
Bottleneck: Computing $\left((\mathcal{H}\odot\mathcal{G})^\top\right)^\dagger$ requires $O(n^6)$ naively.

Solution:

- $((\mathcal{H}\odot\mathcal{G})^\top)^\dagger = (\mathcal{H}\odot\mathcal{G})((\mathcal{H}^\top\mathcal{H}).\star(\mathcal{G}^\top\mathcal{G}))^\dagger$
- Row and column stacked circulant matrices $(\mathcal{H}^\top\mathcal{H}).\star(\mathcal{G}^\top\mathcal{G})$
- The inversion of $(\mathcal{H}^\top\mathcal{H}).\star(\mathcal{G}^\top\mathcal{G})$ reduced to the inversion of row-and-column stacked set of diagonal matrices
- Block matrix inversion theorem [Golub and Van Loan, 2012]

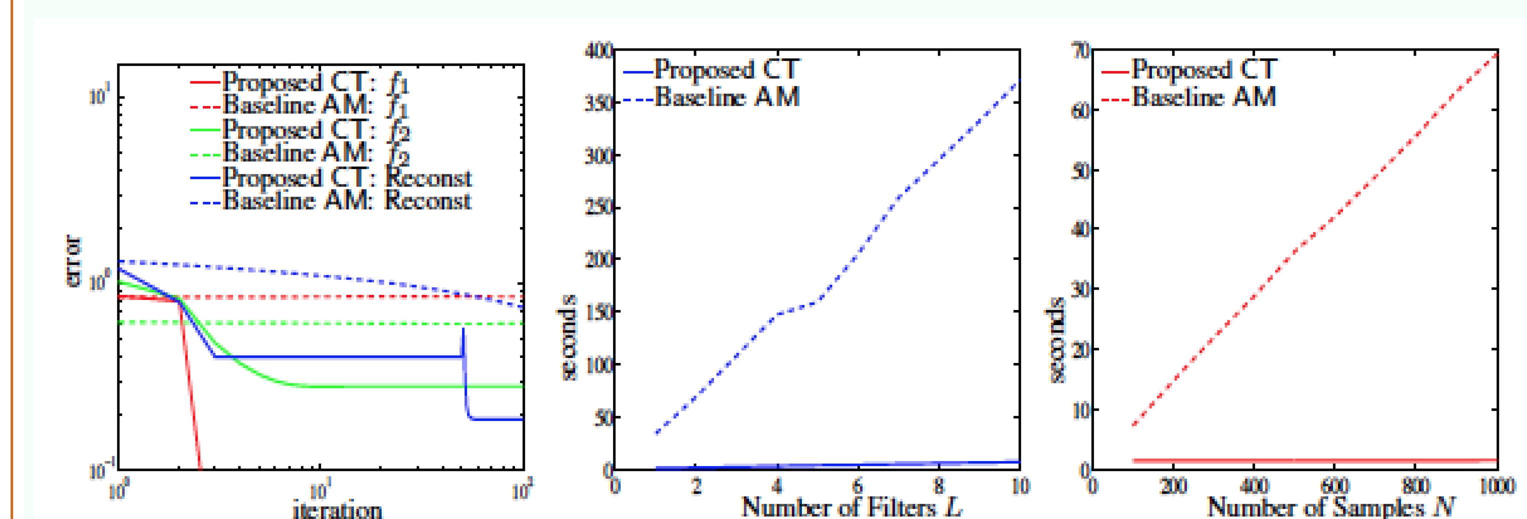
Running Time: $O(\log n + \log L)$ with $O(L^2 n^3)$ processors. Involves $2L$ FFT's, some matrix multiplications, inverse of diagonal matrices.

Objective Visualization: Optimal Points



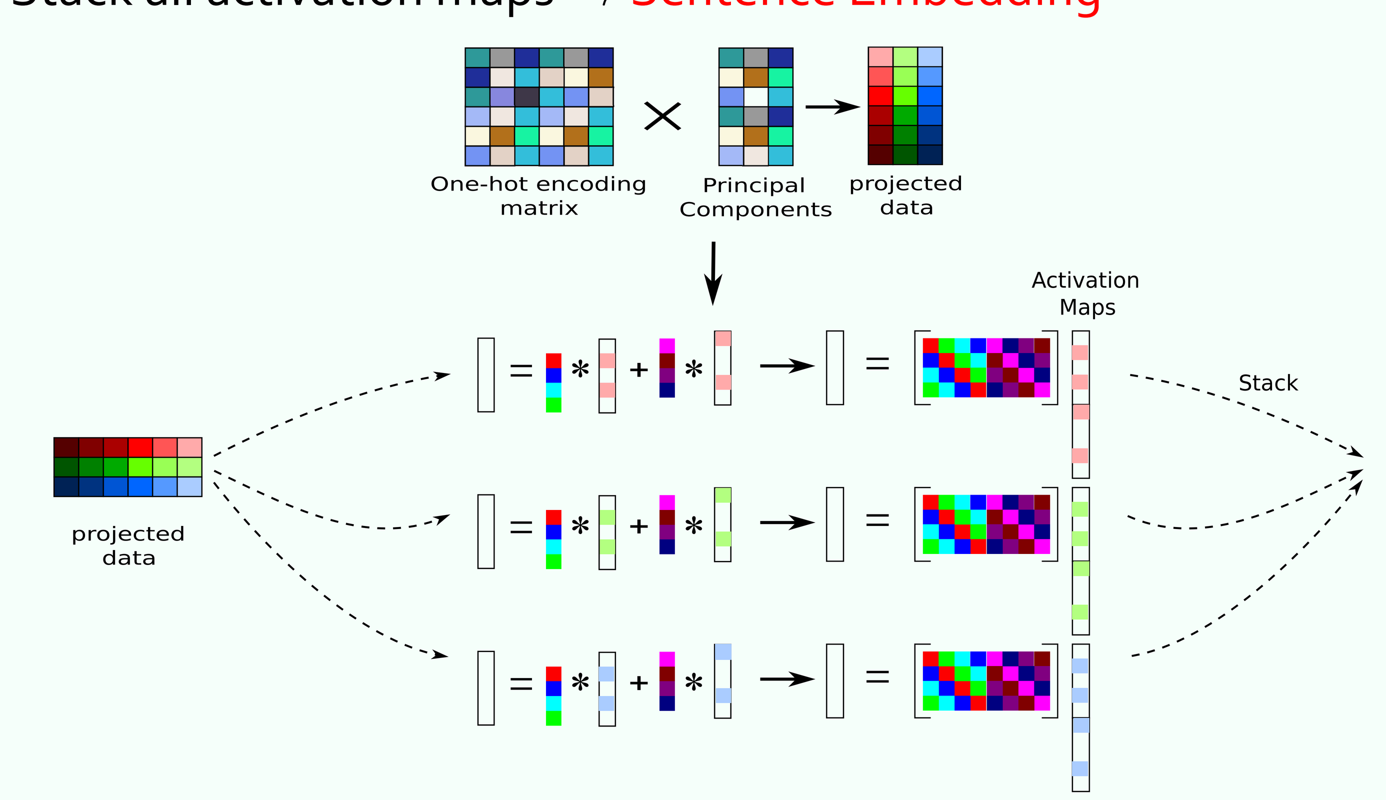
Blue: objective fn has small value $\to$ Red: objective fn has large value

Convolutional Tensor vs Alternating Minimization



Paraphrase Detection: embeddings learned from scratch

- PCA on One-hot Encoding Matrix $\to$ Subspace and Projected data
- CT on each coordinate $\to$ activation map for each coordinate
- Stack all activation maps $\to$ Sentence Embedding



Method	Description	Outside Information [1]	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%

- All other methods use word similarities trained on Wikipedia or from *WordNet*.
- Our algorithm: detects paraphrases from scratch, no side information used.
- Our algorithm: achieves comparable results as we incorporate word orders.

*Email: furongh@uci.edu