Guaranteed Learning of Overcomplete Latent Representations

Anima Anandkumar

U.C. Irvine

Joint work with Alekh Agarwal, Praneeth Netrapalli, Prateek Jain, Rashish, Daniel Hsu, Majid Janzamin, Sham Kakade.
Latent Variable Modeling

Goal: Discover hidden effects from observed measurements

Example: document modeling


Learning latent variable models: efficient methods and guarantees
Other Applications of Latent Variable Modeling

Social Network Modeling
- Observed: social interactions.
- Hidden: communities, relationships

Bio-Informatics
- Observed: gene expressions or neural activity.

Recommendation Systems
- Observed: recommendations: e.g. yelp reviews.
- Hidden: User and business attributes

Applications in Speech, Vision ...
Challenges in Learning Latent Variable Models

Challenges in Identifiability

- When can latent variables be identified?
- Conditions on the model parameters, e.g. on topic-word matrix or dictionary elements?
- Does identifiability also lead to tractable algorithms?
Challenges in Identifiability

- When can latent variables be identified?
- Conditions on the model parameters, e.g. on topic-word matrix or dictionary elements?
- Does identifiability also lead to tractable algorithms?

Challenges in Design of Learning Algorithms

- Maximum likelihood learning NP-hard (Arora et. al.)
- In practice, methods such as Gibbs sampling, variational Bayes etc. but no guarantees.
- Guaranteed learning with minimal assumptions? Efficient methods? Low sample and computational complexities?
Classes of Latent Variable Models

Typical Assumption in Latent Variable Models

- Latent dimensionality $\ll$ observed dimensionality.
- Applicable in community and document modeling.
- Low rank tensor through conditional independence relations.
Classes of Latent Variable Models

Typical Assumption in Latent Variable Models

- Latent dimensionality $\ll$ observed dimensionality.
- Applicable in community and document modeling
- Low rank tensor through conditional independence relations

Classes of Latent Variable Models

Typical Assumption in Latent Variable Models

- Latent dimensionality \( \ll \) observed dimensionality.
- Applicable in community and document modeling
- Low rank tensor through conditional independence relations


Overcomplete Latent Representations

- Latent dimensionality \( \gg \) observed dimensionality
- Flexible modeling, robust to noise
- Applicable in speech and image modeling
- Large amount of unlabeled samples
Classes of Latent Variable Models

Typical Assumption in Latent Variable Models

- Latent dimensionality \(\ll\) observed dimensionality.
- Applicable in community and document modeling
- Low rank tensor through conditional independence relations


Overcomplete Latent Representations

- Latent dimensionality \(\gg\) observed dimensionality
- Flexible modeling, robust to noise
- Applicable in speech and image modeling
- Large amount of unlabeled samples

This talk: Guaranteed Learning of Overcomplete Representations
Linear Overcomplete Latent Variable Models

- Also known as dictionary learning problem
Linear Overcomplete Latent Variable Models

- Also known as \textit{dictionary learning} problem

\textbf{Setup}

- Latent dimensionality \( k > \) observed dimensionality \( n \).
- \( A = [a_1, \ldots, a_k] \): Latent vectors (dictionary elements)
- \( y \in \mathbb{R}^n \): Observation. \( Y = [y_1, \ldots, y_m] \in \mathbb{R}^{n \times m} \): Observation matrix.
- Linear model: \( Y = AX \).
- Learning problem: Given \( Y \), find \( A \) and \( X \).
Linear Overcomplete Latent Variable Models

- Also known as dictionary learning problem

Setup

- Latent dimensionality $k >$ observed dimensionality $n$.
- $A = [a_1, \ldots, a_k]$: Latent vectors (dictionary elements)
- $y \in \mathbb{R}^n$: Observation. $Y = [y_1, \ldots, y_m] \in \mathbb{R}^{n \times m}$: Observation matrix.
- Linear model: $Y = AX$.
- Learning problem: Given $Y$, find $A$ and $X$.

Challenges

- Learning in overcomplete regime: $k > n$.
- Ill-posed without further constraints.
Two Approaches for Learning Overcomplete Models

- Latent dimensionality $k \gg$ observed dimensionality $n$.

Dictionary Learning
- $y \in \mathbb{R}^n$: sample. $Y \in \mathbb{R}^{n \times m}$.

Sparse Topic Models
- $y \in \mathbb{R}^n$: word. $m$ documents.
Two Approaches for Learning Overcomplete Models

- Latent dimensionality $k \gg$ observed dimensionality $n$.

Dictionary Learning
- $y \in \mathbb{R}^n$: sample. $Y \in \mathbb{R}^{n \times m}$.
- $A = [a_1, \ldots, a_k]$: Dictionary.

Sparse Topic Models
- $y \in \mathbb{R}^n$: word. $m$ documents.
- $A = [a_1, \ldots, a_k]$: Topic-word matrix.
Two Approaches for Learning Overcomplete Models

- Latent dimensionality $k \gg$ observed dimensionality $n$.

**Dictionary Learning**
- $y \in \mathbb{R}^n$: sample. $Y \in \mathbb{R}^{n \times m}$.
- $A = [a_1, \ldots, a_k]$: Dictionary.
- $X \in \mathbb{R}^{k \times m}$: mixing matrix.

**Sparse Topic Models**
- $y \in \mathbb{R}^n$: word. $m$ documents.
- $A = [a_1, \ldots, a_k]$: Topic-word matrix.
- $x \in \mathbb{R}^k$: Topic proportions vector.
Two Approaches for Learning Overcomplete Models

- Latent dimensionality $k \gg$ observed dimensionality $n$.

**Dictionary Learning**
- $y \in \mathbb{R}^n$: sample. $Y \in \mathbb{R}^{n \times m}$.
- $A = [a_1, \ldots, a_k]$: Dictionary.
- $X \in \mathbb{R}^{k \times m}$: mixing matrix.
- Linear model: $Y = AX$.

**Sparse Topic Models**
- $y \in \mathbb{R}^n$: word. $m$ documents.
- $A = [a_1, \ldots, a_k]$: Topic-word matrix.
- $x \in \mathbb{R}^k$: Topic proportions vector
- Linear model: $\mathbb{E}[y|x] = Ax$. 
Two Approaches for Learning Overcomplete Models

- Latent dimensionality \( k \gg n \).

**Dictionary Learning**
- \( y \in \mathbb{R}^n \): sample. \( Y \in \mathbb{R}^{n \times m} \).
- \( A = [a_1, \ldots, a_k] \): Dictionary.
- \( X \in \mathbb{R}^{k \times m} \): mixing matrix.
- Linear model: \( Y = AX \).
- Sparse mixing \( X \) and Incoherent dictionary \( A \).

**Sparse Topic Models**
- \( y \in \mathbb{R}^n \): word. \( m \) documents.
- \( A = [a_1, \ldots, a_k] \): Topic-word matrix.
- \( x \in \mathbb{R}^k \): Topic proportions vector
- Linear model: \( \mathbb{E}[y|x] = Ax \).
Two Approaches for Learning Overcomplete Models

- Latent dimensionality $k \gg$ observed dimensionality $n$.

Dictionary Learning
- $y \in \mathbb{R}^n$: sample. $Y \in \mathbb{R}^{n \times m}$.
- $A = [a_1, \ldots, a_k]$: Dictionary.
- $X \in \mathbb{R}^{k \times m}$: mixing matrix.
- Linear model: $Y = AX$.
- Sparse mixing $X$ and Incoherent dictionary $A$.
- Clustering and alt. min.

Sparse Topic Models
- $y \in \mathbb{R}^n$: word. $m$ documents.
- $A = [a_1, \ldots, a_k]$: Topic-word matrix.
- $x \in \mathbb{R}^k$: Topic proportions vector
- Linear model: $\mathbb{E}[y|x] = Ax$. 
Two Approaches for Learning Overcomplete Models

- Latent dimensionality $k \gg$ observed dimensionality $n$.

**Dictionary Learning**
- $y \in \mathbb{R}^n$: sample. $Y \in \mathbb{R}^{n \times m}$.
- $A = [a_1, \ldots, a_k]$: Dictionary.
- $X \in \mathbb{R}^{k \times m}$: mixing matrix.
- Linear model: $Y = AX$.
- Sparse mixing $X$ and Incoherent dictionary $A$.
- Clustering and alt. min.

**Sparse Topic Models**
- $y \in \mathbb{R}^n$: word. $m$ documents.
- $A = [a_1, \ldots, a_k]$: Topic-word matrix.
- $x \in \mathbb{R}^k$: Topic proportions vector
- Linear model: $\mathbb{E}[y|x] = Ax$.
- $y_1, y_2, y_3$: Three words/views
Two Approaches for Learning Overcomplete Models

- Latent dimensionality $k \gg$ observed dimensionality $n$.

**Dictionary Learning**
- $y \in \mathbb{R}^n$: sample. $Y \in \mathbb{R}^{n \times m}$.
- $A = [a_1, \ldots, a_k]$: Dictionary.
- $X \in \mathbb{R}^{k \times m}$: mixing matrix.
- Linear model: $Y = AX$.
- Sparse mixing $X$ and Incoherent dictionary $A$.
- Clustering and alt. min.

**Sparse Topic Models**
- $y \in \mathbb{R}^n$: word. $m$ documents.
- $A = [a_1, \ldots, a_k]$: Topic-word matrix.
- $x \in \mathbb{R}^k$: Topic proportions vector
- Linear model: $\mathbb{E}[y|x] = Ax$.
- $y_1, y_2, y_3$: Three words/views
- $\mathbb{E}[y_1 \otimes y_2 \otimes y_3]$: multi-linear map.
Two Approaches for Learning Overcomplete Models

- Latent dimensionality $k \gg$ observed dimensionality $n$.

Dictionary Learning
- $y \in \mathbb{R}^n$: sample. $Y \in \mathbb{R}^{n \times m}$.
- $A = [a_1, \ldots, a_k]$: Dictionary.
- $X \in \mathbb{R}^{k \times m}$: mixing matrix.
- Linear model: $Y = AX$.
- Sparse mixing $X$ and Incoherent dictionary $A$.
- Clustering and alt. min.

Sparse Topic Models
- $y \in \mathbb{R}^n$: word. $m$ documents.
- $A = [a_1, \ldots, a_k]$: Topic-word matrix.
- $x \in \mathbb{R}^k$: Topic proportions vector.
- Linear model: $\mathbb{E}[y|x] = Ax$.
- $y_1, y_2, y_3$: Three words/views
- $\mathbb{E}[y_1 \otimes y_2 \otimes y_3]$: multi-linear map.
- Multi-view and Persistent topics
Dictionary Learning or Sparse Coding

- Each sample is a \textit{sparse} combination of dictionary atoms.
Dictionary Learning or Sparse Coding

- Each sample is a **sparse** combination of dictionary atoms.

**Setup**

- No. of dictionary elements $k >$ observed dimensionality $n$.
- $A = [a_1, \ldots, a_k]$: dictionary elements
- $y \in \mathbb{R}^n$: Observation. $Y = [y_1, \ldots, y_m] \in \mathbb{R}^{n \times m}$: Observation matrix.
- Linear model: $Y = AX$.
- Learning problem: Given $Y$, find $A$ and $X$.

**Ill-posed without further constraints**
Dictionary Learning or Sparse Coding

Each sample is a **sparse** combination of dictionary atoms.

**Setup**

- No. of dictionary elements $k >$ observed dimensionality $n$.
- $A = [a_1, \ldots, a_k]$: dictionary elements
- $y \in \mathbb{R}^n$: Observation. $Y = [y_1, \ldots, y_m] \in \mathbb{R}^{n \times m}$: Observation matrix.
- Linear model: $Y = AX$.
- Learning problem: Given $Y$, find $A$ and $X$.

Ill-posed without further constraints

**Main Assumptions**

- $X$ is **sparse**: each column is randomly $s$-sparse
  
  Each sample is a combination of $s$ dictionary atoms.
Dictionary Learning or Sparse Coding

- Each sample is a *sparse* combination of dictionary atoms.

**Setup**

- No. of dictionary elements $k >$ observed dimensionality $n$.
- $A = [a_1, \ldots, a_k]$: dictionary elements
- $y \in \mathbb{R}^n$: Observation. $Y = [y_1, \ldots, y_m] \in \mathbb{R}^{n \times m}$: Observation matrix.
- Linear model: $Y = AX$.
- Learning problem: Given $Y$, find $A$ and $X$.

Ill-posed without further constraints

**Main Assumptions**

- $X$ is *sparse*: each column is randomly $s$-sparse
  - Each sample is a combination of $s$ dictionary atoms.
- $A$ is incoherent: $\max_{i \neq j} |\langle a_i, a_j \rangle| \approx 0$. 

Intuitions: how incoherence helps

- Each sample is a combination of dictionary atoms: \( y_i = \sum_j x_{i,j} a_j \).
- Consider \( y_i \) and \( y_j \) s.t. they have no common dictionary atoms.
- What about \( |\langle y_i, y_j \rangle| \)?
Intuitions: how incoherence helps

- Each sample is a combination of dictionary atoms: \( y_i = \sum_j x_{i,j} a_j \).
- Consider \( y_i \) and \( y_j \) s.t. they have no common dictionary atoms.
- What about \( |\langle y_i, y_j \rangle| \)?
- Under incoherence: \( |\langle y_i, y_j \rangle| \approx 0 \).
Intuitions: how incoherence helps

- Each sample is a combination of dictionary atoms: \( y_i = \sum_j x_{i,j} a_j \).
- Consider \( y_i \) and \( y_j \) s.t. they have no common dictionary atoms.
- What about \(|\langle y_i, y_j \rangle|\)?
- Under incoherence: \(|\langle y_i, y_j \rangle| \approx 0\).

Construction of Correlation Graph

- Nodes: Samples \( y_1, \ldots, y_n \).
- Edges: \(|\langle y_i, y_j \rangle| > \tau\) for some threshold \( \tau \).

How does the correlation graph help in dictionary learning?
Main Insight

- \((y_i, y_j)\): edge in correlation graph \(\Rightarrow\) \(y_i\) and \(y_j\) have at least one dictionary element in common.
**Main Insight**

- \((y_i, y_j)\): edge in correlation graph \(\Rightarrow y_i\) and \(y_j\) have at least one dictionary element in common.
Main Insight

- \((y_i, y_j)\): edge in correlation graph \(\Rightarrow y_i \text{ and } y_j \text{ have at least one dictionary element in common.}\)

- Consider a large clique: a large fraction of pairs have exactly one element in common.
Main Insight

- \((y_i, y_j)\): edge in correlation graph \Rightarrow y_i and y_j have at least one dictionary element in common.

- Consider a large clique: a large fraction of pairs have exactly one element in common.

- How to find such a large clique efficiently?
Main Insight

- \((y_i, y_j)\): edge in correlation graph \(\Rightarrow\) \(y_i\) and \(y_j\) have at least one dictionary element in common.
- Consider a large clique: a large fraction of pairs have exactly one element in common.
- How to find such a large clique efficiently? Start with a random edge.
**Main Insight**

- \((y_i, y_j)\): edge in correlation graph \(\Rightarrow y_i\) and \(y_j\) have at least one dictionary element in common.

- Consider a large **clique**: a large fraction of pairs have exactly one element in common.

- How to find such a large clique efficiently? Start with a **random edge**.
Result on Approximate Dictionary Estimation

Procedure

- Start with a random edge \((y_i^*, y_j^*)\).
- \(\hat{S}\) = common nbd. of \(y_i^*\) and \(y_j^*\). If \(\hat{S}\) is close to a clique, accept.
- Estimate a dictionary element via top singular vector of \(\sum_{i \in \hat{S}} y_i y_i^\top\).

Theorem

The dictionary \(A\) can be estimated with bounded error w.h.p. when \(s = o(k^{1/3})\) and number of samples \(m = \omega(k)\).

- Exact estimation when \(X\) is discrete, e.g. Bernoulli.

Exact Estimation via Alternating Minimization

- So far.. approximate dictionary estimation. What about exact estimation for arbitrary $X$?
Exact Estimation via Alternating Minimization

- So far.. approximate dictionary estimation. What about exact estimation for arbitrary $X$?

Alternating Minimization

- Given $Y = AX$, initialize an estimate for $A$.
- Update $X$ via $\ell_1$ optimization.
- Re-estimate $A$ via Least Squares.
Exact Estimation via Alternating Minimization

- So far.. approximate dictionary estimation. What about exact estimation for arbitrary $X$?

Alternating Minimization

- Given $Y = AX$, initialize an estimate for $A$.
- Update $X$ via $\ell_1$ optimization.
- Re-estimate $A$ via Least Squares.

- In general, alternating minimization converges to a local optimum.
Exact Estimation via Alternating Minimization

- So far.. approximate dictionary estimation. What about exact estimation for arbitrary $X$?

Alternating Minimization

- Given $Y = AX$, initialize an estimate for $A$.
- Update $X$ via $\ell_1$ optimization.
- Re-estimate $A$ via Least Squares.

- In general, alternating minimization converges to a local optimum.

Specific Initialization: Through our previous method.
Exact Estimation via Alternating Minimization

- So far.. approximate dictionary estimation. What about exact estimation for arbitrary $X$?

Alternating Minimization

- Given $Y = AX$, initialize an estimate for $A$.
- Update $X$ via $\ell_1$ optimization.
- Re-estimate $A$ via Least Squares.

- In general, alternating minimization converges to a local optimum.

Specific Initialization: Through our previous method.

Theorem

The above method converges to the true solution $(A, X)$ at a linear rate w.h.p. when $s < \min(k^{1/8}, n^{1/9})$ and number of samples $m = \Omega(k^2)$.

Relationship to Previous Results

Previous Results on Guaranteed Recovery
- Spielman et. al. : guaranteed recovery of undercomplete dictionaries.
- Arora et. al: concurrent results for approximate dictionary estimation.

Our Result
- First guarantees for exact recovery of overcomplete dictionary.
- Validates some of empirical success of alternating minimization.
- Propose a new method for initialization.

Simple Methods for Guaranteed Recovery of Overcomplete Dictionaries
Outline

1. Introduction
2. Dictionary Learning
3. Topic Models
4. Conclusion
Two Approaches for Learning Overcomplete Models

- Latent dimensionality $k \gg$ observed dimensionality $n$.

**Dictionary Learning**
- $y \in \mathbb{R}^n$: sample. $Y \in \mathbb{R}^{n \times m}$.
- $A = [a_1, \ldots, a_k]$: Dictionary.
- $X \in \mathbb{R}^{k \times m}$: mixing matrix.
- Linear model: $Y = AX$.
- Sparse mixing $X$ and Incoherent dictionary $A$.
- Clustering and alt. min.

**Sparse Topic Models**
- $y \in \mathbb{R}^n$: word. $m$ documents.
- $A = [a_1, \ldots, a_k]$: Topic-word matrix.
- $x \in \mathbb{R}^k$: Topic proportions vector
- Linear model: $\mathbb{E}[y|x] = Ax$.
- $y_1, y_2, y_3$: Three words/views
- $\mathbb{E}[y_1 \otimes y_2 \otimes y_3]$: multi-linear map.
- Multi-view and Persistent topics
Probabilistic Topic Models

- **Bag of words**: order of words does not matter

**Graphical model representation**

- \( y \in \mathbb{R}^n \): word. \( l \) words in a document.
- \( x \in \mathbb{R}^k \): topic proportions in document.

**Exchangeability**: \( y_1 \perp \perp y_2 \perp \perp \ldots | x \)

- Word \( y_i \) generated from topic \( z_i \).
- Topic \( z_i \) drawn from mixture \( x \).

- \( A(i, j) := \mathbb{P}[y = i|z = j] \): topic-word matrix.

- Linear model: \( \mathbb{E}[y_i|x] = Ax \).
Formulation as Linear Models

Distribution of the topic proportions vector $x$

If there are $k$ topics, distribution over the simplex $\Delta^{k-1}$

$$\Delta^{k-1} := \{ x \in \mathbb{R}^k, x_i \in [0, 1], \sum_i x_i = 1 \}. $$

Distribution of the words $y_1, y_2, \ldots$

- $n$ words in vocabulary. If $y_1$ is $j^{th}$ word, assign $e_j \in \mathbb{R}^n$
- Distribution of each $y_i$: supported on vertices of $\Delta^{n-1}$

Properties

- Linear Model: $\mathbb{E}[y_i | x] = Ax$

- Multiview model: $x$ is fixed and multiple words $(y_i)$ are generated.
Geometric Picture for Topic Models

Topic proportions vector \((x)\)
Geometric Picture for Topic Models

Single topic \((x)\)
Geometric Picture for Topic Models

Topic proportions vector \((x)\)
Geometric Picture for Topic Models

Topic proportions vector \( (x) \)

Word generation \( (y_1, y_2, \ldots) \)
Geometric Picture for Topic Models

Topic proportions vector \((x)\)

Word generation \((y_1, y_2, \ldots)\)

Moment-based estimation: co-occurrences of words in documents
Learning Topic Models

Exchangeable Topic Model

Topic

Mixture

Topics

Words
Allow for general $x$: model arbitrary topic correlations

Constrain topic-word matrix $A$: 
Learning Topic Models

Exchangeable Topic Model

- Allow for general $x$: model arbitrary topic correlations
- Constrain topic-word matrix $A$: Sparsity constraints
Learning Topic Models

Exchangeable Topic Model

- Topic
- Mixture
- Topics
- Words

Topic-word matrix

- Allow for general $x$: model arbitrary topic correlations
- Constrain topic-word matrix $A$: Sparsity constraints
Learning Overcomplete Representations

- Latent dimensionality $k$ and observed dimensionality $n$.

Undercomplete Representation

Overcomplete Representation

When are overcomplete models ($k > n$) learnable?
Moments of a topic model

Linear model: \[ \mathbb{E}[y_i | x] = Ax. \]

Tucker Form of Moments for Topic Models

\[ M_2 := \mathbb{E}(y_1 \otimes y_2) = A \mathbb{E}[xx^\top] A^\top \]
Moments of a topic model

Linear model: $\mathbb{E}[y_i | x] = Ax.$

Tucker Form of Moments for Topic Models

$M_2 := \mathbb{E}(y_1 \otimes y_2) = A \mathbb{E}[xx^\top] A^\top$

$M_4 := \mathbb{E}((y_1 \otimes y_2)(y_3 \otimes y_4)^\top) = (A \otimes A) \mathbb{E}[(x \otimes x)(x \otimes x)^\top](A \otimes A)^\top$
Moments of a topic model

Linear model: \[ \mathbb{E}[y_i | x] = Ax. \]

Tucker Form of Moments for Topic Models

\[ M_2 := \mathbb{E}(y_1 \otimes y_2) = A \mathbb{E}[xx^\top] A^\top \]

\[ M_4 := \mathbb{E}((y_1 \otimes y_2)(y_3 \otimes y_4)^\top) = (A \otimes A) \mathbb{E}[(x \otimes x)(x \otimes x)^\top] (A \otimes A)^\top \]

- Kronecker product: \((A \otimes A) \in \mathbb{R}^{n^2 \times k^2}\)
- \(k > n\): Tucker decomposition not unique: model non-identifiable.
Moments of a topic model

Linear model: \[ \mathbb{E}[y_i|x] = Ax. \]

Tucker Form of Moments for Topic Models

\[ M_2 := \mathbb{E}(y_1 \otimes y_2) = A \mathbb{E}[xx^\top]A^\top \]

\[ M_4 := \mathbb{E}((y_1 \otimes y_2)(y_3 \otimes y_4)^\top) = (A \otimes A)\mathbb{E}[(x \otimes x)(x \otimes x)^\top](A \otimes A)^\top \]

- Kronecker product: \( (A \otimes A) \in \mathbb{R}^{n^2 \times k^2} \)
- \( k > n \): Tucker decomposition not unique: model non-identifiable.

Identifiability of Overcomplete Models

- Possible under the notion of topic persistence
- Includes single topic model as a special case.
Persistent Topic Models

Bag of Words Model

Topic Mixture

Topics

Words

\( y_1 \) \( y_2 \) \( y_3 \) \( y_4 \) \( y_5 \)

\( z_1 \) \( z_2 \) \( z_3 \) \( z_4 \) \( z_5 \)

\( A \) \( A \) \( A \) \( A \) \( A \)
Persistent Topic Models

Bag of Words Model

Persistent Topic Model

Topic Mixture

Topics

Words

Words
Single-topic model is a special case.

Persistence: incorporates locality or order of words.
Identifiability of Overcomplete Models

Recall Form of Moments for Bag-of-Words Model

\[ \mathbb{E}((y_1 \otimes y_2)(y_3 \otimes y_4)^\top) = (A \otimes A)\mathbb{E}[(x \otimes x)(x \otimes x)^\top](A \otimes A)^\top \]
Identifiability of Overcomplete Models

Recall Form of Moments for Bag-of-Words Model

\[ \mathbb{E}\left((y_1 \otimes y_2)(y_3 \otimes y_4)^\top\right) = (A \otimes A)\mathbb{E}\left[(x \otimes x)(x \otimes x)^\top\right](A \otimes A)^\top \]

For Persistent Topic Model

\[ \mathbb{E}\left((y_1 \otimes y_2)(y_3 \otimes y_4)^\top\right) = (A \odot A)\mathbb{E}\left[xx^\top\right](A \odot A)^\top \]
Identifiability of Overcomplete Models

Recall Form of Moments for Bag-of-Words Model

\[ \mathbb{E}((y_1 \otimes y_2)(y_3 \otimes y_4)^\top) = (A \otimes A)\mathbb{E}[(x \otimes x)(x \otimes x)^\top](A \otimes A)^\top \]

For Persistent Topic Model

\[ \mathbb{E}((y_1 \otimes y_2)(y_3 \otimes y_4)^\top) = (A \odot A)\mathbb{E}[xx^\top](A \odot A)^\top \]

Kronecker vs. Khatri-Rao Products

- **A**: Topic-word matrix, is \( n \times k \).
- \((A \otimes A)\): Kronecker product, is \( n^2 \times k^2 \) matrix.
- \((A \odot A)\): Khatri-Rao product, is \( n^2 \times k \) matrix.
Some Intuitions

- **Bag-of-words Model:**
  \[(A \otimes A)E[(x \otimes x)(x \otimes x)^\top](A \otimes A)^\top.\]

- **Persistent Model:**
  \[(A \odot A)E[xx^\top](A \odot A)^\top.\]

**Effective Topic-Word Matrix Given Fourth-Order Moments:**

- **Bag of Words Model:**
  Kronecker Product \(A \otimes A\).

- **Persistent Model:**
  Khatri-Rao Product \(A \odot A\).

Not Identifiable.

Identifiable
Identifiability of Overcomplete Topic Models

- \( A \in \mathbb{R}^{n \times k} \): topic-word matrix.
- Each topic has number of words (degree) \( \in [\log n, \sqrt{n}] \).
- Random connections in \( A \).
- Number of topics \( k = O(n^2) \).

**Corollary**

The above topic model is identifiable from \( M_4 \) when topic persistence level is at least 2.

- Learning: via \( \ell_1 \) optimization.

---

Outline

1. Introduction
2. Dictionary Learning
3. Topic Models
4. Conclusion
Conclusion

Learning Overcomplete Representations
- More flexibility in modeling, robust to noise
- Exploit availability of large number of unlabelled samples, e.g. speech, vision etc

Dictionary Learning/Sparse Coding
- Each sample is a sparse combination of dictionary atoms.
- Guaranteed learning through clique finding and alternating minimization.

Learning Sparse Overcomplete Topic Models
- Learning using higher order moments
- Identifiability under persistence of topics
- Learning via $\ell_1$ optimization.