What’s hot in Machine Learning?

Anima Anandkumar

U.C. Irvine
Feature Engineering

- Learn good features/representations for classification tasks, e.g. image and speech recognition.
- Sparse representations, low dimensional hidden structures.
What’s hot in ML: Optimization Methods

Convex Optimization
- Fast convergence for non-smooth (and not strongly convex) functions.
- Online learning: variance reduction for stochastic gradient methods.

Non-convex problems
- When can we hope to reach global optimum?
- What problem structures make this possible?
- Can we have fast convergence?
Challenges in Feature Learning

In practice
- Deep learning has provided impressive gains.
- Parameter training challenging and not stable.

Theory
- Representational power of networks.
- Guaranteed learning of probabilistic models with latent variables?
- Maximum likelihood is NP-hard.
- Practice: EM, Variational Bayes have no consistency guarantees.
- Efficient computational and sample complexities?
Outline

1. Introduction

2. Representation Learning

3. Tensor Methods for Guaranteed Learning

4. Conclusion
Linear Neural Networks

- Observed sample $x = Ah$.
- $h$ is hidden variable and $A$ is dictionary.
- $x \in \mathbb{R}^d$, $h \in \mathbb{R}^k$ and $A \in \mathbb{R}^{d \times k}$.

Learning through SVD

- Pairwise moments: $M_2 = \mathbb{E}[xx^\top] = A\mathbb{E}[hh^\top]A^\top$.
- SVD: $M_2 = U\Lambda U^\top$: a valid linear representation.

- Learning through SVD: cannot learn overcomplete representations. ($k > d$) learnable?
- SVD cannot enforce sparsity, non-negativity etc.
Linear Neural Networks

- Observed sample $x = Ah$.
- $h$ is hidden variable and $A$ is dictionary.
- $x \in \mathbb{R}^d$, $h \in \mathbb{R}^k$ and $A \in \mathbb{R}^{d \times k}$.

Learning through SVD

- Pairwise moments: $M_2 = \mathbb{E}[xx^\top] = A\mathbb{E}[hh^\top]A^\top$.
- SVD: $M_2 = U\Lambda U^\top$: a valid linear representation.

Learning through SVD: cannot learn overcomplete representations. $(k > d)$ learnable?

SVD cannot enforce sparsity, non-negativity etc.
Learning Overcomplete Dictionaries

\[ X \in \mathbb{R}^{d \times n} \quad A \in \mathbb{R}^{d \times k} \quad H \in \mathbb{R}^{k \times n} \]

- **Linear model:** \( X = AH \), both \( A, H \) unknown.
- **Sparse** \( H \): each column is randomly \( s \)-sparse
- **Overcomplete dictionary** \( A \in \mathbb{R}^{d \times k} : k \geq d \).
- **Incoherence:** \( \max_{i \neq j} |\langle a_i, a_j \rangle| \approx 0 \). (satisfied by random vectors)

"Learning Sparsely Used Overcomplete Dictionaries" by A. Agarwal, A., P. Jain, P. Netrapalli, R. Tandon. COLT 2014.
Intuitions: how incoherence helps

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

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

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

**Construction of Correlation Graph**

- Nodes: Samples \( x_1, \ldots, x_n \).
- Edges: \( |\langle x_i, x_j \rangle| > \tau \) for some threshold \( \tau \).

How does the correlation graph help in dictionary learning?
Correlation Graph and Clique Finding

Main Insight

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

- \((x_i, x_j)\): edge in correlation graph \(\Rightarrow x_i \text{ and } x_j \text{ have at least one dictionary element in common.}\)
Main Insight

- \((x_i, x_j)\): edge in correlation graph \(\Rightarrow\) \(x_i\) and \(x_j\) 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

- \((x_i, x_j)\): edge in correlation graph \(\Rightarrow\) \(x_i\) and \(x_j\) 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

- \((x_i, x_j)\): edge in correlation graph \(\Rightarrow x_i\) and \(x_j\) 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

- $(x_i, x_j)$: edge in correlation graph ⇒ $x_i$ and $x_j$ 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

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

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

- How to find such a large clique efficiently? Start with a random edge.

- Refinement through alternating minimization.
Experiments on MNIST

Original

Reconstruction

Learnt Representation
Outline

1. Introduction
2. Representation Learning
3. Tensor Methods for Guaranteed Learning
4. Conclusion
Warm-up: PCA on Gaussian Mixtures

- Mixture of Spherical Gaussians.
- PCA on pairwise moments: span of mean vectors.

Learning Mean Vectors through Spectral Clustering

- Project samples on to span of mean vectors.
- Distance-based clustering (e.g. \(k\)-means).
Warm-up: PCA on Gaussian Mixtures

- Mixture of Spherical Gaussians.
- PCA on pairwise moments: span of mean vectors.

Learning Mean Vectors through Spectral Clustering
- Project samples on to span of mean vectors.
- Distance-based clustering (e.g. \( k \)-means).

Failure to cluster under large variance.

Learning Gaussian Mixtures Without Separation Constraints?
Beyond PCA: Spectral Methods on Tensors

- How to learn the component means (not just its span) without separation constraints?
- PCA is a spectral method on (covariance) matrices.
  - Are higher order moments helpful?
Beyond PCA: Spectral Methods on Tensors

- How to learn the component means (not just its span) without separation constraints?
- PCA is a spectral method on (covariance) matrices.
  - Are higher order moments helpful?
- What if number of components is greater than observed dimensionality $k > d$?
Beyond PCA: Spectral Methods on Tensors

- How to learn the component means (not just its span) without separation constraints?
- PCA is a spectral method on (covariance) matrices.
  - Are higher order moments helpful?

- What if number of components is greater than observed dimensionality $k > d$?
  - Do higher order moments help to learn overcomplete models?
Beyond PCA: Spectral Methods on Tensors

- How to learn the component means (not just its span) without separation constraints?
- PCA is a spectral method on (covariance) matrices.
  - Are higher order moments helpful?
- What if number of components is greater than observed dimensionality $k > d$?
  - Do higher order moments help to learn overcomplete models?
- What if the data is not Gaussian?
How to learn the component means (not just its span) without separation constraints?

PCA is a spectral method on (covariance) matrices.
  ▶ Are higher order moments helpful?

What if number of components is greater than observed dimensionality $k > d$?
  ▶ Do higher order moments help to learn overcomplete models?

What if the data is not Gaussian?
  ▶ Moment-based Estimation of probabilistic latent variable models?
Multi-variate higher order moments form tensors.

Are there spectral operations on tensors akin to PCA?

Matrix

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

Tensor

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

$$M_2 \approx \sum_i \lambda_i u_i \otimes v_i$$

Matrix $M_2$

$\lambda_1 u_1 \otimes v_1$

$\lambda_2 u_2 \otimes v_2$
Matrices vs. Tensors

Matrix $M_2 \approx \sum_i \lambda_i u_i \otimes v_i$

$= \lambda_1 u_1 \otimes v_1 + \lambda_2 u_2 \otimes v_2 + \ldots$

Tensor $M_3 \approx \sum_i \lambda_i u_i \otimes v_i \otimes w_i$

$= \lambda_1 u_1 \otimes v_1 \otimes w_1 + \lambda_2 u_2 \otimes v_2 \otimes w_2 + \ldots$
Topic Modeling

$k$ topics (distributions over vocab words).

Each document $\leftrightarrow$ mixture of topics.

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

E.g.,

$0.6 \cdot \text{sports} + 0.3 \cdot \text{science} + 0.1 \cdot \text{politics} + 0.0 \cdot \text{business}$

Pr$_\theta$[“play” | sports] = 0.0002
Pr$_\theta$[“game” | sports] = 0.0003
Pr$_\theta$[“season” | sports] = 0.0001

aardvark 0
athlete 3
zygote 1
Tensor Factorizations for Other Models

HMM

ICA

Latent Trees

Method of Moments: Analyze moment tensors under statistical models.

Tensor methods can find overlapping communities in networks
Experimental Results

Friend
Users
Friend

Facebook
$n \sim 20,000$

Business
User
Reviews

Yelp
$n \sim 40,000$

Author
Coauthor

DBLP
$n \sim 1$ million

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$\hat{k}$</th>
<th>Method</th>
<th>Running Time</th>
<th>$\mathcal{E}$</th>
<th>$\mathcal{R}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook ($k=360$)</td>
<td>500</td>
<td>ours</td>
<td>468</td>
<td>0.0175</td>
<td>100%</td>
</tr>
<tr>
<td>Facebook ($k=360$)</td>
<td>500</td>
<td>variational</td>
<td>86,808</td>
<td>0.0308</td>
<td>100%</td>
</tr>
<tr>
<td>Yelp ($k=159$)</td>
<td>100</td>
<td>ours</td>
<td>287</td>
<td>0.046</td>
<td>86%</td>
</tr>
<tr>
<td>Yelp ($k=159$)</td>
<td>100</td>
<td>variational</td>
<td>N.A.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBLP ($k=6000$)</td>
<td>100</td>
<td>ours</td>
<td>5407</td>
<td>0.105</td>
<td>95%</td>
</tr>
</tbody>
</table>

**Experimental Results on Yelp**

**Lowest error business categories & largest weight businesses**

<table>
<thead>
<tr>
<th>Rank</th>
<th>Category</th>
<th>Business</th>
<th>Stars</th>
<th>Review Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Latin American</td>
<td>Salvadoreno Restaurant</td>
<td>4.0</td>
<td>36</td>
</tr>
<tr>
<td>2</td>
<td>Gluten Free</td>
<td>P.F. Chang's China Bistro</td>
<td>3.5</td>
<td>55</td>
</tr>
<tr>
<td>3</td>
<td>Hobby Shops</td>
<td>Make Meaning</td>
<td>4.5</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>Mass Media</td>
<td>KJZZ 91.5FM</td>
<td>4.0</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>Yoga</td>
<td>Sutra Midtown</td>
<td>4.5</td>
<td>31</td>
</tr>
</tbody>
</table>
Experimental Results on Yelp

Lowest error business categories & largest weight businesses

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

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

Top-5 bridging nodes (businesses)

<table>
<thead>
<tr>
<th>Business</th>
<th>Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Four Peaks Brewing</td>
<td>Restaurants, Bars, American, Nightlife, Food, Pubs, Tempe</td>
</tr>
<tr>
<td>Pizzeria Bianco</td>
<td>Restaurants, Pizza, Phoenix</td>
</tr>
<tr>
<td>FEZ</td>
<td>Restaurants, Bars, American, Nightlife, Mediterranean, Lounges, Phoenix</td>
</tr>
<tr>
<td>Matt’s Big Breakfast</td>
<td>Restaurants, Phoenix, Breakfast&amp; Brunch</td>
</tr>
<tr>
<td>Cornish Pasty Co</td>
<td>Restaurants, Bars, Nightlife, Pubs, Tempe</td>
</tr>
</tbody>
</table>
Conclusion

Guaranteed Learning of Latent Variable Models

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

- Tensor approach: mixed membership communities, topic models, latent trees...
- Sparsity-based approach: overcomplete models, e.g. sparse coding and topic models.

http://newport.eecs.uci.edu/anandkumar/MLSS.html