Guaranteed Non-Convex Optimization in Machine Learning

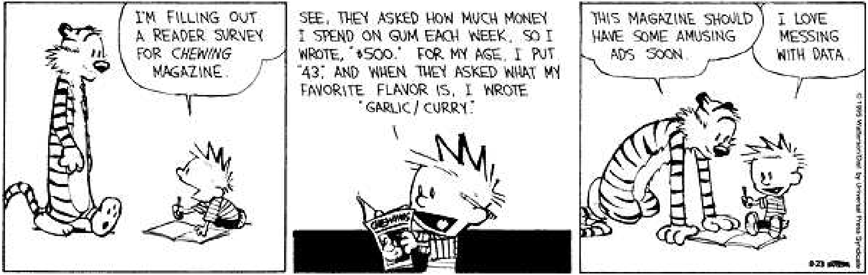
Anima Anandkumar

U.C. Irvine

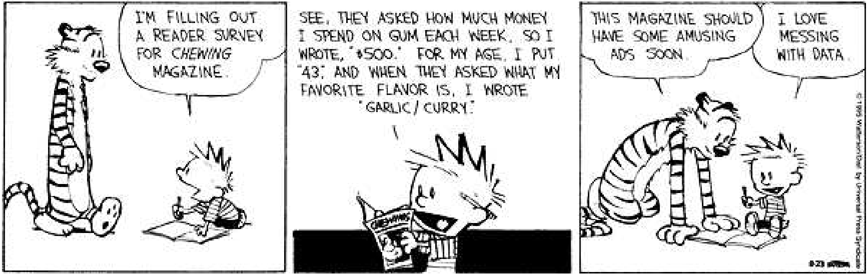
Learning with Big Data

Data vs. Information

Data vs. Information

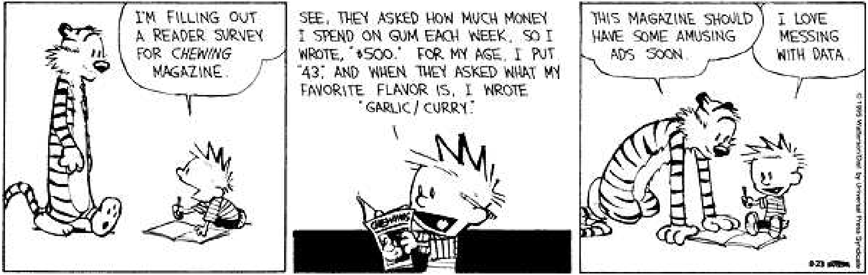


Data vs. Information



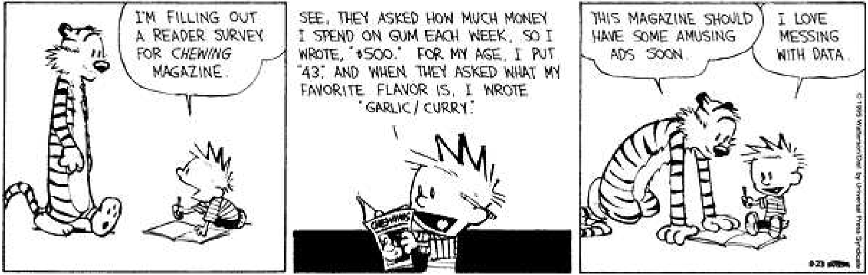
- Missing observations, gross corruptions, outliers.

Data vs. Information



- Missing observations, gross corruptions, outliers.
- High dimensional regime: as data grows, more variables !

Data vs. Information



- Missing observations, gross corruptions, outliers.
- High dimensional regime: as data grows, more variables !

Data deluge also a data desert!

Learning in High Dimensional Regime

- Useful information: low-dimensional structures.
- Learning with big data: ill-posed problem.

Learning in High Dimensional Regime

- Useful information: low-dimensional structures.
- Learning with big data: ill-posed problem.

Learning is finding needle in a haystack

Learning in High Dimensional Regime

- Useful information: low-dimensional structures.
- Learning with big data: ill-posed problem.

Learning is finding needle in a haystack

- Learning with big data: computationally challenging!

Principled approaches for finding low dimensional structures?

How to model information structures?

Latent variable models

- Incorporate hidden or latent variables.
- Information structures: Relationships between latent variables and observed data.

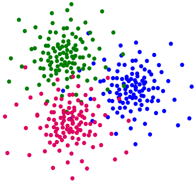
How to model information structures?

Latent variable models

- Incorporate hidden or latent variables.
- Information structures: Relationships between latent variables and observed data.

Basic Approach: mixtures/clusters

- Hidden variable is categorical.



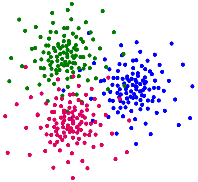
How to model information structures?

Latent variable models

- Incorporate hidden or latent variables.
- Information structures: Relationships between latent variables and observed data.

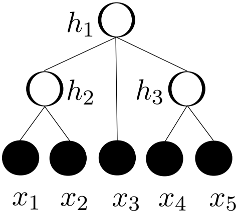
Basic Approach: mixtures/clusters

- Hidden variable is categorical.



Advanced: Probabilistic models

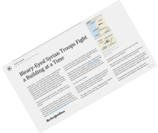
- Hidden variables have more general distributions.
- Can model mixed membership/hierarchical groups.



Latent Variable Models (LVMs)

Document modeling

- Observed: words.
- Hidden: topics.

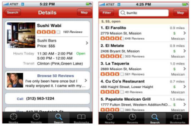


Social Network Modeling

- Observed: social interactions.
- Hidden: communities, relationships.

Recommendation Systems

- Observed: recommendations (e.g., reviews).
- Hidden: User and business attributes



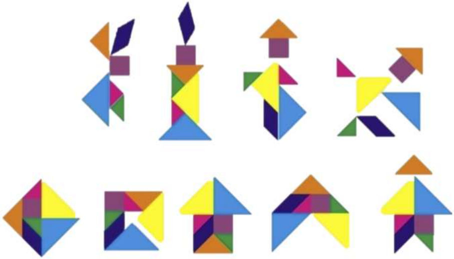
Unsupervised Learning: Learn LVM without labeled examples.

LVM for Feature Engineering

- Learn good features/representations for classification tasks, e.g., computer vision and NLP.

Sparse Coding/Dictionary Learning

- Sparse representations, low dimensional hidden structures.
- A few dictionary elements make complicated shapes.



Challenges in Learning LVMs

Computational Challenges

- Maximum likelihood is NP-hard in most scenarios: non-convex optimization.
- Practice: Local search approaches such as gradient descent, EM, Variational Bayes have no consistency guarantees.

Sample Complexity

- Sample complexity is exponential (w.r.t hidden variable dimension) for many learning methods.

Guaranteed and efficient learning through non-convex methods?

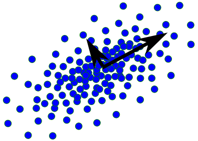
Outline

Classical Spectral Methods: Matrix PCA and CCA

Single-view Setting: PCA

For centered samples $\{x_i\}$, find projection P with $\text{Rank}(P) = k$ s.t.

$$\boxed{\min_P \frac{1}{n} \sum_{i \in [n]} \|x_i - P x_i\|^2.}$$

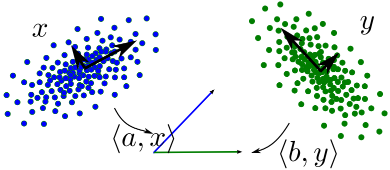


Result: Eigen-decomposition of $S = \text{Cov}(X)$.

Multiview Setting: CCA

For centered samples $\{x_i, y_i\}$, find

$$\boxed{\max_{a,b} \frac{a^\top \hat{\mathbb{E}}[xy^\top] b}{\sqrt{a^\top \hat{\mathbb{E}}[xx^\top] a \; b^\top \hat{\mathbb{E}}[yy^\top] b}}.}$$



Result: Generalized eigen decomposition.

Beyond SVD: Spectral Methods on Tensors

- How to learn the mixture models without separation constraints?
  - PCA uses covariance matrix of data. Are higher order moments helpful?
- Unified framework?
  - Moment-based estimation of probabilistic latent variable models?
- SVD gives spectral decomposition of matrices.
  - What are the analogues for tensors?

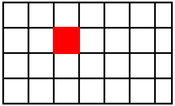
Moment Matrices and Tensors

Multivariate Moments

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

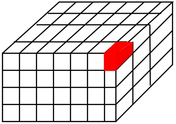
Matrix

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



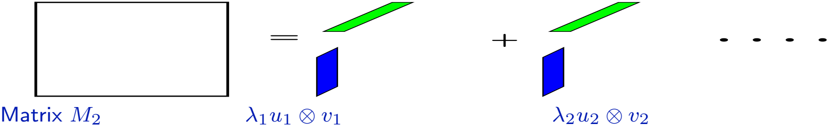
Tensor

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



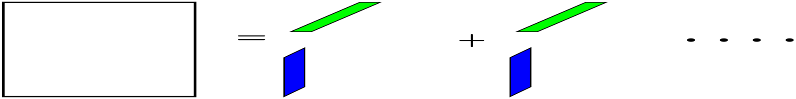
Spectral Decomposition of Tensors

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



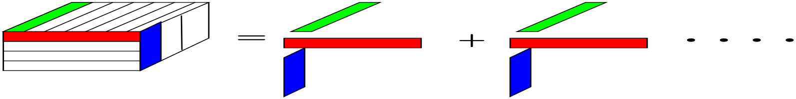
Spectral Decomposition of Tensors

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



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 $u_{i_1} v_{i_2} w_{i_3}$.

How to solve this non-convex problem?

Decomposition of Orthogonal Tensors

$$\boxed{M_3 = \sum_i w_i a_i \otimes a_i \otimes a_i.}$$

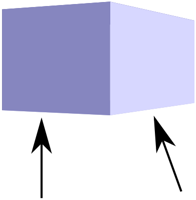
- Suppose A has orthogonal columns.

Decomposition of Orthogonal Tensors

$$\boxed{M_3 = \sum_i w_i a_i \otimes a_i \otimes a_i.}$$

- Suppose A has orthogonal columns.

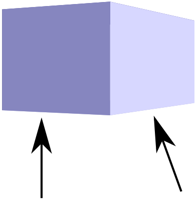
- $M_3(I, a_1, a_1) = \sum_i w_i \langle a_i, a_1 \rangle^2 a_i = w_1 a_1.$



Decomposition of Orthogonal Tensors

$$\boxed{M_3 = \sum_i w_i a_i \otimes a_i \otimes a_i.}$$

- Suppose A has orthogonal columns.

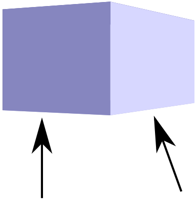


- $M_3(I, a_1, a_1) = \sum_i w_i \langle a_i, a_1 \rangle^2 a_i = w_1 a_1.$
- a_i are eigenvectors of tensor M_3.
- Analogous to matrix eigenvectors: $Mv = M(I, v) = \lambda v.$

Decomposition of Orthogonal Tensors

$$\boxed{M_3 = \sum_i w_i a_i \otimes a_i \otimes a_i.}$$

- Suppose A has orthogonal columns.



- $M_3(I, a_1, a_1) = \sum_i w_i \langle a_i, a_1 \rangle^2 a_i = w_1 a_1.$
- a_i are eigenvectors of tensor M_3.
- Analogous to matrix eigenvectors: $Mv = M(I, v) = \lambda v.$

Two Problems

- How to find eigenvectors of a tensor?
- A is not orthogonal in general.

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: $v \mapsto \dfrac{M(I,v)}{\|M(I,v)\|}$.

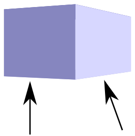
Orthogonal Tensor Power Method

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



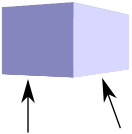
Orthogonal Tensor Power Method

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: $v \mapsto \dfrac{M(I,v)}{\|M(I,v)\|}$.

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.

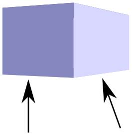
Orthogonal Tensor Power Method

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.

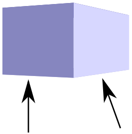
Orthogonal Tensor Power Method

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.

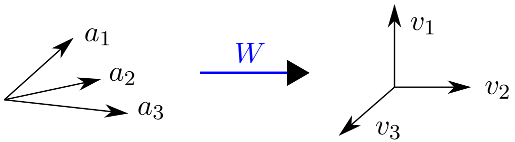
- All other eigenvectors are saddle points.

For an orthogonal tensor, no spurious local optima!

Whitening: Conversion to Orthogonal Tensor

$$M_3 = \sum_i w_i a_i \otimes a_i \otimes a_i, \quad M_2 = \sum_i w_i a_i \otimes a_i.$$

- Find whitening matrix W s.t. $W^\top A = V$ is an orthogonal matrix.
- When $A \in \mathbb{R}^{d \times k}$ has full column rank, it is an invertible transformation.

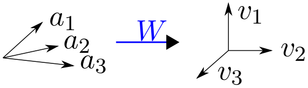


- Use pairwise moments M_2 to find W.
- SVD of M_2 is needed.

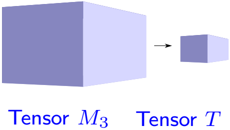
Putting it together

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: $T = M_3(W, W, W)$

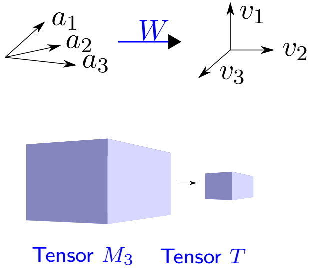


Putting it together

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: $T = M_3(W, W, W)$

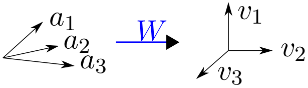


Tensor Decomposition: Guaranteed Non-Convex Optimization!

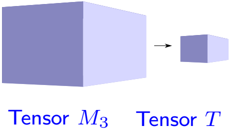
Putting it together

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: $T = M_3(W, W, W)$

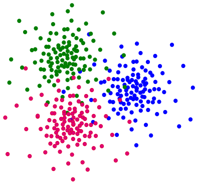


Tensor Decomposition: Guaranteed Non-Convex Optimization!

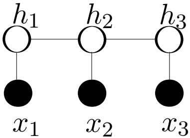
For what latent variable models can we obtain M_2 and M_3 forms?

Tractable Learning for LVMs

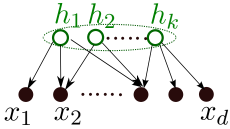
GMM



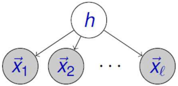
HMM



ICA



Multiview and Topic Models



$h \in [k]$,

$\vec{x}_1 \in \mathbb{R}^{d_1}, \vec{x}_2 \in \mathbb{R}^{d_2}, \ldots, \vec{x}_\ell \in \mathbb{R}^{d_\ell}$.

k = # components, ℓ = # views (*e.g.*, audio, video, text).

View 1: $\vec{x}_1 \in \mathbb{R}^{d_1}$

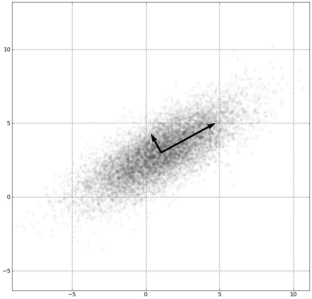
View 2: $\vec{x}_2 \in \mathbb{R}^{d_2}$

View 3: $\vec{x}_3 \in \mathbb{R}^{d_3}$

Outline

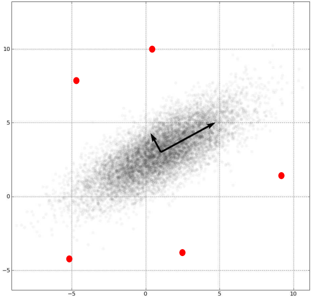
PCA: Classical Method

- Denoising: find hidden low rank structures in data.
- Efficient computation, perturbation analysis.



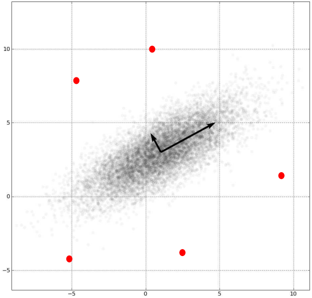
PCA: Classical Method

- Denoising: find hidden low rank structures in data.
- Efficient computation, perturbation analysis.



PCA: Classical Method

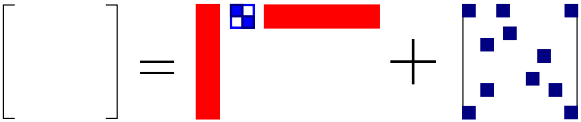
- Denoising: find hidden low rank structures in data.
- Efficient computation, perturbation analysis.



Not robust to even a few outliers

Robust PCA Problem

- Find low rank structure after removing sparse corruptions.
- Decompose input matrix as low rank + sparse matrices.



M L^* S^*

- $M \in \mathbb{R}^{n \times n}$, L^* is low rank and S^* is sparse.
- Applications in computer vision, topic and community modeling.

Convex Relaxation Techniques

(Hard) Optimization Problem, given $M \in \mathbb{R}^{n \times n}$

$$\boxed{\min_{L,S} \mathrm{Rank}(L) + \gamma \|S\|_0, \quad M = L + S.}$$

- $\mathrm{Rank}(L) = \{\#\sigma_i(L) : \sigma_i(L) \neq 0\}$, $\|S\|_0 = \{\#S(i,j) : S(i,j) \neq 0\}$ are not tractable.

Convex Relaxation Techniques

(Hard) Optimization Problem, given $M \in \mathbb{R}^{n\times n}$

$$\boxed{\min_{L,S} \mathrm{Rank}(L) + \gamma \|S\|_0, \quad M = L + S.}$$

- $\mathrm{Rank}(L) = \{\#\sigma_i(L) : \sigma_i(L) \neq 0\}$, $\|S\|_0 = \{\#S(i,j) : S(i,j) \neq 0\}$ are not tractable.

Convex Relaxation

$$\boxed{\min_{L,S} \|L\|_* + \gamma \|S\|_1, \quad M = L + S.}$$

- $\|L\|_* = \sum_i \sigma_i(L)$, $\|S\|_1 = \sum_{i,j} |S(i,j)|$ are convex sets.
- Chandrasekharan et. al, Candes et. al '11: seminal works.

Other Alternatives for Robust PCA?

$$\boxed{\min_{L,S} \|L\|_* + \gamma\|S\|_1, \quad M = L + S.}$$

Shortcomings of convex methods

Other Alternatives for Robust PCA?

$$\boxed{\min_{L,S} \|L\|_* + \gamma\|S\|_1, \quad M = L + S.}$$

Shortcomings of convex methods

- Computational cost: $O(n^3/\epsilon)$ to achieve error of ϵ
  - Requires SVD of $n \times n$ matrix.
- Analysis: requires dual witness style arguments.
- Conditions for success usually opaque.

Other Alternatives for Robust PCA?

$$\boxed{\min_{L,S} \|L\|_* + \gamma\|S\|_1, \quad M = L + S.}$$

Shortcomings of convex methods

- Computational cost: $O(n^3/\epsilon)$ to achieve error of ϵ
  - Requires SVD of $n \times n$ matrix.
- Analysis: requires dual witness style arguments.
- Conditions for success usually opaque.

Non-convex alternatives?

Proposal for Non-convex Robust PCA

$$\boxed{\min_{L,S} \|S\|_0, \quad s.t.\ M = L + S, \quad \mathrm{Rank}(L) = r}$$

Proposal for Non-convex Robust PCA

$$\boxed{\min_{L,S} \|S\|_0, \quad s.t.\ M = L + S, \quad \mathrm{Rank}(L) = r}$$

A non-convex heuristic (AltProj)

- Initialize $L, S = 0$ and iterate:
- $\boxed{L \leftarrow P_r(M - S)}$ and $\boxed{S \leftarrow H_\zeta(M - L)}$.
- $P_r(\cdot)$: rank-r projection. $H_\zeta(\cdot)$: thresholding with ζ.

- Computationally efficient: each operation is just a rank-r SVD or thresholding.

Proposal for Non-convex Robust PCA

$$\boxed{\min_{L,S} \|S\|_0, \quad s.t.\ M = L + S, \quad \mathrm{Rank}(L) = r}$$

A non-convex heuristic (AltProj)

- Initialize $L, S = 0$ and iterate:
- $\boxed{L \leftarrow P_r(M - S)}$ and $\boxed{S \leftarrow H_\zeta(M - L)}$.
- $P_r(\cdot)$: rank-r projection. $H_\zeta(\cdot)$: thresholding with ζ.

- Computationally efficient: each operation is just a rank-r SVD or thresholding.

Any hope for proving guarantees?

Non-convex Robust PCA

A non-convex heuristic (AltProj)

- Initialize $L, S = 0$ and iterate:
- $\boxed{L \leftarrow P_r(M - S)}$ and $\boxed{S \leftarrow H_\zeta(M - L)}$.

Observations regarding Robust PCA

- Projection on to rank and sparse subspaces: non-convex but tractable: SVD and hard thresholding.
- But alternating projections: challenging to analyze

Non-convex Robust PCA

A non-convex heuristic (AltProj)

- Initialize $L, S = 0$ and iterate:
- $\boxed{L \leftarrow P_r(M - S)}$ and $\boxed{S \leftarrow H_\zeta(M - L)}$.

Observations regarding Robust PCA

- Projection on to rank and sparse subspaces: non-convex but tractable: SVD and hard thresholding.
- But alternating projections: challenging to analyze

Our results for (a variant of) AltProj

- Guaranteed recovery of low rank L^* and sparse part S^*.
- Match the bounds for convex methods (deterministic sparsity).
- Reduced computation: only require low rank SVDs!

Non-convex Robust PCA

A non-convex heuristic (AltProj)

- Initialize $L, S = 0$ and iterate:
- $\boxed{L \leftarrow P_r(M - S)}$ and $\boxed{S \leftarrow H_\zeta(M - L)}$.

Observations regarding Robust PCA

- Projection on to rank and sparse subspaces: non-convex but tractable: SVD and hard thresholding.
- But alternating projections: challenging to analyze

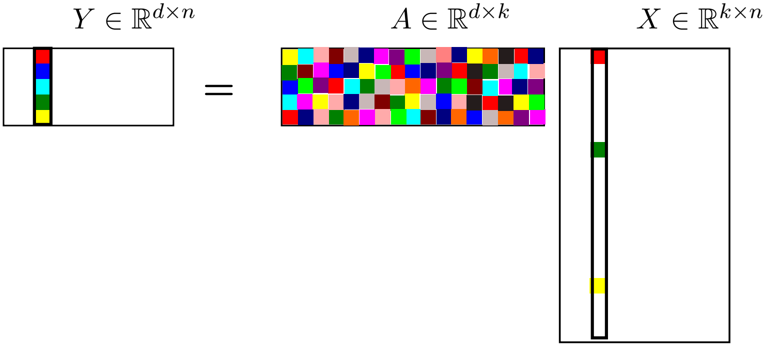
Our results for (a variant of) AltProj

- Guaranteed recovery of low rank L^* and sparse part S^*.
- Match the bounds for convex methods (deterministic sparsity).
- Reduced computation: only require low rank SVDs!

Best of both worlds: reduced computation with guarantees!

Outline

Dictionary Learning or Sparse Coding



- Each sample is a sparse combination of dictionary atoms.

Learning Overcomplete Dictionaries

- 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

Learning Overcomplete Dictionaries

- 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.

Learning Overcomplete Dictionaries

- 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

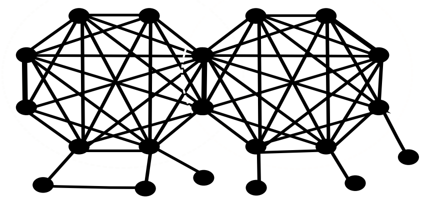
- Each sample is a combination of dictionary atoms: $y_i = \sum_j x_{i,j} a_j$.
- Consider y_i and y_j s.t. they have no common dictionary atoms.
- What about $|\langle y_i, y_j \rangle|$?
- Under incoherence: $|\langle y_i, y_j \rangle| \approx 0$.

Construction of Correlation Graph

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

How does the correlation graph help in dictionary learning?

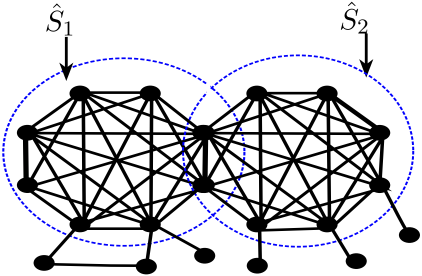
Correlation Graph and Clique Finding



Main Insight

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

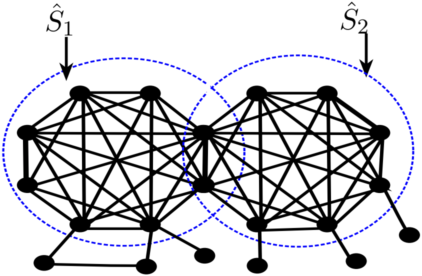
Correlation Graph and Clique Finding



Main Insight

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

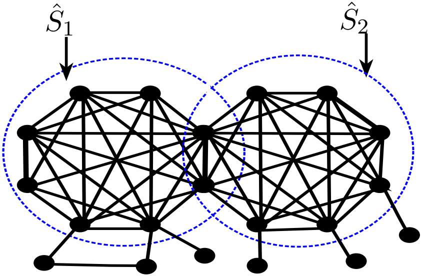
Correlation Graph and Clique Finding



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.

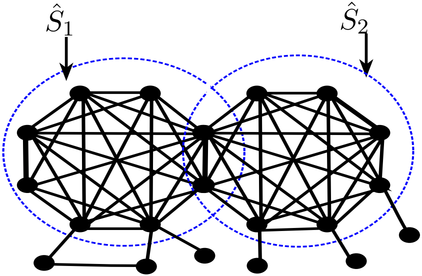
Correlation Graph and Clique Finding



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?

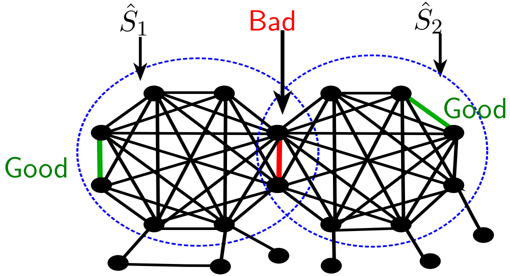
Correlation Graph and Clique Finding



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.

Correlation Graph and Clique Finding



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.

A. Agarwal, A., P. Netrapalli. "Exact Recovery of Sparsely Used Overcomplete Dictionaries," Preprint, Sept. 2013.

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 ℓ_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 ℓ_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 ℓ_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 ℓ_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 $m = \Omega(k^2)$.

A. Agarwal, A., P. Netrapalli, P. Jain, R. Tandon. "Learning Sparsely Used Overcomplete Dictionaries via Alternating Minimization," Preprint, Oct. 2013.

Outline

Observations regarding non-convex analysis

Challenges

- Multiple stable points: bad local optima, solution depends on initialization.
- Method may have very slow convergence or may not converge at all!

Observations regarding non-convex analysis

Challenges

- Multiple stable points: bad local optima, solution depends on initialization.
- Method may have very slow convergence or may not converge at all!

Non-convex Projections vs. Convex Projections

- Projections on to non-convex sets: NP-hard in general.
  - Projections on to rank and sparse sets: tractable.
- Less information than convex projections: zero-order conditions.

$$\|P(M) - M\| \leq \|Y - M\|, \quad \forall\, Y \in C(\text{Non-convex}),$$

$$\|P(M) - M\|^2 \leq \langle Y - M, P(M) - M \rangle, \quad \forall\, Y \in C(\text{Convex}).$$

Non-convex success stories

- Spectral Methods: PCA, Tensor methods.
- Robust PCA: Alternating projections.
- Dictionary learning: Initialize using a “clustering style” method.

Advantages

- Iterative methods, global convergence guarantees.
- Efficient sample and computational complexities
- Competitive performance, easily parallelizable and scalable.

(Somewhat) common theme

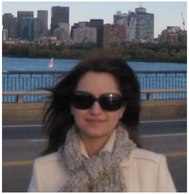
- Characterize basin of attraction for global optimum.
- Obtain a good initialization to “land in the ball”: usually also a non-convex method!

My Research Group and Resources

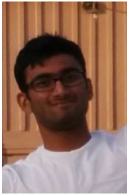
Furong Huang

Majid Janzamin

Hanie Sedghi



Niranjan UN



Forough Arabshahi

- ML summer school lectures available at
  `http://newport.eecs.uci.edu/anandkumar/MLSS.html`