Fitting Convex Sets to Data via Matrix Factorization

Yong Sheng Soh

LCCC Focus Period – May/June 2017

California Institute of Technology
Joint work with Venkat Chandrasekaran

Variational Approach to Inference

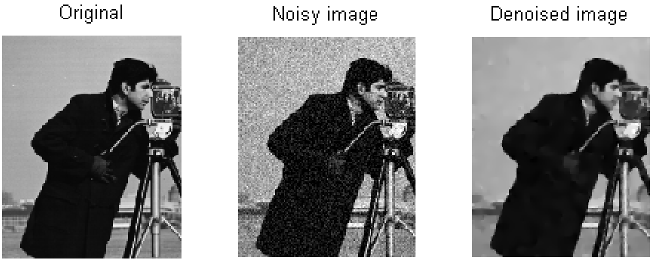
Given **data**, fit **model** (θ) by solving

$$\arg\min_{\theta} \quad \mathrm{Loss}(\theta; \mathsf{data}) + \lambda \cdot \mathrm{Regularizer}(\theta)$$

- **Loss:** ensures fidelity to observed data
  - Based on model of noise that has corrupted observations
- **Regularizer:** useful to induce desired structure in solution
  - Based on prior knowledge, domain expertise

Example

Denoise an image corrupted by noise



- **Loss:** Euclidean-norm
- **Regularizer:** L1-norm of wavelet coefficients
- Natural images are typically sparse in wavelet basis

Photo: [Rudin, Osher, Fatemi]

Example

Complete a partially filled survey

	Life is Beautiful	Goldfinger	Big Lebowski	Shawshank Redemption	Godfather
Alice	5	4	?	?	?
Bob	?	4	1	4	?
Charlie	?	4	4	?	5
Donna	4	?	?	5	?

- **Loss:** Euclidean / Logistic
- **Regularizer:** Nuclear-norm of user-preference matrix
- User-preference matrices often well-approximated as low-rank

This Talk

- **Question:** What if we do not have the domain expertise to design or select an appropriate regularizer for our task?
  - E.g. domains with high-dimensional data comprising different data types
- **Approach:** **Learn** a **suitable** **regularizer** from example **data**
  - E.g. Learn a suitable regularizer for denoising images using examples of clean images
- **Geometric picture:** **Fit** a **convex set** (with **suitable facial structure**) to a **set of points**

This Talk – Pipeline

- **Learn:** Have access to examples of (relatively) clean example data. Use examples to learn a suitable regularizer.
- **Apply:** Faced with subsequent task that involves noisy or incomplete data. Apply learned regularizer.

Outline

A paradigm for designing regularizers

LP-representable regularizers

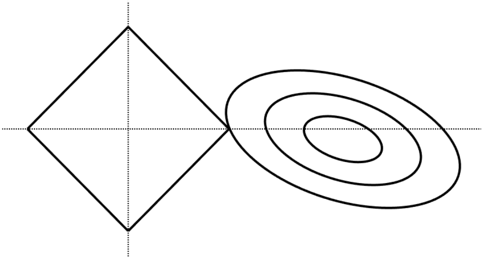
SDP-representable regularizers

Summary and future work

Designing Regularizers

- **Conceptual question:** Given a dataset, how do we identify a regularizer that is effective at enforcing structure that is present in the data?
- **First Step:** What properties of a regularizer make them effective?

Facial Geometry



Key: **Facial geometry** of the **level sets** of the regularizer.

- Optimal solution corresponding to generic data often lie on low-dimensional faces
- In many applications the low-dimensional faces are the structured models we wish to recover e.g. images are sparse in wavelet domain

Approach: Design a regularizer s.t. **data** lies on **low-dimensional faces** of **level sets**. We do so by using **concise representations**.

From Concise Representations to Regularizer

Concise representations:

We say that a datapoint (a vector) $\boldsymbol{y} \in \mathbb{R}^d$ is **concisely represented** by a set $\{\boldsymbol{a}_i\}_{i \in \mathcal{I}} \subset \mathbb{R}^d$ (called **atoms**) if

$$\boldsymbol{y} = \sum_{i \in \mathcal{S}, \mathcal{S} \subset \mathcal{I}} c_i \boldsymbol{a}_i, \quad c_i \geq 0,$$

for $|\mathcal{S}|$ small.

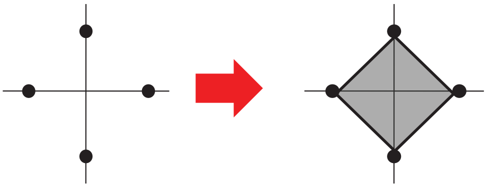
Regularizer:

$$\|\boldsymbol{x}\| = \inf \{ t : \boldsymbol{x} \in t \cdot \mathrm{conv}(\{\boldsymbol{a}_i\}), t > 0 \}.$$

Smallest "blow-up" of $\mathrm{conv}(\{\boldsymbol{a}_i\})$ that includes $\boldsymbol{x}$

[Maurey, Pisier, Jones...]

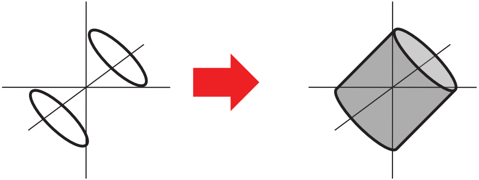
Sparse Representations



- **Concisely represented data:** Sparse vectors
  - Linear sum of **few** standard basis vectors
- **Regularizer:** L1-norm
  - Norm-ball is the convex hull of standard basis vectors

[Donoho, Johnstone, Tibshirani, Chen, Saunders, Candès, Romberg, Tao, Tanner, Meinhausen, Bühlmann]

Sparse Representations



- **Concisely represented data:** Low-rank matrices
  - Linear sum of **few** rank-one unit-norm matrices
- **Regularizer:** Nuclear-norm (sum of singular values)
  - Norm-ball is the convex hull of rank-one unit-norm matrices

[Fazel, Boyd, Recht, Parrilo, Candès, Gross, ...]

From Concise Representations to Regularizer

- From the view-point of optimization, this is the **"correct" convex regularizer** to employ
  - Low-dimensional faces of $\mathrm{conv}(\{\boldsymbol{a}_i\})$ are concisely represented with $\{\boldsymbol{a}_i\}$

[Chandrasekaran, Recht, Parrilo, Willsky]

Designing Regularizers

- **Conceptual question:** Given a dataset, how do we identify a regularizer that is effective at enforcing structure present in the data?

- **Prior work:** If data can be concisely represented wrt a set $\{\boldsymbol{a}_i\} \subset \mathbb{R}^d$ then an effective regularizer is **available**
  - It is the norm induced by $\mathrm{conv}(\{\boldsymbol{a}_i\})$.

- **Approach:** Given a dataset, **identify** a set $\{\boldsymbol{a}_i\} \subset \mathbb{R}^d$ s.t. data permits concise representations.

Polyhedral Regularizers

Approach: Given dataset, how do we identify a set $\{\pm \boldsymbol{a}_i\} \subset \mathbb{R}^d$ such that the data permits concise representations?

Assume: $|\{\boldsymbol{a}_i\}|$ is finite.

Precise mathematical formulation:

Given data $\{\boldsymbol{y}^{(j)}\}_{j=1}^{n} \subset \mathbb{R}^d$, *find* $\{\boldsymbol{a}_i\}_{i=1}^{q} \subset \mathbb{R}^d$ so that

$$\begin{aligned} \boldsymbol{y}^{(j)} &\approx \sum x_i^{(j)} \boldsymbol{a}_i, \quad \text{where } x_i^{(j)} \text{ are mostly zero} \\ &= A\boldsymbol{x}^{(j)} \quad \text{where} \quad A = [\boldsymbol{a}_1 | \ldots | \boldsymbol{a}_q], \quad \text{and} \quad \boldsymbol{x}^{(j)} \text{ is sparse,} \end{aligned}$$

for each j.

Polyhedral Regularizers

Given data $\{\boldsymbol{y}^{(j)}\}_{j=1}^{n} \subset \mathbb{R}^d$, *find* $A \in \mathbb{R}^q \mapsto \mathbb{R}^d$ so that

$$\boldsymbol{y}^{(j)} \quad \approx \quad A\boldsymbol{x}^{(j)}, \quad \text{where} \quad \boldsymbol{x}^{(j)} \text{ is sparse} \quad \forall j.$$

Regularizer:

Natural choice of regularizer is the norm *induced* by

$$\mathrm{conv}(\{\pm \boldsymbol{a}_i\}),$$

or equivalently

$$A(\text{L1 norm ball}), \quad \text{where} \quad A = [\boldsymbol{a}_1 | \dots | \boldsymbol{a}_q].$$

The regularizer can be expressed as a **linear program (LP)**.

Polyhedral Regularizers – Dictionary Learning

Given data $\{\boldsymbol{y}^{(j)}\}_{j=1}^{n} \subset \mathbb{R}^d$, *find* $A \in \mathbb{R}^q \mapsto \mathbb{R}^d$ so that

$$\boldsymbol{y}^{(j)} \quad \approx \quad A\boldsymbol{x}^{(j)}, \quad \text{where} \quad \boldsymbol{x}^{(j)} \text{ is sparse} \quad \forall j.$$

Studied elsewhere as:

- '**Dictionary Learning**' or '**Sparse Coding**'
  - Olshausen, Field ('96); Aharon, Elad, Bruckstein ('06), Spielman, Wang, Wright ('12); Arora, Ge, Moitra ('13); Agarwal, Anandkumar, Netrapalli, Jain ('13); Barak, Kelner, Steurer ('14); ...
- Developed as a procedure for automatically discovering sparse representations with finite dictionaries

Learning an Infinite Set of Atoms?

So far:

- Learning a regularizer corresponds to computing a matrix factorization
- **Finite** set of atoms = dictionary learning

Question: Can we learn an **infinite** set of atoms?

- Richer family of concise representations
- Require
  - Compact description of atoms
  - Computationally tractable description of the convex hull

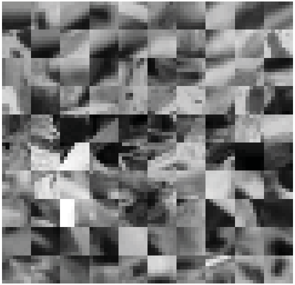
Remainder of the talk:

- Specify infinite atomic set as a **algebraic variety** whose convex hull is computable via **semidefinite programming**

From dictionary learning to our work

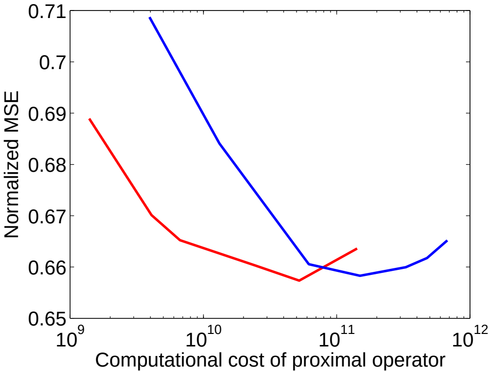
	Dictionary learning	**Our work**
Atoms	$\{\pm A\boldsymbol{e}^{(i)} \mid \boldsymbol{e}^{(i)} \in \mathbb{R}^p$ is a **standard basis vector**$\}$ $A : \mathbb{R}^p \to \mathbb{R}^d$	$\{\mathcal{A}(U) \mid U \in \mathbb{R}^{q\times q}$, U **unit-norm rank-one**$\}$ $\mathcal{A} : \mathbb{R}^{q\times q} \to \mathbb{R}^d$
Compute regularizer by	Find A s.t. $\boldsymbol{y}^{(j)} \approx A\boldsymbol{x}^{(j)}$ for **sparse** $\boldsymbol{x}^{(j)}$	Find $\mathcal{A}$ s.t. $\boldsymbol{y}^{(j)} \approx \mathcal{A}(X^{(j)})$ for **low-rank** $X^{(j)}$
Level set	A(**L1-norm** ball)	$\mathcal{A}$(**nuclear norm** ball)
Regularizer expressed via	**Linear Programming (LP)**	**Semidefinite Programming (SDP)**

Empirical results – Set-up



- **Learn:** Learn a collection of regularizers of varying complexities from 6500 example image patches.
- **Apply:** Denoise 720 **new** data points corrupted by additive Gaussian noise.

Empirical results – Comparison



Denoise 720 new data points corrupted by additive Gaussian noise

Polyhedral regularizer, i.e. dictionary learning

Semidefinite-representable regularizer

Apply proximal denoising (squared-loss + regularizer)

Cost is derived by computing proximal operator via an interior point scheme

Semidefinite-Representable Regularizers

Goal: Compute a matrix factorization problem

Given data $\{\boldsymbol{y}^{(j)}\}_{j=1}^{n} \subset \mathbb{R}^d$ and a target dimension q, *find* $\mathcal{A} : \mathbb{R}^{q \times q} \mapsto \mathbb{R}^d$ so that

$$\boldsymbol{y}^{(j)} \approx \mathcal{A}(X^{(j)}) \quad \text{for low-rank} \quad X^{(j)} \in \mathbb{R}^{q \times q},$$

for each j.

Obstruction: This is a matrix factorization problem. The factors $\mathcal{A}$ and $\{X^{(j)}\}_{j=1}^{n}$ are both unknown, and hence any factorization is **not unique**.

Identifiablity Issues

- Given a factorization of $\{\boldsymbol{y}^{(j)}\}_{j=1}^{n} \subset \mathbb{R}^d$ as $\boldsymbol{y}^{(j)} = \mathcal{A}(X^{(j)})$ for low-rank $X^{(j)}$, there are **many equivalent factorizations**

- Let $\mathcal{M} : \mathbb{R}^{q\times q} \mapsto \mathbb{R}^{q\times q}$ be an invertible linear operator that preserves the rank of matrices
  - Transpose operator $\mathcal{M}(X) = X'$
  - Conjugation by invertible matrices $\mathcal{M}(X) = PXQ'$

  Then

  $$\boldsymbol{y}^{(j)} = \underbrace{\mathcal{A} \circ \mathcal{M}^{-1}}_{\text{Linear map}}\left(\underbrace{\mathcal{M}(X^{(j)})}_{\text{Low rank matrix}} \right)$$

  specifies an equally valid factorization!

- $\{\mathcal{A} \circ \mathcal{M}^{-1}\}$ specifies family of regularizers – require a **canonical choice** of factorization to **uniquely** specify a regularizer

Identifiablity Issues

Theorem (Marcus and Moyls ('59)): An invertible linear operator $\mathcal{M} : \mathbb{R}^{q\times q} \mapsto \mathbb{R}^{q\times q}$ **preserves the rank of matrices** $\Leftrightarrow$ composition of

- Transpose operator $\mathcal{M}(X) = X'$
- Conjugation by invertible matrices $\mathcal{M}(X) = PXQ'$

In our context, the regularizer is induced by

$$\mathcal{A} \circ \mathcal{M}^{-1}(\text{nuclear norm ball})$$

- $\mathcal{M}$ is transpose operator: leaves nuclear norm invariant
- $\mathcal{M}$ is conjugation by invertible matrices: apply polar decomposition to orthogonal + positive definite
  - Orthogonal matrices also leave nuclear norm invariant
  - Ambiguity down to conjugation by <u>positive definite matrices</u>

Identifiablity Issues

Definition: A linear map $\mathcal{A} : \mathbb{R}^{q \times q} \mapsto \mathbb{R}^d$ is **normalized** if

$$\sum_{k=1}^{d} \mathcal{A}_k \mathcal{A}_k' = \sum_{k=1}^{d} \mathcal{A}_k' \mathcal{A}_k = I$$

where $\mathcal{A}_k \in \mathbb{R}^{q \times q}$ is the k-th component linear functional of $\mathcal{A}$.

One should think of $\mathcal{A}$ as

$$\mathcal{A}(X) = \begin{pmatrix} \langle \mathcal{A}_1, X \rangle \\ \vdots \\ \langle \mathcal{A}_d, X \rangle \end{pmatrix}$$

Identifiablity Issues

Definition: A linear map $\mathcal{A} : \mathbb{R}^{q \times q} \mapsto \mathbb{R}^d$ is **normalized** if

$$\sum_{k=1}^{d} \mathcal{A}_k \mathcal{A}_k' = \sum_{k=1}^{d} \mathcal{A}_k' \mathcal{A}_k = I$$

where $\mathcal{A}_k \in \mathbb{R}^{q \times q}$ is the k-th component linear functional of $\mathcal{A}$.

Given a generic linear map $\mathcal{A} : \mathbb{R}^{q \times q} \mapsto \mathbb{R}^d$, normalization entails finding a rank-preserver $\mathcal{M}$ so that

$$\mathcal{A} \circ \mathcal{M} \quad \text{is normalized.}$$

Rank-preserver is unique, and can be computed via Operator Sinkhorn Scaling [Gurvits ('04)].

Operator Sinkhorn Scaling

- **Matrix Scaling:** Given matrix $M \in \mathbb{R}^{q \times q}$, $M_{ij} > 0$, find $\mathrm{diag}(D_1), \mathrm{diag}(D_2)$ so that

$$\mathrm{diag}(D_1) M \mathrm{diag}(D_2) \quad \text{is doubly-stochastic}$$

- **Operator Sinkhorn Scaling:** Operator analog of Matrix Scaling
  - Edmond's problem: Given subspace of $\mathbb{F}^{q \times q}$, decide if there exists nonsingular matrix.

Algorithm – Overview

- **Goal:** Compute $\mathcal{A}$ and $X^{(j)}$'s so that

$$\{\boldsymbol{y}^{(j)}\}_{j=1}^{n} \approx \mathcal{A}(\{X^{(j)}\}_{j=1}^{n})$$

- **Approach:** **alternating updates**
  - Input: Data $\{\boldsymbol{y}^{(j)}\}_{j=1}^{n}$, initial estimate of $\mathcal{A}$
  - Alternate between updating $\{X^{(j)}\}_{j=1}^{n}$, and updating $\mathcal{A}$
  - Generalizes previous algorithms for classical dictionary learning

Algorithm

Input: Data $\{\boldsymbol{y}^{(j)}\}_{j=1}^{n}$, initial estimate of $\mathcal{A}$

1. Fix $\mathcal{A}$, update $X^{(j)}$

$$X^{(j)} \leftarrow \arg\min_{X} \|\boldsymbol{y}^{(j)} - \mathcal{A}(X)\|_2^2 \quad \text{subject to} \quad \mathrm{rank}(X) \leq r$$

   - Computationally intractable in general.
   - Tractable approximations with guarantees available, e.g. convex relaxation (Recht, Fazel, Parrilo ('07)), singular-value projection (Meka, Jain, Dhillon ('10))
   - Updates occur in parallel

2. ...

3. ...

Algorithm

Input: Data $\{\boldsymbol{y}^{(j)}\}_{j=1}^{n}$, initial estimate of $\mathcal{A}$

1. ...

2. Fix $X^{(j)}$, update $\mathcal{A}$, e.g. least squares

$$\mathcal{A} \leftarrow \arg\min_{\mathcal{A}} \sum_{j} \|\boldsymbol{y}^{(j)} - \mathcal{A}(X^{(j)})\|_2^2$$

3. ...

Algorithm

Input: Data $\{\boldsymbol{y}^{(j)}\}_{j=1}^{n}$, initial estimate of $\mathcal{A}$

1. ...

2. ...

3. Normalize using **Operator Sinkhorn Scaling** described earlier

Algorithm

Input: Data $\{\boldsymbol{y}^{(j)}\}_{j=1}^{n}$, initial estimate of $\mathcal{A}$

1. Fix $\mathcal{A}$, update $X^{(j)}$: **Affine-rank minimization**

$$X^{(j)} \leftarrow \arg\min_{X} \|\boldsymbol{y}^{(j)} - \mathcal{A}(X)\|_2^2 \quad \text{subject to} \quad \mathrm{rank}(X) \leq r$$

2. Fix $X^{(j)}$, update $\mathcal{A}$: **Least-squares**

$$\mathcal{A} \leftarrow \arg\min_{\mathcal{A}} \sum_{j} \|\boldsymbol{y}^{(j)} - \mathcal{A}(X^{(j)})\|_2^2$$

3. Normalize via **Operator Sinkhorn Scaling**

Analysis – High Level Description

Assumptions: Data is **generated** by a model

Guarantee: Algorithm recovers the **true regularizer** with **suitable initialization**

Analysis

Suppose: Data $\{\mathbf{y}^{(j)}\}_{j=1}^{n}$ is generated as $\mathbf{y}^{(j)} = \mathcal{A}(X^{(j)})$

- $\mathcal{A} : \mathbb{R}^{q \times q} \mapsto \mathbb{R}^{d}$ is normalized and satisfies restricted isometry property [Recht, Fazel, Parrilo]
- $X^{(j)} \sim UV'$ where $U, V \in \mathbb{R}^{q \times r}$ are partial orthogonal matrices distributed u.a.r.,

If:

- $\#$ data-points is sufficiently many ($\gtrsim q^{10}/d$),
- Lifted dimension is not too high ($\lesssim d^2/r^2$).

Guarantee: Algorithm is **locally linearly convergent** and recovers the **same regularizer** as $\mathcal{A}$ w.h.p..

Here, d = dim of ambient space, and r = rank.

Summary and Future work

Summary

- Described an approach for learning regularizer from data by computing a structured matrix factorization
- $\#$ atoms being finite $=$ polyhedral regularizer
- Described a special case with infinite atoms where learned regularizer is computable via SDP

Future work

- Applying our algorithm as a building block in more complex learning algorithms
- Informed strategies for initializing alternating minimization procedure

arXiv: 1701.01207