Learning Regularizers From Data

Venkat Chandrasekaran

Caltech

Joint work with Yong Sheng Soh

Variational Perspective on Inference

$$\min_{\theta} \quad \text{loss}(\theta \; ; \; \text{data}) \;+\; \lambda \, \text{regularizer}(\theta)$$

- ***Loss*** ensures fidelity to observed data
  - Based on the specific inverse problem one wishes to solve

- ***Regularizer*** useful to induce desired structure in solution
  - Based on prior knowledge via **domain expertise**

This Talk

- What if we don't have domain expertise to design regularizer?
  - Many domains with unstructured, high-dimensional data

- ***Learn regularizer from data?***
  - Eg., learn regularizer for image denoising given many "clean" images?

- <u>Pipeline</u>: (relatively) clean data → **learn regularizer** → use regularizer in subsequent problems with noisy/incomplete data

Outline

- Learning ***computationally tractable*** regularizers from data
- Convex regularizers that can be computed / optimized efficiently by **semidefinite programming**
- Along the way, algorithms for quantum / operator problems
  - Operator Sinkhorn scaling [Gurvits (`03)]
- Contrast with prior work on dictionary learning / sparse coding

Designing Regularizers

- What is a good regularizer?
- What properties do we want of a regularizer?
- When does a regularizer induce the desired structure?
- First, let's understand ***how to transform domain expertise to a suitable regularizer*** …

Example: Image Denoising


Original


Noisy


Denoised

Ideas due to: Meyer, Mallat, Daubechies, Donoho, Johnstone, Crouse, Nowak, Baraniuk, …

- Loss: Euclidean-norm

- Regularizer: L1 norm (sum of magnitudes) of wavelet coefficients
  - Natural images are typically sparse in wavelet basis

Example: Matrix Completion

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

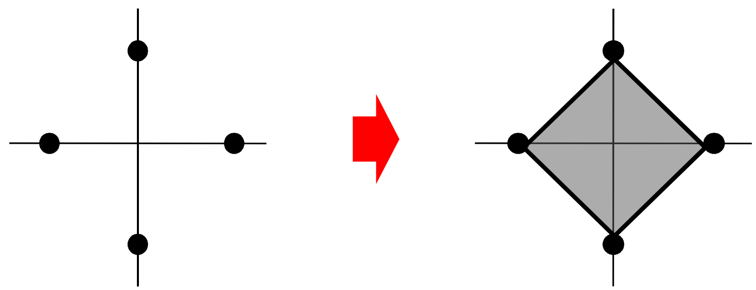
Ideas due to: Srebro, Jaakkola, Fazel, Boyd, Recht, Parrilo, Candes, …

- Loss: Euclidean/logistic
- Regularizer: nuclear norm (sum of singular values) of matrix
  - User-preference matrices often well-approximated as low-rank

What is a Good Regularizer?

- Why the L1 and nuclear norms in these examples?

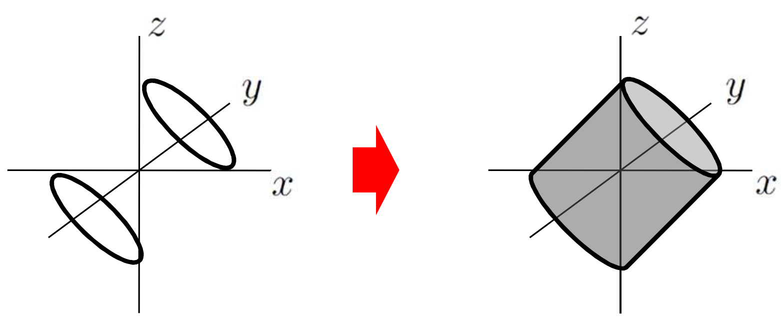
Vectors with one nonzero



L1 norm ball [Santosa, Symes, Donoho, Johnstone, Tibshirani, Chen, Saunders, Candes, Romberg, Tao, Tanner, Meinshausen, Buhlmann, …]

Rank-one matrices

$$\begin{pmatrix} x & y \\ y & z \end{pmatrix}$$



Nuclear norm ball [Fazel, Boyd, Recht, Parrilo, Candes, …]

Atomic Sets and Atomic Norms

- Given a set $\{\mathbf{a}_i\}_{i\in\mathcal{I}} \subset \mathbb{R}^d$ of ***atoms***, concisely described data w.r.t. $\{\mathbf{a}_i\}$ are

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

  for $|\mathcal{S}|$ small

- Given atomic set $\{\mathbf{a}_i\}$, regularize using **atomic norm**

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

C., Recht, Parrilo, Willsky, "The Convex Geometry of Linear Inverse Problems," Foundations of Computational Mathematics, 2012

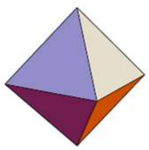
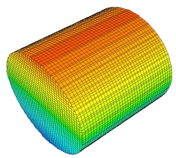
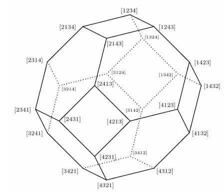
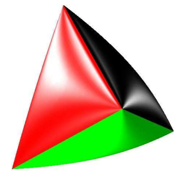
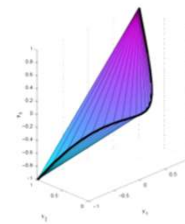
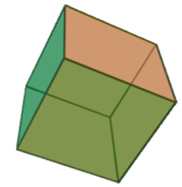
Atomic Norm Regularizers

Underlying model	Application	Atomic norm
sparse vector	lasso, compressed sensing	L1 norm
low-rank matrix	factor modeling, matrix completion	nuclear norm
vector with entries of same magnitude	knapsack, democratic representations [Mangasarian; Studer et al.]	infinity-norm
permutation matrix	ranking, multi-target tracking [Jagabathula et al.; Huang et al.]	norm induced by Birkhoff polytope
orthogonal matrix	visual pose estimation [Horowitz & Matni]	spectral norm

- Line spectral estimation [Bhaskar at al. (`12)]
- Low-rank tensor decomposition [Tang et al. (`15)]

C., Recht, Parrilo, Willsky, "The Convex Geometry of Linear Inverse Problems," Foundations of Computational Mathematics, 2012

Atomic Norm Regularizers

- These norms also have **the 'right' convex-geometric properties**
  - Low-dimensional faces of $\mathrm{conv}(\{\mathbf{a}_i\})$ are concisely described using $\{\mathbf{a}_i\}$
  - Solutions of convex programs with generic data lie on low-dimensional faces

C., Recht, Parrilo, Willsky, "The Convex Geometry of Linear Inverse Problems," Foundations of Computational Mathematics, 2012

Learning Regularizers

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

- **Atomic norms**: If data can be concisely represented wrt a set of atoms $\{\mathbf{a}_i\}$, then an effective regularizer is available
  - It is the atomic norm wrt $\{\mathbf{a}_i\}$

- **Approach**: Given dataset, identify a set of atoms s.t. data permits concise representations

Learning Polyhedral Regularizers

- Assume that the atomic set is finite

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

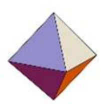
$$\begin{aligned} \mathbf{y}^{(j)} &\approx \sum x_i^{(j)} \mathbf{a}_i, && \text{where } x^{(j)} \text{ are } \textbf{mostly zero} \\ &= A\mathbf{x}^{(j)} && \text{where } A = [\mathbf{a}_1 | \ldots | \mathbf{a}_q] \\ & && \mathbf{x}^{(j)} \text{ is } \textbf{sparse} \end{aligned}$$

Learning Polyhedral Regularizers

Given $\{\mathbf{y}^{(j)}\}_{j=1}^{n} \subset \mathbb{R}^d$ and target dimension q, find $A \in \mathbb{R}^{d\times q}$ such that each $\mathbf{y}^{(j)} \approx A\mathbf{x}^{(j)}$ for ***sparse*** $\mathbf{x}^{(j)} \in \mathbb{R}^q$

- Regularizer is the atomic norm wrt

$$\mathrm{conv}(\{\pm\mathbf{a}_i\})$$

- Level set is $A($  $)$, where $A = [\mathbf{a}_1|\ldots|\mathbf{a}_q]$
  - Expressible as a **linear program**

Learning Polyhedral Regularizers

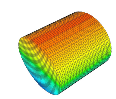
Given $\{\mathbf{y}^{(j)}\}_{j=1}^{n} \subset \mathbb{R}^d$ and target dimension q, find $A \in \mathbb{R}^{d \times q}$ such that each $\mathbf{y}^{(j)} \approx A\mathbf{x}^{(j)}$ for ***sparse*** $\mathbf{x}^{(j)} \in \mathbb{R}^q$

- Extensively studied as '**dictionary learning**' or '**sparse coding**'
  - Olshausen, Field (`96); Aharon, Elad, Bruckstein (`06); Spielman, Wang, Wright (`12); Arora, Ge, Moitra (`13); Agarwal, Anandkumar, Netrapalli (`13); Barak, Kelner, Steurer (`14); Sun, Qu, Wright (`15); …
- Dictionary learning identifies linear programming regularizers!

Learning an Infinite Set of Atoms?

- So far
  - Learning a regularizer corresponds to computing a matrix factorization
  - **Finite** set of atoms = dictionary learning
- Can we learn an **infinite** set of atoms?
  - Richer family of concise representations
  - Require compact description of atoms, tractable description of convex hull
- Specify infinite atomic set as an **algebraic variety** whose convex hull is computable via **semidefinite programming**

In a Nutshell…

	Polyhedral Regularizers (Dictionary Learning)	Semidefinite-Representable Regularizers (Our work)
Atoms	$A(\text{standard basis vectors})$	$\mathcal{A}(\text{unit-norm rank 1 matrices})$
Learn Regularizer	Find $A \in \mathbb{R}^q \mapsto \mathbb{R}^d$ s.t. $\mathbf{y}^{(j)} \approx A\mathbf{x}^{(j)}$ for sparse $\mathbf{x}^{(j)}$	Find $\mathcal{A} \in \mathbb{R}^{q \times q} \mapsto \mathbb{R}^d$ s.t. $\mathbf{y}^{(j)} \approx \mathcal{A}(X^{(j)})$ for low-rank $X^{(j)}$
Level Set	$A($  $)$	$\mathcal{A}($  $)$
Compute regularizer	Linear Programming	Semidefinite Programming

Learning Semidefinite Regularizers

- Learning phase:

Given $\{\mathbf{y}^{(j)}\}_{j=1}^{n} \subset \mathbb{R}^d$ and target dimension q, find $\mathcal{A} : \mathbb{R}^{q \times q} \mapsto \mathbb{R}^d$ such that each $\mathbf{y}^{(j)} \approx \mathcal{A}(X^{(j)})$ for ***low-rank*** $X^{(j)} \in \mathbb{R}^{q \times q}$

- Deployment phase: use image of nuclear norm ball under learned map $\mathcal{A}$ as unit ball of regularizer

Learning Semidefinite Regularizers

- Learning phase:

Given $\{\mathbf{y}^{(j)}\}_{j=1}^{n} \subset \mathbb{R}^d$ and target dimension q, find $\mathcal{A} : \mathbb{R}^{q \times q} \mapsto \mathbb{R}^d$ such that each $\mathbf{y}^{(j)} \approx \mathcal{A}(X^{(j)})$ for ***low-rank*** $X^{(j)} \in \mathbb{R}^{q \times q}$

- Obstruction: This is a matrix factorization problem. The factors are ***not-unique***.

Addressing Identifiability Issues

- **Characterize** the degrees of **ambiguities** in any factorization

- Propose a **normalization** scheme
  - Selects a unique choice of regularizer

- Normalization scheme is **computable** via **Operator Sinkhorn Scaling**

Identifiability Issues

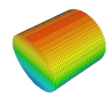
- Given a factorization of $\{\mathbf{y}^{(j)}\}_{j=1}^{n} \subset \mathbb{R}^d$ as $\mathbf{y}^{(j)} \approx \mathcal{A}(X^{(j)})$ for low-rank $X^{(j)} \in \mathbb{R}^{q \times q}$, there are ***many equivalent factorizations***

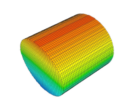
- For any linear map $\mathcal{M} : \mathbb{R}^{q \times q} \rightarrow \mathbb{R}^{q \times q}$ that is a **rank-preserver**, an equivalent factorization is $\mathbf{y}^{(j)} = \mathcal{A}\mathcal{M}^{-1}(\mathcal{M}X^{(j)})$
  - Eg., transpose, conjugation by non-singular matrices

- **Thm** [Marcus, Moyls (`59)]: A linear map $\mathcal{M} : \mathbb{R}^{q \times q} \rightarrow \mathbb{R}^{q \times q}$ is a rank-preserver if and only if we have that (i) $\mathcal{M}(X) = W_1 X W_2$ or (ii) $\mathcal{M}(X) = W_1 X' W_2$ for non-singular $W_1, W_2 \in \mathbb{R}^{q \times q}$

Identifiability Issues

- For a given factorization, the regularizer is specified by

$$\mathcal{A}\mathcal{M}^{-1}($$  $$)$$

- Normalization entails selecting $\mathcal{M}$ so that $\mathcal{A}\mathcal{M}^{-1}($  $)$ is uniquely specified

Identifiability Issues

- **Def**: 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 $\mathcal{A}_i \in \mathbb{R}^{q\times q}$ is the i'th component linear functional of $\mathcal{A}$

- Think of $\mathcal{A}$ as:

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

Identifiability Issues

- **Def**: 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 $\mathcal{A}_i \in \mathbb{R}^{q \times q}$ is the i'th component linear functional of $\mathcal{A}$
  - Analogous to unit-norm columns in dictionary learning

- Generic $\mathcal{A}$ normalizable by conjugating $\mathcal{A}_i$'s by PD matrices
  - Such a conjugation is **unique**
  - Computed via **Operator Sinkhorn Scaling** [Gurvits (`03)]
  - Developed for matroid problems, operator analogs of matching, …

Algorithm for Learning Semidefinite Regularizer

Given $\{\mathbf{y}^{(j)}\}_{j=1}^{n} \subset \mathbb{R}^d$ and target dimension q, find $\mathcal{A} : \mathbb{R}^{q \times q} \mapsto \mathbb{R}^d$ such that each $\mathbf{y}^{(j)} \approx \mathcal{A}(X^{(j)})$ for ***low-rank*** $X^{(j)} \in \mathbb{R}^{q \times q}$

<u>Alternating updates</u>

1) Updating $X^{(j)}$'s -- affine rank-minimization problems
   - NP-hard, but many relaxations available with performance guarantees
2) Updating $\mathcal{A}$ -- least-squares + Operator Sinkhorn scaling

- Direct generalization of dictionary learning algorithms

Convergence Result

- Suppose data $\{\mathbf{y}^{(j)}\}_{j=1}^{n} \subset \mathbb{R}^d$ generated as $\mathbf{y}^{(j)} = \mathcal{A}^\star(X^{(j)^\star})$

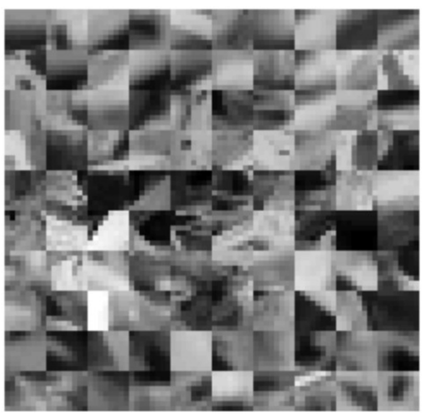
- $\mathcal{A}^\star : \mathbb{R}^{q \times q} \mapsto \mathbb{R}^d$ is a random Gaussian map

- $\mathrm{rank}\left(X^{(j)^\star}\right) = r$ with uniform-at-random row/column spaces

- <u>**Theorem**</u>: Then our algorithm is ***locally linearly convergent*** w.h.p. to the correct regularizer if $d \gtrsim rq, \ n \gtrsim q^{10}/d$
  - Recovery for 'most' regularizers

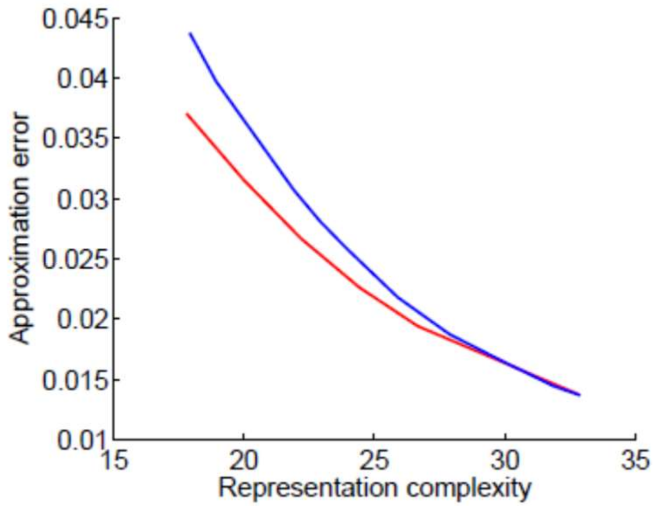
Experiments – Setup

- Pictures taken by Yong Sheng Soh



- Supplied 8x8 patches and their rotations as training set to our algorithm

Experiments – Approximation Power

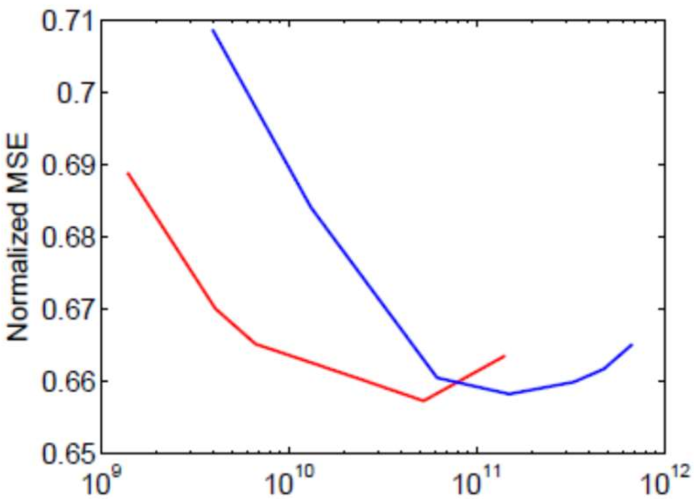


- Train: 6500 points (centered, normalized)
- Learn linear / semidefinite regularizers

- Blue – linear programming (dictionary learning)
- Red – semidefinite programming (our idea)

- Best over many random initializations

Experiments – Denoising Performance

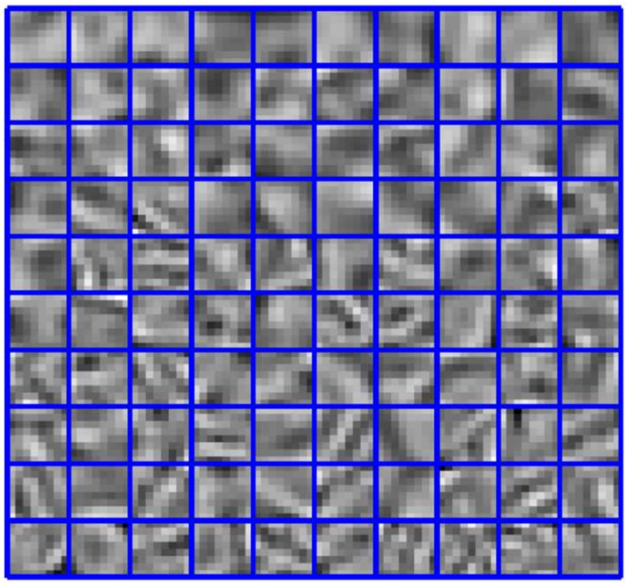


Computational complexity of regularizer

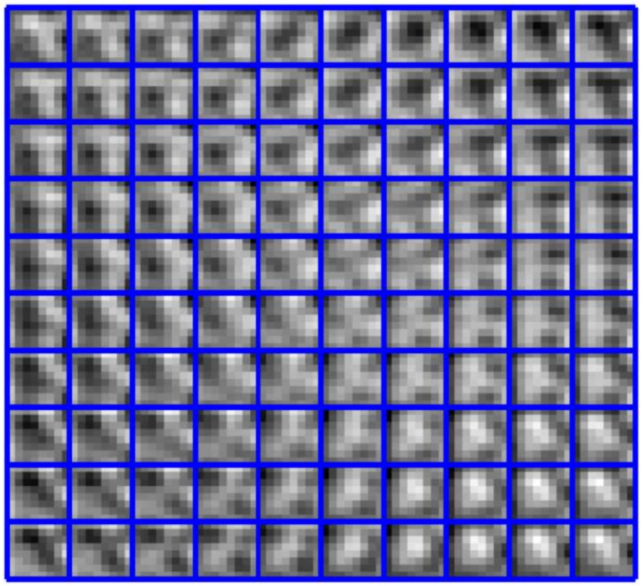
- Test: 720 points corrupted by Gaussian noise
- Denoise with Euclidean loss, learned regularizer

- Blue – linear programming (dictionary learning)
- Red – semidefinite programming (our idea)

Comparison of Atomic Structure



Finite atomic set (dictionary learning)

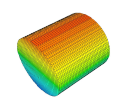


Subset of infinite atomic set (our idea)

Summary

- Learning **semidefinite programming** regularizers from data
  - Generalize dictionary learning, which gives linear programming regularizers

- Q: Data more likely to lie near faces of certain convex sets?

$A($  $)$ vs $\mathcal{A}($  $)$

  - **What do high-dimensional data really look like?**
  - Can physics help us answer this question?

users.cms.caltech.edu/~venkatc