AMATH 840: Advanced Numerical Methods for Computational and Data Sciences

Giang Tran

Department of Applied Mathematics, University of Waterloo

Jan 19, 2021

Recall: Greedy Algorithms for Compressive Sensing

- Given $A \in \mathbb{C}^{m\times n}$ with unit columns and $y \in \mathbb{C}^m$, find a s-sparse vector $w \in \mathbb{C}^n$ s.t. $y = Aw$.
- $\mu_1(s) := \max_{k\in[n]} \max \left\{ \sum_{j\in S} |\langle a_k, a_j\rangle|, S \subset [n], |S| = s, k \notin S \right\}$.
- $\mu(A) := \max_{1\le k\neq j\le n} |\langle a_k, a_j\rangle|$.
- If Coherence condition, then every s-sparse vector $w \in \mathbb{C}^n$ is exactly recovered from $y = Aw$ after at most s iterations of the method.
  - For OMP: $\mu_1(s) + \mu_1(s-1) < 1$ or $\mu(A) < \dfrac{1}{2s-1}$.
  - For IHT: $\mu_1(2s) < 1$ or $\mu(A) < \dfrac{1}{2s}$. .
  - For HTP: $2\mu_1(s) + \mu_1(s-1) < 1$ or $\mu(A) < \dfrac{1}{3s-1}$. .

Lecture 05: ℓ_1-Minimization and Compressive Sensing

- Different ℓ_1-Minimization Problems and Their Relations
- Popular ℓ_1-Minimization Algorithms and Available Codes
- Examples of Sparse Optimization Problems
- Exact Recovery of Sparse Vectors via Basis Pursuit: Null Space Property

ℓ_1-Minimization Problems ⇐ fast alg., stability robustness

- Basis pursuit:

$z^\#$
$$\min_{z \in \mathbb{C}^n} \|z\|_1 \quad s.t. \quad Az = y. \qquad (BP)$$

$O^{\#}_{x}$

- Basis pursuit denoising:

z_{bp}
$$\min_{z \in \mathbb{C}^n} \|z\|_1 \quad s.t. \quad \|Az - y\|_2 \leq \sigma, \qquad (BP_\sigma)$$

$\sigma = \|Az_{qp} - y\|_2$

$T_{z_{qp}}$

or

z_{qp}
$$\min_{z \in \mathbb{C}^n} \frac{1}{2}\|Az - y\|_2^2 + \lambda\|z\|_1. \qquad (QP_\lambda)$$

$\tau = \|z_{bp}\|_1$

- Lasso:

z_{ls}
$$\min_{z \in \mathbb{C}^n} \frac{1}{2}\|Az - y\|_2^2 \quad s.t. \quad \|z\|_1 \leq \tau. \qquad (LS_\tau)$$

ℓ_1-Minimization Problems (cont'd)

Theorem 5.1 (Relations between BP_σ, QP_λ, and LS_τ).

1. If z_{qp} is a minimizer of (QP_λ) with $\lambda > 0$, then there exists $\sigma = \sigma_{z_{qp}} \geq 0$ such that z_{qp} is a minimizer of (BP_σ).
2. If z_{bp} is a unique minimizer of BP_σ with $\sigma \geq 0$, then there exists $\tau = \tau_{z_{bp}} \geq 0$ such that z_{bp} is a unique minimizer of (LS_τ).
3. If z_{ls} is a minimizer of (LS_τ) with $\tau > 0$, then there exists $\lambda = \lambda_{z_{ls}} \geq 0$ such that z_{ls} is a minimizer of QP_λ.

Proof Sketch.

- $(QP_\lambda \Rightarrow BP_\sigma)$. Set $\sigma := \|Az_{qp} - y\|_2$.
- $(BP_\sigma \Rightarrow LS_\tau)$. Set $\tau := \|z_{bp}\|_1$.
- $(LS_\tau \Rightarrow QP_\lambda)$. See Theorem B.28 from "A Mathematical Introduction to Compressive Sensing", by S. Foucart and H. Rauhut.

ℓ_1-Minimization Problems (cont'd)

With suitable values of σ, λ, τ, the solutions of $BP_\sigma, QP_\lambda, LS_\tau$ coincide.

- If A is orthogonal, a suggestion is $\lambda = \sigma\sqrt{2\log(n)}$. [1]
- In general, the relations among σ, λ, τ cannot be known a priori. [2]
- If λ is large enough, the solution of QP_λ problem is $z_\lambda = 0$.

Theorem 5.1. 2 (BP vs QP_λ). Assume that $Aw = y$ has a solution. For each $\lambda > 0$, let z_λ be a minimizer of (QP_λ). If the (BP) problem has a unique solution $z^\#$, then

$$\lim_{\lambda \to 0^+} z_\lambda = z^\#.$$

---

[1] *Atomic Decomposition by Basis Pursuit*, by Chen, Donoho, and Saunders, SIAM Review, 2001.

[2] *Probing the Pareto frontier for basis pursuit solutions*, by E. van den Berg and M. P. Friedlander, SIAM J. on Scientific Computing, 2008.

ℓ_1-Minimization Problems (cont'd)

Proof Sketch. The detailed proof can be found in Proposition 15.1, "A Mathematical Introduction to Compressive Sensing", by S. Foucart and H. Rauhut.

$$\text{BP}: \quad z^{\#} = \underset{z}{\operatorname{argmin}} \{ \|z\|_1 \ \text{s.t} \quad y = Az \}$$

$$\text{QP}_\lambda: \quad z_\lambda = \underset{z}{\operatorname{argmin}} \ \frac{1}{2} \|Az - y\|_2^2 + \lambda \|z\|_1$$

ℓ_1-Algorithms: SPGL1 $\min_z \|z\|_1 \;\; \text{s.t} \;\; \|Az - y\|_2 \leq \sigma$ //

- Paper: E. van den Berg and M. P. Friedlander, Probing the Pareto frontier for basis pursuit solutions, SIAM J. on Scientific Computing, 2008.

- Goal: Solve BP_σ, where σ is approximately known. It is also used to solve the BP ($\sigma = 0$) and Lasso.

- Main idea: Solve a sequence of Lasso problem $(LS_{\tau_k})_k$ using a spectral projected-gradient algorithm, where the τ_k are the Newton iterates of $\phi(\tau) := \|y - Az_\tau\|_2 = \sigma$. Here z_τ is the optimal solution of (LS_τ).

- Matlab codes (from the authors): Download the zip file from
  `https://friedlander.io/spgl1/install`

- Python codes:
  - Link: `https://spgl1.readthedocs.io/en/latest/index.html`.
  - Install the package within your current environment (Google colab, Jupyter notebook,...):

    ```
    pip install spgl1
    ```

ℓ_1-Algorithms for (QP_λ) Problem

$$\min_z \frac{1}{2}\|Az-y\|_2^2 + \lambda\|z\|_1$$

- Algorithms: FISTA [3], Nesterov's 2nd method [4], SpaRSA [5], Primal-dual algorithm[6], Augmented Lagrangian / Split-Bregman algorithm[7], ...

- Python packages: scikit-learn package.
  - Link: `https://scikit-learn.org/stable/modules/linear_model.html` LASSO $- QP_\lambda$
  - Solve the (QP_λ) by coordinate descent method [8].

---

[3] *A Fast Iterative Shrinkage-Thresholding Algorithm*, by Beck & Teboulle, SIAM J. Imaging Sciences, 2009.

[4] *Gradient Methods for Minimizing Composite Objective Function*, by Nesterov.

[5] *Sparse Reconstruction by Separable Approximation*, by Wright, Nowak, and Figueiredo.

[6] *A First-Order Primal-Dual Alg. for Convex Problems with Applications to Imaging*, by Chambolle & Pock.

[7] *The Split Bregman Method for L1-Regularized Problems*, by Goldstein and Osher.

[8] *Regularization Path For Generalized linear Models by Coordinate Descent*, by Friedman, Hastie and Tibshirani.

ℓ_1-Algorithms for (QP_λ) Problem (cont'd)

$\min \frac{1}{2}\|Az-y\|_2^2 + \lambda\|z\|_1$

Remarks:

- Global rate of convergence $\mathcal{O}(1/k^2)$ can be achieved, for example, with FISTA and Nesterov's 2nd method. [9]

- The speed of some algorithms for ℓ_1-minimization problems does not depend on s, such as the primal-dual algorithm $\rightarrow$ Use ℓ_1-minimization solvers for mildly large s.

- Debiasing technique: Suppose z_{sol} is the num. soln. of the (QP_λ) problem. Let $S := \mathrm{supp}(z_{final})$ (sol) and solve $|entry| > 10^{-10}$

$$\min\{\|Az-y\|_2^2 : \mathrm{supp}(z) \subset S\}.$$

$\min\{\|A_S z_S - y\|_2^2\}$

- For SpaRSA, when λ is small (more non-zeros in z), solve a sequence of (QP_{λ_k}) problems, where[10]

$$\|A^T y\|_\infty > \lambda_0 > \lambda_1 > \cdots > \lambda_{final} > 0.$$

[9] http://www.seas.ucla.edu/~vandenbe/236C/lectures/fista.pdf

[10] Sparse Optimization Methods, slides by S. Wright.

Example 1: Solve the BP and BPDN by SPGL1

1 Generate Data

$A \in \mathbb{R}^{m\times n}$ Gaussian Random Matrix

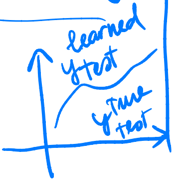
$m=200,\ n=500,\ s=20$

$w_0 \in \mathbb{R}^n$ = a Gaussian Random s-sparse

$y = Aw_0 \in \mathbb{R}^m$

$y_{noise} = Aw_0 + \text{noise}$

2 Solve $\min \|w\|_1$ s.t. $Aw=y$ via spgl1



Solve σ = noiselevel*1.1

$\min \|w\|_1$ s.t. $\|Aw-y\|_2 \le \sigma$

Example 2: Discover Governing Equations from Limited Data Using Monomial Approximation

$\left(x^{(k)}, \dot{x}^{(k)}\right)_{k=1}^{m}$

$x^{(k)} \in \mathbb{R}^{50}$

Example 2: Given $\{x^{(k)}\}_{k=1}^{m} \subset \mathbb{R}^{50}$ are the collections of some short trajectories generated from i.i.d. random initializations of some unknown ODE: $\dfrac{dx}{dt} = F(x,t), \ x \in \mathbb{R}^{50}$. Find F.

- Assume F is well-approximated by a polynomial where only a few monomial terms are important (active).
- Simulated data: Training data are obtained from solving the Lorenz 96 system

$$\frac{dx_k}{dt} = -x_{k-2}x_{k-1} + x_{k-1}x_{k+1} - x_k + 8, k = 1, ..., 50.$$

Approximate $\dfrac{dx_k}{dt}$ by finite difference method.

$\underline{x} = (x_1, \dots, x_{50})$ $\qquad \dfrac{dx_k}{dt} = F_k(x_1, \dots, x_{50})$

- Mathematical formulation:

Given $\left\{ \underline{x}^{(j)}, \dot{\underline{x}}^{(j)} \right\}_{j=1}^{m}$. Find $F_1, \dots, F_{50}$

$$\dot{x}_1 = F_1(x_1, \dots, x_{50}) = c_0 + c_1 x_1 + \dots + c_{50} x_{50} + c_{51} x_1^2 + c_{52} x_1 x_2 + \dots + c_r x_{50}^2$$

$$\begin{pmatrix} \dot{x}_1(t_1) \\ \dot{x}_1(t_2) \\ \vdots \end{pmatrix}_{m\times 1} = \underbrace{\begin{pmatrix} 1 & x_1(t_1) & x_2(t_1) & \dots & x_{50} & x_1^2 & x_1 x_2 & \dots & x_{50}^2 \\ 1 & x_1(t_2) & x_2(t_2) & \dots & & & & & \\ & & & & & & & & \end{pmatrix}}_{A}{}_{m\times(r+1)} \underbrace{\begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_r \end{pmatrix}}_{\vec{c}}$$

Find $\vec{c}$

Find $\vec{c}$ so that

$$\vec{y} : \begin{pmatrix} \dot{x}_1(t_1) \\ \vdots \\ \dot{x}_i(t_m) \end{pmatrix} = A \vec{c}$$

s.t c is sparse

$$\min \|c\|_1 \quad \text{s.t} \quad y = Ac$$

$$\|y - Ac\|_2 \leq \sigma$$

$$\vec{x} = (x_1, x_2, x_3)^T$$

$$A = \begin{pmatrix} 1 & x_1 & x_2 & x_3 & x_1^2 & x_1 x_2 & x_1 x_3 & x_2^2 & x_2 x_3 & x_3^2 . \end{pmatrix}$$

Example 3: Learn Nonlinear Functions from Data Using Random Features

Example 3: NACA Sound Dataset from UCI datasets
`https://archive.ics.uci.edu/ml/datasets/airfoil+self-noise`

- Data description: 1503 data. This problem has the following inputs:

  1. Frequency, in Hertzs.
  2. Angle of attack, in degrees.
  3. Chord length, in meters.
  4. Free-stream velocity, in meters per second.
  5. Suction side displacement thickness, in meters.

  The only output is: Scaled sound pressure level, in decibels.
- Goal: Learn the function F s.t. Output = F(Inputs).

- Approximation: $\vec{x} = (x_1, x_2, x_3, x_4, x_5)^T \xmapsto{F} y \in \mathbb{R}$

$d = 5$

Motivation Shallow network

$$\underset{1\times 1}{y} \approx \underset{1\times H}{W_2}\, \sigma(\underset{H\times d}{W_1}\, \underset{d\times 1}{x} + b_1) + b_2$$

$\uparrow$ learn

$\underbrace{\sigma(W_1 x + b_1)}_{\text{build } A^T}$

Randomize W_1, b_1

build A

fix

$H \to \infty$

$$y \approx W_2 A + b_2$$

$$\boxed{y = A\, W_2^T}, \quad A = (\sigma(W_1 x + b_1))^T \in d \times H$$

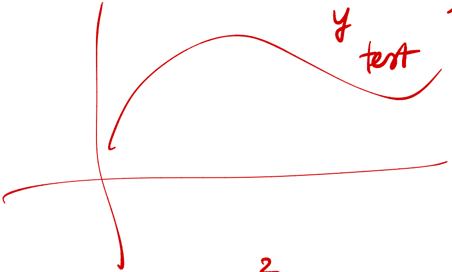
Relu, sin, tanh,

Find $w_2^T = c$ sparse s.t $y = \underline{A} \, \underline{\underline{w_2^T}}$

$\underline{\underline{H >> d}}$

$\underline{\text{solve spgl}_1}$

Plot



$$\frac{1}{\# \text{test}} \| y_{\text{learned}} - y_{\text{true}} \|_2^2$$

Null Space Property

- Provide a necessary and sufficient condition for exact recovery of sparse vectors via basis pursuit.

Definition 5.1. A matrix $A \in K^{m \times n}$ is said to satisfy

- The null space property relative to a set $S \subset [n]$ if

$$\|v_S\|_1 < \|v_{S^c}\|_1 \quad \text{for all } v \in \ker A \setminus \{0\}.$$

- The null space property of order s if it satisfies the null space property relative to any set $S \subset [n]$ with $|S| \leq s$.