AMATH 840: Advanced Numerical Methods for Computational and Data Sciences

Giang Tran

Department of Applied Mathematics, University of Waterloo

Jan 05, 2021

Lecture 01

- Course Introduction
- Basic Steps of a Learning Process
- Least Squares Solution of an Underdetermined Linear System. Pros and Cons.

Course Outline

Goal: Study some computational and mathematical perspectives of machine learning and data science.

1. Sparse Optimization and Compressed Sensing: Underdetermined systems, reconstruction guarantees, sparse approximation, sparse optimization solvers, with applications to image processing, model selection, and parameter estimation.
2. Supervised Learning: Kernel methods, reproducing kernel Hilbert Spaces, learning from data, overfitting, hyperparameter selection.
3. Neural Networks: Mathematical formulations of popular NN architectures, universal approximation, adjoint methods & automatic differentiation, implicit and explicit regularizations, stochastic gradient method and its accelerations, overparametrization, the importance of effective initialization.
4. Randomized Linear Algebra: Johnson-Lindenstrauss lemma, matrix approximation by sampling, randomized QR and SVD, random projections, with applications to model reduction and large-scale problems.

Course Logistics

- Student Assessments: 50% Assignments + 50% Final project.
- Assignments: Theoretical + Computational questions.
- Final Project:
  - Each team = Individual or a group of two students.
  - Each team gives a presentation of 25 minutes.
  - Each team submits the slides and a short report (10-25 pages).

Course Logistics

Course Websites:

- LEARN: To check course outline, course notes, recorded videos, assignments, supplementary materials, and important announcements.
- Crowdmark: To submit and see marked assignments. For each assignment, you will receive an invitation from Crowdmark to submit your assignment.
- Discussion Forum: To pose questions about lectures, assignments, textbooks, ..., please sign up for the course discussion board at Piazza, via the following link: `piazza.com/uwaterloo.ca/winter2022/amath840`.

More details can be found in the course outline on Learn or on `online.uwaterloo.ca`.

Basic Steps of a Learning Process

1. Collect and preprocess data: data cleaning, data augmentation, normalized/standardized data.
   - Data resources: public datasets (UCI dataset, Kaggle,...), data from experiments, simulated data.
2. From raw or preprocessed data, generate
   - Training data = a collection of samples that will be used to learn the model. For example, in a regression problem, given:

$$(\underline{x_i}, y_i)_{i=1}^{m}: \quad \text{where} \quad \begin{aligned} m &:= \#\text{training samples} \\ \underline{x_i} &:= \text{sample's features/input data} \in \mathbb{R}^n \\ y_i &:= \text{sample's label/output data} \in \mathbb{R}. \end{aligned}$$

   - Validation (test) data = a collection of samples that will be used to validate(test) the learned model.

Basic Steps of a Learning Process (cont'd)

3. Choose
   - A learning model: $\hat{y}_i = f(\underline{x_i}) \approx y_i, \quad \forall i \in [m]$. For example,
     - Linear model:
       $$f(x) = f(x; w) = x_1 w_1 + \ldots + x_n w_n = x^T w.$$
     - Generalized linear model: learn a nonlinear function f,
       $$f(x) = \varphi_1(x) w_1 + \ldots + \varphi_p(x) w_p,$$
       where $\varphi_k(x)$ are prescribed nonlinear functions.
     - Neural network model:
       $$f(x) = W_2 \sigma(W_1 x + b_1) + b_2.$$
   - A loss function + (optional) a regularization: How well a model fits the data. For example,
     $$\mathcal{L} = \frac{1}{2m} \sum_{i=1}^{m} |y_i - \hat{y}_i|^2.$$
4. Learn the model (model parameters) to minimize the loss on training data.
5. Compute the generalization error, i.e., error of the trained model on new data.

Linear Models: $Aw = \hat{y}$

- From $y_i \approx \hat{y}_i = \underline{x_i}^T w$ for all $i = 1, \ldots, m$, we can rewrite as

$$\mathbf{y} \approx \hat{y} = Aw,$$

where $\mathbf{y} = [y_1, \ldots, y_m]^T \in \mathbb{R}^m, \hat{y} = [\hat{y}_1, \ldots, \hat{y}_m]^T \in \mathbb{R}^m$, and

$$A = \begin{bmatrix} \underline{x_1}^T \\ \underline{x_2}^T \\ \vdots \\ \underline{x_m}^T \end{bmatrix} \in \mathbb{R}^{m \times n}.$$

Linear Models: $Aw = y$

- A classical problem in linear algebra: m measurements, n unknowns. Given $y \in \mathbb{C}^m$ and $A \in \mathbb{C}^{m \times n}$,

$$\text{Find } w \in \mathbb{C}^n : Aw = y.$$

- Case 1: # measurements $\geq$ # unknowns. The system is overdetermined or determined $\Rightarrow$ The problem is easily solved.

- Case 2: # measurements $<$ # unknowns. The system is underdetermined. Assume that A is full rank. The solution $w \in$ an $(n - m)$ dimensional subspace.
  - Without additional information, it is impossible to recover w from y.
  - Under certain assumptions, it is possible to reconstruct w from y. Moreover, efficient reconstruction algorithms do exist.

Underdetermined system $Aw = y$: Least Squares Solution

- If we assume w has the smallest Euclidean norm:

$$\min_{w \in \mathbb{C}^n} \frac{1}{2}\|w\|_2^2 \quad s.t. \quad Aw = y,$$

  then $w_{ls} = A^*(AA^*)^{-1}y$.
  - 1^{st} Approach: Use the Lagrange multiplier. (Prove in class).
  - 2^{nd} Approach: Use the Fundamental Theorem of Linear Algebra and Best Approximation Theorem.

- Matlab code: $w = A \backslash y$.
- Python code:

```
import numpy as np
# Load A and y .... #
w_ls = np.linalg.lstsq(A, y, rcond=None)[0]
```

Underdetermined system $Aw = y$: Least Squares Solution

- Pros: A closed-form solution, $w_{ls} = A^*(AA^*)^{-1}y$.
- Cons:
  1. Least squares solutions likely overfit the data (See the codes).
  2. Least squares solutions are not robust to noisy measurements.
  3. In many applications, the solution with smallest Euclidean norm is not the expected solution. For example, reconstruct a one-dimensional discrete signal $f : \{1, \ldots, n\} \to \mathbb{C}$ from a partial collection of its Fourier coefficients $\{\hat{f}(\xi_1), \ldots, \hat{f}(\xi_m)\}$. Note that $m < n$.

Lecture 02

- Sparse Solutions of an Underdetermined Linear System
- Introduction to Compressive Sensing: Main Questions
- Why ℓ_1 for Sparsity? Illustration and Mathematical Proof.
- Compressive Sensing: Minimum Number of Measurements

Underdetermined system $Aw = y$: Sparse Solution

- Another assumption: w is a sparse vector, i.e., most components of w are 0. Note that we don't know the locations of the nonzero entries.
- Does the sparsity assumption valid?
  - Related to simplicity, bet-on-sparsity principle, sparsity-of-effects principle, Pareto principle.
    - "Use a procedure that does well in sparse problems, since no procedure does well in dense problems." (The Elements of Statistical Learning, by Hastie, Tibshirani, and Friedman).
    - A system is usually dominated by main effects and low-order interactions.
    - Pareto principle: 80/20 rule or the law of the vital few.
  - Many real-world signals and images are compressible, i.e., well-approximated by sparse signals after an appropriate change of basis: MP3 signals, JPEG images,... (See the codes).

Example: Sampling Theory

Reconstruct

$$f(t) = \sum_{k=-M}^{M} w_k e^{2\pi i k t}, \quad t \in [0, 1].$$

from m samples $f(t_1), \ldots, f(t_m)$, where $\{t_1, \ldots, t_m\} \subset [0, 1]$.

- Formulation:

  $$y = Aw,$$

  where $A \in \mathbb{C}^{m \times n}$, $n = 2M + 1$, and

  $$A_{l,k} = e^{2\pi i k t_l}, \quad l = 1, \ldots, m; \quad k = -M, \ldots, M.$$

- It is possible to recover f from a few samples, under certain conditions $\rightarrow$ Compressive sensing beats Shannon sampling theorem.

Example: Sparse Approximation

- Suppose a vector $y \in \mathbb{C}^m$ is well-approximated by a few term from prescribed elements $\underline{a_1}, \ldots, \underline{a_p} \in \mathbb{C}^m$.
- Formulation:
$$y = c_1 \underline{a_1} + \cdots + c_p \underline{a_p},$$
s.t. $c = (c_1, \ldots, c_p)^T \in \mathbb{C}^p$ is sparse.
- Applications: Compression, denoising, data separation, model discovery.

Compressive Sensing Problem

- Goal: Compress and Sense (acquire) data at the same time.
  - Acquire the compressed version of a signal directly via much fewer measured data than the signal length.
  - Reconstruct an s-sparse vector $w \in \mathbb{C}^n$ from an underdetermined system $y = Aw \in \mathbb{C}^m$, where $m \ll n$.
- Definition: A vector $w \in \mathbb{C}^n$ is called s-sparse if at most s of its entries are nonzero.
- Challenges: The locations of the non-zero entries of w is unknown $\longrightarrow$ Introduce the nonlinearity.

Compressive Sensing Problem

- Main questions:
  1. What matrices A are suitable? $\leftarrow$ Need to design a suitable linear measurement process.
  2. What is the optimal value for $\#$ measurements? $\leftarrow$ Should depends on the compressed size, not on its uncompressed size!
  3. What are efficient (fast, stable, robust) reconstruction algorithms?
- Advantages of compressive sensing:
  1. Measurements are sparse in a known basis or compressible.
  2. Measurements are expensive or require a lot of time.

Other Problems of Compressive Sensing

1. Robustness: Output measurements are contaminated by noise.

$$y = Aw + z, \quad \|z\|_2 \leq \varepsilon.$$

2. Stability: w is not sparse, but is well-approximated by a sparse vector (compressibility).

Notations

- The support of a vector $w \in \mathbb{C}^n$, $\text{supp}(w)$, is the index set of its nonzero entries:

$$\text{supp}(w) := \{j \in [n] : w_j \neq 0\}.$$

- A vector $w \in \mathbb{C}^n$ is called s-sparse if at most s of its entries are nonzero.

- The p-norm of a vector $w \in \mathbb{C}^n$ for $p \geq 1$:

$$\|w\|_p := \Big(\sum_{j=1}^{n} |w_j|^p \Big)^{1/p}.$$

  Convention: $\|w\|_0 := |\text{supp} w|$.

- The ℓ_p-error of best s-term approximation to a vector $w \in \mathbb{C}^n$:

$$\sigma_s(w)_p := \inf\{\|w - z\|_p : \ z \in \mathbb{C}^n \text{ is } s\text{-sparse}\}.$$

Compressive Sensing Problem: Models

- ℓ_0-minimization: NP-hard in general.

$$\min \|z\|_0 \quad s.t. \quad Az = y.$$

- ℓ_1-minimization (convex relaxation of the ℓ_0-minimization):

$$\min \|z\|_1 \quad s.t. \quad Az = y \quad \text{(Basis Pursuit)}.$$

Other models:

$$\min \|z\|_1 \quad s.t. \quad \|Az - y\|_2 \leq \eta \quad \text{(Basis Pursuit Denoising)}.$$

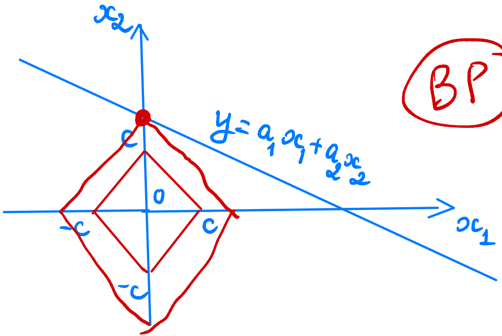
or

$$\min \|Az - y\|_2^2 \quad s.t. \quad \|z\|_1 \leq \tau \quad \text{(Lasso)}.$$

Why ℓ_1 for Sparsity?

- The ℓ_1-norm $\|\cdot\|_1$ is a convex function $\Rightarrow$ The ℓ_1-minimization problem can be solved by efficient algorithms from convex optimization.

- Ilustration of the ℓ_1-minimization induces sparse solutions:



BP

$n = 2$ $\quad m = 1$

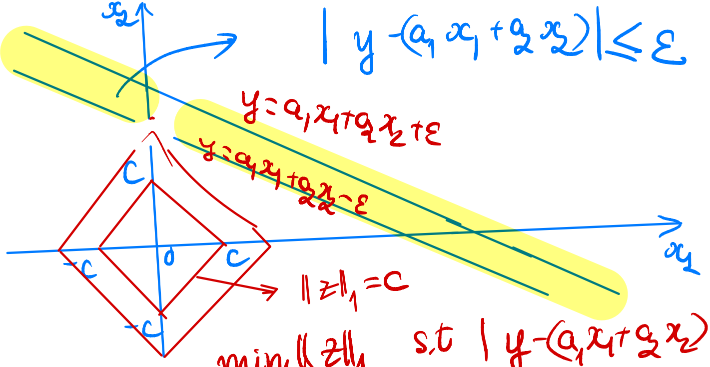
$$\min_{z \in \mathbb{R}^2} \|z\|_1 \ \text{s.t.} \ y = Az = a_1 z_1 + a_2 z_2 \in \mathbb{R}$$

$$\{\|z\|_1 = |z_1| + |z_2| = t, \ t \geq 0\} \cap \{z : y = Az\}$$

Why ℓ_1 for Sparsity?

BPDN

- The ℓ_1-norm $\|\cdot\|_1$ is a convex function $\Rightarrow$ The ℓ_1-minimization problem can be solved by efficient algorithms from convex optimization.
- Ilustration of the ℓ_1-minimization induces sparse solutions:



$$\min_{z\in\mathbb{R}^2} \|z\|_1 \quad \text{s.t } |y-(a_1x_1+a_2x_2)| \le \varepsilon$$

Why ℓ_1 for Sparsity?

Theorem 1. Given $A \in \mathbb{C}^{m\times n}$ and $y \in \mathbb{C}^m$. If the basis pursuit problem:

$$\min_{w\in\mathbb{C}^n} \|w\|_1 \quad s.t. \quad y = Aw$$

has a unique solution w^*, then $|\operatorname{supp} w^*| \leq m$. In other words, $w^* \in \mathbb{C}^n$ is at most m-sparse.

Sketch Proof. Denote a_j the columns of A, where $1 \leq j \leq n$. We will prove that $\{a_j : j \in \operatorname{supp}(w^*)\}$ is linearly independent by contradiction. Indeed, assume

$$\sum_{j\in\underline{S}} c_j a_j = 0, \quad \text{where} \quad c_j \in \mathbb{C}, \quad \sum_{j\in\underline{S}} |c_j|^2 > 0.$$

Denote $\underline{c} \in \mathbb{C}^n$ s.t. $\underline{c}_{\underline{S}} = [c_j : j \in \underline{S}]$ and $\operatorname{supp}(\underline{c}) = \underline{S}$.

Consider $z = w^* + t\underline{c}$, where $t \in \mathbb{C}$. Then

$$Az = Aw = y \quad \text{and} \quad \mathrm{supp}(z) = \underline{S}.$$

By the uniqueness assumption,

$$\|w^*\|_1 < \|z\|_1 = \sum_{k=1}^{n} |z_k| = \sum_{k=1}^{n} z_k \mathrm{sign}(z_k) = \sum_{k \in \underline{S}} (w_k^* + tc_k)\,\mathrm{sign}(w_k^* + tc_k)$$

With t small enough, $\forall |t| < \varepsilon$, we have

$$\mathrm{sign}(w_k^* + tc_k) = \mathrm{sign}(w_k^*).$$

Then

$$\|w^*\|_1 < \sum_{k \in \underline{S}} (w_k^* + tc_k)\,\mathrm{sign}(w_k^*) = \|w^*\|_1 + t \sum_{k \in \underline{S}} \mathrm{sign}(w_k^*) c_k.$$

The last inequality can't be hold for all t s.t. $|t| < \varepsilon$ since $\sum_{k \in \underline{S}} \mathrm{sign}(w_k^*) c_k$ is nonzero (due to the uniqueness of w^*) and we can choose t small enough on the opposite sign of $\sum_{k \in \underline{S}} \mathrm{sign}(w_k^*) c_k$.

Compressive Sensing Problem: Number of Measurements

Theorem 2. Given $A \in \mathbb{C}^{m \times n}$, the following properties are equivalent:

1. Every s-sparse vector $w \in \mathbb{C}^n$ is the unique s-sparse solution of $Az = Aw$. That is, if $Az = Aw$ and both z and w are s-sparse, then $z = w$.
2. Every set of $2s$ columns of A is linearly independent.

Proof. (1) $\Rightarrow$ (2) Suppose $a_1, \ldots, a_n$ are the columns of A. Wlog, we will prove the first (2s) columns of A are linearly independent. Consider:

$$c_1 a_1 + c_2 a_2 + \ldots + c_{2s} a_{2s} = 0, \quad \text{where} \quad c_1, \ldots, c_{2s} \in \mathbb{C}.$$

Then

$$c_1 a_1 + \ldots + c_s a_s = -c_{s+1} a_{s+1} - \ldots - c_{2s} a_{2s}.$$

$$A[c_1, \ldots, c_s, 0, \ldots, 0]^T = A[0, \ldots, 0, -c_{s+1}, -c_{s+2}, \ldots, -c_{2s}, 0, \ldots, 0]^T.$$

By the assumption on (1), we have

$$[c_1, \ldots, c_s, 0, \ldots, 0]^T = [0, \ldots, 0, -c_{s+1}, -c_{s+2}, \ldots, -c_{2s}, 0, \ldots, 0]^T$$

$$c_1 = \ldots = c_{2s} = 0.$$

(2) $\Leftarrow$ (1) Exercise.

Compressive Sensing Problem: Number of Measurements

Theorem 2. Given $A \in \mathbb{C}^{m \times n}$, the following properties are equivalent:

1. Every s-sparse vector $w \in \mathbb{C}^n$ is the unique s-sparse solution of $Az = Ax$. That is, if $Az = Ax$ and both z and x are s-sparse, then $z = x$.
2. Every set of $2s$ columns of A is linearly independent.

Corollary 2.1. If it is possible to reconstruct every s-sparse vector $w \in \mathbb{C}^n$ from the measurements $y = Ax \in \mathbb{C}^m$, then $m \geq 2s$.

Proof.

We have:

$$m \geq \operatorname{rank}(A) \geq 2s.$$

□

Compressive Sensing Problem: Number of Measurements

Theorem 3. For any integer $n \geq 2s$, there exists a measurement matrix $A \in \mathbb{C}^{m \times n}$ with $m = 2s$ rows such that every s-sparse vector $w \in \mathbb{C}^n$ can be recovered from its measurement vector $y = Aw \in \mathbb{C}^m$ as a solution of

$$\min \|z\|_0 \quad s.t. \quad Az = y.$$

Proof. Example 1: Vandermonde matrix. We will construct a matrix A of size $(2s) \times n$ such that every $(2s)$ columns of A are linearly independent. Pick $t_1 < t_2 < \ldots < t_n$, $t_k \in \mathbb{R}$.

Compressive Sensing Problem: Number of Measurements

Consider

$$A = \begin{bmatrix} 1 & 1 & \dots & 1 \\ t_1 & t_2 & \dots & t_n \\ t_1^2 & t_2^2 & \dots & t_n^2 \\ \vdots & \vdots & & \vdots \\ t_1^{2s-1} & t_2^{2s-1} & \dots & t_n^{2s-1} \end{bmatrix} \in \mathbb{C}^{2s \times n}.$$

Let $S = \{j_1, \dots, j_s\} \subset \{1, \dots, n\}$, $j_k \neq j_l \ \forall k \neq l$ and let $A_S = [a_{j_1} \ a_{j_2} \ \dots a_{j_{2s}}]$ be the submatrix of A formed from the columns $a_{j_1}, a_{j_2}, \dots, a_{j_{2s}}$ of A. Then A_S is a Vandermonde matrix and

$$|\det(A_S)| = \left| \prod_{k<j} (t_j - t_k) \right| \neq 0.$$

Therefore, A_S is invertible and the columns $a_{j_1}, a_{j_2}, \dots, a_{j_{2s}}$ are linearly independent.

Example 2: Sparse recovery from $2s$ Fourier measurements

Given $y_1, \dots, y_{m=\underline{2s}}$ Fourier coefficients

$$y_j = \sum_{k=0}^{n-1} \underline{x(k)}\, \underbrace{e^{-2\pi i j k}}$$

$y_j = \hat{x}(j)$, $x(k) = W$

$x : \{0, \dots, n-1\} \to \mathbb{C}$.

Find $x(0), x(1), \dots, x(n-1)$ s.t x is $\underline{s\text{-sparse}}$

$$\vec{y} = \overset{m\times n}{A} x$$

<u>Prof</u> Prony method

<u>Step 1</u> Find support set of x

<u>Step 2</u> Find x

Example 2: Sparse recovery from $2s$ Fourier measurements

Trick $p(t) = \frac{1}{n} \prod_{k \in S} \left(1 - e^{-2\pi i k/n} e^{+2\pi i t/n}\right)$

where $S = \operatorname{supp}(x)$ don't know yet

Clearly $p(t) = 0$ for all $t \in S$ $t \in \{0, \dots, n-1\}$

$x(t) = 0$ for all $t \notin S$

$p(t)\, x(t) = 0 \quad \forall\, t \in \{0, 1, \dots, n-1\}$

$p.x(t) := p(t)\, x(t)$

$\hat{p} * \hat{x} = \widehat{p \cdot x}$

$\hat{p} * \hat{x}(j) = \sum_{k=0}^{n-1} \hat{p}(k)\, \hat{x}(j-k), \quad 0 \leq j \leq n-1$

$$p(t) = \frac{1}{n} \prod_{k \in \underline{S}} \left(1 - e^{-2\pi i k/n} e^{2\pi i t/n}\right)$$

$$= \sum_{0 \le j \le |\underline{S}|} c_j e^{2\pi i j t/n}$$

$\hat{p}(0) =$ scalar coefficient of $np(t)$

$= nc_0 = 1.$

$\hat{p}(t) = 0$ for all $t > S$

To find p, we only need to find $\hat{p}(1), \ldots, \hat{p}(s)$

Write $\hat{p} * \hat{x} = 0$ explicitly

$$\sum_{k=0}^{n-1} \hat{p}(k)\, \hat{x}(j-k) = 0 \quad \forall j = 0 \to n-1$$

since $\hat{p}(t) = 0 \quad \forall t > s$

$$\sum_{k=0}^{s} \hat{p}(k)\, \hat{x}(j-k) = 0 \quad j = 0 \to s-1.$$

$$\underbrace{\begin{pmatrix} \hat{x} \text{ part} & & \\ & & \\ & & \end{pmatrix} \begin{pmatrix} \hat{p}(1) \\ \vdots \\ \hat{p}(s) \end{pmatrix}}_{} = \underbrace{\begin{pmatrix} \hat{x} \\ \text{part} \\ \text{vector} \\ \\ \end{pmatrix}}_{}$$

$\hat{p} \longrightarrow p \longrightarrow \operatorname{supp} p = \operatorname{supp} x \xrightarrow{\text{Fourier}(x) \quad 2s} x.$ $\boxed{\text{//}}$

Compressive Sensing Problem: Number of Measurements

- Key result: With high probability on the random draw of an $m \times N$ Gaussian or Bernoulli matrix A, all s-sparse vector w can be reconstructed from $y = Aw$ using a variety of algorithms provided that

$$m \geq Cs \ln(n/s),$$

where C is a universal constant (does not depend on s, m, n).

- In practice, $m \geq 4s \ln(n/s)$ works numerically.

References

- A Mathematical Introduction to Compressive Sensing, by S. Foucart and H. Rauhut. Chapters 1 and 2.
- Statistical Learning with Sparsity, The Lasso and Generalizations, by T. Hastie, R. Tibshirani, and M. Wainwright. Chapter 1.
- Data-driven Sicence and Engineering, Machine Learning, Dynamical Systems, and Control, by S. L. Brunton and J. N. Kutz.
- Talks and lectures by T. Tao, S. Foucart, R. Willett, S. Brunton, H. Schaeffer,....