Math 6880/7875: Advanced Optimization
Examples, Part 2

Akil Narayan[1]

[1] Department of Mathematics, and Scientific Computing and Imaging (SCI) Institute
University of Utah

Feburary 1, 2022

Examples in optimization

We'll take a short tour of some examples in optimization:

- Machine learning
  - Model training to deep learning: Training of model parameters
  - Classification: Building of classification models
- Statisics
  - Bayesian inference: Updating beliefs with data
  - Gaussian Processes: Hyperparameter optimization
- PDE-constrained optimization: Optimization with PDE constraints
- Sparse approximation: Compressed sensing and matrix completion
- ~~Stochastic programming: Optimization under uncertainty~~

PDE-constrained optimization

PDE-constrained optimization is an instance of an optimal control problem:

- A system is governed by a PDE, whose solution we wish to behave in a certain way.
- We cannot directly control the solution, but instead can control an input to the PDE.
- Optimization proceeds over the joint control/solution state, subject to the PDE as a constraint.

Examples:

- Desired temperature distribution with input heat/boundary control
- Optimize drug delivery, with drug administration the control
- Shape optimization: optimize pressure distribution subject to aerodynamic shape

PDE-constrained optimization

PDE-constrained optimization is an instance of an optimal control problem:

- A system is governed by a PDE, whose solution we wish to behave in a certain way.
- We cannot directly control the solution, but instead can control an input to the PDE.
- Optimization proceeds over the joint control/solution state, subject to the PDE as a constraint.

Examples:

- Desired temperature distribution with input heat/boundary control
- Optimize drug delivery, with drug administration the control
- Shape optimization: optimize pressure distribution subject to aerodynamic shape

PDE-constrained optimization setup

In a simple setting, PDE-constrained optimization has 3 main ingredients:

- The state variable u, the solution to a PDE (with an input control)
- The control z, an input to a PDE
- The objective function $\mathcal{L}(u; z)$

Additional constraints on the control may also be imposed. The optimization problem to be solved is frequently of the form:

$$\underset{z,u}{\arg\min}\, \mathcal{L}(z, u) = \|\mathcal{S}(u) - s\| + \lambda\|z\|$$

with $\|\cdot\|$ appropriate norms.

- s is some desired/observed solution behavior
- $\mathcal{S}$ is an observation operator, mapping PDE solutions to observations
- $\|z\|$ penalizes "complex" behavior of the control
- u depends on z, implicitly through a PDE $\longrightarrow$ this is a constraint

PDE-constrained optimization setup

In a simple setting, PDE-constrained optimization has 3 main ingredients:

- The state variable u, the solution to a PDE (with an input control)
- The control z, an input to a PDE
- The objective function $\mathcal{L}(u; z)$

Additional constraints on the control may also be imposed. The optimization problem to be solved is frequently of the form:

$$\arg\min_{z,u} \mathcal{L}(z, u) = \|\mathcal{S}(u) - s\|^2 + \lambda \|z\|^2$$

(maybe subject to $0 \leq z(x) \leq 1$)

with $\|\cdot\|$ appropriate norms.

- s is some desired/observed solution behavior
- $\mathcal{S}$ is an observation operator, mapping PDE solutions to observations
- $\|z\|$ penalizes "complex" behavior of the control
- u depends on z, implicitly through a PDE $\longrightarrow$ this is a constraint

alternative: $\arg\min_{z} \|S(u(z)) - s\|^2 + \lambda \|z\|^2$

A simple example

Optimize heat forcing to achieve desired temperature distribution.

$$\begin{aligned} \Delta u &= f, \quad \text{in } \Omega \\ u|_{\partial\Omega} &= d \end{aligned}$$

- u is the PDE solution state
- f is the control
- d is given data, known behavior of u at the boundary
- We are given a target temperature distribution u_*

The optimization problem reads,

was λ

$$\min_{u,f} \|u - u_*\|^2 + \mu \|f\|^2 \quad \text{subject to} \quad \Delta u = f,$$

for appropriate norms. In such problems, there are two broad strategies:

- *Optimize then discretize* – derive optimality conditions and discretize them
- *Discretize then optimize* – discretize the PDE first, then derive (finite-dimensional) optimality conditions

These are not the same!

A simple example

Optimize heat forcing to achieve desired temperature distribution.

$$\begin{aligned} \Delta u &= f, \quad \text{in } \Omega \\ u|_{\partial\Omega} &= d \end{aligned}$$

- u is the PDE solution state
- f is the control
- d is given data, known behavior of u at the boundary
- We are given a target temperature distribution u_*

The optimization problem reads,

$$\min_{u,f} \|u - u_*\|^2 + \mu \|f\|^2 \quad \text{subject to} \quad \Delta u = f,$$

for appropriate norms. In such problems, there are two broad strategies:

- *Optimize then discretize* – derive optimality conditions and discretize them
- *Discretize then optimize* – discretize the PDE first, then derive (finite-dimensional) optimality conditions

These are not the same!

PDE discretization

It's typically easier to discretize then optimize:

$$\Delta u = f, \quad \Longrightarrow \quad \boldsymbol{S}\boldsymbol{u} = \boldsymbol{f},$$

- $\boldsymbol{u}$, $\boldsymbol{f}$ are vector discretizations of u, f
- $\boldsymbol{S}$ is the discretization of Δ

$$\min_{\boldsymbol{u},\boldsymbol{f}} \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{u}_*\|_2^2 + \cancel{\frac{\mu}{\|}}\boldsymbol{f}\|_2^2 \quad \text{subject to} \quad \boldsymbol{S}\boldsymbol{u} = \boldsymbol{f}.$$

$$+ \frac{\mu}{2}\|f\|_2^2$$

Generally: $\|\underline{u}\|_2^2$ is not the right measure for norm.

Instead $\|\underline{u}\|_M^2 = \underline{u}^T \underline{\underline{M}}\, \underline{u}$ for some p.d. matrix $\underline{\underline{M}}$.

The KKT conditions

Optimality conditions: stationarity of the Lagrangian.

$$L(\boldsymbol{u}, \boldsymbol{f}, \boldsymbol{\lambda}) = \frac{1}{2} \|\boldsymbol{u} - \boldsymbol{u}_*\|_{\boldsymbol{M}}^2 + \frac{\mu}{2} \|\boldsymbol{f}\|_M^2 + \boldsymbol{\lambda}^T (\boldsymbol{S}\boldsymbol{u} - \boldsymbol{f}),$$

where $\|\boldsymbol{x}\|_{\boldsymbol{M}}^2 := \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}$ is the finite-dimensional norm defined by discretization.

Stationarity of the Lagrangian requires:

$$\begin{aligned} \left(\frac{\partial \mathcal{L}}{\partial u}\right) \qquad \boldsymbol{M}(\boldsymbol{u} - \boldsymbol{u}_*) + S^T \boldsymbol{\lambda} &= 0 \\ \left(\frac{\partial \mathcal{L}}{\partial F}\right) \qquad \mu \boldsymbol{M} \boldsymbol{f} - \boldsymbol{\lambda} &= 0 \\ \left(\frac{\partial \mathcal{L}}{\partial \lambda}\right) \qquad \boldsymbol{S}\boldsymbol{u} - \boldsymbol{f} &= 0 \end{aligned}$$

Solving this equation yields stationary points.

This problem is frequently very large and expensive to solve.

Formulations

$$\begin{pmatrix} \boldsymbol{M} & 0 & \boldsymbol{S}^T \\ 0 & \mu \boldsymbol{M} & -\boldsymbol{I} \\ \boldsymbol{S} & -\boldsymbol{I} & 0 \end{pmatrix} \overset{\begin{pmatrix} u \\ f \\ \lambda \end{pmatrix}}{\downarrow} = \begin{pmatrix} \boldsymbol{M}\boldsymbol{u}_* \\ 0 \\ 0 \end{pmatrix}$$

There are typically two strategies to proceed:

The "primal-dual" approaches solves the above system directly:

- Is a very large linear system: typically system is not formed directly
- Iterative methods are used: efficient/accurate preconditioners are required

Formulations

$$\begin{pmatrix} \boldsymbol{M} & 0 & \boldsymbol{S}^T \\ 0 & \mu \boldsymbol{M} & -\boldsymbol{I} \\ \boldsymbol{S} & -\boldsymbol{I} & 0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{M}\boldsymbol{u}_* \\ 0 \\ 0 \end{pmatrix}$$

There are typically two strategies to proceed:

The "dual" approach first isolates the dual variables ($\boldsymbol{\lambda}$):

$$\left(\frac{1}{\mu}\boldsymbol{M} + \boldsymbol{S}\boldsymbol{M}^{-1}\boldsymbol{S}^T\right)\boldsymbol{\lambda} = \boldsymbol{S}\boldsymbol{u}_*,$$

and subsequently uses them to solve for the primal variables $\boldsymbol{f}, \boldsymbol{u}$:

$$\boldsymbol{f} = \frac{1}{\mu}\boldsymbol{M}^{-1}\boldsymbol{\lambda}, \qquad \boldsymbol{u} = \boldsymbol{u}_* - \boldsymbol{M}^{-1}\boldsymbol{S}\boldsymbol{\lambda}$$

This requires typically 3 linear algebra solves, and the dual variables equation can be more difficult to invert.

Heat equation results

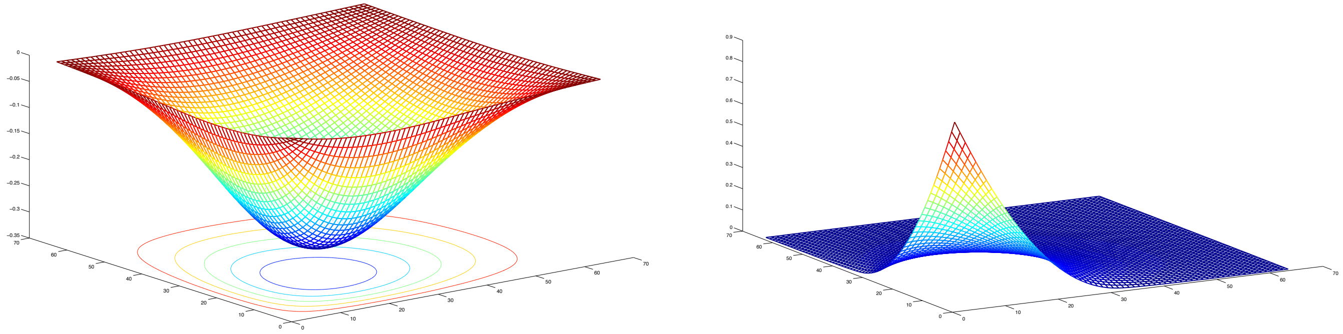


Image: Block-triangular preconditioners for PDE-constrained optimization, Rees & Stoll, 2010

Left: control   Right: state

(The computed state u and target u_* are visually indistinguishable.)

Odds and ends

For a general PDE $\mathcal{P}(z)u = 0$,

$u = \text{"}p^{-1}\text{"} z$

$$\min_{u,z} \|\mathcal{S}u - d\|^2 + \mu \|z\|^2$$

- If $\mathcal{P}$ is a nonlinear PDE (in either z or u), the stationarity conditions become more difficult to compute
- The KKT conditions are nonlinear equations: iterative methods for nonlinear systems used
- The KKT conditions already contain first-order derivatives: gradients for iterative methods involve second derivatives
- These are only necessary optimality conditions in general
- If problems are time-dependent, the discretization size frequently is multiplied by the number of time steps

Odds and ends

For a general PDE $\mathcal{P}(z)u = 0$,

$$\min_{u,z} \|\mathcal{S}u - d\|^2 + \mu \|z\|^2$$

- If $\mathcal{P}$ is a nonlinear PDE (in either z or u), the stationarity conditions become more difficult to compute
- The KKT conditions are nonlinear equations: iterative methods for nonlinear systems used
- The KKT conditions already contain first-order derivatives: gradients for iterative methods involve second derivatives
- These are only necessary optimality conditions in general
- If problems are time-dependent, the discretization size frequently is multiplied by the number of time steps

Odds and ends

For a general PDE $\mathcal{P}(z)u = 0$,

$$\min_{u,z} \|\mathcal{S}u - d\|^2 + \mu \|z\|^2$$

- If $\mathcal{P}$ is a nonlinear PDE (in either z or u), the stationarity conditions become more difficult to compute
- The KKT conditions are nonlinear equations: iterative methods for nonlinear systems used
- The KKT conditions already contain first-order derivatives: gradients for iterative methods involve second derivatives
- These are only necessary optimality conditions in general
- If problems are time-dependent, the discretization size frequently is multiplied by the number of time steps

Odds and ends

For a general PDE $\mathcal{P}(z)u = 0$,

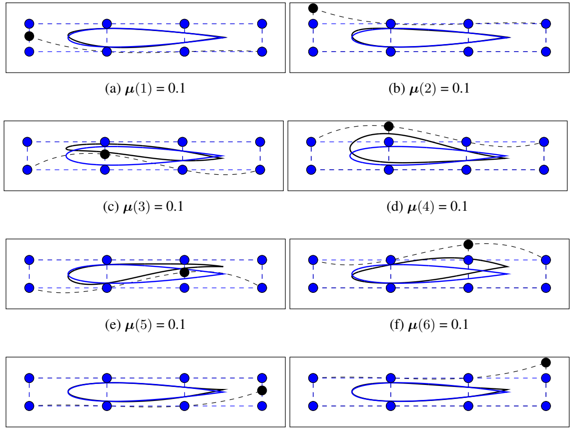
$$\min_{u,z} \left\| \mathcal{S}u - d \right\|^2 + \mu \left\| z \right\|^2$$

- If $\mathcal{P}$ is a nonlinear PDE (in either z or u), the stationarity conditions become more difficult to compute
- The KKT conditions are nonlinear equations: iterative methods for nonlinear systems used
- The KKT conditions already contain first-order derivatives: gradients for iterative methods involve second derivatives
- These are only necessary optimality conditions in general
- If problems are time-dependent, the discretization size frequently is multiplied by the number of time steps

But this works in many cases

Goal: design an airfoil from a parametric class whose steady-state pressure distribution matches a desired target.

PDE model: compressible 3D Euler equations (nonlinear, hyperbolic, time-dependent)



(a) $\mu(1) = 0.1$ (b) $\mu(2) = 0.1$

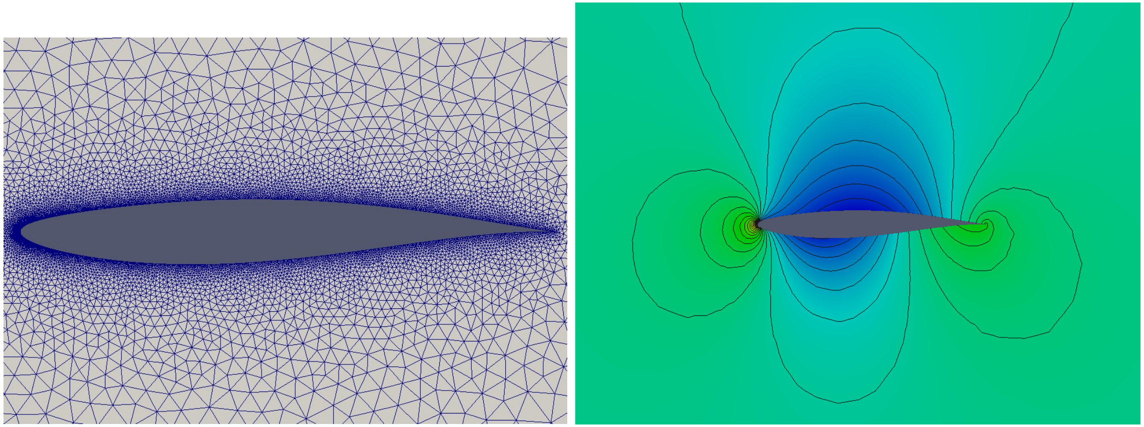
(c) $\mu(3) = 0.1$ (d) $\mu(4) = 0.1$

(e) $\mu(5) = 0.1$ (f) $\mu(6) = 0.1$

(g) $\mu(7) = 0.1$ (h) $\mu(8) = 0.1$

Control: 8-dimensional parameter defining wing shape
State: Pressure (with given target data), which is derived from PDE solution

But this works in many cases



(a) CFD mesh for the Cub-RAE2822 airfoil

(b) Pressure field ($M_\infty = 0.5$, $\alpha = 0.0^\circ$)

Image: Progressive construction of a parametric reduced-order model for PDE-constrained optimization, Zahr & Farhat, 2015

Sparse recovery and compressed sensing

Compressed or *compressive* sampling is a decoding strategy to identify a signal from a small number of measurements.

The basic idea is understandable from Nyquist-Shannon sampling concepts:

Sampling at twice the maximum frequency is necessary and sufficient for general signal recovery

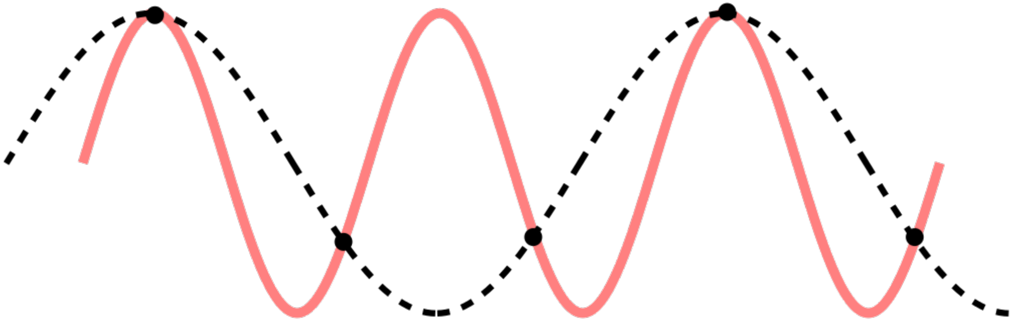


Image: Wikipedia

I.e., it is generally <u>not</u> possible to uniquely recover signals from fewer equispaced measurements.

Encoding and decoding

In the simplest setting, signals are fully represented by a finite-dimensional vector of Fourier Series coefficients:

$$\boldsymbol{c} \in \mathbb{R}^{2N+1} \qquad \longrightarrow \qquad x(t) = c_0 + \sum_{j=1}^{N} c_j \cos(2\pi j t) + c_{j+N} \sin(2\pi j t).$$

The process of *encoding* is transformation of $\boldsymbol{c}$ into a new, typically compressed, representation, e.g.,

$$\boldsymbol{c} \longrightarrow \boldsymbol{x} = (x_1, \ldots, x_M)^T, \qquad x_m = x(t_m),$$

for some sampling times t_m.

Decoding is the process of transforming the encoded representation back into $\boldsymbol{c}$ (hopefully without error).

The simplest form of encoding-decoding is temporal sampling (and its corresponding decoding).

Encoding and decoding

In the simplest setting, signals are fully represented by a finite-dimensional vector of Fourier Series coefficients:

$$\boldsymbol{c} \in \mathbb{R}^{2N+1} \qquad \longrightarrow \qquad x(t) = c_0 + \sum_{j=1}^{N} c_j \cos(2\pi j t) + c_{j+N} \sin(2\pi j t).$$

The process of *encoding* is transformation of $\boldsymbol{c}$ into a new, typically compressed, representation, e.g.,

$$\boldsymbol{c} \longrightarrow \boldsymbol{x} = (x_1, \ldots, x_M)^T, \qquad x_m = x(t_m),$$

for some sampling times t_m.

Decoding is the process of transforming the encoded representation back into c (hopefully without error).

The simplest form of encoding-decoding is temporal sampling (and its corresponding decoding).

Encoding and decoding

In the simplest setting, signals are fully represented by a finite-dimensional vector of Fourier Series coefficients:

$$\boldsymbol{c} \in \mathbb{R}^{2N+1} \qquad \longrightarrow \qquad x(t) = c_0 + \sum_{j=1}^{N} c_j \cos(2\pi j t) + c_{j+N} \sin(2\pi j t).$$

The process of *encoding* is transformation of $\boldsymbol{c}$ into a new, typically compressed, representation, e.g.,

$$\boldsymbol{c} \longrightarrow \boldsymbol{x} = (x_1, \ldots, x_M)^T, \qquad x_m = x(t_m),$$

for some sampling times t_m.

Decoding is the process of transforming the encoded representation back into $\boldsymbol{c}$ (hopefully without error).

The simplest form of encoding-decoding is temporal sampling (and its corresponding decoding).

Decoding under Nyquist-Shannon conditions

$$\boldsymbol{c} \xrightarrow{Encoding} \boldsymbol{x} = (x(t_1), \ldots, x(t_M))^T, \xrightarrow{Decoding} \widetilde{\boldsymbol{c}}$$

According to Shannon-Nyquist if the sampling t_j is equispaced and $M/2 \geqslant F$, where F is the maximum frequency in the signal, then this process is exact.

But we're greedy: this requires $2F$ samples, which is expensive if F is large. Can we do better?

In general, **no**, without suffering *lossy* decoding.

Decoding under Nyquist-Shannon conditions

$$\boldsymbol{c} \xrightarrow{Encoding} \boldsymbol{x} = (x(t_1), \ldots, x(t_M))^T, \xrightarrow{Decoding} \widetilde{\boldsymbol{c}}$$

According to Shannon-Nyquist if the sampling t_j is equispaced and $M/2 \geqslant F$, where F is the maximum frequency in the signal, then this process is exact.

But we're greedy: this requires $2F$ samples, which is expensive if F is large. Can we do better?

In general, **no**, without suffering *lossy* decoding.

Linear decoding

One way to observe the Shannon-Nyquist rate condition is writing this as a linear problem:

$$\begin{aligned} \boldsymbol{A}\boldsymbol{c} &= \boldsymbol{x}, \\ (A)_{m,j} &= \cos(2\pi j t_m), \quad j \leqslant N, \\ (A)_{m,j} &= \sin(2\pi j t_m), \quad j > N. \end{aligned}$$

If $M \geqslant 2N + 1$ and is equispaced over $[0, 1)$, there is a unique solution for $\boldsymbol{c}$.

If $M < 2N + 1$, we violate Shannon-Nyquist. In this case, $\ker(\boldsymbol{A})$ is nonempty, and therefore,

$$\boldsymbol{c} = \boldsymbol{c}_0 + \boldsymbol{v}, \qquad\qquad \boldsymbol{v} \in \ker(\boldsymbol{A}),$$

solves the problem, where $\boldsymbol{c}_0$ is the original signal.

I.e., there are infinitely many (perfectly reasonble) solutions – unique decoding is not possible.

In particular, recovery of the unknown original signal $\boldsymbol{c}$ is practically infeasible.

This is essentially as far as we can go with linear decoding.

Linear decoding

One way to observe the Shannon-Nyquist rate condition is writing this as a linear problem:

$$\boldsymbol{A}\boldsymbol{c} = \boldsymbol{x}, = A c_0 \quad \text{(for some exact } x_0\text{)}$$

$$\begin{aligned}
(A)_{m,j} &= \cos(2\pi j t_m), \quad j \leqslant N, \\
(A)_{m,j} &= \sin(2\pi j t_m), \quad j > N.
\end{aligned}$$

If $M \geqslant 2N+1$ and is equispaced over $[0,1)$, there is a unique solution for $\boldsymbol{c}$.

If $M < 2N+1$, we violate Shannon-Nyquist. In this case, $\ker(\boldsymbol{A})$ is nonempty, and therefore,

$$\boldsymbol{c} = \boldsymbol{c}_0 + \boldsymbol{v}, \qquad\qquad \boldsymbol{v} \in \ker(\boldsymbol{A}),$$

solves the problem, where $\boldsymbol{c}_0$ is the original signal.

I.e., there are infinitely many (perfectly reasonble) solutions – unique decoding is not possible.

In particular, recovery of the unknown original signal c is practically infeasible.

This is essentially as far as we can go with linear decoding.

Linear decoding

One way to observe the Shannon-Nyquist rate condition is writing this as a linear problem:

$$\begin{aligned} \boldsymbol{A}\boldsymbol{c} &= \boldsymbol{x}, \\ (A)_{m,j} &= \cos(2\pi j t_m), \quad j \leqslant N, \\ (A)_{m,j} &= \sin(2\pi j t_m), \quad j > N. \end{aligned}$$

If $M \geqslant 2N + 1$ and is equispaced over $[0, 1)$, there is a unique solution for $\boldsymbol{c}$.

If $M < 2N + 1$, we violate Shannon-Nyquist. In this case, $\ker(\boldsymbol{A})$ is nonempty, and therefore,

$$\boldsymbol{c} = \boldsymbol{c}_0 + \boldsymbol{v}, \qquad \boldsymbol{v} \in \ker(\boldsymbol{A}),$$

solves the problem, where $\boldsymbol{c}_0$ is the original signal.

I.e., there are infinitely many (perfectly reasonble) solutions – unique decoding is not possible.

In particular, recovery of the unknown original signal $\boldsymbol{c}$ is practically infeasible.

This is essentially as far as we can go with linear decoding.

Q: Solving $Ac = x$ ($x = Ac_0$ is given)

$M < 2N+1$: underdetermined

$$c = c_0 + v \quad , v \in \ker(A)$$

$$Ac = x$$

$c = A^+ x$, A^+: Moore-Penrose pseudoinverse

$A = U\Sigma V^T$ is the reduced SVD of A,

then $A^+ = V\Sigma^{-1}U^T$

What kind of a solution to $Ac=x$ is A^+x?

A: A^+x solves: $\min \|c\|_2$ s.t. $Ac = x$

Sparsity

Compressed sensing is a *nonlinear*, optimization-based decoding paradigm.

The argument: assuming extra signal structure allows one to circumvent Shannon-Nyquist.

Define

$$\|\boldsymbol{c}\|_0 := \# \text{ of nonzero entries in } \boldsymbol{c},$$

which is not a norm.

Given $s \in \mathbb{N}$, we say $\boldsymbol{c}$ is an s-sparse vector if $\|\boldsymbol{c}\|_0 \leqslant s$.

The high-level idea: if $\boldsymbol{c}$ is s-sparse, there are only s pieces of information, so probably we can decode with only s pieces of data?

(This is not quite correct since we don't know the support of $\boldsymbol{c}$, the locations of the nonzeros.)

Sparsity

Compressed sensing is a *nonlinear*, optimization-based decoding paradigm.

The argument: assuming extra signal structure allows one to circumvent Shannon-Nyquist.

Define

$$\|\boldsymbol{c}\|_0 := \#\text{ of nonzero entries in } \boldsymbol{c},$$

which is not a norm.

Given $s \in \mathbb{N}$, we say $\boldsymbol{c}$ is an s-sparse vector if $\|\boldsymbol{c}\|_0 \leqslant s$.

The high-level idea: if $\boldsymbol{c}$ is s-sparse, there are only s pieces of information, so probably we can decode with only s pieces of data?

(This is not quite correct since we don't know the support of $\boldsymbol{c}$, the locations of the nonzeros.)

Sparsity

Compressed sensing is a *nonlinear*, optimization-based decoding paradigm.

The argument: assuming extra signal structure allows one to circumvent Shannon-Nyquist.

Define

$$\|\boldsymbol{c}\|_0 := \# \text{ of nonzero entries in } \boldsymbol{c},$$

which is not a norm.

Given $s \in \mathbb{N}$, we say $\boldsymbol{c}$ is an s-sparse vector if $\|\boldsymbol{c}\|_0 \leqslant s$.

The high-level idea: if $\boldsymbol{c}$ is s-sparse, there are only s pieces of information, so probably we can decode with only s pieces of data?

(This is not quite correct since we don't know the support of $\boldsymbol{c}$, the locations of the nonzeros.)

Sparse approximation

This suggests the following optimization problem:

Let $\boldsymbol{c}$ be an unknown s-sparse vector. Assume we have M samples of $\boldsymbol{c}$ in the vector $\boldsymbol{x}$, with associated design matrix $\boldsymbol{A}$.

Establishing that a successful decoder is *possible* leverages the so-called robust null-space property.

Theorem

If $\ker(\boldsymbol{A})$ contains no $2s$-sparse vectors, then there is some decoder that uniquely recovers $\boldsymbol{c}$.

Note that this is a condition on what types of measurements $\boldsymbol{A}$ are permissible.

Such decoders are typically not numerically useful.

Sparse approximation

This suggests the following optimization problem:

Let $\boldsymbol{c}$ be an unknown s-sparse vector. Assume we have M samples of $\boldsymbol{c}$ in the vector $\boldsymbol{x}$, with associated design matrix $\boldsymbol{A}$.

Establishing that a successful decoder is *possible* leverages the so-called robust null-space property.

Theorem

If $\ker(\boldsymbol{A})$ contains no $2s$-sparse vectors, then there is some decoder that uniquely recovers $\boldsymbol{c}$.

Note that this is a condition on what types of measurements $\boldsymbol{A}$ are permissible.

Such decoders are typically not numerically useful.

Why? $\tilde{c} = c + v$, $v \in \ker(A)$

$\Rightarrow \|\tilde{c}\|_0 \geq s+1$ if $v \neq 0$.

c: s-sparse

v: At least $2s+1$ non-zeros

Decoding via optimization

One decoder we might consider minimizes sparsity:

$$\min \|\boldsymbol{c}\|_0 \text{ subject to } \boldsymbol{A}\boldsymbol{c} = \boldsymbol{x}.$$

This provides a reasonable initial point for investigation.

The major problem with this optimization is implementation: $\|\cdot\|_0$ is not convex.

One might consider a *relaxation* of this problem, such as

$$\min \|\boldsymbol{c}\|_* \text{ subject to } \boldsymbol{A}\boldsymbol{c} = \boldsymbol{x},$$

where $\| * \|_*$ is a more "friendly" function to work with, such as a convex function.

Decoding via optimization

One decoder we might consider minimizes sparsity:

$$\min \|\boldsymbol{c}\|_0 \;\; \text{subject to} \;\; \boldsymbol{A}\boldsymbol{c} = \boldsymbol{x}.$$

This provides a reasonable initial point for investigation.

The major problem with this optimization is implementation: $\|\cdot\|_0$ is not convex.

One might consider a *relaxation* of this problem, such as

$$\min \|\boldsymbol{c}\|_* \;\; \text{subject to} \;\; \boldsymbol{A}\boldsymbol{c} = \boldsymbol{x},$$

where $\| * \|_*$ is a more "friendly" function to work with, such as a convex function.

Decoding via optimization

One decoder we might consider minimizes sparsity:

$$\min \|\boldsymbol{c}\|_0 \;\text{ subject to }\; \boldsymbol{A}\boldsymbol{c} = \boldsymbol{x}.$$

This provides a reasonable initial point for investigation.

The major problem with this optimization is implementation: $\|\cdot\|_0$ is not convex.

One might consider a *relaxation* of this problem, such as

$$\min \|\boldsymbol{c}\|_* \;\text{ subject to }\; \boldsymbol{A}\boldsymbol{c} = \boldsymbol{x},$$

where $\| * \|_*$ is a more "friendly" function to work with, such as a convex function.

ℓ^1 minimization

The closest convex ℓ^p-type norm to $\|\cdot\|_0$ is $\|\cdot\|_1$. So we could consider the problem:

$$\min \|\boldsymbol{c}\|_1 \;\text{ subject to }\; \boldsymbol{A}\boldsymbol{c} = \boldsymbol{x}.$$

It is geometrically plausible that this decodes sparse vectors.

But does it correctly decode them?

ℓ^1 minimization

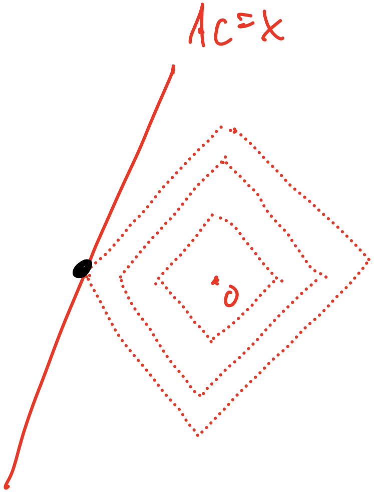
The closest convex ℓ^p-type norm to $\|\cdot\|_0$ is $\|\cdot\|_1$. So we could consider the problem:

$$\min \|\boldsymbol{c}\|_1 \;\text{ subject to }\; \boldsymbol{A}\boldsymbol{c} = \boldsymbol{x}.$$

It is geometrically plausible that this decodes sparse vectors.

But does it correctly decode them?

$c \in \mathbb{R}^{2N+1}$



RIP and decoding

The seminal foundation of compressed sensing is the Restricted Isometry Property (RIP).

Definition

A matrix $\boldsymbol{A}$ satisfies the (s, δ) RIP if

$$(1-\delta)\|\boldsymbol{c}\|^2 \leqslant \|\boldsymbol{A}\boldsymbol{c}\|^2 \leqslant (1+\delta)\|\boldsymbol{c}\|^2,$$

for all vectors $\boldsymbol{c}$ that are s-sparse, where $\|\cdot\|$ is the ℓ^2 norm.

This condition on measurements ensures ℓ^1 optimization is sparsity-promoting:

Theorem

Assume $\boldsymbol{c}_0$ is s-sparse, and assume the measurement matrix $\boldsymbol{A}$ satisfies the RIP condition with constants $(4s, \frac{1}{3})$. Then the optimization

$$\min \|\boldsymbol{c}\|_1 \quad \textit{subject to} \quad \boldsymbol{A}\boldsymbol{c} = \boldsymbol{x}.$$

uniquely recovers $\boldsymbol{c}_0$.

RIP and decoding

The seminal foundation of compressed sensing is the Restricted Isometry Property (RIP).

Definition

A matrix $\boldsymbol{A}$ satisfies the (s, δ) RIP if

$$(1-\delta)\|\boldsymbol{c}\|^2 \leqslant \|\boldsymbol{A}\boldsymbol{c}\|^2 \leqslant (1+\delta)\|\boldsymbol{c}\|^2,$$

for all vectors $\boldsymbol{c}$ that are s-sparse, where $\|\cdot\|$ is the ℓ^2 norm.

This condition on measurements ensures ℓ^1 optimization is sparsity-promoting:

Theorem

Assume $\boldsymbol{c}_0$ is s-sparse, and assume the measurement matrix $\boldsymbol{A}$ satisfies the RIP condition with constants $(4s, \frac{1}{3})$. Then the optimization

$$\min \|\boldsymbol{c}\|_1 \;\; \textit{subject to} \;\; \boldsymbol{A}\boldsymbol{c} = \boldsymbol{x}.$$

uniquely recovers $\boldsymbol{c}_0$.

RIP in practice

The RIP condition is actually quite strong, and significantly constrains the type of permissible measurement matrices.

To mitigate pathological configurations that violate the RIP, randomness is typically employed.

For example, if $\boldsymbol{c} \in \mathbb{R}^{2N+1}$, then let $\boldsymbol{F}$ denote the $(2N+1) \times (2N+1)$ Fourier measurement matrix.

Let $\boldsymbol{A}$ be formed by randomly selecting $M \ll 2N+1$ rows of $\boldsymbol{F}$ (with renormalization of columns).

If $M \geqslant CS(\log N)^6$, then with high probability $\boldsymbol{A}$ satisfies an RIP condition.

RIP in practice

The RIP condition is actually quite strong, and significantly constrains the type of permissible measurement matrices.

To mitigate pathological configurations that violate the RIP, randomness is typically employed.

For example, if $\boldsymbol{c} \in \mathbb{R}^{2N+1}$, then let $\boldsymbol{F}$ denote the $(2N+1) \times (2N+1)$ Fourier measurement matrix.

Let $\boldsymbol{A}$ be formed by randomly selecting $M \ll 2N+1$ rows of $\boldsymbol{F}$ (with renormalization of columns).

If $M \geqslant CS(\log N)^6$, then with high probability $\boldsymbol{A}$ satisfies an RIP condition.

S-sparse

Compressed sensing in practice

By putting all this together, one can investigate the efficacy of compressed sensing methods.

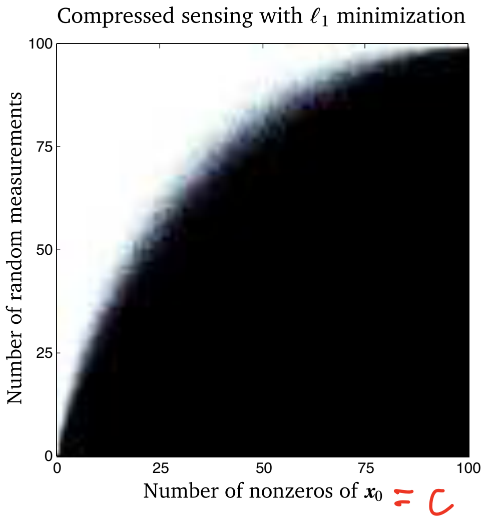


FIGURE 1.1: **Empirical phase transition in compressed sensing.** The colormap indicates the empirical probability that the ℓ_1 minimization problem (1.1) successfully recovers a sparse vector $x_0 \in \mathbb{R}^{100}$ from the vector $z_0 = Ax_0$ of random linear measurements, where A is a standard normal matrix. The probability of success increases with brightness from certain failure (black) to certain success (white).

Image: Living on the edge: A geometric theory of phase transitions in convex optimization, Amelunxen et al, 2013

An interesting observation: there are fairly clear phase transitions delineating the region where recovery happens with probability 1, and where it happens with probability 0.

More recent compressed sensing

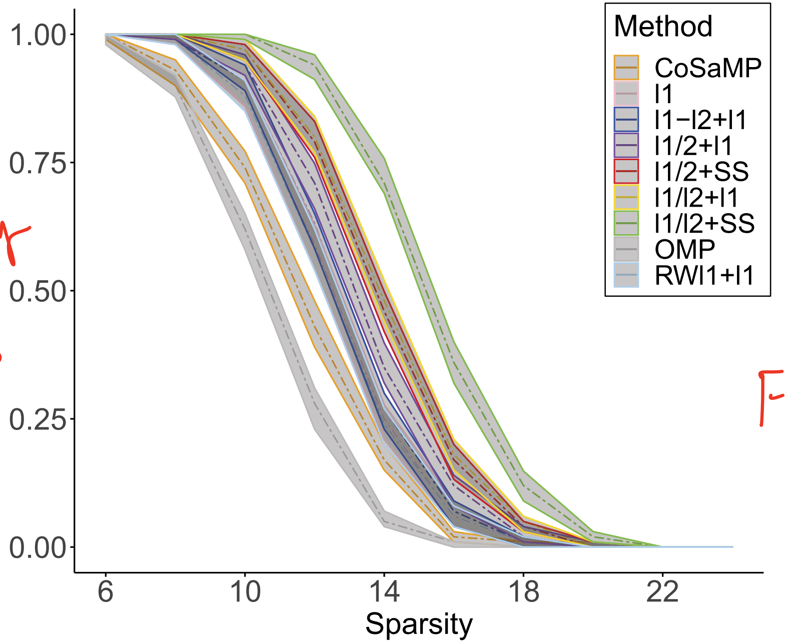
Recent methods in compressed sensing attempt to solve more difficult problems,

$$\min \|\boldsymbol{c}\|_* \ \text{ subject to } \ \boldsymbol{A}\boldsymbol{c} = \boldsymbol{x},$$

where $\|\cdot\|_*$ is a "sparsity-promoting" function.

Such problems are non-convex, but can produce better results.

Probability of success



Fix M

Image: Analysis of the ratio of ℓ^1 and ℓ^2 norms in compressed sensing, Xu et al, 2021

Practical compressed sensing

Many methods are robust to noise, solving, e.g.,

$$\min \|\boldsymbol{c}\|_1 \text{ subject to } \left\|\boldsymbol{A}\boldsymbol{c} - \boldsymbol{x}\right\|_2 \leqslant \epsilon.$$

There are theoretical guarantees ensuring accuracy up to ϵ.

Most realistic problems are "approximately sparse" or *compressible* and not exactly sparse.

Compressed sensing theory extends to ensuring accurate decoding in these cases.

Practical compressed sensing

Many methods are robust to noise, solving, e.g.,

$$\min \|\boldsymbol{c}\|_1 \text{ subject to } \left\|\boldsymbol{A}\boldsymbol{c} - \boldsymbol{x}\right\|_2 \leqslant \epsilon.$$

There are theoretical guarantees ensuring accuracy up to ϵ.

Most realistic problems are "approximately sparse" or *compressible* and not exactly sparse.

Compressed sensing theory extends to ensuring accurate decoding in these cases.

Matrix completion

A problem related to compressed sensing and sparse recovery: matrix completion.

Let $\boldsymbol{A}$ an unknown $m \times n$ matrix. We have access to a small number of entries:

$$\boldsymbol{A}_S, \qquad S \subset [m] \times [n],$$

and our goal is reconstruct $\boldsymbol{A}$ as well as possible.

Again, we should not expect this is possible in general without some assumptions on $\boldsymbol{A}$.

Matrix completion examples

The matrix completion problem is inspired by several real-world examples:

- *Collaborative filtering* – inference about individual preferences from observed group preference.
  This is the "Netflix problem": how much will someone like a new movie? User preferences are frequently determined by a small number of considerations, suggesting low-rank structure.
- Social networks: Abstract "distances" between agents can be measured sparsely. Can we fill in missing data to identify cliques, social patterns, emergent behavior, etc?
- Remote sensing: A full correlation matrix for incoming EM signals cannot be measured, but sensors located at certain locations give partial information.

Matrix completion examples

The matrix completion problem is inspired by several real-world examples:

- *Collaborative filtering* – inference about individual preferences from observed group preference.
  This is the "Netflix problem": how much will someone like a new movie? User preferences are frequently determined by a small number of considerations, suggesting low-rank structure.
- Social networks: Abstract "distances" between agents can be measured sparsely. Can we fill in missing data to identify cliques, social patterns, emergent behavior, etc?
- Remote sensing: A full correlation matrix for incoming EM signals cannot be measured, but sensors located at certain locations give partial information.

Matrix completion examples

The matrix completion problem is inspired by several real-world examples:

- *Collaborative filtering* – inference about individual preferences from observed group preference.
  This is the "Netflix problem": how much will someone like a new movie? User preferences are frequently determined by a small number of considerations, suggesting low-rank structure.
- Social networks: Abstract "distances" between agents can be measured sparsely. Can we fill in missing data to identify cliques, social patterns, emergent behavior, etc?
- Remote sensing: A full correlation matrix for incoming EM signals cannot be measured, but sensors located at certain locations give partial information.

Low-rank matrix completion

If we can only observe a few entries, it seems plausible that we can exactly recover low-rank matrices.

Like in the compressed sensing case regarding sparsity, this is not quite true without additional properties.

Given $\boldsymbol{A}_S$, then as a first step we might consider the optimization,

$$\min \operatorname{rank}(\boldsymbol{B}) \quad \text{subject to} \quad \boldsymbol{B}_S = \boldsymbol{A}_S.$$

We don't have good algorithms for this problem. (It's NP hard.)

So like before, let's relax the problem.

Low-rank matrix completion

If we can only observe a few entries, it seems plausible that we can exactly recover low-rank matrices.

Like in the compressed sensing case regarding sparsity, this is not quite true without additional properties.

Given $\boldsymbol{A}_S$, then as a first step we might consider the optimization,

$$\min \operatorname{rank}(\boldsymbol{B}) \quad \text{subject to} \quad \boldsymbol{B}_S = \boldsymbol{A}_S.$$

We don't have good algorithms for this problem. (It's NP hard.)

So like before, let's relax the problem.

Low-rank matrix completion

If we can only observe a few entries, it seems plausible that we can exactly recover low-rank matrices.

Like in the compressed sensing case regarding sparsity, this is not quite true without additional properties.

Given $\boldsymbol{A}_S$, then as a first step we might consider the optimization,

$$\min \operatorname{rank}(\boldsymbol{B}) \quad \text{subject to} \quad \boldsymbol{B}_S = \boldsymbol{A}_S.$$

We don't have good algorithms for this problem. (It's NP hard.)

So like before, let's relax the problem.

Low-rank matrix completion, II

A closest convex relaxation to the low-rank constraint is nuclear norm minimization,

$$\min \|\boldsymbol{B}\|_{NN} \quad \text{subject to} \quad \boldsymbol{B}_S = \boldsymbol{A}_S,$$

where

$$\|\boldsymbol{B}\|_{NN} = \mathrm{Tr}(\sqrt{\boldsymbol{B}^* \boldsymbol{B}}) = \sum_{j=1}^{r} \sigma_r(\boldsymbol{B}),$$

is the nuclear norm of a matrix.

Low-rank matrix completion, II

$$\min \|\boldsymbol{B}\|_{NN} \quad \text{subject to} \quad \boldsymbol{B}_S = \boldsymbol{A}_S,$$

Like in compressed sensing, exact recovery is possible with an optimal number of samples, subject to some additional assumptions.

Theorem

Define $N := \max\{n, m\}$, and let $\boldsymbol{A}$ have fixed rank r that is "small". Assuming the left- and right-singular vectors of $\boldsymbol{A}$ are not too "peaked", and if,

$$|S| \gtrsim C N \log^2 N,$$

then sampling these $|S|$ samples uniformly at random from $\boldsymbol{A}$ ensures that the nuclear norm minimization exactly recovers $\boldsymbol{A}$ exactly with high probability.

References I

- Dennis Amelunxen, Martin Lotz, Michael B. McCoy, and Joel A. Tropp, *Living on the edge: A geometric theory of phase transitions in convex optimization*, arXiv e-print 1303.6672, March 2013.

- Harbir Antil and Dmitriy Leykekhman, *A Brief Introduction to PDE-Constrained Optimization*, Frontiers in PDE-Constrained Optimization (Harbir Antil, Drew P. Kouri, Martin-D. Lacasse, and Denis Ridzal, eds.), The IMA Volumes in Mathematics and its Applications, Springer, 2018, pp. 3–40.

- E. J. Candes and Y. Plan, *Matrix Completion With Noise*, Proceedings of the IEEE **98** (2010), no. 6, 925–936.

- Emmanuel J. Candes and Terence Tao, *The Power of Convex Relaxation: Near-Optimal Matrix Completion*, IEEE Transactions on Information Theory **56** (2010), no. 5, 2053–2080, Conference Name: IEEE Transactions on Information Theory.

- Emmanuel J. Candès, Justin K. Romberg, and Terence Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Communications on Pure and Applied Mathematics **59** (2006), no. 8, 1207–1223 (en).

References II

- Albert Cohen, Wolfgang Dahmen, and Ronald DeVore, *Compressed sensing and best k-term approximation*, Journal of the American Mathematical Society **22** (2009), no. 1, 211–231.
- D.L. Donoho, *Compressed sensing*, IEEE Transactions on Information Theory **52** (2006), no. 4, 1289–1306.
- Tyrone Rees, H. Sue Dollar, and Andrew J. Wathen, *Optimal Solvers for PDE-Constrained Optimization*, SIAM Journal on Scientific Computing **32** (2010), no. 1, 271–298.
- Tyrone Rees and Martin Stoll, *Block-triangular preconditioners for PDE-constrained optimization*, Numerical Linear Algebra with Applications **17** (2010), no. 6, 977–996 (en).
- Yiming Xu, Akil Narayan, Hoang Tran, and Clayton G. Webster, *Analysis of the ratio of ℓ^1 and ℓ^2 norms in compressed sensing*, Applied and Computational Harmonic Analysis **55** (2021), 486–511, arXiv: 2004.05873.

References III

Matthew J. Zahr and Charbel Farhat, *Progressive construction of a parametric reduced-order model for PDE-constrained optimization*, International Journal for Numerical Methods in Engineering **102** (2015), no. 5, 1111–1135.