Large-scale Optimization Algorithms for Missing Data Completion and Inverse Problems

Curt Da Silva

PhD Defence - Aug. 21, 2017
Inverse problems

Estimate the unknown parameters of a physical system via indirect measurements

- seismology - estimate sound speed of the earth

- medical imaging - infer conductivity of tissue via surface measurements
Inverse problems

Given a model described by parameters $m$, find the parameters $m^\star$ that minimize the misfit between your observed and predicted data
3D seismic experiments

Seismic Source

Receiver Array

Subsurface
Inverse problems

Measured data
- multidimensional (e.g., 5D for seismic problems)
- expensive to acquire fully (budget, environmental, time constraints)
- fully sampled data required for parameter inversion
Compressed sensing / Matrix completion

Acquire a sub-Nyquist number of *randomized* samples

Use *signal structure* (sparsity, low-rank) to recover the signal via

an associated *optimization* problem

Tensor completion

Low rank tensor completion requires a tractable notion of rank
- There are a number of nonequivalent extensions of matrix rank to tensors
- no unique extension of the SVD to multiple dimensions

Optimization in the Hierarchical Tucker format - Chapter 2
- efficient tensor format with low number of parameters
- parametrizes a low rank manifold -> suitable for optimization
Convex composite optimization

Problems of the form

$$\min_{x} h(c(x))$$

$h(z)$ - convex (typically nonsmooth) function

c$(x)$ - smooth mapping
Convex composite optimization

The overall problem is non-convex in general

Non-smooth outer function
  • subgradient methods converge slowly

Chapter 4 - We develop a level set method for efficiently solving this class of problems
Software design for inverse problems

Academic software
- Oriented towards mathematical rigor, less so performance
- Often written for a single paper, no emphasis on extensibility

Industrial software
- Problem sizes are so large -> performance at all costs
- Difficult to implement new algorithms, slow uptake of new technologies

We will bridge these gaps in Chapter 5
Chapter 2
Low-rank tensor completion
Tensor completion

We aim to complete a multidimensional tensor $X \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_d}$ given a subset of its entries on an index set

$$\Omega \subset \{1, \ldots, n_1\} \times \cdots \times \{1, \ldots, n_d\}$$

Our measured data is $b = AX$, where

$$AX = \begin{cases} 
X_{i_1, \ldots, i_d} & \text{if } (i_1, \ldots, i_d) \in \Omega \\
0 & \text{otherwise}
\end{cases}$$
Hierarchical Tucker format

\[ X - n_1 \times n_2 \times n_3 \times n_4 \text{ tensor} \]

\[ X^{(1,2)} = U_{12} B_{1234} U_{34}^T \]

“SVD”-like decomposition
Hierarchical Tucker format

\[ X \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times n_4} \text{ tensor} \]

\[ U_{12} \quad \rightarrow \quad n_1 \]

\[ U_{12} \quad n_2 \quad k_{12} \]

\[ n_1 n_2 \]

\[ k_{12} \]
Hierarchical Tucker format

\[ X = n_1 \times n_2 \times n_3 \times n_4 \] tensor
Hierarchical Tucker format

Intermediate matrices don’t need to be stored

$U_t, B_t$ - small parameter matrices/tensors
  - recursive definition specifies the tensor completely

Separating groups of dimensions from each other
  - dimension tree
Example

\[ B_{12345} \{1, 2, 3, 4, 5\} = t_r \]

\[ U_{123} \quad B_{123} \quad \{1, 2, 3\} \]

\[ U_{23} \quad B_{23} \quad \{2, 3\} = t \]

\[ U_{4} \quad \{4\} \]

\[ U_{5} \quad \{5\} \]

\[ \sqrt{1} \quad \{1\} \]

\[ \sqrt{2} \quad \{2\} = t_1 \]

\[ \sqrt{3} \quad \{3\} = t_2 \]

\[ \sqrt{4} \quad \{4\} \]

\[ \sqrt{5} \quad \{5\} \]
Hierarchical Tucker format

Storage $\leq dN K + (d - 2) K^3 + K^2$

Compare to $N^d$ storage for the full tensor

Effectively breaking the curse of dimensionality when $K \ll N \quad d \geq 4$
Differential geometry

[1] HT tensors parametrize a submanifold of full tensor space $\mathbb{C}^{n_1 \times \cdots \times n_d}$

- Smooth nonlinear nonconvex space
- HT parameters are redundant via a group action (induces a quotient manifold)

In [2], we construct a Riemannian metric on this manifold that respects the underlying quotient topology

Optimization

Given data $b$ with missing sources and/or receivers, subsampling operator $A$, full tensor expansion operator

\[
\phi : (U_t, B_t) \mapsto \mathbb{C}^{n_1 \times \cdots \times n_d}
\]

solve

\[
\min_{x=(U_t, B_t)} \frac{1}{2} \| A\phi(x) - b \|^2_2
\]
Optimization program

Parameter space

$C^D$

$x = (U_t, B_t)$

$x_{best}$

$\mathbb{C}^{n_1 \times \ldots \times n_d}$

Full-tensor space

$\phi(x)$

$\phi(x_{best})$

$\phi(x)$
Optimization program

\[ \mathbb{C}^{n_1 \times \ldots \times n_d} \]

\[ A\phi(x) \]

\[ b \]

\[ \phi(x) \]

\[ \phi(x_{\text{best}}) \]

\[ -\nabla f \]
Numerical Example
Synthetic BG Compass data

Synthetic data from the BG Compass Model
- 68 x 68 sources with 401 x 401 receivers, data at 4.68Hz

Receivers subsampled to 201 x 201

Recovered with Gauss-Newton
4.68 Hz - 75% missing receivers

Fixed source coordinates, varying receiver coordinates

True data

Subsampled data
4.68 Hz - 75% missing receivers

Fixed source coordinates, varying receiver coordinates

True data

Recovered data - SNR 20 dB
4.68 Hz - 75% missing receivers

Fixed source coordinates, varying receiver coordinates

True data

Difference
Chapter 4
A level set, variable projection approach for composite convex optimization
Convex composite optimization

We aim to solve problems of the form

$$\min_x h(c(x))$$

where

- $h(z)$ is convex, non-smooth
- $c(x)$ is a smooth mapping
Convex composite optimization

Here we assume that $h(z)$ has an easy to compute projection, that is

$$\arg \min_z \frac{1}{2} \| z - \hat{z} \|^2_2$$

such that $h(z) \leq \tau$

is efficient to solve for each $\tau$.
Many applications

\[
\min_x \| Ax - b \|_1
\]

Least Absolute Deviation regression

\[
\min_{X, S} \| X + S - D \|_F + \lambda \| S \|_1 + \gamma \| X \|_*
\]

Robust PCA

\[
\min_x \max_{i=1,\ldots,p} f_i(x) \quad f_i \text{ smooth}
\]

Finite min-max optimization

\[
\min_x f(x) + g(x)
\quad f \text{ smooth}, g \text{ non-smooth, convex}
\]

Additive composite minimization
Level set methods

The issue with the problem

$$\min_x h(c(x))$$

is the non-smooth outer function $h(z)$.
Level set methods

Introduce the variable $z = c(x)$ so the problem becomes

\[
\min_{x,z} h(z)
\]

s.t. $z = c(x)$

Simple, but non-smooth objective

Difficult, but smooth constraints
Level set methods

Consider the problem where we flip the objective and constraints

\[ v(\tau) = \min_{x,z} \frac{1}{2} \| z - c(x) \|^2 \]

\[ \text{s.t. } h(z) \leq \tau \]

This is the value function associated to the previous problem

Approach first introduced with SPGL1 for the basis pursuit problem
Level set methods

The value function is easy + efficient to evaluate
  • smooth objective
  • easy to project on constraints

The first value \( \tau^* \) such that \( v(\tau^*) = 0 \) is the optimal value of the original problem
  • \((x, z)\) that solve this subproblem satisfy \( z = c(x) \), \( x \) is the solution to the original composite problem
Updating $\tau$

In the most general case, the secant method

$$\tau_{k+1} = \tau_k - v(\tau_k) \frac{\tau_k - \tau_{k-1}}{v(\tau_k) - v(\tau_{k-1})}$$

converges superlinearly, only requires evaluations of $v(\tau)$

Value function

Projecting out the $z$–variable and rearranging gives us that

$$v(\tau) = \min_x \frac{1}{2} d_h(x) \leq \tau(c(x))$$

Here $d_C(y) = \inf_{w \in C} \|y - w\|_2$ is the distance function to the convex set $C$.
Convergence analysis

We’ll look at the convergence of first order methods to solve the subproblems

\[
\min_x \frac{1}{2} d_h^2() \leq \tau(c(x))
\]
Convergence analysis - Proposition 4.4

Suppose that $h(z)$ has compact level sets, $c(x)$ is $C^1$ and coercive and is $\beta$—Lipschitz continuous with $\gamma$—Lipschitz cont. gradient.

Define

$$L_\tau := \{ z : h(z) \leq \tau \}$$

$$\alpha := \max_{x \in L_\tau} \| c(x) - P_{L_\tau}(c(x)) \|_2$$

$$\kappa := \max_{x \in L_\tau} \sigma_{\max}(\nabla c(x))$$

$$\lambda := \min_{x \in L_\tau} \sigma_{\min}(\nabla c(x))$$
Convergence analysis - Proposition 4.4

Gradient descent with step size \( \frac{1}{\alpha \gamma + \kappa \beta} \) converges linearly with the estimate

\[
\tilde{g}(x_k) - \min \tilde{g} \leq \left( 1 - \frac{\lambda^2}{\alpha \gamma + \kappa \beta} \right)^k (\tilde{g}(x_0) - \min \tilde{g})
\]
Convergence analysis - Proposition 4.4

Still linear convergence, even though $\frac{1}{2} d_h^2(\cdot) \leq \tau$ is not strongly convex

- follows from work in Chapter 3
Numerical Examples
Applications - Robust tensor PCA/completion

We want to recover a tensor

\[ \mathbf{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d} \]

from subsampled, noisy measurements

\[ b = \mathcal{A}(\mathbf{X}) + n \]

\( \mathcal{A} \) - subsampling operator

\( n \) - noise
Applications - Robust tensor PCA/completion

If \( n \) is impulsive (high amplitude, but spatially sparse) and \( X \) is low-rank, then we can solve

\[
\min_{X \in \mathcal{H}} \| A(X) - b \|_1
\]

\( \mathcal{H} \) - class of low rank tensors
Seismic example

BG Data Set

- 68 x 68 sources on a 150m grid, 201 x 201 receivers on a 50m grid, ocean bottom setup
- 75% receivers decimated randomly
- 5% of remaining receivers corrupted with noise = energy of decimated signal
- Hierarchical Tucker interpolation with previous L1 formulation
Seismic example

We compare to

- L2 misfit - original HT tensor completion
- Huber misfit - smoothed L1

\[ H_\delta(x) = \begin{cases} 
  x^2 & \text{if } |x| \leq \delta \\
  2\delta|x| - \delta^2 & \text{if } |x| \geq \delta 
\end{cases} \]
Robust tensor completion
75% Missing receivers with 5% impulsive noise

True Data

Input Data
Robust tensor completion
75% Missing receivers with 5% impulsive noise

True Data

L2 norm - SNR 8.8 dB
Robust tensor completion
75% Missing receivers with 5% impulsive noise

True Data

L1 norm - SNR 16.8 dB
Robust tensor completion
75% Missing receivers with 5% impulsive noise

True Data

Huber penalty - best parameter - SNR 16.7 dB
### Robust tensor completion

<table>
<thead>
<tr>
<th>Method</th>
<th>Recovery SNR (dB)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\ell^2$</td>
<td>7.68</td>
<td>632</td>
</tr>
<tr>
<td>$\ell^1$</td>
<td>16.2</td>
<td>1072</td>
</tr>
<tr>
<td>Huber - best $\delta$</td>
<td>15.9</td>
<td>1003</td>
</tr>
</tbody>
</table>
Huber performance versus $\delta$

<table>
<thead>
<tr>
<th>$5 \cdot 10^{-6}$</th>
<th>13.4</th>
<th>1578</th>
</tr>
</thead>
<tbody>
<tr>
<td>$5 \cdot 10^{-5}$</td>
<td>15.9</td>
<td>1003</td>
</tr>
<tr>
<td>$5 \cdot 10^{-4}$</td>
<td>8.32</td>
<td>928</td>
</tr>
</tbody>
</table>
More applications in my thesis - Section 4.4

Analysis-based compressed sensing

TV denoising

Audio declipping

One-bit compressed sensing
Chapter 5
A unified 2D/3D large scale software environment for nonlinear inverse problems
Solving the inverse problem

Complicated process
- large 3D models, multidimensional data sets
- computationally intensive
- requires large amount of programmer effort to write fast code
- in industry, often speed is the tradeoff for correctness
Software organization

Software hierarchy manages complexity

- human brains have very limited working memory

- if a particular part of a program only has one function, people using/debugging it only have to think about that one function

- if software is easier to reason about -> it’s easier to work with, easier to test
Software organization

Software hierarchy manages complexity

- we don’t have to sacrifice performance
- performance critical operations implemented in C w/ multithreading
Software organization for inverse problems

User Facing Functions

Distributed Parallel Computation

PDEfunc

Construct System Matrix

Abstract Matrix

Linear solve

Abstract linear solver

H\(q\)

H\(\times q\)

2D

Sparse Matrix

3D

C-based MVP

Objective function construction
Jacobian, Hessian operators

PDE-related quantities
Parallel version

PDE-related quantities
Serial version

System matrix: multiplication/division

Multithreaded
Mat-vec multiply
Benefits of this approach

Modular design

- easy to integrate a new preconditioner, parallelization scheme, PDE discretization, misfit function
- speedups in solving PDEs propagate to whole system

Abstract user-facing interfaces

- suitable for use with black-box optimization methods
3D Helmholtz equation

The Helmholtz equation (with PML)

\[(\frac{\partial_x}{x}\eta(x)\frac{\partial_x}{x} + \frac{\partial_y}{y}\eta(y)\frac{\partial_y}{y} + \frac{\partial_z}{z}\eta(z)\frac{\partial_z}{z} + \omega^2 v^{-2})u = q\]

is difficult to discretize + solve numerically

- minimum number of points per wavelength needed
  - high memory, computational costs

- resulting system is unsymmetric & indefinite, conditioning isn’t great
  - tricky for classical Krylov solvers

- need to use complicated stencils to avoid numerical dispersion
Recursive multigrid Helmholtz preconditioner

[1] uses traditional multigrid components arranged in a recursive fashion to precondition Helmholtz discretized with the standard 7pt stencil

- good performance but very specific to the 7pt stencil
- ill-suited for the compact stencil of [2]

In this chapter, we propose a new recursive multigrid preconditioner that is suitable for the 27pt stencil

Multilevel-GMRES

Smoother

GMRES

GMRES

Coarse solve

GMRES

Preconditioned by

GMRES

GMRES

GMRES

Discretization Spacing

$h$

$2h$

$4h$
Numerical Examples
3D FWI Example

Overthrust model

- 20 km x 20 km x 4.6 km - 50 m spacing, 500m water layer
- 50 x 50 sources, 200m spacing - 2500 shots
- 401 x 401 receivers, 50m spacing
- 3Hz - 6Hz frequency range, 0.25 Hz spacing, single freq. inverted at a time
Computational environment

SENAI Yemoja cluster

- 100 nodes, 128 GB RAM each, 20-core processors

- 400 Parallel Matlab workers (4 per node), Helmholtz MVP uses 5 threads - full core utilization
z=1000m slice

True model

Initial model
z=1000m slice

True model

Stochastic LBFGS
x=17.5km slice

True model

Initial model
$x=17.5\text{km slice}$

**True model**

**Stochastic LBFGS**
Conclusion

In this thesis, I have developed

- manifold optimization methods for large-scale tensor completion
- an algorithm for convex-composite optimization
- a modern software framework for PDE-constrained inverse problems
Publications

C. Da Silva, F. Herrmann “A unified 2D/3D large scale software environment for nonlinear inverse problems”, Submitted, 2017

Y. Zhang, C. Da Silva, R. Kumar, F. Herrmann “Massive 3D seismic data compression and inversion with hierarchical Tucker”, SEG Conference 2017


Z. Fang, C. Da Silva, R. Kuske, F. Herrmann “Uncertainty quantification for inverse problems with a weak wave-equation constraint”, WAVES 2017

C. Da Silva, F. Herrmann “A unified 2D/3D software framework for large scale time-harmonic full waveform inversion”, SEG Conference 2016


C. Da Silva, F. Herrmann “Irregular grid tensor completion”, Workshop on Low-rank Optimization and Applications, 2015

C. Da Silva, F. Herrmann “Low-rank promoting transformations and tensor interpolation – applications to seismic data denoising”, EAGE Conference 2014

C. Da Silva, F. Herrmann “Hierarchical Tucker tensor optimization – applications to tensor completion”, SAMPTA Conference 2013

C. Da Silva, F. Herrmann “Hierarchical Tucker tensor optimization – applications to 4D seismic data interpolation”, EAGE Conference 2013

C. Da Silva, F. Herrmann “Matrix probing and simultaneous sources: a new approach for preconditioning the Hessian”, EAGE Conference 2012
Thank you for your attention