Compression and Reconstruction using Simulated Annealing

Mentors: Michael Lamoureux, Pooya Ronagh
Aritra Dutta, Geonwoo Kim, Meiqin Li
Carlos Ortiz, Mohit Sinha, Cole Stiegler

Math Modeling in Industry XIX

Institute for Mathematics and its Applications

August 14, 2015

Outline

1. Introduction
2. Simulated Annealing
3. Compression Problem
4. Image Compression
5. Sound Compression
6. SPGL1 and Reconstruction
7. Tuning the Algorithm

Introduction

Our Idea

- Our goal is to apply an optimized simulated annealing algorithm to image compression and reconstruction.

Our Idea

- Our goal is to apply an optimized simulated annealing algorithm to image compression and reconstruction.
- This investigation is motivated by the assumption that our algorithm could be implemented in the processor produced by D-Wave Systems, Inc.

Simulated Annealing

Simulated Annealing

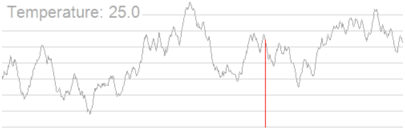
- Simulated annealing is a search method that explores a search space randomly in order to find a desired object.

Simulated Annealing

- Simulated annealing is a search method that explores a search space randomly in order to find a desired object.
- One can think of this technique as analogous to the way molecules in a metal cool down into a crystalline structure.

Simulated Annealing

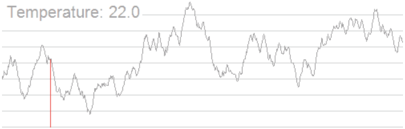
- Simulated annealing is a search method that explores a search space randomly in order to find a desired object.
- One can think of this technique as analogous to the way molecules in a metal cool down into a crystalline structure.
- This algorithm has been particularly successful at providing approximate solution to NP-Hard problems.



Optimizing the objective function

Simulated Annealing

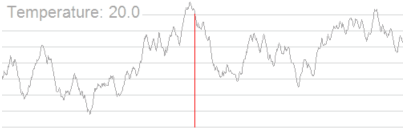
- Simulated annealing is a search method that explores a search space randomly in order to find a desired object.
- One can think of this technique as analogous to the way molecules in a metal cool down into a crystalline structure.
- This algorithm has been particularly successful at providing approximate solution to NP-Hard problems.



Optimizing the objective function

Simulated Annealing

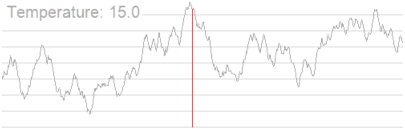
- Simulated annealing is a search method that explores a search space randomly in order to find a desired object.
- One can think of this technique as analogous to the way molecules in a metal cool down into a crystalline structure.
- This algorithm has been particularly successful at providing approximate solution to NP-Hard problems.



Optimizing the objective function

Simulated Annealing

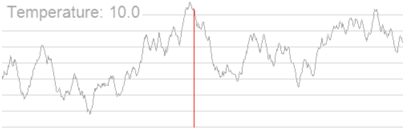
- Simulated annealing is a search method that explores a search space randomly in order to find a desired object.
- One can think of this technique as analogous to the way molecules in a metal cool down into a crystalline structure.
- This algorithm has been particularly successful at providing approximate solution to NP-Hard problems.



Optimizing the objective function

Simulated Annealing

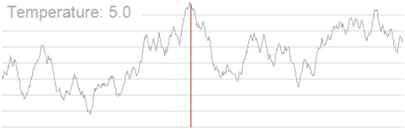
- Simulated annealing is a search method that explores a search space randomly in order to find a desired object.
- One can think of this technique as analogous to the way molecules in a metal cool down into a crystalline structure.
- This algorithm has been particularly successful at providing approximate solution to NP-Hard problems.



Optimizing the objective function

Simulated Annealing

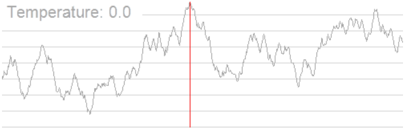
- Simulated annealing is a search method that explores a search space randomly in order to find a desired object.
- One can think of this technique as analogous to the way molecules in a metal cool down into a crystalline structure.
- This algorithm has been particularly successful at providing approximate solution to NP-Hard problems.



Optimizing the objective function

Simulated Annealing

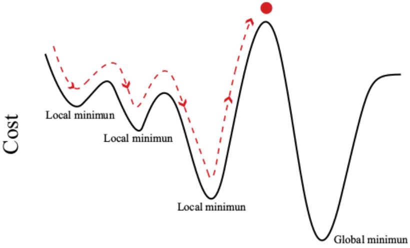
- Simulated annealing is a search method that explores a search space randomly in order to find a desired object.
- One can think of this technique as analogous to the way molecules in a metal cool down into a crystalline structure.
- This algorithm has been particularly successful at providing approximate solution to NP-Hard problems.



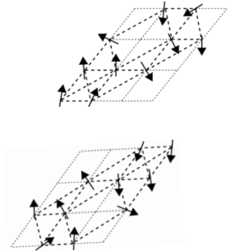
Optimizing the objective function

Physical Annealing

- Thermal annealing
  - Uses thermal fluctuations to reach the ground state.



- Quantum Annealing
  - Uses quantum fluctuations (e.g tunneling) to reach the ground state.



Simulated Annealing Algorithm

We use an implementation of simulated annealing optimized to solve the Ising spin problem.

Definition

Let $G = (V, E)$ be a graph on n vertices with vertex set V, edge set E, and let $s_i \in \{-1, 1\}$ for $i \in V$. For a given configuration $s \in \{-1, 1\}^n$, the energy of this system is given by,

$$H(s) = \sum_{k \in V} h_k s_k + \sum_{(i,j) \in E} J_{ij} s_i s_j = \langle h, s \rangle + \langle s, Js \rangle$$

where $h = (h_1, \ldots, h_n) \in \mathbb{R}^n$ and $J = (J_{ij}) \in M_n(\mathbb{R}^n)$.

Simulated Annealing Algorithm

We use an implementation of simulated annealing optimized to solve the Ising spin problem.

Definition

Let $G = (V, E)$ be a graph on n vertices with vertex set V, edge set E, and let $s_i \in \{-1, 1\}$ for $i \in V$. For a given configuration $s \in \{-1, 1\}^n$, the energy of this system is given by,

$$H(s) = \sum_{k \in V} h_k s_k + \sum_{(i,j) \in E} J_{ij} s_i s_j = \langle h, s \rangle + \langle s, Js \rangle$$

where $h = (h_1, \ldots, h_n) \in \mathbb{R}^n$ and $J = (J_{ij}) \in M_n(\mathbb{R}^n)$.

The objective here is to minimize $H(s)$.

Compression Problem

Compression Problem

Consider the following problem,

$$
\begin{aligned}
\underset{x\in\{0,1\}^n}{\text{minimize}} \quad & ||x||_0 \\
\text{subject to} \quad & ||Ax-b||_2^2 \le \sigma
\end{aligned}
\tag{1}
$$

Compression Problem

Consider the following problem,

$$\begin{aligned} \underset{x \in \{0,1\}^n}{\text{minimize}} \quad & ||x||_0 \\ \text{subject to} \quad & ||Ax - b||_2^2 \leq \sigma \end{aligned} \tag{1}$$

- This problem can be interpreted as an image compression problem where your goal is to find a sparse binary vector x such that its image under A is close to the original image b.

Relaxation

It turns out that (1) is equivalent to the following unconstrained optimization problem,

$$\min_{x \in \{0,1\}^n} \quad ||x||_0 + \lambda ||Ax - b||_2^2 \tag{2}$$

for $\lambda > 0$.

Reformulation

Let $e := (1, \ldots, 1) \in \{0, 1\}^n$ and consider the following reformulation of (2),

$$\begin{aligned} ||x||_0 + \lambda ||Ax - b||_2^2 &= \langle e, x \rangle + \langle Ax - b, Ax - b \rangle \\ &= \langle e, x \rangle + \langle Ax, Ax \rangle - 2\langle Ax, b \rangle + \langle b, b \rangle \\ &= \langle e, x \rangle + \langle A^* Ax, x \rangle - 2\langle Ax, b \rangle + \langle b, b \rangle \\ &= \langle e - 2A^* b, x \rangle + \langle A^* Ax, x \rangle + \langle b, b \rangle \end{aligned}$$

Change of variables

In order to use the algorithm we need to formulate our problem as an Ising spin problem. Consider the following change of variables,

$$x = \frac{1}{2}(s + e) \implies x \in \{0, 1\}^n$$

where $e := (1, \ldots, 1) \in \{0, 1\}^n$. Now equation (2) becomes,

$$\min_{s \in \{-1,1\}^n} \langle e - 2A^* b, \frac{1}{2}(s + e)\rangle + \langle A^* A \frac{1}{2}(s + e), \frac{1}{2}(s + e)\rangle + \langle b, b\rangle$$

Notice that this reformulation is an Ising spin problem.

Image Compression

Quantification of Error

- We use the normalized mean square error to quantify our results

$$\text{NMSE} = \frac{||Ax - b||_2}{||b||_2},$$

where $\|\cdot\|_2$ is the ℓ_2-norm of the vectors.

Quantification of Error

- We use the normalized mean square error to quantify our results

$$\text{NMSE} = \frac{||Ax - b||_2}{||b||_2},$$

where $\|\cdot\|_2$ is the ℓ_2-norm of the vectors.

- We also measured the sparsity (non-black percentage) of the x vector by taking the sum of the entries and dividing by the total number of entries.

$$\text{Sparsity} = \frac{\sum_{i,j} x_{i,j}}{\text{length of } x}$$

Results

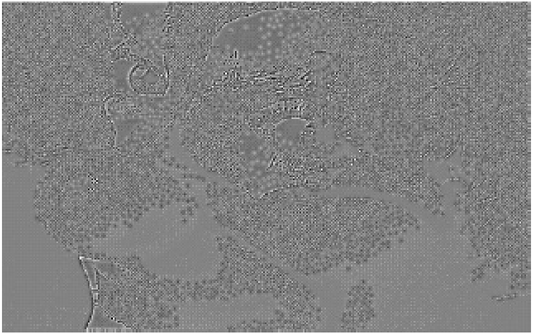
- Recall the objective function is

$$\min_x ||x||_0 + \lambda||Ax - b||_2^2$$

- Overall, our best results were with a high penalty (100) λ on the least squares term and using a 5x5 gaussian kernel with standard deviation 0.9 for our convolution.
- This produced a NMSE$= 0.1573$.

(a) Blurred image NMSE=0.1573

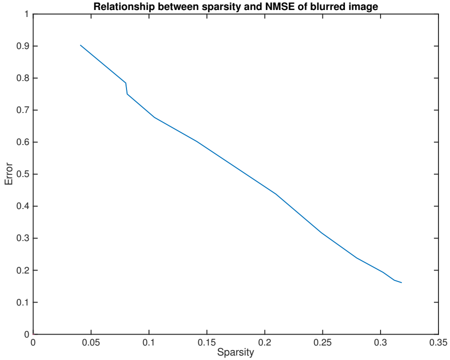
(b) Original image



(c) Error Image

Sparsity of binary image

Further sparsity in the binary solution x would decrease memory usage, but increases error much more quickly than would be worthwhile.



The maximum sparsity is roughly 0.33 since the compression algorithm produced no image with higher sparsity percentage.

Sound Compression

1D signals - Sound

We try compression on a sinusoidal ramp (left), smoothly varying from 100 to 400 Hz. A rough reconstruction is obtained (right).

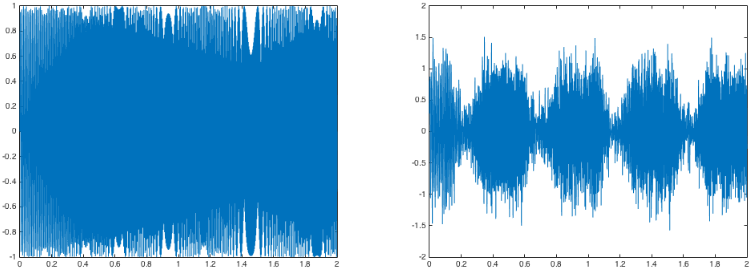
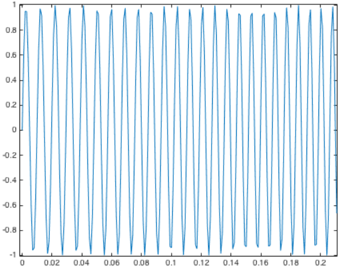
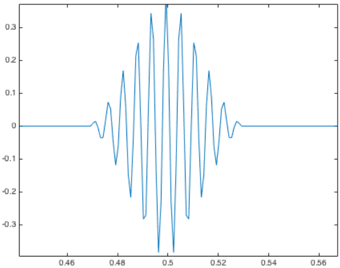


Figure: Original and reconstructed signal.

Sinusoidal atoms

Zooming in on the original sound, we see sinusoids of various widths (left). We reconstruct using atoms that are localized sinusoids (right), of five discrete frequencies from 100 to 400 Hz.





1D compression details

- 2,000 samples.
- Used 5 atoms to reconstruct, representing 5 frequencies.
- From 100 to 400Hz, good reconstruction near those freqs.
- Had to include the negative atoms as well.
- Sparse reconstruction matrix is 2,000 x 20,000.
- Compression ratio is only 2 to 1.
- Error measure is 30%.
- Not so good!

SPGL1 and Reconstruction

Sparse Reconstruction Problem

Consider the following problem,

$$\begin{aligned} \underset{x \in \{0,1\}^n}{\text{minimize}} \quad & ||x||_0 \\ \text{subject to} \quad & ||Ax - b||_2^2 \leq \sigma \end{aligned}$$

Sparse Reconstruction Problem

Consider the following problem,

$$\begin{aligned} \underset{x \in \{0,1\}^n}{\text{minimize}} \quad & ||x||_0 \\ \text{subject to} \quad & ||Ax - b||_2^2 \leq \sigma \end{aligned}$$

- Alternatively, you can think of this problem as a image reconstruction problem where your goal is to find a sparse binary vector x that was corrupted by A, but we only have access to the corrupted image b.

Classic Sparse Optimization Problems

Solving the system $Ax = b$ where $A \in \mathbb{R}^{m \times n}$ such that $m << n$, suffers from ill-posedness.

- $\min_x \|x\|_1$ subject to $Ax = b$. (BP)

Classic Sparse Optimization Problems

Solving the system $Ax = b$ where $A \in \mathbb{R}^{m \times n}$ such that $m << n$, suffers from ill-posedness.

- $\min_x \|x\|_1$ subject to $Ax = b$. (BP)
- $\min_x \|x\|_1$ subject to $\|Ax - b\|_2 \leq \sigma$. (BP_σ)

Classic Sparse Optimization Problems

Solving the system $Ax = b$ where $A \in \mathbb{R}^{m \times n}$ such that $m << n$, suffers from ill-posedness.

- $\min_x \|x\|_1$ subject to $Ax = b$. (BP)
- $\min_x \|x\|_1$ subject to $\|Ax - b\|_2 \leq \sigma$. (BP_σ)
- $\min_x \|Ax - b\|_2$ subject to $\|x\|_1 \leq \tau$. (LS_τ)

Classic Sparse Optimization Problems

Solving the system $Ax = b$ where $A \in \mathbb{R}^{m \times n}$ such that $m << n$, suffers from ill-posedness.

- $\min_x \|x\|_1$ subject to $Ax = b$. (BP)
- $\min_x \|x\|_1$ subject to $\|Ax - b\|_2 \leq \sigma$. (BP_σ)
- $\min_x \|Ax - b\|_2$ subject to $\|x\|_1 \leq \tau$. (LS_τ)

Related Work

- Homotopy approches
- BPDN as a cone program
- BPDN as a linear program
- Projected gradient

Let x_τ denote the optimal solution of (LS_τ). Consider the single-parameter function

$$\phi(\tau) = \|r_\tau\|_2, \quad \text{with } r_\tau := b - Ax_\tau;$$

gives an optimal value (LS_τ) for each $\tau > 0$.

Let x_τ denote the optimal solution of (LS_τ). Consider the single-parameter function

$$\phi(\tau) = \|r_\tau\|_2, \quad \text{with } r_\tau := b - Ax_\tau;$$

gives an optimal value (LS_τ) for each $\tau > 0$. So the approach lies in finding a root of

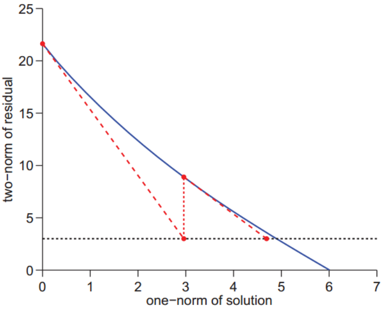
$$\phi(\tau) = \sigma.$$

Let x_τ denote the optimal solution of (LS_τ). Consider the single-parameter function

$$\phi(\tau) = \|r_\tau\|_2, \quad \text{with } r_\tau := b - Ax_\tau;$$

gives an optimal value (LS_τ) for each $\tau > 0$. So the approach lies in finding a root of

$$\phi(\tau) = \sigma.$$



Framework of SPGL1

In order to derive the dual of (LS_τ), we discuss an equivalent problem

$$min_{r,x}\|r\|_2 \text{ subject to } Ax + r = b,\ \|x\|_1 \leq \tau.$$

The dual of this problem is

$$\max_{y,\lambda} b^T y - \tau\lambda \ \text{ subject to } \|y\|_2 \leq 1, \|A^T y\|_\infty \leq \lambda$$

Original Image

Original Image

Blurred Image

SA+Ridge Reconstruction Image

SPGL Reconstruction Image

Tuning the Algorithm

Optimizing the Algorithm: Reducing Rows

$BAx = Bb$

- The original linear system is $Ax = b$, where $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and $m << n$.

Optimizing the Algorithm: Reducing Rows

$BAx = Bb$

- The original linear system is $Ax = b$, where $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and $m << n$.
- Find $B \in \mathbb{R}^{k \times m}$, with $k < m$, such that for a predefined $\epsilon > 0$,

$$\|\hat{x} - x\|_2 < \epsilon,$$

where $\hat{A}\hat{x} = \hat{b}$ and $\hat{A} = BA$, $\hat{b} = Bb$.

Optimizing the Algorithm: Reducing Rows

How to choose B

- Form $\tilde{b}$ such that
$$\tilde{b} = (b^{(i)})^m \in \mathbb{R}^m$$
where
$$|b^{(1)}| \geq |b^{(2)}| \geq \cdots \geq |b^{(m)}| \geq 0.$$

Optimizing the Algorithm: Reducing Rows

How to choose B

- Form $\tilde{b}$ such that
$$\tilde{b} = (b^{(i)})^m \in \mathbb{R}^m$$
where
$$|b^{(1)}| \geq |b^{(2)}| \geq \cdots \geq |b^{(m)}| \geq 0.$$

- For a tolerance $0 < \alpha \leq 1$, choose $\tilde{b}(1 : k)$ if $\frac{\|\tilde{b}(1:k)\|_2}{\|b\|_2} > \alpha$.

Optimizing the Algorithm: Reducing Rows

How to choose B

- Form $\tilde{b}$ such that
$$\tilde{b} = (b^{(i)})^m \in \mathbb{R}^m$$
where
$$|b^{(1)}| \geq |b^{(2)}| \geq \cdots \geq |b^{(m)}| \geq 0.$$

- For a tolerance $0 < \alpha \leq 1$, choose $\tilde{b}(1:k)$ if $\frac{\|\tilde{b}(1:k)\|_2}{\|b\|_2} > \alpha$.
- Suppose $S = \{i : \frac{\|\tilde{b}(1:k)\|_2}{\|b\|_2} > \alpha\}$ form $\hat{A}$, such that $\hat{A} \in \mathbb{R}^{|S| \times n}$.

Optimizing the Algorithm: Reducing Rows

How to choose B

- Form $\tilde{b}$ such that
$$\tilde{b} = (b^{(i)})^m \in \mathbb{R}^m$$
where
$$|b^{(1)}| \geq |b^{(2)}| \geq \cdots \geq |b^{(m)}| \geq 0.$$
- For a tolerance $0 < \alpha \leq 1$, choose $\tilde{b}(1:k)$ if $\frac{\|\tilde{b}(1:k)\|_2}{\|b\|_2} > \alpha$.
- Suppose $S = \{i : \frac{\|\tilde{b}(1:k)\|_2}{\|b\|_2} > \alpha\}$ form $\hat{A}$, such that $\hat{A} \in \mathbb{R}^{|S| \times n}$.

We choose
$$B = \left(e_{i_1} \, e_{i_2} \cdots e_{i_k}\right)^T$$
Where e_{i_j} is a $1 \times m$ vector with 1 on the i_jth position and 0 elsewhere,

Recovery

We use SA, we get the recovered image as:

No. of rows: 64000->10843

Optimizing the Algorithm: Memory Efficiency

- We divide x in to subblocks x_j, such that $1 \leq j \leq k << m$. For each block x_j, solve $A_j x_j = b_j$.

Optimizing the Algorithm: Memory Efficiency

- We divide x in to subblocks x_j, such that $1 \leq j \leq k << m$. For each block x_j, solve $A_j x_j = b_j$.
- Form $x_{Recovered} = (x_j : 1 \leq j \leq k << m)$.

Optimizing the Algorithm: Memory Efficiency

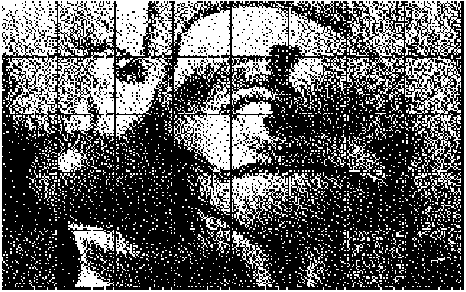
- We divide x in to subblocks x_j, such that $1 \leq j \leq k << m$. For each block x_j, solve $A_j x_j = b_j$.
- Form $x_{Recovered} = (x_j : 1 \leq j \leq k << m)$.
- We can use the same process to reduce the size of each block A_j. Hence, for each block, we solve

$$\hat{A}_j x_j = \hat{b}_j, \ \text{ for } 1 \leq j \leq k << m,$$

where $\hat{A}_j = B_j A_j$ and $B_j b_j = \hat{b}_j, \forall j$.

Optimizing the Algorithm: Memory Efficiency

recovery--by reducing the NO. of rows for each block



Overlapping the blocks

Using the overlapping block technique the recovered image is

Future Work

- Evaluate our algorithm using Simulated Quantum Annealing (no implementation in the literature yet).
- Implement these ideas in a D-Wave system.

Future Work

- Evaluate our algorithm using Simulated Quantum Annealing (no implementation in the literature yet).
- Implement these ideas in a D-Wave system.

Special thanks to Michael, Pooya, the IMA, and PIMS.