Numerical Algorithms for Solving Nonsmooth Optimization Problems and Applications in Image Reconstructions

Karina Rodriguez
(Joint work with Lewis Hicks, Mike Wells, and Nam Nguyen)

Department of Mathematics and Statistics
Portland State University
Image Reconstruction

3-D Reconstruction of the Heart

(2018, December 12). 3D reconstruction of the heart by advanced image processing algorithms. RSIP. Retrieved from https://www.rsipvision.com/3d-reconstruction-of-the-heart/

3-D Reconstruction of Skull

Related Work and Our Approach


patching

<table>
<thead>
<tr>
<th>80</th>
<th>88</th>
<th>97</th>
<th>102</th>
<th>102</th>
<th>99</th>
<th>96</th>
<th>90</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>97</td>
<td>105</td>
<td>104</td>
<td>99</td>
<td>96</td>
<td>95</td>
<td>89</td>
</tr>
<tr>
<td>84</td>
<td>101</td>
<td>108</td>
<td>103</td>
<td>95</td>
<td>93</td>
<td>93</td>
<td>91</td>
</tr>
<tr>
<td>91</td>
<td>101</td>
<td>103</td>
<td>98</td>
<td>92</td>
<td>91</td>
<td>92</td>
<td>95</td>
</tr>
<tr>
<td>99</td>
<td>99</td>
<td>96</td>
<td>91</td>
<td>89</td>
<td>90</td>
<td>91</td>
<td>100</td>
</tr>
<tr>
<td>04</td>
<td>97</td>
<td>90</td>
<td>87</td>
<td>87</td>
<td>89</td>
<td>91</td>
<td>103</td>
</tr>
<tr>
<td>98</td>
<td>90</td>
<td>84</td>
<td>82</td>
<td>84</td>
<td>89</td>
<td>92</td>
<td>99</td>
</tr>
<tr>
<td>93</td>
<td>86</td>
<td>81</td>
<td>80</td>
<td>85</td>
<td>92</td>
<td>96</td>
<td>107</td>
</tr>
<tr>
<td>86</td>
<td>80</td>
<td>76</td>
<td>79</td>
<td>87</td>
<td>97</td>
<td>103</td>
<td>115</td>
</tr>
<tr>
<td>78</td>
<td>75</td>
<td>74</td>
<td>80</td>
<td>90</td>
<td>102</td>
<td>109</td>
<td>116</td>
</tr>
<tr>
<td>73</td>
<td>74</td>
<td>78</td>
<td>85</td>
<td>96</td>
<td>106</td>
<td>112</td>
<td>109</td>
</tr>
<tr>
<td>71</td>
<td>78</td>
<td>86</td>
<td>94</td>
<td>102</td>
<td>108</td>
<td>112</td>
<td>101</td>
</tr>
<tr>
<td>71</td>
<td>83</td>
<td>95</td>
<td>104</td>
<td>108</td>
<td>109</td>
<td>109</td>
<td>98</td>
</tr>
<tr>
<td>72</td>
<td>87</td>
<td>101</td>
<td>110</td>
<td>112</td>
<td>110</td>
<td>107</td>
<td>99</td>
</tr>
<tr>
<td>77</td>
<td>96</td>
<td>107</td>
<td>116</td>
<td>120</td>
<td>113</td>
<td>101</td>
<td>111</td>
</tr>
</tbody>
</table>
Vectorizing Small Image

$\begin{bmatrix}
  x_{1,1} & \cdots & x_{1,n} \\
  \vdots & \ddots & \vdots \\
  x_{n,1} & \cdots & x_{n,n}
\end{bmatrix} \rightarrow 
\begin{bmatrix}
  x_{1,1} \\
  \vdots \\
  x_{n,1} \\
  x_{1,n} \\
  \vdots \\
  x_{n,n}
\end{bmatrix}$

Original

patch

vectorization
Create Blurry Image

\[ Ax + E = b \]

Identity matrix with missing rows

\[
A = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
1 & 2 & 3 & 4
\end{bmatrix}, \quad b = \begin{bmatrix}
1 \\
2 \\
4
\end{bmatrix}
\]

A deletes entries from the vectorized image x.

---

% extract patch size rows:251-282, col:251-282
%X is considered as the image
patch = X(251:282,251:282);
figure
dezo(patch) imshow patch
tezize([300 300]);
title('Original'); add title
sampleRate = .5;
%returns array [1 204] with random numbers 1 thru 64
Omega = randi([1,32*32], 1,floor(32*32/5));
%vector of numbers 1-1024
index = 1:1024;
%assign 0 (makes pixel black) to random (Omega) indices of index vector
index(Omega) = 0;
%gets indices ~0
pos = index(index > 0);
%create a 1024*1024 w/ 1's along diagonal
T = eye(32*32);
%make A matrix to be all zeros, excluding random indices greater than 0
% i.e. indices 3 and 6, a matrix 16*16 the values of row1 col3 & row2 col6
A = T(pos,:);
%create vector from image patch
X3 = patch();
%create image with missing pixels
b = A*X3;
c = reshape(A'*b, size(patch));
Dictionary to Create a Good Image

Properties of a “good image”:

• $Ax = A(Dy) = Ay$ is very close to $b$:
  \[
  \frac{1}{2} \|Ay - b\|^2 \text{ is small}
  \]

• The vector $y$ is sparse:
  \[
  \|y\|_0 \approx \|y\|_1 - \|y\|_2 \text{ is small}
  \]

We seek to minimize

\[
  f(y) = \frac{1}{2} \|Ay - b\|^2 + \lambda(\|y\|_1 - \|y\|_2)
\]

Reconstructed image $x = Dy$: a linear combination of the columns (“atoms”) of $D$

Define $A := AD$. 

Dictionary (D)
A function $f : \mathbb{R}^n \to \mathbb{R}$ is called \textit{convex} if for all $x_1, x_2 \in \mathbb{R}^n$ and for all $t \in [0,1]$

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$$
DC Functions

A Difference of Convex (DC) function, f

- $g(x) = x^4 - 100$
- $h(x) = 10(x-2)^2 + 20$
- $f(x) = g(x) - h(x)$
Nesterov Smoothing

Equation being smoothed:

\[ y = |x| \]
DCA Algorithm

Algorithm 1 The DCA

1: Input: $x_0 \in \mathbb{R}^n$, $N \in \mathbb{N}$
2: for $k = 1, \ldots, N$ do
3:   Find $y_k \in \partial h(x_{k-1})$
4:   Find $x_k \in \partial g^*(y_k)$
5: end for
6: Output: $x_N$.

Function: $f = \frac{1}{2} \|Ax - b\|^2 + \lambda \|x\|_1 - \lambda \|x\|_2$
which has
$g = \frac{1}{2} \|Ax - b\|^2 + \lambda \|x\|_1$
and
$h = \lambda \|x\|_2$. 

```
x=0*A+(b+.5;
f=2*norm(A(x)-b)^2+norm(x,1);
for LP=1:N
  for i=1:3
    proj=max(-1, min(x/mu,1));
    if norm(x)==0
      w=0*x;
    else
      w=x/norm(x);
    end
    arg=x/mu;
    proj=max(-1, min(arg,1));
    y=w+gam*x-v*AT(A(x)-b)+x/mu-proj);
    x=mu/(1+gam*mu)*y;
    ct=ct+1;
    FPLD(opt)=f(x);
  end
  mu=mu*sig;
end
```
DCA with Smoothing

Since $\|x\|_1$ is non-smooth, we use Nesterov’s Smoothing Technique to find a smooth approximation:

$$p_\mu(x) = \frac{1}{2\mu} \|x\|^2 - \frac{\mu}{2} d\left(\frac{x}{\mu}; Q\right)^2$$

where $Q = \{x \in \mathbb{R}^n \mid |x_i| \leq 1\}$. The function

$$f(x) = \frac{1}{2} \|Ax - b\|_2^2 + \lambda (\|x\|_1 - \|x\|_2)$$

is then approximated by

$$f_\mu(x) = \frac{\lambda}{2\mu} \|x\|^2 + \frac{1}{2} \|Ax - b\|^2 - \frac{\lambda \mu}{2} d\left(\mu^{-1} x; Q\right)^2 - \lambda \|x\|.$$
DCA with Smoothing

We set

\[ g(x) = \left( \frac{\lambda + \gamma \mu}{2 \mu} \right) \| x \|^2 \]

and

\[ h(x) = \frac{\gamma}{2} \| x \|^2 - \frac{1}{2} \| Ax - b \|^2 + \frac{\lambda \mu}{2} d(\mu^{-1} x; Q)^2 + \lambda \| x \|. \]

Gamma is chosen large enough so \( \frac{\gamma}{2} \| x \|^2 - \| Ax - b \|^2 \) is convex.

We have

\[ \partial h(x) = \left( \frac{\lambda + \gamma \mu}{\mu} \right) x - A^T (Ax - b) - \lambda \Pi_Q (\mu^{-1} x) + \lambda \partial \| x \| \]

where \( \Pi_Q \) is projection onto \( Q \).
We substitute
\[ \omega(x) = \begin{cases} \frac{x}{\|x\|} & x \neq 0, \\ 0 & x = 0, \end{cases} \]
for $\partial\|x\|$ since it is a subgradient for all $x$.

We use the fact that $x \in \partial g^*(y)$ if and only if $y \in \partial g(x)$. Thus,
\[ y = \left( \frac{\lambda + \gamma \mu}{\mu} \right) x \]
implies
\[ x = \left( \frac{\mu}{\lambda + \gamma \mu} \right) y. \]
The Boosted DCA

**Boosted DCA Algorithm**

**INPUT:** $x_0$, $N \in \mathbb{N}$, 
$\alpha > 0$, $\bar{\lambda} > 0$, $0 < \beta < 1$.

**for** $k = 0, \ldots, N$ **do**

1. Find $z_k \in \partial h(x_k)$.
2. Solve $y_k = \arg\min_{x \in \mathbb{R}^n} \{g(x) - \langle z_k, x \rangle\}$.
3. Set $d_k = y_k - x_k$. If $d_k = 0$, stop, return $x_k$. Else, continue.
4. Set $\lambda_k = \bar{\lambda}$.
   While $\phi(y_k + \lambda_k d_k) > \phi(y_k) - \alpha \lambda_k \|d_k\|^2$, set $\lambda_k = \beta \lambda_k$
5. Set $x_{k+1} = y_k + \lambda_k d_k$.

If $x_{k+1} = x_k$, stop, return $x_k$. Else, set $k = k + 1$ and return to step 1.

**end for**

**OUTPUT:** $x_{N+1}$
minimize $f(x) = x^4 - 2x^2 + 2x - 3$, $x \in \mathbb{R}$
RE and PSNR

\[ RE = 100\% \times \frac{\|M - \hat{M}\|}{\|M\|} \]

\[ PSNR = 20 \log_{10}(\frac{\sqrt{N_1N_2}}{\|M - \hat{M}\|}) \]
Convergence Graph

Boosted DCA with Nesterov Smoothing

DCA with Nesterov Smoothing
Results

Sampled Image

Boosted DCA

DCA
Future Work

DCT Dictionary

Learned Dictionary
Computational Modeling Serving the City

(2016, March 7). Retrieved from https://www.youtube.com/watch?v=qMhGzaPhTKk

Low-quality video
Acknowledgements

Lewis Hicks
Michael Wells
Advisor: Mau Nam Nguyen

The REU Site is supported by the National Science Foundation under grant no 1758006.
References


• Vandenberghe L. Optimization methods for large-scale systems, EE236C lecture notes, UCLA.
