CS475 / CS675
Lecture 7: May 24, 2016

Image Denoising and

Iterative Methods

Reading: [Saad] Chapt 4

Total variation regularization

- Let $R(u) = \int_{\Omega} |\nabla u| dx$
- Then $\min_{u} \alpha \int_{\Omega} |\nabla u| dx + \left|\left|u - u^0\right|\right|_2^2$
- Idea: still minimize slopes, but don't get punished too much by it.

Total Variation Regularization

- Solution: Euler-Lagrange equation

$$-\alpha\nabla\left(\frac{1}{\left\|\nabla u\right\|_2}\nabla u\right)+u=u^0$$

NB: with Laplacian regularization, $\frac{1}{\left\|\nabla u\right\|_2}$ is replaced by 1

- For fixed α, near edges: $\left\|\nabla u_{ij}\right\|_2$ is large $\rightarrow \frac{1}{\left\|\nabla u_{ij}\right\|_2}$ is small

$$\Longrightarrow -\alpha\nabla\left(\frac{1}{\left\|\nabla u_{ij}\right\|_2}\nabla u_{ij}\right)\approx-\frac{\alpha}{\left\|\nabla u_{ij}\right\|_2}\Delta u_{ij}\approx 0$$

$$\Longrightarrow u_{ij}\approx u_{ij}^0$$

Total Variation Regularization

- On flat surfaces: $\left\|\nabla u_{ij}\right\|_2 \approx 0 \Longrightarrow \frac{1}{\left\|\nabla u_{ij}\right\|_2}$ is large

  $\Longrightarrow -C\Delta u_{ij} + u_{ij} = u_{ij}^0$ $\qquad$ $C =$ large constant
  $\Longrightarrow$ more diffusion at (i,j)
  $\Longrightarrow u_{ij}$ is flat
- Euler-Lagrange equation is non-linear
  - We cannot solve the equation directly
  - Instead use fixed-point iteration
- Idea: pick an initial guess, improve it by a simple procedure and repeat this procedure iteratively.

Iterative Methods

- Splitting
  - Let $A = M - N$
  - Then $Ax = b$
    $$(M - N)x = b$$
    $$Mx = Nx + b$$

- Define an iterative method by: $Mx^{k+1} = Nx^k + b$
  - Then $x^{k+1} = M^{-1}Nx^k + M^{-1}b$
    $$= M^{-1}(M - A)x^k + M^{-1}b$$
    $$= x^k + M^{-1}(b - Ax^k)$$

Splitting

- Note: if $M = A$,
  then $x^{k+1} = x^k + A^{-1}\left(b - Ax^k\right)$
  $= x^k + x - x^k = x$
  → one step convergence
  but one needs to compute $A^{-1}(b - Ax^k)$

- Goals: select M such that
  1. $M \approx A$
  2. M^{-1} is easy to compute

Richardson Iteration

- Select $M = \frac{1}{\theta} I$ ($\theta \in \Re$ is appropriately chosen)

  Then $M^{-1} = \theta I$

  $\therefore x^{k+1} = x^k + \theta I(b - Ax^k)$

- Consider the i[th] entry of x^{k+1}:

  $x_i^{k+1} = x_i^k + \theta(b_i - \sum_{j=1}^{n} a_{ij} x_j^k)$

Richardson Iteration

- Algorithm

  $x^0 =$ initial guess
  for $k = 0,1,2,\ldots$
      for $i = 1,2,\ldots,n$
          $x_i^{k+1} = x_i^k + \theta\left(b_i - \sum_{j=1}^{n} a_{ij} x_j^k\right)$
      end
  end

- Note: need 2 separate vectors x^k, x^{k+1}

Jacobi Iteration

- Select $M = D =$ diagonal of $A = \begin{bmatrix} a_{11} & & \\ & \ddots & \\ & & a_{nn} \end{bmatrix}$
- Then $M^{-1} = \begin{bmatrix} a_{11}^{-1} & & \\ & \ddots & \\ & & a_{nn}^{-1} \end{bmatrix}$
- $\therefore x^{k+1} = x^k + D^{-1}\left(b - Ax^k\right)$

$$x_i^{k+1} = x_i^k + \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{n} a_{ij} x_j^k\right)$$
$$= x_i^k + \frac{1}{a_{ii}}\left(b_i - \sum_{j \neq i} a_{ij} x_j^k - a_{ii} x_i^k\right)$$
$$= \frac{b_i - \sum_{j \neq i} a_{ij} x_j^k}{a_{ii}}$$

Jacobi Iteration

- Interpretation
  - Let $r^k = b - Ax^k$ (residual vector of x^k)
  - Then $x^k = x \iff r^k = 0$
  - Thus $\left\|r^k\right\|_2 \approx 0 \implies x^k \approx x$
- Consider $r_i^k \equiv b_i - \sum_{j=1}^{n} a_{ij} x_j^k$

  $= b_i - \sum_{j \neq i} a_{ij} x_j^k - a_{ii} x_i^k$
- In general $r_i^k \neq 0$

Jacobi Iteration

- Now modify x_i^k so that $r_i^k = 0$
  - i.e., $b_i - \sum_{j \neq i} a_{ij} x_j^k - a_{ii} x_i^{k+1} = 0$

$$x_i^{k+1} = \frac{b_i - \sum_{j \neq i} a_{ij} x_j^k}{a_{ii}}$$

$\rightarrow$ Jacobi iteration

Jacobi Iteration

- Algorithm

  $x^0 =$ initial guess

  For $k = 0,1,2,\dots$

  For $i = 1,2,\dots,n$

  $$x_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{j\neq i} a_{ij}x_j^k\right)$$

  End

  End

- Note: need separate storage for x^k, x^{k+1}

Gauss-Seidel Iteration

- Let $A = D - L - U$ where
  - $D =$ diag of A
  - $L =$ strictly lower Δ part
  - $U =$ strictly upper Δ part
- i.e. $A = \begin{bmatrix} \ddots & & -U \\ & D & \\ -L & & \ddots \end{bmatrix}$
- Then GS iteration: $M = D - L =$ lower Δ part of A
  - i.e., $x^{k+1} = x^k + (D - L)^{-1}(b - Ax^k)$

Gauss-Seidel Iteration

- Interpretation
  - Modify x_i^k so that $r_i^k = 0$
  - Use the new $x_j^{k+1}, j < i$ from previous updates

- i.e. $b_i - \sum_{i<j} a_{ij} x_j^{k+1} - a_{ii} x_i^{k+1} - \sum_{j>i} a_{ij} x_j^k = 0$

  $\therefore x_i^{k+1} = \frac{b_i - \sum_{j<i} a_{ij} x_j^{k+1} - \sum_{j>i} a_{ij} x_j^k}{a_{ii}}$

Gauss-Seidel Iteration

- Algorithm

  $x^0 =$ initial guess

  For $k = 0,1,2,\ldots$

  For $i = 1,2,\ldots,n$

  $$x_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{j<i} a_{ij} x_j^{k+1} - \sum_{j>i} a_{ij}\, x_j^k\right)$$

  End

  End

- Note: no extra storage for x^{k+1} since x^{k+1} can be overwritten immediately.

Gauss-Seidel Iteration

- Backward Gauss-Seidel

$M = D - U$: $\qquad x^{k+1} = x^k + (D-U)^{-1}(b - Ax^k)$

- Symmetric Gauss-Seidel
  - A forward sweep followed by a backward sweep:

$$\begin{cases} x^{k+\frac{1}{2}} = x^k + (D-L)^{-1}\left(b - Ax^k\right) \\ x^{k+1} = x^{k+\frac{1}{2}} + (D-U)^{-1}\left(b - Ax^{k+\frac{1}{2}}\right) \end{cases}$$

$$\Leftrightarrow x^{k+1} = x^k + (D-U)^{-1}D(D-L)^{-1}(b - Ax^k)$$