Math 4/896: Seminar in Mathematics
Topic: Inverse Theory

Instructor: Thomas Shores
Department of Mathematics

Lecture 25, April 13, 2006
AvH 10

Image Recovery

Problem:

An image is blurred and we want to sharpen it. Let intensity function $I_{true}(x, y)$define the true image and $I_{blurred}(x, y)$ define the blurred image.

- A typical model results from convolving true image with Gaussian point spread function

$$I_{blurred}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I_{true}(x - u, y - v) \Psi(u, v) \, du \, dv$$

where $\Psi(u, v) = e^{-(u^2+v^2)/(2\sigma^2)}$.

- Think about discretizing this over an SVGA image (1024×768).
- But the discretized matrix should be sparse!

Image Recovery

Problem:

An image is blurred and we want to sharpen it. Let intensity function $I_{true}(x, y)$define the true image and $I_{blurred}(x, y)$ define the blurred image.

- A typical model results from convolving true image with Gaussian point spread function

$$I_{blurred}(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I_{true}(x-u, y-v)\,\Psi(u, v)\; du\, dv$$

  where $\Psi(u, v) = e^{-(u^2+v^2)/(2\sigma^2)}$.
- Think about discretizing this over an SVGA image (1024×768).
- But the discretized matrix should be sparse!

Image Recovery

Problem:

An image is blurred and we want to sharpen it. Let intensity function $I_{true}(x, y)$define the true image and $I_{blurred}(x, y)$ define the blurred image.

- A typical model results from convolving true image with Gaussian point spread function

$$I_{blurred}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I_{true}(x - u, y - v)\, \Psi(u, v)\; du\, dv$$

  where $\Psi(u, v) = e^{-(u^2+v^2)/(2\sigma^2)}$.
- Think about discretizing this over an SVGA image (1024×768).
- But the discretized matrix should be sparse!

Image Recovery

Problem:

An image is blurred and we want to sharpen it. Let intensity function $I_{true}(x, y)$define the true image and $I_{blurred}(x, y)$ define the blurred image.

- A typical model results from convolving true image with Gaussian point spread function

$$I_{blurred}(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I_{true}(x-u, y-v)\,\Psi(u, v)\,du\,dv$$

  where $\Psi(u, v) = e^{-\left(u^2+v^2\right)/\left(2\sigma^2\right)}$.
- Think about discretizing this over an SVGA image (1024×768).
- But the discretized matrix should be sparse!

Sparse Matrices and Iterative Methods

Sparse Matrix:

A matrix with sufficiently many zeros that we should pay attention to them.

- There are efficient ways of storing such matrices and doing linear algebra on them.
- Given a problem $A\mathbf{x} = \mathbf{b}$ with A sparse, iterative methods become attractive because they usually only require storage of A, $\mathbf{x}$ and some auxillary vectors, and saxpy, gaxpy, dot algorithms – ("scalar a*x+y" , "general A*x+y", "dot product")
- Classical methods: Jacobi, Gauss-Seidel, Gauss-Seidel SOR and conjugate gradient.
- Methods especially useful for tomographic problems: Kaczmarz's method, ART (algebraic reconstruction technique).

Sparse Matrices and Iterative Methods

Sparse Matrix:

A matrix with sufficiently many zeros that we should pay attention to them.

- There are efficient ways of storing such matrices and doing linear algebra on them.
- Given a problem $A\mathbf{x} = \mathbf{b}$ with A sparse, iterative methods become attractive because they usually only require storage of A, $\mathbf{x}$ and some auxillary vectors, and saxpy, gaxpy, dot algorithms – ("scalar a*x+y" , "general A*x+y", "dot product")
- Classical methods: Jacobi, Gauss-Seidel, Gauss-Seidel SOR and conjugate gradient.
- Methods especially useful for tomographic problems: Kaczmarz's method, ART (algebraic reconstruction technique).

Sparse Matrices and Iterative Methods

Sparse Matrix:

A matrix with sufficiently many zeros that we should pay attention to them.

- There are efficient ways of storing such matrices and doing linear algebra on them.
- Given a problem $A\mathbf{x} = \mathbf{b}$ with A sparse, iterative methods become attractive because they usually only require storage of A, $\mathbf{x}$ and some auxillary vectors, and saxpy, gaxpy, dot algorithms – ("scalar a*x+y" , "general A*x+y", "dot product")
- Classical methods: Jacobi, Gauss-Seidel, Gauss-Seidel SOR and conjugate gradient.
- Methods especially useful for tomographic problems: Kaczmarz's method, ART (algebraic reconstruction technique).

Sparse Matrices and Iterative Methods

Sparse Matrix:

A matrix with sufficiently many zeros that we should pay attention to them.

- There are efficient ways of storing such matrices and doing linear algebra on them.
- Given a problem $A\mathbf{x} = \mathbf{b}$ with A sparse, iterative methods become attractive because they usually only require storage of A, $\mathbf{x}$ and some auxillary vectors, and saxpy, gaxpy, dot algorithms – ("scalar a*x+y" , "general A*x+y", "dot product")
- Classical methods: Jacobi, Gauss-Seidel, Gauss-Seidel SOR and conjugate gradient.
- Methods especially useful for tomographic problems: Kaczmarz's method, ART (algebraic reconstruction technique).

Sparse Matrices and Iterative Methods

Sparse Matrix:

A matrix with sufficiently many zeros that we should pay attention to them.

- There are efficient ways of storing such matrices and doing linear algebra on them.
- Given a problem $A\mathbf{x} = \mathbf{b}$ with A sparse, iterative methods become attractive because they usually only require storage of A, $\mathbf{x}$ and some auxillary vectors, and saxpy, gaxpy, dot algorithms – ("scalar a*x+y" , "general A*x+y", "dot product")
- Classical methods: Jacobi, Gauss-Seidel, Gauss-Seidel SOR and conjugate gradient.
- Methods especially useful for tomographic problems: Kaczmarz's method, ART (algebraic reconstruction technique).

Yet Another Regularization Idea

To regularize in face of iteration:

Use the number of iteration steps taken as a regularization parameter.

- Conjugate gradient methods are designed to work with SPD coefficient matrices A in the equation $A\mathbf{x} = \mathbf{b}$.
- So in the unregularized least squares problem $G^T G\mathbf{m} = G^T\mathbf{d}$ take $A = G^T G$ and $\mathbf{b} = G^T\mathbf{d}$, resulting in the CGLS method, in which we avoid explicitly computing $G^T G$.
- Key fact: in exact arithmetic, if we start at $\mathbf{m}^{(0)} = \mathbf{0}$, then $\left\|\mathbf{m}^{(k)}\right\|$ is monotone increasing in k and $\left\|G\mathbf{m}^{(k)} - \mathbf{d}\right\|$ is monotonically decreasing in k. So we can make an L-curve in terms of k.

Do Example 6.3 from the CD. Change startupfile path to Examples/chap6/examp3

Yet Another Regularization Idea

To regularize in face of iteration:

Use the number of iteration steps taken as a regularization parameter.

- Conjugate gradient methods are designed to work with SPD coefficient matrices A in the equation $A\mathbf{x} = \mathbf{b}$.
- So in the unregularized least squares problem $G^T G\mathbf{m} = G^T\mathbf{d}$ take $A = G^T G$ and $\mathbf{b} = G^T\mathbf{d}$, resulting in the CGLS method, in which we avoid explicitly computing $G^T G$.
- Key fact: in exact arithmetic, if we start at $\mathbf{m}^{(0)} = \mathbf{0}$, then $\left\|\mathbf{m}^{(k)}\right\|$ is monotone increasing in k and $\left\|G\mathbf{m}^{(k)} - \mathbf{d}\right\|$ is monotonically decreasing in k. So we can make an L-curve in terms of k.

Do Example 6.3 from the CD. Change startupfile path to Examples/chap6/examp3

Yet Another Regularization Idea

To regularize in face of iteration:

Use the number of iteration steps taken as a regularization parameter.

- Conjugate gradient methods are designed to work with SPD coefficient matrices A in the equation $A\mathbf{x} = \mathbf{b}$.
- So in the unregularized least squares problem $G^T G\mathbf{m} = G^T\mathbf{d}$ take $A = G^T G$ and $\mathbf{b} = G^T\mathbf{d}$, resulting in the CGLS method, in which we avoid explicitly computing $G^T G$.
- Key fact: in exact arithmetic, if we start at $\mathbf{m}^{(0)} = \mathbf{0}$, then $\left\|\mathbf{m}^{(k)}\right\|$ is monotone increasing in k and $\left\|G\mathbf{m}^{(k)} - \mathbf{d}\right\|$ is monotonically decreasing in k. So we can make an L-curve in terms of k.

Do Example 6.3 from the CD. Change startupfile path to Examples/chap6/examp3

Yet Another Regularization Idea

To regularize in face of iteration:

Use the number of iteration steps taken as a regularization parameter.

- Conjugate gradient methods are designed to work with SPD coefficient matrices A in the equation $A\mathbf{x} = \mathbf{b}$.
- So in the unregularized least squares problem $G^T G\mathbf{m} = G^T\mathbf{d}$ take $A = G^T G$ and $\mathbf{b} = G^T\mathbf{d}$, resulting in the CGLS method, in which we avoid explicitly computing $G^T G$.
- Key fact: in exact arithmetic, if we start at $\mathbf{m}^{(0)} = \mathbf{0}$, then $\left\|\mathbf{m}^{(k)}\right\|$ is monotone increasing in k and $\left\|G\mathbf{m}^{(k)} - \mathbf{d}\right\|$ is monotonically decreasing in k. So we can make an L-curve in terms of k.

Do Example 6.3 from the CD. Change startupfile path to Examples/chap6/examp3

Outline

Regularization...Sort Of

Basic Idea:

Use prior knowledge about the nature of the solution to restrict it:

- Most common restrictions: on the magnitude of the parameter values. Which leads to the problem:
- Minimize $f(\mathbf{m})$
  subject to $l \leq m \leq u$.
- One could choose $f(\mathbf{m}) = \|G\mathbf{m} - \mathbf{d}\|_2$(BVLS)
- One could choose $f(\mathbf{m}) = \mathbf{c}^T \cdot \mathbf{m}$with additional constraint $\|G\mathbf{m} - \mathbf{d}\|_2 \leq \delta$.

Regularization...Sort Of

Basic Idea:

Use prior knowledge about the nature of the solution to restrict it:

- Most common restrictions: on the magnitude of the parameter values. Which leads to the problem:
- Minimize $f(\mathbf{m})$
  subject to $l \leq m \leq u$.
- One could choose $f(\mathbf{m}) = \|G\mathbf{m} - \mathbf{d}\|_2$(BVLS)
- One could choose $f(\mathbf{m}) = \mathbf{c}^T \cdot \mathbf{m}$with additional constraint $\|G\mathbf{m} - \mathbf{d}\|_2 \leq \delta$.

Regularization...Sort Of

Basic Idea:

Use prior knowledge about the nature of the solution to restrict it:

- Most common restrictions: on the magnitude of the parameter values. Which leads to the problem:
- Minimize $f(\mathbf{m})$
  subject to $l \leq m \leq u$.
- One could choose $f(\mathbf{m}) = \|G\mathbf{m} - \mathbf{d}\|_2$(BVLS)
- One could choose $f(\mathbf{m}) = \mathbf{c}^T \cdot \mathbf{m}$with additional constraint $\|G\mathbf{m} - \mathbf{d}\|_2 \leq \delta$.

Regularization...Sort Of

Basic Idea:

Use prior knowledge about the nature of the solution to restrict it:

- Most common restrictions: on the magnitude of the parameter values. Which leads to the problem:
- Minimize $f(\mathbf{m})$
  subject to $l \leq m \leq u$.
- One could choose $f(\mathbf{m}) = \|G\mathbf{m} - \mathbf{d}\|_2$(BVLS)
- One could choose $f(\mathbf{m}) = \mathbf{c}^T \cdot \mathbf{m}$with additional constraint $\|G\mathbf{m} - \mathbf{d}\|_2 \leq \delta$.

Regularization...Sort Of

Basic Idea:

Use prior knowledge about the nature of the solution to restrict it:

- Most common restrictions: on the magnitude of the parameter values. Which leads to the problem:
- Minimize $f(\mathbf{m})$
  subject to $l \leq m \leq u$.
- One could choose $f(\mathbf{m}) = \|G\mathbf{m} - \mathbf{d}\|_2$(BVLS)
- One could choose $f(\mathbf{m}) = \mathbf{c}^T \cdot \mathbf{m}$with additional constraint $\|G\mathbf{m} - \mathbf{d}\|_2 \leq \delta$.

Example 3.3

Contaminant Transport

Let $C(x,t)$ be the concentration of a pollutant at point x in a linear stream, time t, where $0 \le x < \infty$ and $0 \le t \le T$. The defining model

$$
\begin{aligned}
\frac{\partial C}{\partial t} &= D\frac{\partial^2 C}{\partial x^2} - v\frac{\partial C}{\partial x} \\
C(0,t) &= C_{in}(t) \\
C(x,t) &\to 0, \ \ x \to \infty \\
C(x,0) &= C_0(x)
\end{aligned}
$$

Solution

Solution:

In the case that $C_0(x) \equiv 0$, the explicit solution is

$$C(x, T) = \int_0^T C_{in}(t) f(x, T - t)\, dt,$$

where

$$f(x, \tau) = \frac{x}{2\sqrt{\pi D \tau^3}} e^{-(x - v\tau)^2/(4D\tau)}$$

Inverse Problem

Problem:

Given simultaneous measurements at time T, to estimate the contaminant inflow history. That is, given data

$$d_i = C\left(x_i, T\right), \, i = 1, 2, \ldots, m,$$

to estimate

$$C_{in}\left(t\right), \, 0 \leq t \leq T.$$

Change the startupfile path to Examples/chap7/examp1 execute it and examp.

Outline

A Better Idea (?)

Entropy:

$$E(\mathbf{m}) = -\sum_{j=1}^{n} m_j \ln(w_j m_j),$$ **w** a vector of positive weights.

- Motivated by Shannon's information theory and Bolzmann's theory of entropy in statistical mechanics. A measure of uncertainty about which message or physical state will occur.
- Shannon's entropy function for a probability distribution $\{p_i\}_{i=1}^{n}$ is $H(\mathbf{p}) = -\sum_{i=1}^{n} p_i \ln(p_i)$.
- Bayesian Maximimum Entropy Principle: least biased model is one that maximizes entropy subject to constraints of testable information like bounds or average values of parameters.

A Better Idea (?)

Entropy:

$$E(\mathbf{m}) = -\sum_{j=1}^{n} m_j \ln(w_j m_j),$$ **w** a vector of positive weights.

- Motivated by Shannon's information theory and Bolzmann's theory of entropy in statistical mechanics. A measure of uncertainty about which message or physical state will occur.
- Shannon's entropy function for a probability distribution $\{p_i\}_{i=1}^{n}$ is $H(\mathbf{p}) = -\sum_{i=1}^{n} p_i \ln(p_i)$.
- Bayesian Maximimum Entropy Principle: least biased model is one that maximizes entropy subject to constraints of testable information like bounds or average values of parameters.

A Better Idea (?)

Entropy:

$$E(\mathbf{m}) = -\sum_{j=1}^{n} m_j \ln(w_j m_j),$$ $\mathbf{w}$ a vector of positive weights.

- Motivated by Shannon's information theory and Bolzmann's theory of entropy in statistical mechanics. A measure of uncertainty about which message or physical state will occur.
- Shannon's entropy function for a probability distribution $\{p_i\}_{i=1}^{n}$ is $H(\mathbf{p}) = -\sum_{i=1}^{n} p_i \ln(p_i)$.
- Bayesian Maximimum Entropy Principle: least biased model is one that maximizes entropy subject to constraints of testable information like bounds or average values of parameters.

A Better Idea (?)

Entropy:

$$E(\mathbf{m}) = -\sum_{j=1}^{n} m_j \ln(w_j m_j),\ \mathbf{w} \text{ a vector of positive weights.}$$

- Motivated by Shannon's information theory and Bolzmann's theory of entropy in statistical mechanics. A measure of uncertainty about which message or physical state will occur.
- Shannon's entropy function for a probability distribution $\{p_i\}_{i=1}^{n}$ is $H(\mathbf{p}) = -\sum_{i=1}^{n} p_i \ln(p_i)$.
- Bayesian Maximimum Entropy Principle: least biased model is one that maximizes entropy subject to constraints of testable information like bounds or average values of parameters.

Maximum Entropy Regularization

Maximize Entropy:

That is, our version. So problem looks like:

- Maximize $-\sum_{j=1}^{n} m_j \ln(w_j m_j)$
- Subject to $\|G\mathbf{m} - \mathbf{d}\|_2 \leq \delta$ and $\mathbf{m} \geq \mathbf{0}$.
- In absence of extra information, take $w_i = 1$. Lagrange multipliers give:
- Minimize $\|G\mathbf{m} - \mathbf{d}\|_2^2 + \alpha^2 \sum_{j=1}^{n} m_j \ln(w_j m_j)$,
- subject to $\mathbf{m} \geq \mathbf{0}$.

Change the startupfile path to Examples/chap7/examp2 execute it and examp.

Maximum Entropy Regularization

Maximize Entropy:

That is, our version. So problem looks like:

- Maximize $-\sum_{j=1}^{n} m_j \ln(w_j m_j)$
- Subject to $\|G\mathbf{m} - \mathbf{d}\|_2 \leq \delta$ and $\mathbf{m} \geq \mathbf{0}$.
- In absence of extra information, take $w_i = 1$. Lagrange multipliers give:
- Minimize $\|G\mathbf{m} - \mathbf{d}\|_2^2 + \alpha^2 \sum_{j=1}^{n} m_j \ln(w_j m_j)$,
- subject to $\mathbf{m} \geq \mathbf{0}$.

Change the startupfile path to Examples/chap7/examp2 execute it and examp.

Maximum Entropy Regularization

Maximize Entropy:

That is, our version. So problem looks like:

- Maximize $-\sum_{j=1}^{n} m_j \ln(w_j m_j)$
- Subject to $\|G\mathbf{m} - \mathbf{d}\|_2 \leq \delta$ and $\mathbf{m} \geq \mathbf{0}$.
- In absence of extra information, take $w_i = 1$. Lagrange multipliers give:
- Minimize $\|G\mathbf{m} - \mathbf{d}\|_2^2 + \alpha^2 \sum_{j=1}^{n} m_j \ln(w_j m_j)$,
- subject to $\mathbf{m} \geq \mathbf{0}$.

Change the startupfile path to Examples/chap7/examp2 execute it and examp.

Maximum Entropy Regularization

Maximize Entropy:

That is, our version. So problem looks like:

- Maximize $-\sum_{j=1}^{n} m_j \ln(w_j m_j)$
- Subject to $\|G\mathbf{m} - \mathbf{d}\|_2 \leq \delta$ and $\mathbf{m} \geq \mathbf{0}$.
- In absence of extra information, take $w_i = 1$. Lagrange multipliers give:
- Minimize $\|G\mathbf{m} - \mathbf{d}\|_2^2 + \alpha^2 \sum_{j=1}^{n} m_j \ln(w_j m_j)$,
- subject to $\mathbf{m} \geq \mathbf{0}$.

Change the startupfile path to Examples/chap7/examp2 execute it and examp.

Maximum Entropy Regularization

Maximize Entropy:

That is, our version. So problem looks like:

- Maximize $-\sum_{j=1}^{n} m_j \ln(w_j m_j)$
- Subject to $\|G\mathbf{m} - \mathbf{d}\|_2 \leq \delta$ and $\mathbf{m} \geq \mathbf{0}$.
- In absence of extra information, take $w_i = 1$. Lagrange multipliers give:
- Minimize $\|G\mathbf{m} - \mathbf{d}\|_2^2 + \alpha^2 \sum_{j=1}^{n} m_j \ln(w_j m_j)$,
- subject to $\mathbf{m} \geq \mathbf{0}$.

Change the startupfile path to Examples/chap7/examp2 execute it and examp.

Maximum Entropy Regularization

Maximize Entropy:

That is, our version. So problem looks like:

- Maximize $-\sum_{j=1}^{n} m_j \ln(w_j m_j)$
- Subject to $\|G\mathbf{m} - \mathbf{d}\|_2 \leq \delta$ and $\mathbf{m} \geq \mathbf{0}$.
- In absence of extra information, take $w_i = 1$. Lagrange multipliers give:
- Minimize $\|G\mathbf{m} - \mathbf{d}\|_2^2 + \alpha^2 \sum_{j=1}^{n} m_j \ln(w_j m_j)$,
- subject to $\mathbf{m} \geq \mathbf{0}$.

Change the startupfile path to Examples/chap7/examp2 execute it and examp.

Outline

TV Regularization

We only consider total variation regularization from this section.

Regularization term:

$\mathrm{DV}(\mathbf{m}) = \sum_{j=1}^{n-1} |m_{j+1} - m_j| = \|L\mathbf{m}\|_1$, where L is the matrix used in first order Tikhonov regularization.

- Problem becomes: minimize $\|G\mathbf{m} - \mathbf{d}\|_2^2 + \alpha \|\mathbf{m}\|_1$
- Better yet: minimize $\|G\mathbf{m} - \mathbf{d}\|_1 + \alpha \|\mathbf{m}\|_1$.
- Equivalently: minimize $\left\| \begin{bmatrix} G \\ \alpha L \end{bmatrix} \mathbf{m} - \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \right\|_1$.
- Now just use IRLS (iteratively reweighted least squares) to solve it and an L-curve of sorts to find optimal α.

TV Regularization

We only consider total variation regularization from this section.

Regularization term:

$\mathrm{DV}(\mathbf{m}) = \sum_{j=1}^{n-1} |m_{j+1} - m_j| = \|L\mathbf{m}\|_1$, where L is the matrix used in first order Tikhonov regularization.

- Problem becomes: minimize $\|G\mathbf{m} - \mathbf{d}\|_2^2 + \alpha \|\mathbf{m}\|_1$
- Better yet: minimize $\|G\mathbf{m} - \mathbf{d}\|_1 + \alpha \|\mathbf{m}\|_1$.
- Equivalently: minimize $\left\| \begin{bmatrix} G \\ \alpha L \end{bmatrix} \mathbf{m} - \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \right\|_1$.
- Now just use IRLS (iteratively reweighted least squares) to solve it and an L-curve of sorts to find optimal α.
-

TV Regularization

We only consider total variation regularization from this section.

Regularization term:

$\mathrm{DV}(\mathbf{m}) = \sum_{j=1}^{n-1} |m_{j+1} - m_j| = \|L\mathbf{m}\|_1$, where L is the matrix used in first order Tikhonov regularization.

- Problem becomes: minimize $\|G\mathbf{m} - \mathbf{d}\|_2^2 + \alpha \|\mathbf{m}\|_1$
- Better yet: minimize $\|G\mathbf{m} - \mathbf{d}\|_1 + \alpha \|\mathbf{m}\|_1$.
- Equivalently: minimize $\left\| \begin{bmatrix} G \\ \alpha L \end{bmatrix} \mathbf{m} - \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \right\|_1$.
- Now just use IRLS (iteratively reweighted least squares) to solve it and an L-curve of sorts to find optimal α.

TV Regularization

We only consider total variation regularization from this section.

Regularization term:

$\mathrm{DV}(\mathbf{m}) = \sum_{j=1}^{n-1} |m_{j+1} - m_j| = \|L\mathbf{m}\|_1$, where L is the matrix used in first order Tikhonov regularization.

- Problem becomes: minimize $\|G\mathbf{m} - \mathbf{d}\|_2^2 + \alpha \|\mathbf{m}\|_1$
- Better yet: minimize $\|G\mathbf{m} - \mathbf{d}\|_1 + \alpha \|\mathbf{m}\|_1$.
- Equivalently: minimize $\left\| \begin{bmatrix} G \\ \alpha L \end{bmatrix} \mathbf{m} - \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \right\|_1$.
- Now just use IRLS (iteratively reweighted least squares) to solve it and an L-curve of sorts to find optimal α.
-

TV Regularization

We only consider total variation regularization from this section.

Regularization term:

$\mathrm{DV}(\mathbf{m}) = \sum_{j=1}^{n-1} |m_{j+1} - m_j| = \|L\mathbf{m}\|_1$, where L is the matrix used in first order Tikhonov regularization.

- Problem becomes: minimize $\|G\mathbf{m} - \mathbf{d}\|_2^2 + \alpha \|\mathbf{m}\|_1$
- Better yet: minimize $\|G\mathbf{m} - \mathbf{d}\|_1 + \alpha \|\mathbf{m}\|_1$.
- Equivalently: minimize $\left\| \begin{bmatrix} G \\ \alpha L \end{bmatrix} \mathbf{m} - \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \right\|_1$.
- Now just use IRLS (iteratively reweighted least squares) to solve it and an L-curve of sorts to find optimal α.

TV Regularization

We only consider total variation regularization from this section.

Regularization term:

$\mathrm{DV}(\mathbf{m}) = \sum_{j=1}^{n-1} |m_{j+1} - m_j| = \|L\mathbf{m}\|_1$, where L is the matrix used in first order Tikhonov regularization.

- Problem becomes: minimize $\|G\mathbf{m} - \mathbf{d}\|_2^2 + \alpha \|\mathbf{m}\|_1$
- Better yet: minimize $\|G\mathbf{m} - \mathbf{d}\|_1 + \alpha \|\mathbf{m}\|_1$.
- Equivalently: minimize $\left\| \begin{bmatrix} G \\ \alpha L \end{bmatrix} \mathbf{m} - \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \right\|_1$.
- Now just use IRLS (iteratively reweighted least squares) to solve it and an L-curve of sorts to find optimal α.

Total Variation

Key Property:

- TV doesn't smooth discontinuities as much as Tikhonov regularization.

Change startupfile path to Examples/chap7/examp3 execute it and examp.

Total Variation

Key Property:

- TV doesn't smooth discontinuities as much as Tikhonov regularization.

Change startupfile path to Examples/chap7/examp3 execute it and examp.

Outline

Basic Problems

Root Finding:

Solve the system of equations represented in vector form as

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}.$$

for point(s) $\mathbf{x}^*$ for which $\mathbf{F}(\mathbf{x}^*) = \mathbf{0}$.

- Here $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ and $x = (x_1, \ldots, x_m)$
- Gradient notation: $\nabla f_j(\mathbf{x}) = \left(\frac{\partial f_j}{\partial x_1}(\mathbf{x}), \ldots, \frac{\partial f_j}{\partial x_m}(\mathbf{x})\right)$.
- Jacobian notation: $\nabla \mathbf{F}(\mathbf{x}) = [\nabla f_1(\mathbf{x}), \ldots, \nabla f_m(\mathbf{x})]^T = \left[\frac{\partial f_i}{\partial x_j}\right]_{i,j=1,\ldots,m}$.

Basic Problems

Root Finding:

Solve the system of equations represented in vector form as

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}.$$

for point(s) $\mathbf{x}^*$ for which $\mathbf{F}(\mathbf{x}^*) = \mathbf{0}$.

- Here $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ and $x = (x_1, \ldots, x_m)$
- Gradient notation: $\nabla f_j(\mathbf{x}) = \left(\frac{\partial f_j}{\partial x_1}(\mathbf{x}), \ldots, \frac{\partial f_j}{\partial x_m}(\mathbf{x})\right)$.
- Jacobian notation:
  $\nabla \mathbf{F}(\mathbf{x}) = [\nabla f_1(\mathbf{x}), \ldots, \nabla f_m(\mathbf{x})]^T = \left[\frac{\partial f_i}{\partial x_j}\right]_{i,j=1,\ldots,m}$.

Basic Problems

Root Finding:

Solve the system of equations represented in vector form as

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}.$$

for point(s) $\mathbf{x}^*$ for which $\mathbf{F}(\mathbf{x}^*) = \mathbf{0}$.

- Here $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ and $x = (x_1, \ldots, x_m)$
- Gradient notation: $\nabla f_j(\mathbf{x}) = \left(\frac{\partial f_j}{\partial x_1}(\mathbf{x}), \ldots, \frac{\partial f_j}{\partial x_m}(\mathbf{x})\right)$.
- Jacobian notation:
$\nabla \mathbf{F}(\mathbf{x}) = [\nabla f_1(\mathbf{x}), \ldots, \nabla f_m(\mathbf{x})]^T = \left[\frac{\partial f_i}{\partial x_j}\right]_{i,j=1,\ldots m}$.

Basic Problems

Root Finding:

Solve the system of equations represented in vector form as

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}.$$

for point(s) $\mathbf{x}^*$ for which $\mathbf{F}(\mathbf{x}^*) = \mathbf{0}$.

- Here $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ and $x = (x_1, \ldots, x_m)$
- Gradient notation: $\nabla f_j(\mathbf{x}) = \left(\frac{\partial f_j}{\partial x_1}(\mathbf{x}), \ldots, \frac{\partial f_j}{\partial x_m}(\mathbf{x})\right)$.
- Jacobian notation:
  $\nabla \mathbf{F}(\mathbf{x}) = [\nabla f_1(\mathbf{x}), \ldots, \nabla f_m(\mathbf{x})]^T = \left[\frac{\partial f_i}{\partial x_j}\right]_{i,j=1,\ldots m}$.

Basic Problems

Optimization:

Find the minimum value of scalar valued function $f(\mathbf{x})$, where $\mathbf{x}$ ranges over a feasible set Ω.

- Set $\mathbf{F}(\mathbf{x}) = \nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}(\mathbf{x}), \ldots, \frac{\partial f}{\partial x_m}(\mathbf{x})\right)$
- Hessian of f: $\nabla(\nabla f(\mathbf{x})) \equiv \nabla^2 f(\mathbf{x}) = \left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right]$.

Basic Problems

Optimization:

Find the minimum value of scalar valued function $f(\mathbf{x})$, where $\mathbf{x}$ ranges over a feasible set Ω.

- Set $\mathbf{F}(\mathbf{x}) = \nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}(\mathbf{x}), \ldots, \frac{\partial f}{\partial x_m}(\mathbf{x}) \right)$
- Hessian of f: $\nabla(\nabla f(\mathbf{x})) \equiv \nabla^2 f(\mathbf{x}) = \left[\frac{\partial^2 f}{\partial x_i \partial x_j} \right]$.

Basic Problems

Optimization:

Find the minimum value of scalar valued function $f(\mathbf{x})$, where $\mathbf{x}$ ranges over a feasible set Ω.

- Set $\mathbf{F}(\mathbf{x}) = \nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}(\mathbf{x}), \ldots, \frac{\partial f}{\partial x_m}(\mathbf{x})\right)$
- Hessian of f: $\nabla(\nabla f(\mathbf{x})) \equiv \nabla^2 f(\mathbf{x}) = \left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right]$.

Taylor Theorems

First Order

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ has continuous second partials and $\mathbf{x}^*, \mathbf{x} \in \mathbb{R}^n$. Then
$f(\mathbf{x}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T (\mathbf{x} - \mathbf{x}^*) + \mathcal{O}\left(\|\mathbf{x} - \mathbf{x}^*\|^2\right), \mathbf{x} \to \mathbf{x}.$

Second Order

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ has continuous third partials and $\mathbf{x}^*, \mathbf{x} \in \mathbb{R}^n$. Then $f(\mathbf{x}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T (\mathbf{x} - \mathbf{x}^*) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^T \nabla^2 f(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) + \mathcal{O}\left(\|\mathbf{x} - \mathbf{x}^*\|^3\right), \mathbf{x} \to \mathbf{x}.$
(See Appendix C for versions of Taylor's theorem with weaker hypotheses.)

Taylor Theorems

First Order

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ has continuous second partials and $\mathbf{x}^*, \mathbf{x} \in \mathbb{R}^n$. Then

$$f(\mathbf{x}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T (\mathbf{x} - \mathbf{x}^*) + \mathcal{O}\left(\|\mathbf{x} - \mathbf{x}^*\|^2\right), \ \mathbf{x} \to \mathbf{x}.$$

Second Order

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ has continuous third partials and $\mathbf{x}^*, \mathbf{x} \in \mathbb{R}^n$. Then $f(\mathbf{x}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T (\mathbf{x} - \mathbf{x}^*) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^T \nabla^2 f(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) + \mathcal{O}\left(\|\mathbf{x} - \mathbf{x}^*\|^3\right), \ \mathbf{x} \to \mathbf{x}.$
(See Appendix C for versions of Taylor's theorem with weaker hypotheses.)

Newton Algorithms

Root Finding

Input $\mathbf{F}$, $\nabla\mathbf{F}$, $\mathbf{x}^0$, N_{max}
for $k = 0, \ldots, N_{max}$
  $\mathbf{x}^{k+1} = \mathbf{x}^k - \nabla\mathbf{F}\left(\mathbf{x}^k\right)^{-1}\mathbf{F}\left(\mathbf{x}^k\right)$
  if $\mathbf{x}^{k+1}, \mathbf{x}^k$ pass a convergence test
    return($\mathbf{x}^k$)
  end
end
return($\mathbf{x}^{N_{max}}$)

Convergence Result

Theorem

Let $\mathbf{x}^*$ *be a root of the equation* $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, *where* $\mathbf{F}, \mathbf{x}$ *are m-vectors,* $\mathbf{F}$ *has continuous first partials in some neighborhood of* $\mathbf{x}^*$ *and* $\nabla\mathbf{F}(\mathbf{x}^*)$ *is non-singular. Then Newton's method yields a sequence of vectors that converges to* $\mathbf{x}^*$, *provided that* $\mathbf{x}^0$ *is sufficiently close to* $\mathbf{x}^*$. *If, in addition,* $\mathbf{F}$ *has continuous second partials in some neighborhood of* $\mathbf{x}^*$, *then the convergence is quadratic in the sense that for some constant* $K > 0$,

$$\left\|\mathbf{x}^{k+1} - \mathbf{x}^*\right\| \leq K \left\|\mathbf{x}^k - \mathbf{x}^*\right\|^2 .$$

Newton for Optimization

Bright Idea:

We know from calculus that where $f(\mathbf{x})$ has a local minimum, $\nabla f = \mathbf{0}$. So just let $\mathbf{F}(\mathbf{x}) = \nabla f(\mathbf{x})$ and use Newton's method.

- Result is iteration formula: $\mathbf{x}^{k+1} = \mathbf{x}^k - \nabla^2 f\left(\mathbf{x}^k\right)^{-1} \nabla f\left(\mathbf{x}^k\right)$
- We can turn this approach on its head: root finding is just a special case of optimization, i.e., solving $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ is the same as minimizing $f(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$.
- Downside of root finding point of view of optimization: saddle points and local maxima $\mathbf{x}$ also satisfy $\nabla f(\mathbf{x}) = \mathbf{0}$.
- Upside of optimization view of root finding: if $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ doesn't have a root, minimizing $f(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$ finds the next best solutions – least squares solutions!
- In fact, least squares problem for $\|G\mathbf{m} - \mathbf{d}\|^2$ is optimization!

Newton for Optimization

Bright Idea:

We know from calculus that where $f(\mathbf{x})$ has a local minimum, $\nabla f = \mathbf{0}$. So just let $\mathbf{F}(\mathbf{x}) = \nabla f(\mathbf{x})$ and use Newton's method.

- Result is iteration formula: $\mathbf{x}^{k+1} = \mathbf{x}^k - \nabla^2 f\left(\mathbf{x}^k\right)^{-1} \nabla f\left(\mathbf{x}^k\right)$
- We can turn this approach on its head: root finding is just a special case of optimization, i.e., solving $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ is the same as minimizing $f(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$.
- Downside of root finding point of view of optimization: saddle points and local maxima $\mathbf{x}$ also satisfy $\nabla f(\mathbf{x}) = \mathbf{0}$.
- Upside of optimization view of root finding: if $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ doesn't have a root, minimizing $f(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$ finds the next best solutions – least squares solutions!
- In fact, least squares problem for $\|G\mathbf{m} - \mathbf{d}\|^2$ is optimization!

Newton for Optimization

Bright Idea:

We know from calculus that where $f(\mathbf{x})$ has a local minimum, $\nabla f = \mathbf{0}$. So just let $\mathbf{F}(\mathbf{x}) = \nabla f(\mathbf{x})$ and use Newton's method.

- Result is iteration formula: $\mathbf{x}^{k+1} = \mathbf{x}^k - \nabla^2 f\left(\mathbf{x}^k\right)^{-1} \nabla f\left(\mathbf{x}^k\right)$
- We can turn this approach on its head: root finding is just a special case of optimization, i.e., solving $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ is the same as minimizing $f(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$.
- Downside of root finding point of view of optimization: saddle points and local maxima $\mathbf{x}$ also satisfy $\nabla f(\mathbf{x}) = \mathbf{0}$.
- Upside of optimization view of root finding: if $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ doesn't have a root, minimizing $f(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$ finds the next best solutions – least squares solutions!
- In fact, least squares problem for $\|G\mathbf{m} - \mathbf{d}\|^2$ is optimization!

Newton for Optimization

Bright Idea:

We know from calculus that where $f(\mathbf{x})$ has a local minimum, $\nabla f = \mathbf{0}$. So just let $\mathbf{F}(\mathbf{x}) = \nabla f(\mathbf{x})$ and use Newton's method.

- Result is iteration formula: $\mathbf{x}^{k+1} = \mathbf{x}^k - \nabla^2 f\left(\mathbf{x}^k\right)^{-1} \nabla f\left(\mathbf{x}^k\right)$
- We can turn this approach on its head: root finding is just a special case of optimization, i.e., solving $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ is the same as minimizing $f(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$.
- Downside of root finding point of view of optimization: saddle points and local maxima $\mathbf{x}$ also satisfy $\nabla f(\mathbf{x}) = \mathbf{0}$.
- Upside of optimization view of root finding: if $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ doesn't have a root, minimizing $f(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$ finds the next best solutions – least squares solutions!
- In fact, least squares problem for $\|G\mathbf{m} - \mathbf{d}\|^2$ is optimization!

Newton for Optimization

Bright Idea:

We know from calculus that where $f(\mathbf{x})$ has a local minimum, $\nabla f = \mathbf{0}$. So just let $\mathbf{F}(\mathbf{x}) = \nabla f(\mathbf{x})$ and use Newton's method.

- Result is iteration formula: $\mathbf{x}^{k+1} = \mathbf{x}^k - \nabla^2 f\left(\mathbf{x}^k\right)^{-1} \nabla f\left(\mathbf{x}^k\right)$
- We can turn this approach on its head: root finding is just a special case of optimization, i.e., solving $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ is the same as minimizing $f(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$.
- Downside of root finding point of view of optimization: saddle points and local maxima $\mathbf{x}$ also satisfy $\nabla f(\mathbf{x}) = \mathbf{0}$.
- Upside of optimization view of root finding: if $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ doesn't have a root, minimizing $f(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$ finds the next best solutions – least squares solutions!
- In fact, least squares problem for $\|G\mathbf{m} - \mathbf{d}\|^2$ is optimization!

Newton for Optimization

Bright Idea:

We know from calculus that where $f(\mathbf{x})$ has a local minimum, $\nabla f = \mathbf{0}$. So just let $\mathbf{F}(\mathbf{x}) = \nabla f(\mathbf{x})$ and use Newton's method.

- Result is iteration formula: $\mathbf{x}^{k+1} = \mathbf{x}^k - \nabla^2 f\left(\mathbf{x}^k\right)^{-1} \nabla f\left(\mathbf{x}^k\right)$
- We can turn this approach on its head: root finding is just a special case of optimization, i.e., solving $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ is the same as minimizing $f(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$.
- Downside of root finding point of view of optimization: saddle points and local maxima $\mathbf{x}$ also satisfy $\nabla f(\mathbf{x}) = \mathbf{0}$.
- Upside of optimization view of root finding: if $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ doesn't have a root, minimizing $f(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$ finds the next best solutions – least squares solutions!
- In fact, least squares problem for $\|G\mathbf{m} - \mathbf{d}\|^2$ is optimization!

Remarks on Newton

About Newton:

This barely scratches the surface of optimization theory (take Math 4/833 if you can!!).

- Far from a zero, Newton does not exhibit quadratic convergence. It is accelerated by a line search in the Newton direction $-\nabla \mathbf{F}\left(\mathbf{x}^k\right)^{-1} \mathbf{F}\left(\mathbf{x}^k\right)$ for a point that (approximately) minimizes a merit function like $m(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$.
- Optimization is NOT a special case of root finding. There are special characteristics of the min $f(\mathbf{x})$ problem that get lost if one only tries to find a zero of ∇f .
- For example, $-\nabla f$ is a search direction that leads to the method of steepest descent. This is not terribly efficient, but well understood.
- There is an automatic merit function, namely $f(\mathbf{x})$, in any search direction. Using this helps avoid saddle points, maxima.

Remarks on Newton

About Newton:

This barely scratches the surface of optimization theory (take Math 4/833 if you can!!).

- Far from a zero, Newton does not exhibit quadratic convergence. It is accelerated by a line search in the Newton direction $-\nabla \mathbf{F}\left(\mathbf{x}^k\right)^{-1} \mathbf{F}\left(\mathbf{x}^k\right)$ for a point that (approximately) minimizes a merit function like $m(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$.
- Optimization is NOT a special case of root finding. There are special characteristics of the min $f(\mathbf{x})$ problem that get lost if one only tries to find a zero of ∇f .
- For example, $-\nabla f$ is a search direction that leads to the method of steepest descent. This is not terribly efficient, but well understood.
- There is an automatic merit function, namely $f(\mathbf{x})$, in any search direction. Using this helps avoid saddle points, maxima.

Remarks on Newton

About Newton:

This barely scratches the surface of optimization theory (take Math 4/833 if you can!!).

- Far from a zero, Newton does not exhibit quadratic convergence. It is accelerated by a line search in the Newton direction $-\nabla \mathbf{F}\left(\mathbf{x}^k\right)^{-1} \mathbf{F}\left(\mathbf{x}^k\right)$ for a point that (approximately) minimizes a merit function like $m(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$.
- Optimization is NOT a special case of root finding. There are special characteristics of the min $f(\mathbf{x})$ problem that get lost if one only tries to find a zero of ∇f .
- For example, $-\nabla f$ is a search direction that leads to the method of steepest descent. This is not terribly efficient, but well understood.
- There is an automatic merit function, namely $f(\mathbf{x})$, in any search direction. Using this helps avoid saddle points, maxima.

About Newton:

This barely scratches the surface of optimization theory (take Math 4/833 if you can!!).

- Far from a zero, Newton does not exhibit quadratic convergence. It is accelerated by a line search in the Newton direction $-\nabla \mathbf{F}\left(\mathbf{x}^k\right)^{-1}\mathbf{F}\left(\mathbf{x}^k\right)$ for a point that (approximately) minimizes a merit function like $m(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$.
- Optimization is NOT a special case of root finding. There are special characteristics of the min $f(\mathbf{x})$ problem that get lost if one only tries to find a zero of ∇f .
- For example, $-\nabla f$ is a search direction that leads to the method of steepest descent. This is not terribly efficient, but well understood.
- There is an automatic merit function, namely $f(\mathbf{x})$, in any search direction. Using this helps avoid saddle points, maxima.

Remarks on Newton

About Newton:

This barely scratches the surface of optimization theory (take Math 4/833 if you can!!).

- Far from a zero, Newton does not exhibit quadratic convergence. It is accelerated by a line search in the Newton direction $-\nabla \mathbf{F}\left(\mathbf{x}^k\right)^{-1} \mathbf{F}\left(\mathbf{x}^k\right)$ for a point that (approximately) minimizes a merit function like $m(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$.
- Optimization is NOT a special case of root finding. There are special characteristics of the min $f(\mathbf{x})$ problem that get lost if one only tries to find a zero of ∇f .
- For example, $-\nabla f$ is a search direction that leads to the method of steepest descent. This is not terribly efficient, but well understood.
- There is an automatic merit function, namely $f(\mathbf{x})$, in any search direction. Using this helps avoid saddle points, maxima.

Outline

The Problem:

Given a function $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$, minimize

$$f(\mathbf{x}) = \sum_{k=0}^{m} f_k(\mathbf{x})^2 = \|\mathbf{F}(\mathbf{x})\|^2.$$

- Newton's method can be very expensive, due to derivative evaluations.
- For starters, one shows $\nabla f(\mathbf{x}) = 2(\nabla \mathbf{F}(\mathbf{x}))^T \mathbf{F}(\mathbf{x})$
- Then, $\nabla^2 f(\mathbf{x}) = 2(\nabla \mathbf{F}(\mathbf{x}))^T \nabla \mathbf{F}(\mathbf{x}) + Q(\mathbf{x})$, where $Q(\mathbf{x})$contains all the second derivatives.

The Problem:

Given a function $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$, minimize

$$f(\mathbf{x}) = \sum_{k=0}^{m} f_k(\mathbf{x})^2 = \|\mathbf{F}(\mathbf{x})\|^2.$$

- Newton's method can be very expensive, due to derivative evaluations.
- For starters, one shows $\nabla f(\mathbf{x}) = 2(\nabla \mathbf{F}(\mathbf{x}))^T \mathbf{F}(\mathbf{x})$
- Then, $\nabla^2 f(\mathbf{x}) = 2(\nabla \mathbf{F}(\mathbf{x}))^T \nabla \mathbf{F}(\mathbf{x}) + Q(\mathbf{x})$, where $Q(\mathbf{x})$contains all the second derivatives.

The Problem:

Given a function $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$, minimize

$$f(\mathbf{x}) = \sum_{k=0}^{m} f_k(\mathbf{x})^2 = \|\mathbf{F}(\mathbf{x})\|^2.$$

- Newton's method can be very expensive, due to derivative evaluations.
- For starters, one shows $\nabla f(\mathbf{x}) = 2(\nabla \mathbf{F}(\mathbf{x}))^T \mathbf{F}(\mathbf{x})$
- Then, $\nabla^2 f(\mathbf{x}) = 2(\nabla \mathbf{F}(\mathbf{x}))^T \nabla \mathbf{F}(\mathbf{x}) + Q(\mathbf{x})$, where $Q(\mathbf{x})$contains all the second derivatives.

Gauss-Newton and Levenberg-Marquardt

The Problem:

Given a function $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$, minimize

$$f(\mathbf{x}) = \sum_{k=0}^{m} f_k(\mathbf{x})^2 = \|\mathbf{F}(\mathbf{x})\|^2.$$

- Newton's method can be very expensive, due to derivative evaluations.
- For starters, one shows $\nabla f(\mathbf{x}) = 2(\nabla \mathbf{F}(\mathbf{x}))^T \mathbf{F}(\mathbf{x})$
- Then, $\nabla^2 f(\mathbf{x}) = 2(\nabla \mathbf{F}(\mathbf{x}))^T \nabla \mathbf{F}(\mathbf{x}) + Q(\mathbf{x})$, where $Q(\mathbf{x})$contains all the second derivatives.

- Comment 5.