Optimization and Approximation

Optimization and approximation are two closely related topics. Generally, in optimization, a real-valued function $f(x)$, called objective function, is minimized. In approximation, a function $f(x)$ is approximated by another function, usually a simpler one. We must have a measurement for the closeness of the approximation. The measurement is usually a real valued function $\phi(x)$. We find a "best" approximation by minimizing this function in some sense.

1 Continuous Optimization

We study some general methods for solving the problem:

$$\min_{x \in S} f(x)$$

$f(x)$: objective function, single variable and real-valued
S: support

Golden section search
Assumption: $f(x)$ has a unique global minimum in $[a, b]$.

- If x_* is the minimizer, then $f(x)$ monotonically decreases in $[a, x_*]$ and monotonically increases in $[x_*, b]$, for otherwise we would have local minimums.

Algorithm (Generic)

Choose interior points c and d:
    $c = a + r(b - a)$
    $d = a + (1 - r)(b - a)$, $0 < r < 0.5$
if $f(c) \le f(d)$
    $b = d$
else
    $a = c$
end

Each step, the length of the interval is reduced by a factor of $(1-r)$.
The choice of r:

- When $f(c) \leq f(d)$, $d_+ = c$ (the next d is c).
- When $f(c) > f(d)$, $c_+ = d$ (the next c is d).

Why? Reduce function evaluations by reusing the function values computed in the previous step.
When $f(c) \leq f(d)$, $b_+ = d$,

$$d_+ = a + (1-r)(b_+ - a) = a + (1-r)(d-a),$$

then $d_+ = c$ means

$$a + (1-r)(d-a) = a + r(b-a),$$

which implies $(1-r)^2 = r$.
When $f(c) > f(d)$, $a_+ = c$, then $c_+ = d$ means

$$c_+ = c + r(b-c) = a + (1-r)(b-a),$$

which also implies $(1-r)^2 = r$. Thus we have

$$r = \frac{3-\sqrt{5}}{2}$$

Algorithm (Golden Section)

```
c = a + r*(b - a); fc = f(c);
d = a + (1-r)*(b - a); fd = f(d);
if fc <= fd
   b = d; fb = fd;
   d = c; fd = fc;
   c = a + r*(b-a); fc = f(c);
else
   a = c; fa = fc;
   c = d; fc = fd;
   d = a + (1-r)*(b-a); fd = f(d);
end
```

Each step reduces the length of the interval by a factor of

$$1 - r = 1 - \frac{3 - \sqrt{5}}{2} \approx 0.618.$$

Termination criteria:

$$(d - c) \leq u \cdot \max(|c|, |d|).$$

Multivariate functions

$f(\mathbf{x})$ where $\mathbf{x}$ is a vector.
Gradient

$$\nabla f(\mathbf{x}_c) = \begin{bmatrix} \frac{\partial f(\mathbf{x}_c)}{\partial x_1} \\ \vdots \\ \frac{\partial f(\mathbf{x}_c)}{\partial x_n} \end{bmatrix}$$

$-\nabla f(\mathbf{x}_c)$: the direction of greatest decrease from $\mathbf{x}_c$.
Steepest descent: $\mathbf{s}_c = -\nabla f(\mathbf{x}_c)$, find λ_c such that $f(\mathbf{x}_c + \lambda_c \mathbf{s}_c) \leq f(\mathbf{x}_c + \lambda \mathbf{s}_c)$, for all $\lambda \in \mathcal{R}$. Then we set $\mathbf{x}_+ = \mathbf{x}_c + \lambda_c \mathbf{s}_c$.
Linear search: a single real variable minimization problem is a nontrivial task.

2 Approximation

Suppose $f(x)$ is to be approximated by $p(x)$. What is the measurement for the closeness? The first one we will introduce is $\max|f(x) - p(x)|$, also called ∞-norm. Thus, the approximation method is called minmax approximation, since we try to minimize the maximum error. The second measurement is $\sum_1^n (f(x_i) - p(x_i))^2$, also called 2-norm, the sum of the squares at discrete points. Thus the method is called least squares approximation, since we try to minimize the sum of squares.

Minmax

Example. Approximate $\sin(x)$ on $[0, \pi/2]$ by a linear polynomial $p_1(x) = c_0 + c_1 x$ minimizing the function

$$\phi(x) = \max_{x \in [0, \pi/2]} |\sin(x) - p_1(x)|.$$

Solution: $p_1(x) = 0.105 + 2x/\pi$.
The maximum error occurs at three points:

x	0.000	0.881	$\pi/2$
$\sin(\pi x/2) - p_1(x)$	-0.105	0.105	-0.105

Observation (with generalization):

- For a polynomial $p_k(x)$ of degree k ($k = 1$ this case), there are $k+2$ points:
$$0 \le x_0 < x_1 < \cdots < x_{k+1} \le 0$$
such that
  1. $|e(x)| = |f(x) - p_k(x)|$ is maximized at x_i, $i = 0, ..., k+1$.
  2. $\mathrm{sign}(e(x_i))$ alternate.

Chebyshev Polynomials

Chebyshev polynomials are used in minmax approximation. They can be defined in terms of trigonometric and hyperbolic functions:

$$c_k(t) = \begin{cases} \cos(k\cos^{-1} t) & |t| \le 1 \\ \cosh(k\cosh^{-1} t) & |t| \ge 1 \end{cases}$$

- $c_k(t)$ is a polynomial given by the recursion
$$c_{k+1}(t) = 2tc_k(t) - c_{k-1}(t), \quad c_0(t) = 1, \; c_1(t) = t. \tag{1}$$
Thus $c_k(t)$ is of degree k and its leading term is $2^{k-1}t^k$.
- $c_k(1) = 1$.
- When $|t| \le 1$, $|c_k(t)| \le 1$ and attains one at
$$t_{i,k} = \cos\frac{(k-i)\pi}{k}, \quad i = 0, 1, ..., k.$$
Specifically, $c_k(t_{i,k}) = (-1)^{k-i}$.
- The zeros of $c_k(t)$ are $t_i = \cos((2i+1)\pi/(2k))$ for $i = 0, 1, .., k-1$.

- The recursion (1) can be used to evaluating $c_k(t)$ at point t in $O(k)$ operations. Since all $c_i(t)$ $(i \leq k)$ are evaluated, the linear combination $\sum_{i=0}^{k} b_i c_i(t)$, the approximation, can be evaluated in $O(k)$ operations.
- The coefficients $c_{i,k}$ $(i = 0, ..., k)$ of the explicit polynomial form

  $$c_k(t) = c_{0,k} + c_{1,k}t + \cdots + c_{k,k}t^k$$

  are given by $c_{0,0} = 1$, $c_{0,1} = 0$, $c_{1,1} = 1$, and

  $$c_{0,k} = -c_{0,k-2}, \quad c_{i,k} = 2c_{i-1,k-1} - c_{i,k-2}.$$

  The resulting $C = [c_{i,k}]$ is an upper triangular matrix.

Applications

Interpolation

Runge function revisited. Suppose that the function $f(x)$ to be approximated is expanded in Chebyshev polynomials:

$$f(x) = \sum_{0}^{\infty} b_i c_i(x).$$

We truncate the above series and get

$$C_n(x) = \sum_{0}^{n} b_i c_i(x).$$

Then

$$f(x) - C_n(x) = \sum_{n+1}^{\infty} b_i c_i(x) \approx b_{n+1} c_{n+1}(x),$$

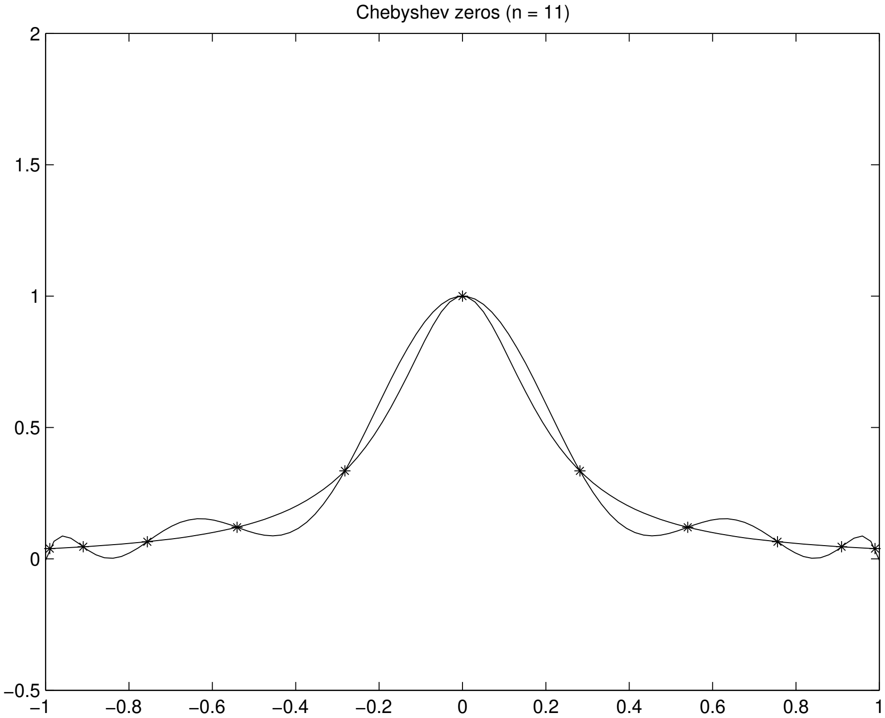
since b_i decrease very fast. Now, we interpolate Runge function at the zeros of $c_{n+1}(x)$:

$$x_i = \cos\left(\frac{2i+1}{2n+2}\pi\right), \quad i = 0, 1, ..., n.$$

Let $p_n(x)$ be the interpolating polynomial, then

$$p_n(x_i) - f(x_i) = 0,$$

which implies that $p_n(x)$ is approximately $C_n(x)$ and we expect the error is small. Note that this is not a universal cure.



Economization of Power Series

Chebyshev approximation can be applied to the evaluation of special functions. Suppose we have a power series expansion of a function

$$f(x) = a_0 + a_1 x + \cdots + \cdots$$

and we want to approximate $f(x)$ to an accuracy ϵ. We first truncate the series so that we know the truncation error is smaller than ϵ. We may make a conservative estimate of m:

$$p_m(x) \equiv a_0 + a_1 x + \cdots + a_m x^m \approx f(x).$$

We then find Chebyshev polynomials to approximate $p_m(x)$:

$$b_0 c_0(x) + b_1 c_1(x) + \cdots + b_k c_k(x) \approx p_m(x)$$

such that

- We have a good estimate of the accuracy;

- k is smaller than m, thus we approximation is economized;

This is how we proceed:

- Compute the $(m+1)$-by-$(m+1)$ upper triangular matrix C of the coefficients of the Chebyshev polynomials;

- Let $\mathbf{x} = [1, x, ..., x^m]^{\mathrm{T}}$, $\mathbf{a} = [a_0, a_1, ..., a_m]^{\mathrm{T}}$, and $\mathbf{b} = [b_0, b_1, ..., b_m]^{\mathrm{T}}$, then $p_m(x) = \mathbf{x}^{\mathrm{T}}\mathbf{a}$ and $\mathbf{x}^{\mathrm{T}}C\mathbf{b}$ is the Chebyshev polynomial representation of $p_m(x)$, i.e., $\mathbf{x}^{\mathrm{T}}C\mathbf{b} = p_m(x)$;

- Let $\mathbf{c}^{\mathrm{T}} = \mathbf{x}^{\mathrm{T}}C$, then $\mathbf{c}$ is the vector of the Chebyshev polynomials and thus $p_m(x) = \mathbf{c}^{\mathrm{T}}C^{-1}\mathbf{a}$ and $\mathbf{x}^{\mathrm{T}}C\mathbf{b}$. So, we have $\mathbf{c}^{\mathrm{T}}C^{-1}\mathbf{a} = \mathbf{c}^{\mathrm{T}}\mathbf{b}$. Consequently, $C^{-1}\mathbf{a} = \mathbf{b}$, i.e., $\mathbf{b}$ is the solution of the linear system $C\mathbf{b} = \mathbf{a}$, since the Chebyshev polynomials $c_i(x)$ are independent;

- Solve the upper triangular system $C\mathbf{b} = \mathbf{a}$;

- Find the smallest k, such that $|b_{k+1}| < \epsilon$;

- $b_0c_0(x) + b_1c_1(x) + \cdots + b_kc_k(x)$ is the polynomial approximation satisfying the accuracy.

Example. Approximate e^t.

Linear least squares

Problem: Given a matrix A (m-by-n, $m \geq n$) and $\mathbf{b}$ (m-by-1), find $\mathbf{x}$ (n-by-1) minimizing

$$\|A\mathbf{x} - \mathbf{b}\|_2^2.$$

Assume the columns of A are linearly independent, in other words, A is of full column rank.

Example. Square root problem revisited.

Find a_1 and a_2 in $y(x) = a_1x + a_2$, such that the sum

$$(y(0.25) - \sqrt{0.25})^2 + (y(0.5) - \sqrt{0.5})^2 + (y(1.0) - \sqrt{1.0})^2$$

of squares at the points 0.25, 0.5, and 1.0 is minimized.

In matrix-vector form:

$$A = \begin{bmatrix} 0.25 & 1 \\ 0.5 & 1 \\ 1.0 & 1 \end{bmatrix}, \ \mathbf{x} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} \sqrt{0.25} \\ \sqrt{0.5} \\ \sqrt{1.0} \end{bmatrix}.$$

Idea: Transform A into a triangular matrix:

$$PA = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

where R is upper triangular. Then the problem becomes

$$\|A\mathbf{x} - \mathbf{b}\|_2^2 = \|P^{-1}(\widehat{R}\mathbf{x} - P\mathbf{b})\|_2^2$$

where

$$\widehat{R} = \begin{bmatrix} R \\ 0 \end{bmatrix}.$$

Desirable properties of P:

- P^{-1} is easy to compute;
- $\|P^{-1}\mathbf{z}\|_2^2 = \|\mathbf{z}\|_2^2$ for any $\mathbf{z}$.

Partitioning

$$P\mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix},$$

we have

$$\|A\mathbf{x} - \mathbf{b}\|_2^2 = \|R\mathbf{x} - \mathbf{b}_1\|_2^2 + \|\mathbf{b}_2\|_2^2.$$

Since $\mathbf{b}_2$ is independent of the solution $\mathbf{x}$, the LLS solution is the solution of the triangular system

$$R\mathbf{x} = \mathbf{b}_1.$$

What about the matrix P? Orthogonal matrix (transformation) $Q = P^{-1}$: $Q^{-1} = Q^T$ and $\|Q\mathbf{z}\|_2^2 = \|\mathbf{z}\|_2^2$ for any $\mathbf{z}$.

Partitioning $Q = [Q_A, Q_C]$, where Q_A is m-by-n, we get a short form

$$A = Q_A R.$$

Also, we have $\mathbf{b}_1 = Q_A^{\mathrm{T}}\mathbf{b}$. It then follows that the LLS solution $\mathbf{x}$ is the solution of $R^{\mathrm{T}}R\mathbf{x} = R^{\mathrm{T}}Q_A^{\mathrm{T}}\mathbf{b}$, which is equivalent to

$$A^{\mathrm{T}}A\mathbf{x} = A^{\mathrm{T}}\mathbf{b}.$$

Since A is of full column rank, $A^{\mathrm{T}}A$ is symmetric and positive definite. Thus, Cholesky factorization can be used to solve the problem. This is called the method of normal equations.

Since R is upper triangular, we can compute the columns of Q_A by orthogonalizing the columns of A.

- Begin with $\mathbf{q}_1 = \mathbf{a}_1/\|\mathbf{a}_1\|_2$.
- Find $\mathbf{q}_2$: $\mathbf{a}_2 = r_{12}\mathbf{q}_1 + r_{22}\mathbf{q}_2$ and $\mathbf{q}_2^{\mathrm{T}}\mathbf{q}_1 = 0$. Thus $r_{12} = \mathbf{q}_1^{\mathrm{T}}\mathbf{a}_2$, $r_{22} = \|\mathbf{a}_2 - r_{12}\mathbf{q}_1\|_2$ and $\mathbf{q}_2 = (\mathbf{a}_2 - r_{12}\mathbf{q}_1)/r_{22}$.
- The above process can continue.

This is called Gram-Schmidt orthogonalization.

Algorithm. (Classical Gram-Schmidt)

for $i = 1$ to n
    $\mathbf{q}_i = \mathbf{a}_i$;
    for $j = 1$ to $i-1$
        $r_{ji} = \mathbf{q}_j^{\mathrm{T}}\mathbf{a}_i$
    end;
    for $j = 1$ to $i-1$
        $\mathbf{q}_i = \mathbf{q}_i - r_{ji}\mathbf{q}_j$
    end;
    $r_{ii} = \|\mathbf{q}_i\|_2$;
    if $r_{ii} = 0$ quit end;
    $\mathbf{q}_i = \mathbf{q}_i/r_{ii}$
end

This is amazing that the two loops can be combined, resulting in the modified Gram-Schmidt algorithm.

Algorithm. (Modified Gram-Schmidt)

for $i = 1$ to n
    $\mathbf{q}_i = \mathbf{a}_i$;
    for $j = 1$ to $i - 1$
        $r_{ji} = \mathbf{q}_j^{\mathrm{T}} \mathbf{q}_i$
        $\mathbf{q}_i = \mathbf{q}_i - r_{ji} \mathbf{q}_j$
    end;
    $r_{ii} = \|\mathbf{q}_i\|_2$;
    if $r_{ii} = 0$ quit end;
    $\mathbf{q}_i = \mathbf{q}_i / r_{ii}$
end

The classical GS is numerically unstable in floating-point arithmetic when A is ill-conditioned. The modified GS is more stable than the classical GS, but the computed Q may still be far from orthogonal when A is ill-conditioned.

Instead of orthogonalizing the columns of A, we can find a sequence of orthogonal transformations, such as Givens rotations or Householder transformations, to upper triangularize A to obtain the QR factorization:

$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix}.$$

A framework of Solving linear least squares problems

$$A\mathbf{x} \approx \mathbf{b}.$$

Similar to solving linear systems.

- Using orthogonal transformations to triangularize A, simultaneously applying the transformations to $\mathbf{b}$;
- Solving the resulting triangular system.

Perturbation theory for the least squares problem

We define

$$\kappa_2(A) := \frac{\sigma_{\max}}{\sigma_{\min}},$$

the ratio of the largest singular value and the smallest singular value, as the condition number. When A is nonsingular, $\kappa_2(A) = \|A\|_2 \|A^{-1}\|_2$. If

$$\epsilon := \max\left(\frac{\|\delta A\|_2}{\|A\|_2}, \frac{\|\delta \mathbf{b}\|_2}{\|\mathbf{b}\|}\right) < \frac{1}{\kappa_2(A)},$$

then $A + \delta A$ is also of full rank. Thus the least squares problem $\min \|(A + \delta A)\mathbf{x} - (\mathbf{b} + \delta \mathbf{b})\|_2$ has a unique minimum norm solution, denoted by $\tilde{\mathbf{x}}$. Let $\mathbf{r} := \mathbf{b} - A\mathbf{x}$ be the residual, then

$$\frac{\|\tilde{\mathbf{x}} - \mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leq \epsilon \frac{\kappa_2(A)}{1 - \epsilon\, \kappa_2(A)} \left(2 + (\kappa_2(A) + 1) \frac{\|\mathbf{r}\|_2}{\|A\|_2 \|\mathbf{x}\|_2}\right)$$

and

$$\frac{\|\tilde{\mathbf{r}} - \mathbf{r}\|_2}{\|\mathbf{r}\|_2} \leq (1 + 2\epsilon\kappa_2(A)).$$

The above result says:

- If $\|\mathbf{r}\|_2$ is small, then the effective condition number is about $2\kappa_2(A)$.
- If $\|\mathbf{r}\|_2$ is moderately large, then the efficitve condition number can be large: $\kappa_2^2(A)$.
- If the true solution $\mathbf{x}$ is close to zero, then the effective condition number can be unbounded even if $\kappa_2(A)$ is small.

The condition number for solving the normal equations $A^{\mathrm{T}} A\mathbf{x} = A^{\mathrm{T}}\mathbf{b}$ is always $\kappa_2^2(A)$. Moreover, the method of normal equations is not necessarily stable, i.e., the computed solution $\tilde{\mathbf{x}}$ is not generally the solution of $\min \|(A + \delta A)\mathbf{x} - (\mathbf{b} + \delta \mathbf{b})\|_2$ for small δA and $\delta \mathbf{b}$. However, the method of normal equations is the fastest way of solving the least squares problem. It is the method of choice when A is well-conditioned.

What if A is rank deficient? Pivoting technique should be used in QR decomposition. A more stable and accurate but more expensive alternative is the SVD.

Let $A = U\Sigma V^T$ be the SVD of A. When A is rank deficient, $\mathrm{rank}(A) = r < n$, and

$$\Sigma = \mathrm{diag}(\sigma_1, ..., \sigma_n), \quad \sigma_1 \geq ... \geq \sigma_r > \sigma_{r+1} = ... = \sigma_n = 0.$$

The minimal norm solution for the linear least-squares problem $\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_2$ is given by

$$\mathbf{x} = V \begin{bmatrix} \sigma_1^{-1} & & 0 & 0 \\ & \ddots & & \\ 0 & & \sigma_r^{-1} & 0 \\ 0 & & 0 & 0 \end{bmatrix} U^T \mathbf{b}.$$

Total least squares

The least squares problem

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_2$$

can be recast as

$$\min \|\mathbf{r}\|_2, \quad \text{for} \quad (\mathbf{b} + \mathbf{r}) \in \text{range}(A),$$

which allows perturbation in $\mathbf{b}$ only. To generalize it, we allow perturbations in both A and $\mathbf{b}$:

$$\min \|[E \;\; r]\|_F, \quad \text{for} \quad (\mathbf{b} + \mathbf{r}) \in \text{range}(A + E).$$

This is called the total least squares problem. The condition $(\mathbf{b} + \mathbf{r}) \in \text{range}(A+E)$ implies that there exists an $\mathbf{x}$ such that $(A+E)\mathbf{x} = \mathbf{b}+\mathbf{r}$. The $\mathbf{x}$ minimizing $\|[E \;\; r]\|_F$ is the solution of the total least squares problem.

Example. Find a line to fit the following three points:

x	1	2	3
y	2	2	4

In matrix form

$$\begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \approx \begin{bmatrix} 2 \\ 2 \\ 4 \end{bmatrix}.$$

In the ordinary least squares, we assume exact x and perturnbation in y only. Thus we minimize the sum of the squares of the distances along the y direction. The ordinary least squares solution is $a_0 = 0.67$ and $a_1 = 1.0$ and the line equation is $y = 0.67 + x$. In the total least squares, we assume perturbations in both x and y. Thus we minimize the sum of the squares of the shortest distances between the data points and the line.

Under the conditions:

1. $A \in R^{m\times n}$ $(m > n)$, $\mathbf{b} \in R^m$
2. The singular values of $[A\ \mathbf{b}]$ satisfy $\sigma_1 \geq \cdots \geq \sigma_n > \sigma_{n+1}$

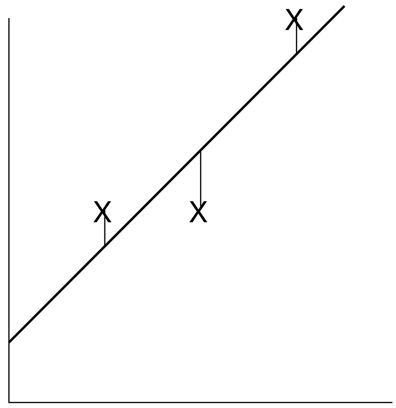
The following algorithm finds the total least squares solution.

Algorithm (Total least squares)

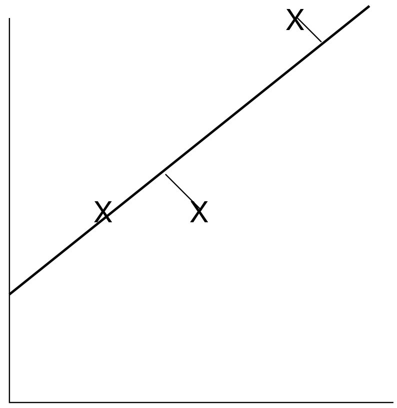
1. SVD decomposition $U^{\mathrm{T}}[A\ \ \mathbf{b}]V = \mathrm{diag}(\sigma_1, ..., \sigma_{n+1})$;
2. If $v_{n+1,n+1} \neq 0$, the total least squares solution $\mathbf{x}$ is given by

$$x_i = -v_{i,n+1}/v_{n+1,n+1}, \quad i = 1, ..., n.$$

In this example, $a_0 = 1.2$ and $a_1 = 0.80$. The line equation is $y = 1.2 + 0.8x$



least squares



total least squares

Nonlinear least squares

Multivariate vector-valued function

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix} \in R^m, \quad \mathbf{x} \in R^n,$$

find the solution of

$$\rho(\mathbf{x}) = \min_{\mathbf{x}} \frac{1}{2} \sum_{i=1}^{m} f_i(\mathbf{x})^2$$

Model fitting problem.

Newton's method

Basic idea: Set the gradient $\nabla\rho(\mathbf{x})$ to zero, then solve $\nabla\rho(\mathbf{x}) = 0$ using Newton's method.

At each step, find the correction $\mathbf{s}_c$ ($\mathbf{x}_+ = \mathbf{x}_c + \mathbf{s}_c$) satisfying

$$\nabla^2\rho(\mathbf{x}_c)\mathbf{s}_c = -\nabla\rho(\mathbf{x}_c).$$

Note. This is Newton's method for solving nonlinear systems.

Gradient

$$\nabla\rho(\mathbf{x}_c) = J(\mathbf{x}_c)^{\mathrm{T}}\mathbf{f}(\mathbf{x}_c)$$

where

$$J(\mathbf{x}_c) = \left[\frac{\partial f_i(\mathbf{x}_c)}{\partial x_j}\right]$$

is the Jacobian. It is expensive to compute the Jacobian and solving systems at each step.

Gauss-Newton method

Observe that

$$\nabla^2\rho(\mathbf{x}_c) = J(\mathbf{x}_c)^{\mathrm{T}}J(\mathbf{x}_c) + \sum_{i=1}^{m} f_i(\mathbf{x}_c)\nabla^2 f_i(\mathbf{x}_c).$$

If $\mathbf{x}_*$ fits the model well ($f_i(\mathbf{x}_*) \approx 0$) and $\mathbf{x}_c$ is close to $\mathbf{x}_*$, then $f_i(\mathbf{x}_c) \approx 0$.

Algorithm (Gauss-Newton)

Evaluate $\mathbf{f}_c = \mathbf{f}(\mathbf{x}_c)$ and $J_c = J(\mathbf{x}_c)$;
Solve $(J_c^T J_c)\mathbf{s}_c = -J_c^{\mathrm{T}}\mathbf{f}_c$ for $\mathbf{s}_c$;
Update $\mathbf{x}_+ = \mathbf{x}_c + \mathbf{s}_c$.

Note. $\mathbf{s}_c$ is the solution to the normal equations for the linear least squares problem:

$$\min_{\mathbf{s}}(\|J_c\mathbf{s} + \mathbf{f}_c\|_2)$$

Reliable methods can be used to solve for $\mathbf{s}_c$.

Remark. Gauss-Newton method works well on small residual problems.

Summary

- Golden section search for minimum.
- Minmax and Chebyshev polynomials.
- Linear least squares: QR decomposition using Givens rotations or Householder transformations. Perturbation theory.
- Nonlinear least squares.

Software packages

IMSL uvmif, uminf, umiah, unlsf

MATLAB fminbnd, fminsearch, fminunc, fmincon, lsqnonlin

NAG e04abf, e04fyf

MINPACK lmdif1

NETLIB varpro, dqed

References

George E. Forsythe, Michael A. Malcolm, and Cleve B. Moler. *Computer Methods for Mathematical Computations*. Prentice-Hall, Inc., 1977.

G.W. Stewart. *Afternotes goes to Graduate School*. Society for Industrial and Applied Mathematics, Philadelphia, 1998.