Lecture 2: Optimization methods

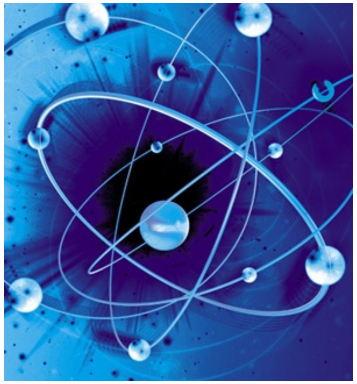
Overview

- Introduction
- Mathematical formulation
- One-dimensional minimization
- Newton's Method
- Conjugate gradient method

Introduction

Optimization is used everywhere in nature:

- Physics
- Chemistry
- Economics
- Engineering
- Etc…



and of course STATISTICS!

Introduction

- Example 1: Industry

How to produce a cylindrical beer can 0.5L so it requires minimum material?

Continuous optimization

Introduction

- Example 2: Economics

Factories F1, F2

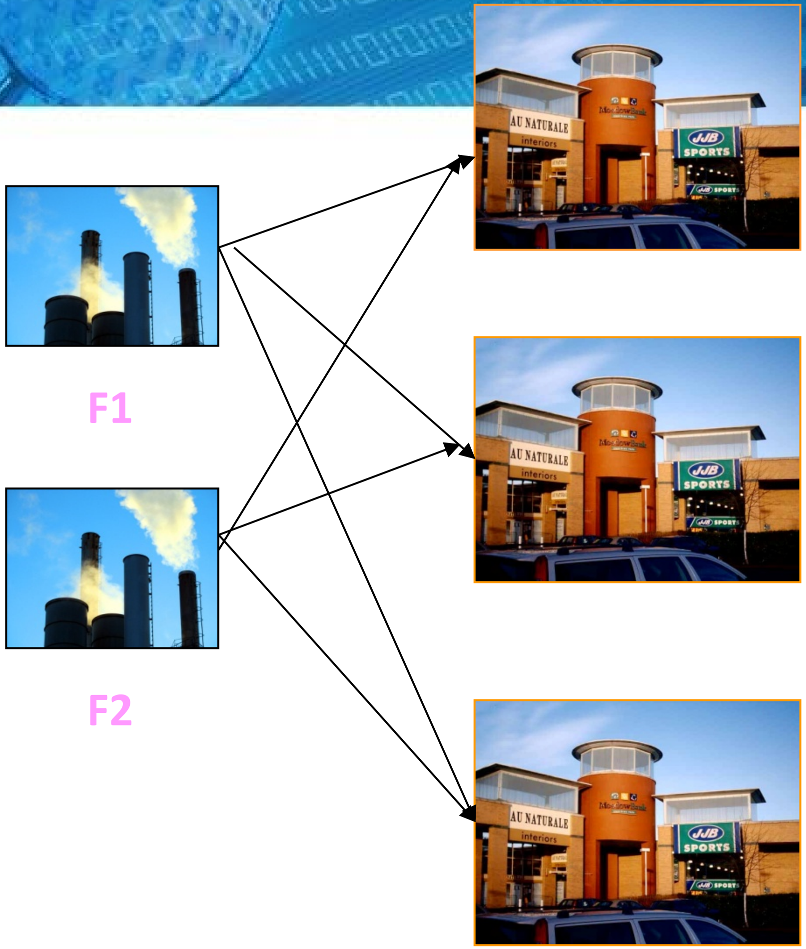
Retail outlets R1, R2, R3

Cost of shipping a product c_{ij}

Production a_i each week

Requirement b_j each week

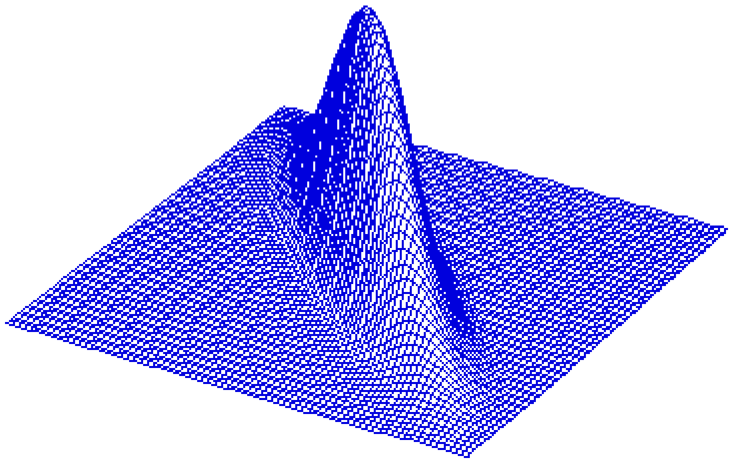
Network flow optimization



Introduction

Example 3: Statistics

Maximize Likelihood $L(X, \theta)$



- Almost all model fitting requires optimization!

Maximum likelihood

Consider a sample (X_1, ..., X_n) which is drawn from a probability distribution $P(X|\Theta)$ where Θ are parameters.

If the Xs are independent with probability density function $P(X_i|\Theta)$ then the joint probability of the whole set is

$$P(X_1,..,X_n|\Theta) = \prod_{i=1}^{n} P(X_i|\Theta)$$

Find the parameters that maximize this function

Mathematical formulation

We need to minimize or maximize

- Objective function $f(x)$ (I - cost, II - profit, III- likelihood)

dependent on

- Parameters or Unknowns x (I-height & diameter, II- supply, III – parameters

Mathematical formulation

- Sometimes we have constraints $c_i(x)$ satisfying equations or inequalities. Formulation:

$$\boxed{\min_{x \in R^n} f(x) \quad subject\ to \quad \begin{matrix} c_i(x) = 0,\ i \in E \\ c_i(x) \geq 0,\ i \in I \end{matrix}}$$

What if:

- Max instead of min
- Constraints are not like these

Mathematical formulation

- Example 1: Constraints – volume=0.5L
- Example 2- cont.

$$\boxed{\begin{aligned} &\min \sum_{ij} c_{ij} x_{ij} \\ &\qquad \sum_{j=1}^{3} x_{ij} \le a_i,\ i=1,2 \\ &s.t. \sum_{i=1}^{2} x_{ij} \ge b_j,\ j=1,2,3 \\ &\qquad\qquad x_{ij} \ge 0 \end{aligned}}$$

- Example 3 – no constraints UNCONSTRAINED MINIMIZATION

Exercise

- Split into groups of three-four and

1. Find an application when optimization is needed (your personal experience, research, university courses)
2. State your problem
    - Objective function
    - Parameters
    - Constraints if any
3. You have **max 10 minutes**

Where we are

Optimization

Unconstrained

~~Constrained~~

Steepest descent

Newton method

Quasi-Newton-Methods

Conjugate gradients

Why different algorithms?

- Speed
- Memory
- Historically

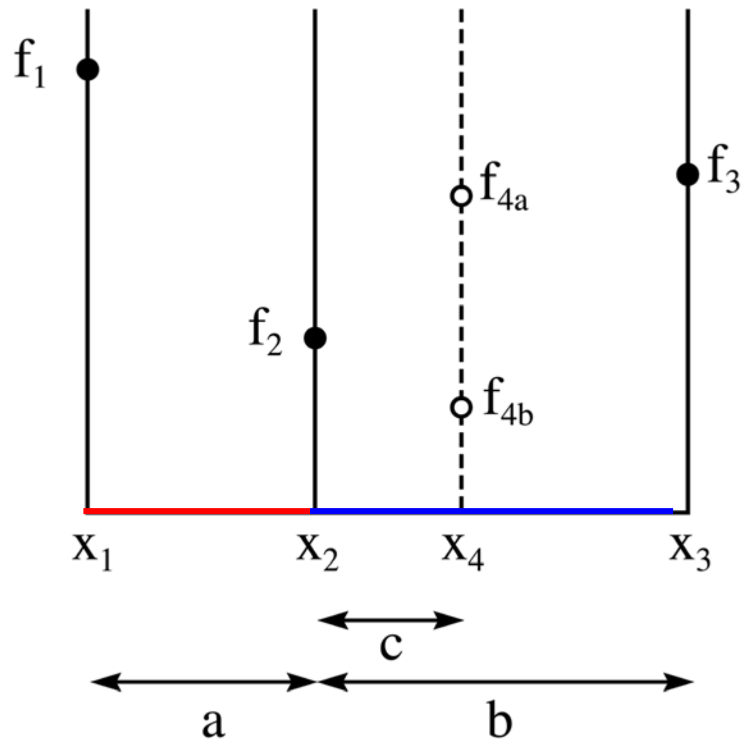
One-dimensional minimization

- One-dimensional minimization=one parameter
- Algorithm Golden Section: finds local minimum on interval [A,B]
- It narrows down the search interval, constant reduction factor $1-\alpha=(\sqrt{5}-1)/2\approx 0.62$

One-dimensional minimization

Golden section

1. Choose interval $[x_1, x_3]$
2. Choose $a=\alpha(x_3 - x_1)$
3. $x_2=x_1+a$, $x_4=x_3-a$
4. If $f_4>f_2$ select RED
5. If $f_4<f_2$ select BLUE
6. Continue with new interval until it is small



Note: f should be unimodal

R: One-dimensional minimization

- Brent’s method – improved golden search

```
optimize(f, interval,...)
```

Multidimensional optimization

The problem:

$$\min_{\mathbf{x}\in R^n} f(\mathbf{x})$$

Gradient

$$\nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \dots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

Hessian

$$\nabla^2 f(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_1} & \dots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ \dots & & \dots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \dots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_n} \end{pmatrix}$$

General methodology:

1. Given starting point x_0, $x=x_0$
2. Choose direction p and step α
3. Move to *x:=x+ α p*
4. Repeat from 2 until convergence

Multidimensional optimization

- How to choose direction leading to function decrease ?

Taylor theorem $$f(x+\alpha p) = f(x) + \alpha p^T \nabla f(x_k) + o(\alpha^2)$$

The minimum is $$\boxed{p = \frac{-\nabla f(x)}{\|\nabla f(x)\|}}$$

Should be minimized

Any direction having $\angle(d, -\nabla f(x)) < \frac{\pi}{2}$ is descent direction

Multidimensional optimization

- How to choose step size α?
  - Find global minimum along direction *p* (expensive)
  - Find a sufficient decrease

BACKTRACKING

Choose $\alpha_0 > 0$, ρ in (0,1), c in (0,1), $\alpha := \alpha_0$

REPEAT until $f(x_k + \alpha p_k) \leq f(x_k) + c\, \alpha \nabla f_k^T (p_k)$

$\alpha := \rho\, \alpha$

END

Newton's method

- In statistics called Newton-Raphson method

General idea:

Quadratic model

$$f(\mathbf{p}) = \frac{1}{2}\mathbf{p}^T A\mathbf{p} + b^T\mathbf{p} + c$$

Minimum

$$\mathbf{p}^* = A^{-1}b$$

- When general function,

Tailor expansion

$$f(x+\alpha p) \approx f(x) + a\nabla f(x)^T p + \frac{\alpha^2}{2} p^T \nabla^2 f(x) p$$

Proceed to next point

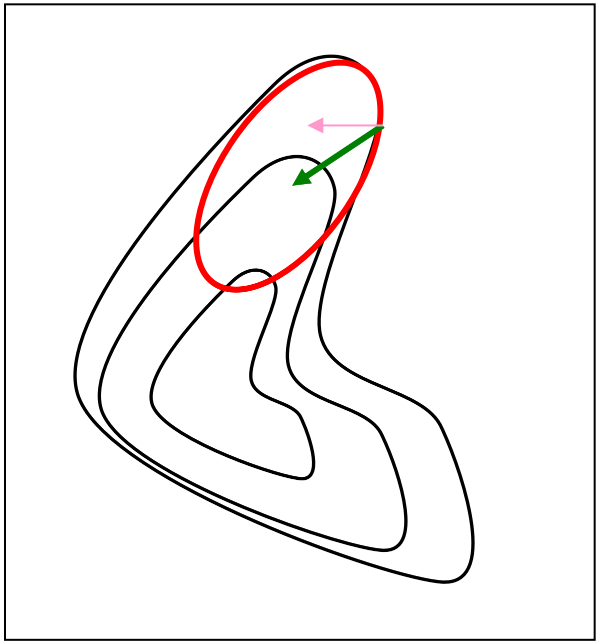
x:=x+αp

$$\boxed{p = -\left(\nabla^2 f(x)\right)^{-1} \nabla f(x)}$$

Newton's method

Illustration:

- Steepest descent
- Newton's direction



Newton’s method

Comments

- Under mild conditions: Converges quickly, especially near optimum
- For *p* to be a descent direction Hessian should be **positive definite** (see why)–strong requirement!
- Can be very expensive to compute reverse of Hessian on each iteration!
- Need to store n*n matrix (Hessian) – memory requirements

Quasi-Newton methods

Idea:

In Newton's method instead computing inverse of Hessian on each step

- Compute approximate Hessian B_k and reverse H_k
- **BFGS**: Using knowledge about H_k, function and gradient in x_k and x_{k+1}, compute H_{k+1}

$$p_k = -H_k \nabla f(x_k)$$

BFGS

How to compute H_{k+1}?

- Quadratic model

$$m_{k+1}(p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T B_{k+1} p$$

should have the same function values and gradients as f(x) in points x_k and x_{k+1}

-> Secant condition

$$\boxed{H_{k+1}\underbrace{(\nabla f_{k+1} - \nabla f_k)}_{y_k} = \underbrace{x_{k+1} - x_k}_{s_k}}$$

BFGS

How to compute H_{k+1}?

- Distance between H_k and H_{k+1} should be minimal

$$\min_{H} \|H - H_k\|$$

$$s.t.\ H = H^T, \text{ secant condition}$$

- Updating formula

$$\boxed{H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k}}$$

BFGS

Comments

- Typically takes more iterations than Newton's method
- Each iteration takes less time (no matrix inversion!)
- Quasi-Newton Methods are particularly good for large-scale problems.
- How to choose initial Hessian?

Conjugate Gradient method

Quadratic function

$$f(x) = \frac{1}{2} x^T A x - b^T x$$

Gradient

$$\nabla f(x) = Ax - b \overset{def}{=} r(x)$$

A- symmetric, positive definite

Def. Directions p and q are conjugate with respect to A if

$$\boxed{p^T A q = 0}$$

Conjugate Gradient method

Conjugate gradient method:

Choose $\boxed{p_{k+1} = -r_{k+1} + \beta_{k+1} p_k}$ $p_0 = -r_0$

p_i should satisfy conjugacy condition, therefore

- $\beta_k = \dfrac{r_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}}$

- Converges in dim(A) steps

Nonlinear CG method

Idea: Consider general f(x) and substitute r_k with ∇f_k

Given x_0, f_0, ∇f_0

$p_0 := -\nabla f_0$

while $\nabla f_k != 0$

compute α_k, $x_{k+1} = x_k + \alpha_k p_k$

$\beta_{k+1} = (\nabla f^T_{k+1} \nabla f_{k+1}) / (\nabla f^T_k \nabla f_k)$

$p_{k+1} = -\nabla f_{k+1} + \beta_{k+1} p_k$

$k = k+1$

end

Nonlinear CG method

- Converges to local minimum
- Much faster than steepest descent in general
- Slower than Newton and Quasi-Newton but much less memory

R: Multidimensional optimization

- Quasi-Newton and CG incorporated in one procedure

```
optim(par, fn, gr=Null, method, ...)
```

- Look also

```
nls(...)
```