

Problem:

we wish to compute

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

where A is Positive semi definite

1) this is a local model of some function

2) solve when $Ax = b$
(Newton, linear alg)

We assume that solving $Ax = b$ is hard
(eg can't store A)

Conjugate directions

a set of dir's $p_0 \dots p_n$ is

conjugate if

$$p_i^T A p_j = 0 \quad i \neq j$$

and $p_i^T A p_i \neq 0$ ~~otherwise~~

Notice: It is quite straightforward to make conjugate dir's

$$p_0 = e_0$$

$$p_1 = e_1 - \frac{(e_1^T A p_0)}{(p_0^T A p_0)} p_0$$

$$p_i = e_i - \sum_{j < i} \frac{(e_i^T A p_j)}{(p_j^T A p_j)} p_j$$

(3)

Notice

- there are many sets of conj. directions (~~start~~^{run} alg. above w/ a different set of basis vectors)

- each set contains N c.d.'s if A is positive definite. You can get this by diagonalizing A

Conjugate directions simplify problem

write $x = \sum_i \alpha_i p_i$ \swarrow conj. wrt A .

$$\text{then } f(x) = \sum_i \alpha_i^2 p_i^T A p_i - \sum_i \alpha_i b^T p_i$$

\uparrow no $\alpha_i \alpha_j$ terms

because $p_i^T A p_j = 0$ $i \neq j$

so we can minimize the directions independently.

Algorithm skeleton

$$x_0 = 0$$

→ (get conj dir p_i)

$$x_{i+1} = x_i + \alpha_i p_i$$

chosen to minimize in p_i

- Need go round only N times

because the p_i form a basis

$$\text{span}\{p_0, \dots, p_N\} = \text{span}\{e_0 \dots e_N\}$$

↑
the basis you chose to form C.D.s in.

α_i is easy to get

$$\underset{x_i}{\text{argmin}} f(x_i + \alpha_i p_i) = \underset{\alpha_i}{\text{argmin}} \frac{x_i^T A x_i}{2} + \alpha_i \frac{p_i^T A p_i}{2} - b^T x_i - \alpha_i b^T p_i$$

BUT $x_i = [\text{linear comb of } p_0 \dots p_{i-1}]$

$$\text{so } x_i^T A p_i = 0$$

so we must minimize

$$\alpha_i \frac{p_i^T A p_i}{2} - \alpha_i b^T p_i + [\text{stuff that doesn't depend on } \alpha_i]$$

$$\text{so } \alpha_i = \frac{b^T p_i}{p_i^T A p_i}$$

Notice if A is P.D., then $p_i^T A p_i > 0$
 - this will come in useful.

Algorithm

$$x_0 = 0$$

→ ~~x_i~~ (get new conj. dir) ←

$$\alpha_i = \frac{b^T p_i}{p_i^T A p_i}$$

$$x_{i+1} = x_i + \alpha_i p_i$$

This is the bad bit

- we currently need to store all old conj. directions
- we must fix this.

Write

$$r_i = Ax_i - b \quad \leftarrow \text{residual}$$

Start at $x_0 = 0$

$$r_0 = -b$$

then assume we have $p_0 \dots p_k$

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

$$\alpha_k = \frac{b^T p_k}{p_k^T A p_k}$$

Notice:

$$r_k^T p_k = \underbrace{x_k^T A p_k}_{\uparrow} - b^T p_k$$

but $x_k \in \text{span}\{p_0 \dots p_{k-1}\}$

$$\text{so } -r_k^T p_k = b^T p_k$$

$$\text{so } \alpha_k = \frac{-r_k^T p_k}{p_k^T A p_k}$$

Now

$$\begin{aligned} r_{k+1} &= Ax_{k+1} - b \\ &= A(x_k + \alpha p_k) - b \\ &= r_k + \alpha_k A p_k \end{aligned}$$

So that:

$$\begin{aligned} r_{k+1}^T p_k &= r_k^T p_k - \frac{r_k^T p_k}{p_k^T A p_k} \cdot (p_k^T A p_k) \\ &= 0 \end{aligned}$$

i.e. r_{k+1} is perp to p_k .

which means we can cook up
a new conj direction out of
 r_{k+1} and p_k

and make it conj. to p_k

9

our new dirn

$$p_{k+1} = -r_{k+1} + \beta_k p_k$$

now, we want

$$p_{k+1}^T A p_k = 0$$

$$\text{so } r_{k+1}^T A p_k = \beta_k p_k^T A p_k$$

i.e.

$$\beta_k = \frac{r_{k+1}^T A p_k}{p_k^T A p_k}$$

This direction is special in
important ways

IF we start with

$$p_0 = -r_0$$

~~It~~ turns

I have already shown that this construction yields

$$p_{k+1}^T A p_k = 0$$

BUT we can also show that

$$p_{k+1}^T A p_j = 0 \quad 0 \leq j \leq k-1.$$

We do this by showing

(1) that

$$M_{k+1}^T p_j = 0, \quad 0 \leq j \leq k.$$

(2)

$$\begin{aligned} & \text{span} \{ r_0, r_1, r_2, \dots, r_k \} \\ & \parallel \\ & \text{span} \{ r_0, A p_0, A p_1, \dots, A p_{k-1} \} \\ & \parallel \\ & \text{span} \{ p_0, p_1, p_2, \dots, p_k \}. \end{aligned}$$

(4)

Now assume (1) and (2)

then

$$p_{k+1}^T A p_j = \begin{cases} 0 & \text{for } j = k \\ -r_{k+1}^T A p_j + \beta_k (p_k^T A p_j) & j < k. \end{cases}$$

Now we apply an induction.

(1) we have $p_2^T A p_1 = 0$

(2) assume $p_k^T A p_j = 0$, $j < k-1$
(we're ok for $j = k-1$)

(3) then

$$p_{k+1}^T A p_j = -r_{k+1}^T A p_j$$

BUT

so $r_{k+1}^T p_j = 0$ by (1)

$r_{k+1} \perp \text{span}\{p_0, \dots, p_k\}$
at right angles to

~~So~~ But

$$\text{span} \{ p_0, \dots, p_k \}$$

$$\parallel \text{span} \{ r_0, Ap_0, \dots, Ap_{k-1} \}$$

$$\text{So } M_{k+1} \perp \text{span} \{ r_0, Ap_0, \dots, Ap_{k-1} \}$$

$$\text{So } M_{k+1}^T Ap_j = 0, \quad j \leq k-1$$

$$\underline{\text{So}} \quad p_{k+1} Ap_j = 0 \quad \text{and we are done}$$

Must now prove ① and ②

$$\textcircled{1} \quad M_{k+1}^T p_k = 0$$

Already established this

must show

$$M_{k+1}^T p_j = 0, \quad j < k.$$

Notice

$$M_{k+1} = M_k + \alpha_k A_k p_k$$

$$\text{so } M_{k+1} = M_{j+1} + \left[\text{a sum of } \alpha A p \text{ terms} \right]$$

for $j+1 \dots k$

$$\text{so } M_{k+1}^T p_j = \underbrace{M_{j+1}^T p_j}_{=0} + \left[\begin{array}{c} \phantom{M_{j+1}^T p_j} \\ \phantom{M_{j+1}^T p_j} \\ \phantom{M_{j+1}^T p_j} \end{array} \right]^T p_j$$

$$\neq 0$$

from above

by conjugacy

so we are done on $\textcircled{1}$

We must deal with claim 2.

$$V_k^P = \text{Span}\{p_0, \dots, p_k\}$$

$$V_k^r = \text{Span}\{r_0, \dots, r_k\}$$

$$V_k^A = \text{Span}\{r_0, Ap_0, \dots, Ap_{k-1}\}$$

Example

$$V_0^P = \text{Span}\{p_0\}; \quad V_0^r = \text{Span}\{r_0\}; \quad V_0^A = \text{Span}\{r_0\}$$

but $p_0 = -r_0$

$$V_1^P = \text{Span}\{p_0, p_1\} = \text{Span}\{p_0, r_1 + \beta p_0\}$$

$$V_1^r = \text{Span}\{r_0, r_1\} = \text{Span}\{r_0, r_0 + \alpha Ap_0\}$$

$$V_1^A = \text{Span}\{r_0, Ap_0\} =$$

Now $V_1^r = V_1^P$ by inspection (Notice β has nothing to do with it)

$V_1^r = V_1^A$ by inspection

AND THAT IS THE FORM OF THE INDUCTION

We have

start

$$\begin{aligned}
 x_0 &= 0 \\
 r_0 &= -b \\
 p_0 &= -r_0
 \end{aligned}$$

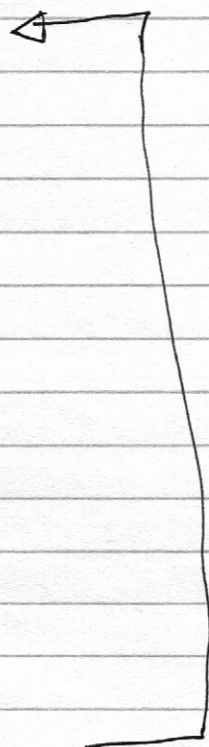
$$\alpha_k = \frac{-r_k^T p_k}{p_k^T A p_k}$$

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

$$r_{k+1} = r_k + \alpha_k A p_k$$

$$\beta_k = \frac{r_{k+1}^T A p_k}{p_k^T A p_k}$$

$$p_{k+1} = -r_{k+1} + \beta_k p_k$$



Terminating if

- r_{k+1} is small "enough"
- or k iterations

A little notation

we computed β_k as

$$\beta_k = \frac{\Gamma_{k+1}^T A p_k}{p_k^T A p_k}$$

But we could use other expressions
we require

$$p_{k+1}^T A p_k = 0$$

Notice: ~~for~~ $\mu_{k+1} - \Gamma_k = \alpha A p_k$

so we could require

$$(\Gamma_{k+1} - \Gamma_k)^T p_{k+1} = 0$$

This gets us

$$\beta_k = (\Gamma_{k+1} - \Gamma_k)$$

$$(\Gamma_{k+1} - \Gamma_k)^T p_{k+1} = -(\Gamma_{k+1} - \Gamma_k)^T \Gamma_k + \beta_k (\Gamma_{k+1} - \Gamma_k)^T p_k$$

But

$$\Gamma_{k+1}^T p_k = 0$$

AND

$$\begin{aligned} \Gamma_k^T p_k &= \Gamma_k^T \left(\Gamma_k - \Gamma_k + \beta_{k-1} p_{k-1} \right) \\ &= -\Gamma_k^T \Gamma_k \end{aligned}$$

So that :

$$\beta_k = \frac{(\Gamma_{k+1} - \Gamma_k)^T \Gamma_k}{\Gamma_k^T \Gamma_k}$$

This comes in useful in non-linear cases

We originally worked with

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

But imagine we have a function $g(x)$ that is modelled as

$$g(x) \approx g_0 + \nabla g|_0^T x + \frac{x^T H x}{2}$$

We can build a powerful analogy

f side

A

b

$$r_k = Ax + b$$

α obtained
by exact
min

g side

H

$-\nabla g|_0$

$$\nabla g \approx \nabla g|_0 + Hx$$

α - linesearch

This isn't perfect, because as x changes, H will change

- we cannot expect to take N steps

- we may need to restart

This leads to 3 algorithms

(Polak-Ribiere; Fletcher-Reeves; ?)

I describe only Polak-Ribiere:

- start at x_0

$$r_0 = -\nabla g \Big|_{x=x_0}$$

$$p_0 = -r_0$$

$$\alpha_k = \text{linesearch} \left(g(x_k + \alpha p_k) \right)$$

$$x_{k+1} = x_k + \alpha p_k$$

$$r_{k+1} = -\nabla g \Big|_{x=x_{k+1}}$$

$$\beta_k = \frac{(r_{k+1} - r_k)^T r_k}{r_k^T r_k}$$

$$p_{k+1} = -r_{k+1} + \beta_k p_k$$

But we may need to reset

options

- every N^{th} iteration,

restart

(i.e. $x_0 \leftarrow \text{current } x_{k+1}$
and go again)

- OR if $\beta_k < \text{small threshold}$

$$\beta_k = \min \beta_k < 0$$

(small threshold is
often 0.2)

NOTICE:

- Method gets second order info, BUT we do not need to compute or store H !