

# Operations Research Techniques and Algorithms (620-361)

Dr Yao-ban Chan

y.chan@ms.unimelb.edu.au

Telephone: 8344 9073

Office: Room 198, Richard Berry Building

Christina Burt

c.burt@ms.unimelb.edu.au

Telephone: 8344 1797

Office: 139 Barry St

Wednesday 12th March, 2008

# Today's Lecture

Introduction

Unimodal 1-D unconstrained optimisation  
Minimum of a half-open interval  
Armijo-Goldstein and Wolff conditions

Unimodal  $n$ -D unconstrained optimisation  
Review of basic definitions and notation

## Finding the Minimum on a Half-Open Interval

In optimisation algorithms for minimising functions on  $\mathbb{R}^N$ , we frequently find ourselves in the situation where we want to find the minimum of a continuous and unimodal function on the interval  $[0, \infty)$ .

For example, many of these algorithms work by choosing a direction and then minimising the function along the chosen direction. If we consider the direction as parametrised by the single variable  $t$ , then this requires us to minimise a function of  $t$  in the interval  $[0, \infty)$ .

Apart from Newton's Method, this situation is not quite the same as the ones that we have been talking about, where we started out with the assumption that the minimum occurred somewhere in a closed interval  $[a, b]$ .

Our first task in attacking this new class of problem is to find a number  $b$ , such that we know the minimum of our function  $f$  lies in the interval  $[0, b]$ .

One obvious way to do this is to increase  $t$  by some increment, say  $T$ , until we find a point of the form  $kT$ , where  $k$  is an integer, such that  $f(kT) \geq f(k-1)T$ .

Because  $f$  is assumed to be continuous and unimodal, we would then know that the minimum lies in the interval  $[0, kT]$  and we are then in a position to use one of the algorithms that we discussed above.

## Procedure for Finding an Upper Bound on the Location of the Minimum

For a continuous, unimodal function  $f$  on  $[0, \infty)$ , to find a point  $b$  such that the minimum  $x_{min} < b$ .

1. Choose some increment value  $T$ .

$$k = 1$$

$$p = 0$$

$$q = T$$

Calculate  $f(p)$  and  $f(q)$ . If  $f(p) \leq f(q)$ , then stop. Else...

2. Set  $k = k + 1$ .

$$p = q$$

$$q = kT$$

Calculate  $f(q)$ .

Repeat until  $f(p) \leq f(q)$ .

However, there is a problem with this procedure and that is in choosing a suitable increment size  $T$ . Since we know nothing about the function  $f$ , if we choose  $T$  to be too small, we will be wasting a lot of time and resources calculating function values unnecessarily.

On the other hand, if we take  $T$  to be too big, we may end up with an interval  $[0, b]$  which is much bigger than necessary and so we will have to make many more  $f$ -calculations than necessary to reduce the size of this interval.

The answer to this problem is to use a variable increment size. Usually we start out with a small increment size and increase it exponentially, for example by doubling it. We thus get the following algorithm:

## An Improved Procedure for Finding an Upper Bound on the Location of the Minimum

For a continuous, unimodal function  $f$  on  $[0, \infty)$ , to find a point  $b$  such that the minimum  $x_{min} < b$ .

1. Choose some small initial increment value  $T$ .

$$k = 1$$

$$p = 0$$

$$q = T$$

Calculate  $f(p)$  and  $f(q)$ . If  $f(p) \leq f(q)$ , then stop. Else...

2. Set  $k = k + 1$ .

$$\begin{aligned}p &= q \\q &= p + 2^{k-1}T\end{aligned}$$

Calculate  $f(q)$ .

Repeat until  $f(p) \leq f(q)$ .

The improved algorithm has the flexibility to “discover” the appropriate scale for the function  $f$ .

It actually does better than just find an upper bound for the minimum of  $f$ . It will stop at a point  $b$  which is of the form

$$b = T \sum_{k=0}^n 2^k$$

Since  $f(p) > f(q)$  at the previous iteration, we know that  $x_{min}$  lies in the interval  $[a, b]$  where

$$a = T \sum_{k=0}^{n-2} 2^k.$$

We are then in a position to apply any of the methods that we discussed earlier (eg: Fibonacci, Golden Section) to reduce the width of the interval in which we know the minimum lies.

A slight modification of the above method will also work if we know only that  $x_{min}$  lies in  $\mathcal{R}$ .

We evaluate  $f$  at two points  $a$  and  $a + T$ . If  $f(a) > f(a + T)$ , then we know  $x_{min} > a$  and we can apply the above algorithm starting at  $a$  instead of at 0.

If  $f(a) \leq f(a + T)$ , then we know  $x_{min} < a + T$  and we apply the above algorithm in the negative direction starting at  $a + T$ .

# Today's Lecture

Introduction

Unimodal 1-D unconstrained optimisation

Minimum of a half-open interval

Armijo-Goldstein and Wolff conditions

Unimodal  $n$ -D unconstrained optimisation

Review of basic definitions and notation

## The Armijo-Goldstein and Wolff conditions

Single-variable optimisation procedures are often used as components of algorithms for minimising functions on  $\mathbb{R}^N$ . These work by choosing a descent direction and moving along that descent direction to find a point where the function value is lower than it was before.

The minimum value of the function restricted along the descent direction is not (in general) the minimum of the function over  $\mathbb{R}^N$ .

For this reason, it often does not make a lot of sense to spend a lot of time finding the minimum along any given descent direction. Rather, it can be enough to find a point that is a significant improvement on the one we had before.

One way of expressing what is meant by “significant improvement” is given by the Armijo-Goldstein and Wolff conditions.

Let  $\sigma \in (0, 1)$ . For a unimodal, continuous and differentiable function  $f$  on  $[0, \infty)$ , we say that the step size  $t$  satisfies the Armijo-Goldstein condition with weight  $\sigma$  if

$$f(t) \leq f(0) + t\sigma f'(0). \quad (1)$$

Remembering that  $f'(0)$  will be negative (unless the minimum occurs at  $t = 0$ ), the Armijo-Goldstein condition ensures that the step size  $t$  cannot be too large.

Now let's consider the following example. Suppose  $f(x) = (x - 3)^2$  and at the  $k$ th iteration of our search we choose a step size of  $t_k = 1/2^k$  in the positive direction. Then after  $N$  steps, we will be at the point

$$\begin{aligned}\sum_{k=0}^N 1/2^k &= \frac{1 - (1/2)^{N+1}}{1 - 1/2} \\ &= 2 - (1/2)^N\end{aligned}$$

In particular after “infinitely many” steps, we will still only be at the point  $x = 2$ , which is not the minimum of the function.

It is not too hard to show that despite this failure, the stepsize scheme  $t_k = 1/2^k$  satisfies the A-G condition for  $f(x)$ .

What has gone wrong?

The problem is the opposite to the one that the Armijo-Goldstein condition addresses. The step size has become too small.

To get around this we introduce a new condition - the Wolff condition.

For a unimodal, continuous and differentiable function  $f$  on  $[0, \infty)$ , we say that the step size  $t$  satisfies the Wolff condition with weight  $\mu$  if

$$f'(t) \geq \mu f'(0) \quad (2)$$

where  $\mu \in [\sigma, 1)$ .

To understand what the Wolff condition means, observe that since  $f'(0) < 0$  and  $\mu \in (0, 1)$ , then

$$f'(0) < \mu f'(0) < 0$$

Also

$$f'(t) \rightarrow f'(0)$$

as  $t \rightarrow 0$  so, by making sure that  $f'(t) \geq \mu f'(0)$ , the Wolff condition forces  $t$  not be too close to 0.

That is, the step size has to be “sufficiently big”.

The following line search procedure starts out by reducing a large step size so that it becomes small enough to satisfy the Armijo-Goldstein condition and then increasing a small step size so that it becomes large enough to satisfy the Wolff condition.

## A Procedure to Find a Step Size that Satisfies The Armijo-Goldstein and Wolff Conditions

For a differentiable, unimodal function  $f$  on  $[0, \infty)$ . Input an initial step size  $T$ , a number  $\sigma \in (0, 1)$  and a number  $\mu \in [\sigma, 1)$ .

1.

$$t_{lo} = 0$$

$$t_{hi} = \infty$$

$$t = T$$

2. If  $f(t) > f(0) + t\sigma f'(0)$ , then

$$\begin{aligned}t_{hi} &= t \\t &= 1/2(t_{lo} + t)\end{aligned}$$

Else if  $f'(t) < \mu f'(0)$ , then

$$\begin{aligned}t_{lo} &= t \\t_{hi} &= \begin{cases} 1/2(t_{lo} + t_{hi}) & \text{if } t_{hi} < \infty \\ 2t & \text{otherwise} \end{cases}\end{aligned}$$

Repeat until  $f(t) \leq f(0) + t\sigma f'(0)$  and  $f'(t) \geq \mu f'(0)$ .

The line search procedure is justified by the following proposition.

*Proposition:*

Let  $0 < \sigma \leq \mu < 1$ . The above line search procedure either finds a point such that  $f(t) \leq f(0) + t\sigma f'(0)$  and  $f'(t) \geq \mu f'(0)$  in finitely many steps or it produces a sequence of  $t$ s, say  $\{t_k\}$  such that  $f(t_k) \rightarrow -\infty$  as  $k \rightarrow \infty$ .

This result says that either the line search procedure works as intended, or  $f$  is unbounded below; in the latter situation, minimizing  $f$  doesn't make sense.

# Today's Lecture

Introduction

Unimodal 1-D unconstrained optimisation  
Minimum of a half-open interval  
Armijo-Goldstein and Wolff conditions

Unimodal  $n$ -D unconstrained optimisation  
Review of basic definitions and notation

**Review of basic definitions and notation** A vector  $v \in \mathbb{R}^n$  is a column vector consisting of  $n$  components (numbers):

$$v = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix},$$

sometimes written  $v = (v_1, \dots, v_n)$ .

This is to be distinguished from the row vector

$$v^T = [v_1 \ \dots \ v_n].$$

The inner or dot product of  $u, v \in \mathbb{R}^n$  is

$$\langle u, v \rangle = u \cdot v = \sum_{i=1}^n u_i v_i.$$

We also write  $u^T v$  for the inner product.

If  $\theta$  is the angle between  $u$  and  $v$  then

$$u \cdot v = \cos(\theta) \|u\| \|v\|.$$

where for  $u \in \mathbb{R}^n$ ,

$$\|u\| = \sqrt{u \cdot u} = \left( \sum_i u_i^2 \right)^{1/2}.$$

The vector space of  $m \times n$  matrices is written  $\mathfrak{R}^{m \times n}$ .

Let  $B \in \mathfrak{R}^{n \times n}$ , then  $B_{ij}$  denotes the component of  $B$  in row  $i$  and column  $j$ .

$B$  is *symmetric* if  $B = B^T$  (transpose of  $B$ ).

$B$  is *positive definite* if  $u^T B u > 0$  for each nonzero vector  $u \in \mathfrak{R}^n$ .

$B$  is *positive semi-definite* if  $u^T B u \geq 0$  for each nonzero  $u \in \mathfrak{R}^n$ .

A theorem from linear algebra: If  $B$  is symmetric then it is positive (semi-)definite if and only if each of its eigenvalues is positive (respectively nonnegative). This fact gives us an easier way of checking the positive definiteness or semidefiniteness of symmetric matrices.

## The gradient vector

The gradient vector of a function  $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ , is defined as the column vector

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

For a given point  $y \in \mathbb{R}^n$ , it can be shown that the gradient vector at that point,  $\nabla f(y)$ , is a vector normal (at right angles) to the tangent plane of the curve defined by the set of points  $\{x \in \mathbb{R}^n : f(x) = f(y)\}$ , at the point  $y$  (see 2nd year vector calculus).

## The directional derivative

The directional derivative of  $f$  at  $x$  in the direction  $d$  is defined as

$$f'(x; d) \stackrel{\text{def}}{=} \lim_{t \rightarrow 0} \frac{f(x + td) - f(x)}{t}$$

This is essentially the one-dimensional derivative of the slice of  $f$  along the (arbitrary) direction  $d$ . To say that  $f$  is differentiable at  $x$ , we require that directional derivatives exist for all directions  $d$ .

It is not difficult to show that

$$f'(x; d) = \nabla f(x)^T d.$$

We say  $f$  is  $C^1$  or continuously differentiable if it is differentiable and the gradient function  $\nabla f$  is continuous.

## The Hessian matrix

The second derivate matrix  $\nabla^2 f(x)$  is called the *Hessian* of  $f$  at  $x$ .

It can be written using partial derivatives, namely the component in row  $i$  and column  $j$  of  $\nabla^2 f(x)$  is

$$\nabla^2 f(x)_{ij} = \frac{\partial}{\partial x_j} \frac{\partial f}{\partial x_i} = \frac{\partial^2 f(x)}{\partial x_j \partial x_i}. \quad (3)$$

We say  $f$  is  $C^2$  (twice continuously differentiable) if  $f$  is twice differentiable and the Hessian function  $\nabla^2 f$  is continuous.

## Taylor series approximations

Consider a small variation  $\Delta x$  from a given vector  $x^*$ . Using Taylor's theorem (see notes), the second order approximation of  $f$  at  $x^* + \Delta x$  is given by

$$f(x^* + \Delta x) \approx f(x^*) + \nabla f(x^*)^T \Delta x + \frac{1}{2} \Delta x^T \nabla^2 f(x^*) \Delta x$$

Compare with the second order Taylor series approximation of a function of one variable

$$f(x^* + \delta x) \approx f(x^*) + \delta x f'(x^*) + \frac{(\delta x)^2}{2} f''(x^*)$$