1 Quasi-Newton methods

Last time we said that conjugate direction methods have cost slightly larger than gradient descent, but should converge faster. They are in this sense intermediate between gradient descent and Newton method.

Quasi-Newton methods were invented earlier than nonquadratic conjugate gradient method. Historically they were the first approach to construct method which tried to converge faster than gradient descent but have lower per iteration cost than Newton method. They try to get good convergence by approximating second derivative using differences of gradients.

Core idea of quasi-Newton methods is to use descent

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

in the search direction d_i given by

$$d_i = -S_i \nabla f(x_i).$$

where S_i is a strictly positive definite matrix. When $S_i = I$, this gives gradient descent. When $S_i = (\nabla^2 f(x_i))^{-1}$, this gives Newton method. Fixed S_i gives preconditioned gradient descent. When $S_i = (\nabla^2 f(x_0))^{-1}$ we get modified Newton method. In general, when S_i approximates $(\nabla^2 f)^{-1}$, then we expect better convergence.

Another view is that we use S_i^{-1} as a new metric. Therefore the first name used for such methods was *variable metric methods*.

Viewing quasi-Newton methods as gradient descent with new metric we can get convergence estimate. First, since d_i are descent direction we see that quasi-Newton methods are descent methods, so values of goal function are non-increasing. Second, write $S_i = Q_i^T Q_i$ and $A(x) = Q_i^T \nabla^2 f(x) Q_i$. Now using $\langle S_i^{-1}x, y\rangle = \langle Q_i^{-1}x, Q_i^{-1}y\rangle$ as a scalar product we see that $A(x)$ gives Hessian matrix corresponding to this scalar product. So speed of convergence depends on conditioning of $A(x)$. In particular, when $A(x)$ is well conditioned we get convergence to stationary point.

More precisely we have the following:

Lemma 1.1 *Assume that $mI \leq A(x) \leq MI$, and x_{i+1} uses step $\frac{2}{M+m}$ (or exact line search), then*

$$f(x_{i+1}) - f(x_\infty) \leq C(f(x_i) - f(x_\infty))$$

where $C = 1 - \frac{m}{M}$.

Note: this is lemma from Lecture 5 applied to metric above. Using inexact line search we get similar result with slightly larger C.

Note: We can compute m and M from eigenvalues of $S_i^{-1}\nabla^2 f(x)$. In particular m and M depend only on S_i and f and are independent of Q_i.

How to find reasonable S_k? Write $p_i = x_{i+1} - x_i$, $g_i = \nabla f(x_i)$, $q_i = g_{i+1} - g_i$. For quadratic f we have

$$q_i = \nabla^2 f(x_i) p_i.$$

So reasonable condition is

$$S_{i+1}q_i = p_i.$$

Equivalently when $B_i = S_i^{-1}$:

$$q_i = B_{i+1}p_i.$$

Those equations are called quasi-Newton equations (or secant equations).

Lemma 1.2 *Let A be positive definite matrix and $f(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle$ be quadratic function. Quasi-Newton method using exact line search with S_i satisfying quasi-Newton equations satisfies*

$$\langle Ad_{i+1}, d_i\rangle = 0.$$

Proof: For quadratic f we have $q_i = Ap_i$. By quasi-Newton equations we have

$$S_{i+1}Ap_i = S_{i+1}q_i = p_i.$$

Since line search is exact we have $\langle\nabla f(x_{i+1}), d_i\rangle = 0$.

We have

$$p_{i+1} = x_{i+2} - x_{i+1} = \alpha_{i+1}d_{i+1}$$

and $d_{i+1} = -S_{i+1}\nabla f(x_{i+1})$ so

$$\langle Ap_{i+1}, p_i\rangle = -\alpha_{i+1}\langle AS_{i+1}\nabla f(x_{i+1}), p_i\rangle = -\alpha_{i+1}\langle\nabla f(x_{i+1}), S_{i+1}Ap_i\rangle$$

$$= -\alpha_{i+1}\langle\nabla f(x_{i+1}), p_i\rangle = \alpha_{i+1}\alpha_i\langle\nabla f(x_{i+1}), d_i\rangle = 0.$$

In other words

$$\langle Ad_{i+1}, d_i\rangle = 0.$$

□

Quasi-Newton equations specify S_i only in single direction, so there is infinite number of solutions. Other reasonable condition is that S_{i+1} should be close to S_i. One meaning of close is to require $U_i = S_{i+1} - S_i$ to be of low rank. It is possible to find symmetric S_{i+1} such that U_i is of rank 1. However, such rule may produce S_{i+1} which is not positive definite and may lead to numerical difficulties. Next level of complexity is rank 2 update: we require update U_i to be of rank at most 2.

Quasi-Newton equations give

$$p_i = S_{i+1}q_i = U_iq_i + S_iq_i$$

so

$$U_iq_i = p_i - S_iq_i$$

That again admits infinite number of solutions.

Quasi-Newton equations involve $S_i q_i$ and p_i. In "general position" $S_i q_i$ and p_i are linearly independent and it is natural to request that update acts only in this plane and maps orthogonal complement to zero. Still admits one dimensional family of solutions. Simplest possibility is

$$U_k(x) = aS_i q_i \langle S_i q_i, x\rangle + bp_i\langle p_i, x\rangle$$

where a and b are numeric parameters. Then

$$p_i - S_i q_i = aS_i q_i\langle S_i q_i, q_i\rangle + bp_i\langle p_i, q_i\rangle$$

so $a = \frac{-1}{\langle S_i q_i, q_i\rangle}$, $b = \frac{1}{\langle p_i, q_i\rangle}$ and

$$S_{i+1}x = S_i x - \frac{\langle S_i q_i, x\rangle}{\langle S_i q_i, q_i\rangle} S_i q_i + \frac{\langle p_i, x\rangle}{\langle p_i, q_i\rangle} p_i.$$

This is called Davidon-Fletcher-Powell update or in short DFP update.

In quasi-Newton equations S_i and B_i play symmetric role, that is

$$q_i = B_{i+1} p_i.$$

so alternatively to DFP update we can request update to B_i build from q_i and $B_i p_i$. This leads to formula invented independently by Broyden, Fletcher, Goldfarb and Shanno or in short BFGS update:

$$B_{i+1}x = B_i x - \frac{\langle B_i p_i, x\rangle}{\langle B_i p_i, p_i\rangle} B_i p_i + \frac{\langle q_i, x\rangle}{\langle p_i, q_i\rangle} q_i.$$

Clearly both DFP and BFGS updates lead to symmetric matrices. If $\langle p_i, q_i\rangle > 0$, then both DFP and BFGS updates lead to positive definite matrices. Namely, when $\langle p_i, q_i\rangle > 0$ then the third term in both equations is positive definite.

One may worry that the second term is clearly negative. However, for DFP update looking at sum of first two terms we get

$$\left\langle S_i x - \frac{(S_i q_i, x)}{(S_i q_i, q_i)} S_i q_i, x \right\rangle = \langle S_i x, x\rangle - \frac{\langle S_i q_i, x\rangle^2}{\langle S_i q_i, q_i\rangle}$$

$$= \frac{1}{\langle S_i q_i, q_i\rangle}\left(\langle S_i q_i, q_i\rangle\langle S_i x, x\rangle - \langle S_i q_i, x\rangle^2\right).$$

If S_i is positive definite, then term in parentheses is nonnegative due to Schwartz inequality for the scalar product $\langle S_i x, x\rangle$. Moreover, when S_i is strictly positive definite, then we get zero only for multiples of q_i. But for multiples of q_i the last term is positive, hence S_{i+1} is strictly positive definite. So, starting from strictly positive definite S_0, say $S_0 = I$ all subsequent S_i are strictly positive definite. Similar argument works for B_i.

Above we needed $\langle p_i, q_i\rangle > 0$. One can show that when x_i is not a stationary point and S_i is strictly positive definite, then exact line search gives $\langle p_i, q_i\rangle > 0$.

In general, to have strictly positive definite S_{i+1} we need to check if $\langle p_i, q_i\rangle > 0$ during line search.

We introduced DFP and BFGS updates in somewhat ad-hoc manner, but one can show that they minimize special norm. For BFGS we can take any strictly positive definite matrix such that $Wp_i = q_i$. then S_{i+1} minimizes

$$\|W^{1/2}(S - S_i)W^{1/2}\|_{HS}$$

where is positive definite S satisfying $Sq_i = p_i$, $W^{1/2}$ is positive definite square root and

$$\|A\|_{HS} = \sum_{j=1}^{n}\sum_{l=1}^{n}|a_{j,l}|^2$$

is Hilbert-Schmidt (also called Frobenius) norm.

For DFP we need to write condition in term of B_i.

At first glance BFGS update requires solving equation $B_i d_i = \nabla f(x_i)$ in each step or inverting B_i to obtain S_i. However, we can formulate update to B_i directly in terms of S_i. This uses Sherman-Morrison-Woodbury formula:

Lemma 1.3 *Assume A is invertible n by n matrix and U, V are n by k matrices. $A + UV^T$ is invertible if and only if $I + V^T A^{-1} U$ is invertible. Moreover*

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1}U)^{-1}V^T A^{-1}.$$

Proof: When $I + V^T A^{-1} U$ is invertible by direct calculation we check that formula for inverse holds. In particular, if $I + V^T A^{-1} U$ is invertible, then $A + UV^T$ is invertible.

Write $\tilde{A} = I$, $\tilde{U} = V^T$, $\tilde{V} = (A^{-1}U)^T$. Next

$$I + \tilde{V}^T \tilde{A}^{-1} \tilde{U} = I + A^{-1}UV^T = A^{-1}(A + UV^T)$$

so by first part, when $A + UV^T$ is invertible, then

$$\tilde{A} + \tilde{U}\tilde{V}^T = I + V^T A^{-1} U$$

is invertible. □

Writing

$$V_1^T(x) = \frac{\langle B_i p_i, x\rangle}{\langle B_i p_i, p_i\rangle},$$

$$V_2^T(x) = \frac{\langle q_i, x\rangle}{\langle p_i, q_i\rangle},$$

$$U_1^T(x) = -\langle B_i p_i, x\rangle,$$

$$U_2^T(x) = \langle q_i, x\rangle$$

we can rewrite BFGS formula as

$$B_{i+1} = B_i + U_1 V_1^T + U_2 V_2^T$$

Then direct but tedious calculation using twice Sherman-Morrison-Woodbury formula gives

$$S_{i+1} = S_i - \frac{\langle S_i q_i, x\rangle}{\langle S_i q_i, q_i\rangle} S_i q_i + \frac{\langle p_i, x\rangle}{\langle p_i, q_i\rangle} p_i + \langle S_i q_i, q_i\rangle \langle v_i, x\rangle v_i$$

where

$$v_i = \frac{p_i}{\langle p_i, q_i\rangle} - \frac{S_i q_i}{\langle S_i q_i, q_i\rangle}.$$

Formula that we obtained differs from DFP update only by last term. So it is reasonable to interpolate between the two formulas obtaining so called Broyden family:

$$S_{i+1} = S_i - \frac{\langle S_i q_i, x\rangle}{\langle S_i q_i, q_i\rangle} S_i q_i + \frac{\langle p_i, x\rangle}{\langle p_i, q_i\rangle} p_i + \phi \langle S_i q_i, q_i\rangle \langle v_i, x\rangle v_i.$$

When $\phi = 0$ this is DFP update, when $\phi = 1$ this is BFGS update and other values give intermediate formulas. When $\phi \geq 0$ the formula above is sum of DFP update and positive definite term, so lead to positive definite S_i. When $\phi < 0$ in principle S_i may become singular.

Conjugate gradient method have similar goal as quasi-Newton methods so it is interesting to compare them.

Lemma 1.4 *Broyden methods with $S_0 = I$ are first order methods, that is $x_{i+1} \in x_0 + \mathrm{lin}\{\nabla f(x_0), \ldots, \nabla f(x_i)\}$.*

Proof: Put $W_i = \mathrm{lin}\{\nabla f(x_0), \ldots, \nabla f(x_i)\}$. We prove by induction

$$S_i x \in x + W_i$$

Namely

$$S_{i+1} x = S_i x + a S_i q_i + b p_i$$

for some a and b. We have $p_i = -\alpha_i S_i \nabla f(x_i)$ and $q_i = \nabla f(x_{i+1}) - \nabla f(x_i)$. Now by inductive assumption

$$S_i x \in x + W_i,$$

$$p_i = \alpha_i S_i \nabla f(x_i) \in W_i,$$

$$S_i q_i \in \nabla f(x_{i+1}) - \nabla f(x_i) + W_i \subset W_{i+1}$$

so

$$S_{i+1} x = x + W_i.$$

Since $x_{i+1} - x_i$ is multiple of $S_i f(x_i)$ claim follows by another induction. □

Lemma 1.5 *Let A be positive definite matrix and $f(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle$ be quadratic function. Broyden method using exact line search with $S_0 = I$ gives the same points as conjugate gradient method.*

Proof: We inductively prove that

$$\langle Ap_i, p_j\rangle = 0,$$

$$S_i Ap_j = p_j$$

for $j = 0, \ldots, i-1$.

$S_{i+1}Ap_i = p_i$ by quasi-Newton equation. Using inductive assumption we get

$$\langle p_i, Ap_j\rangle = 0$$

for $j < i$. Also

$$\langle S_i q_i, Ap_j\rangle = \langle q_i, S_i Ap_j\rangle = \langle q_i, p_j\rangle = \langle Ap_i, p_j\rangle = 0$$

so by formula for S_{i+1} we see that

$$S_{i+1}Ap_j = S_i Ap_j = p_j.$$

Note that p_i is a multiple of d_i so it is enough to show that d_i are A-orthogonal. We proved that quasi-Newton equations and exact line search imply

$$\langle Ad_{i+1}, d_i\rangle = 0.$$

For $j < i$ we have

$$\langle Ad_{i+1}, d_j\rangle = -\langle AS_{i+1}\nabla f(x_{i+1}), d_j\rangle = -\langle \nabla f(x_{i+1}), S_{i+1}Ad_j\rangle$$

$$= -\langle \nabla f(x_{i+1}), d_j\rangle = -\langle q_i, d_j\rangle - \langle \nabla f(x_i), d_j\rangle$$

For the second term we have

$$\langle \nabla f(x_i), d_j\rangle = \langle \nabla f(x_i), S_i Ad_j\rangle = \langle AS_i\nabla f(x_i), dj\rangle = -\langle Ad_i, d_j\rangle = 0.$$

For the first

$$\langle q_i, d_j\rangle = \langle Ap_i, d_j\rangle = 0$$

which ends inductive proof. □

Our results imply that with exact line search quasi-Newton methods are optimal first order methods for quadratic functions. However, this changes dramatically when inexact line search is in use. One can see on example that quasi-Newton using inexact line search may behave much worse than steepest descent. This is when $f(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle$ and all eigenvalues of A are large (or all are small). Quasi-Newton method brings some eigenvalues close to 1, but other remain large, so $S_i A$ becomes badly conditioned. In other words, simple rescaling may significantly worsen behaviour of quasi-Newton methods. There is simple way to correct this problem: one needs to multiply S_i by appropriate scale factor. One can estimate needed factor and in practice method with scaling behaves much better.

Since Broyden methods are first order methods there is lower bound for for convergence speed as long as number of iterations is less than $n/2$ where n is dimension of the problem. However, when number of iterations is larger than n, then we can get superlinear convergence. We will present one recent result in this direction. Recall that studying Newton method we used scalar product

$$\langle v, w\rangle_x = \langle(\nabla^2 f(x))v, w\rangle$$

and norm

$$\|v\|_x = \langle v, v\rangle_x.$$

We also used

$$\lambda(f, x) = \|((\nabla^2 f(x))^{-1}\nabla f(x)\|_x.$$

We assume that f is strongly convex with Lipschitz continous gradient, that is there is $m > 0$ and M such that

$$mI \leq \nabla^2 f \leq M$$

and

$$\nabla^2 f(y) - \nabla^2 f(x) \leq L\|y - x\|_z \nabla^2 f(w)$$

for all x, y, z, w.

Note: the condition above is statisfied when f is strongly convex and $\nabla^2 f$ is Lipschitz continous with constant C, in such case we can take $L = C/m$.

We consider quasi-Newton method with DFS and BGFS update and $S_0 = I/M$. In each step of the method we take constant $\alpha = 1$.

Lemma 1.6 *Let f and Broyden method be a above. Assume that*

$$L\lambda(f, x_0) \leq \frac{\log(3/2)}{4}\frac{m}{M}.$$

Then

$$\lambda(f, x_k) \leq C^{k/2}(\frac{11nM}{mk})^{k/2}\lambda(f, x_0)$$

Where $C = 1$ for BGFS update and $C = M/m$ for DFS update.

Remark: Clearly, estimate for BGFS update is much better.

Some remarks:

- Under reasonable assumptions (like above) quasi-Newton method exhibits superlinear local convergence
- Quasi-Newton methods avoid cost of computing and inverting second derivative, but need storage for S_i

- In practice BFGS update behaves better than DFP update
- Quasi-Newton methods are sensitive to numerical errors

BFGS update for S_i can be written in different form

$$S_{i+1}x = R_i^T S_i R_i x + \frac{\langle p_i, x\rangle}{\langle p_i, q_i\rangle} p_i$$

where

$$R_i x = x - \frac{\langle p_i, x\rangle}{\langle p_i, q_i\rangle} q_i$$

This means that we can represent S_{i+1} by sequence of p_j and q_j, $j = 0, \dots, i$. To save memory in LBFGS method we only store m (say 20) most recent p_j and q_j.

1.1 Further reading 1

David G. Luenberger, Yinyu Ye, Linear and Nonlinear Programming, chapter 10.

Jorge Nocedal, Stephen J. Wright, Numerical Optimization, chapter 6 and section 2 of chapter 7.

2 Momentum

We are going to present another improvement on steepest descent. When problem is badly conditioned steepest descent may abruptly change direction, leading to slow convergence. Idea: remember previous search direction and combine it with gradient. Namely:

$$x_{i+1} = x_i + \alpha_i d_i,$$

$$d_i = -\nabla f(x_i) + \beta_i (x_i - x_{i-1})$$

Equivalently, with different β_i we can write

$$d_i = -\nabla f(x_i) + \beta_i d_{i-1}.$$

This is called heavy ball or momentum method. In general this is not a descent method.

This is similar to conjugate gradient method. Conjugate gradient method give optimal choice of β_i for quadratic functions (and exact line search), but does not guarantee good behaviour in general.

For analysis we rewrite momentum into another equivalent form

$$x_{i+1} = x_i + \eta_i \nabla f(x_i) + \theta_i (x_i - x_{i-1})$$

We are interested in difference to optimum, so put $y_i = x_i - x_\infty$ where x_∞ is optimal point. In terms of y_i we have

$$y_{i+1} = y_i + \eta_i \nabla f(y_i + x_\infty) + \theta_i (y_i - y_{i-1}).$$

Since $\nabla f(x_\infty) = 0$ we can write

$$\nabla f(y_i + x_\infty) = (\int_0^1 \nabla^2 f(ty_i + x_\infty)dt)y_i = A_i y_i$$

where

$$A_i = \int_0^1 \nabla^2 f(ty_i + x_\infty)dt.$$

So

$$y_{i+1} = y_i + \eta_i A_i y_i + \theta_i(y_i - y_{i-1}).$$

When f is a quadratic function, then A_i does not depend on i, that is $A_i = A$. Since we do not have y_{-1} by necessity we take $\theta_0 = 0$. Then we can write

$$y_i = P(A)y_0$$

where P is a polynomial of degree i with constant term 1. Of course P depends on θ_i and η_i. One can easily prove this by induction. From this we see that momentum have trouble when A A has eigenvector v corresponding to very small positive eigenvalue λ. In such case, with $y_0 = v$ we have

$$y_i = P(\lambda)v$$

and since λ is very close to 0 value of $P(\lambda)$ is very close to 1.

This is expected, because Nesterov example shows that convergence of y_i may need number of iterations of order n where n is dimension of the space. More interesting is situation when f is strictly positive definite:

Lemma 2.1 *Assume that $mI \le A \le MI$. Put $\kappa = \frac{M}{m}$, $\gamma = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$. There exit θ_i, η_i such that*

$$\|y_i\| \le \gamma^i \|y_0\|.$$

Such θ_i, η_i can be chosen independently of A and y_0 and depend only on m and M.

Remark: This is significant improvement over gradient descent since we replace κ by $\sqrt{\kappa}$. For example when $\kappa = 100$ we get almost 10 times better estimate.

Note:

- Rather complicated choice of parameters, known as Chebyshev acceleration
- Unlike conjugate gradient choice of parameters does not depend on A and x_0
- But we need to know or estimate m and M
- Since conjugate gradient is optimal for quadratic functions conjugate gradient gives at least the same improvement of goal function

- But above we talk about $\|x_i - x_\infty\|$, minimizing this is different than minimizing goal function.

It is convenient to use η_i and θ_i which does not depend on i, for this we have:

Lemma 2.2 *Assume that* $mI \leq \nabla^2 f(x) \leq MI$. *Put* $\kappa = \frac{M}{m}$, $\gamma = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$, $\eta_i = \frac{4}{(\sqrt{M}+\sqrt{m})^2}$, $\theta_i = \gamma^2$. *Then for* $i > 0$

$$\left\| \begin{pmatrix} x_{i+1} - x_\infty \\ \gamma(x_i - x_\infty) \end{pmatrix} \right\| \leq (3i-1)\gamma^i \left\| \begin{pmatrix} x_1 - x_\infty \\ \gamma(x_0 - x_\infty) \end{pmatrix} \right\|$$

Remark: Factor $(3i-1)$ is not optimal, but we can not avoid linearly growing factor. So we need extra iterations to overcome this. Still substantial gain compared to steepest descent.

Remark: In limiting case, when κ goes to 1 instead of convergence we may get growth of distance.

We expect similar results for convex function when $\nabla^2 f$ is regular so A_i are close to constant. However, when $\nabla^2 f$ is irregular situation is unclear.

2.1 Nesterov acceleration

Nesterov proposed the following method

$$p_{i+1} = \beta_i p_i - \alpha_i \nabla f(x_i + \beta_i p_i)$$

$$x_{i+1} = x_i + p_{i+1}$$

Standard choice of parameters is $\alpha_i = \frac{1}{M}$ where M is Lipschitz constant of ∇f and $\beta_i = \frac{i}{i+3}$.

Like momentum, but updates x_i before computing ∇f.

Lemma 2.3 *When* f *is convex, with* α_i *and* β_i *as above we have*

$$f(x_i) - f(x_\infty) \leq \frac{4M\|x_0 - x_\infty\|^2}{(i+2)^2}$$

There is also result for strictly convex functions, with slightly different choice of parameters.

Notes about Nesterov acceleration:

- proven upper bound for convergence rate is comparable to lower bound for convex functions
- works well in practice
- low computational cost, comparable to momentum or conjugate gradient

2.2 Further reading 2

Yurii Nesterov, Introductory lectures on convex optimization, Springer 2004, Section 2.2