then we start to check whether the inequality $f(x + t\Delta x) \leq f(x) + \alpha t \nabla f(x)^T \Delta x$ holds.

The parameter α is typically chosen between 0.01 and 0.3, meaning that we accept a decrease in f between 1% and 30% of the prediction based on the linear extrapolation. The parameter β is often chosen to be between 0.1 (which corresponds to a very crude search) and 0.8 (which corresponds to a less crude search).

9.3 Gradient descent method

A natural choice for the search direction is the negative gradient $\Delta x = -\nabla f(x)$. The resulting algorithm is called the *gradient algorithm* or *gradient descent method*.

Algorithm 9.3 *Gradient descent method.*

given a starting point $x \in \mathbf{dom}\, f$.

repeat

1. $\Delta x := -\nabla f(x)$.
2. *Line search.* Choose step size t via exact or backtracking line search.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.

The stopping criterion is usually of the form $\|\nabla f(x)\|_2 \leq \eta$, where η is small and positive. In most implementations, this condition is checked after step 1, rather than after the update.

9.3.1 Convergence analysis

In this section we present a simple convergence analysis for the gradient method, using the lighter notation $x^+ = x + t\Delta x$ for $x^{(k+1)} = x^{(k)} + t^{(k)}\Delta x^{(k)}$, where $\Delta x = -\nabla f(x)$. We assume f is strongly convex on S, so there are positive constants m and M such that $mI \preceq \nabla^2 f(x) \preceq MI$ for all $x \in S$. Define the function $\tilde{f} : \mathbf{R} \to \mathbf{R}$ by $\tilde{f}(t) = f(x - t\nabla f(x))$, *i.e.*, f as a function of the step length t in the negative gradient direction. In the following discussion we will only consider t for which $x - t\nabla f(x) \in S$. From the inequality (9.13), with $y = x - t\nabla f(x)$, we obtain a quadratic upper bound on $\tilde{f}$:

$$\tilde{f}(t) \leq f(x) - t\|\nabla f(x)\|_2^2 + \frac{Mt^2}{2}\|\nabla f(x)\|_2^2. \tag{9.17}$$

Analysis for exact line search

We now assume that an exact line search is used, and minimize over t both sides of the inequality (9.17). On the lefthand side we get $\tilde{f}(t_{\text{exact}})$, where t_{exact} is the step length that minimizes $\tilde{f}$. The righthand side is a simple quadratic, which

is minimized by $t = 1/M$, and has minimum value $f(x) - (1/(2M))\|\nabla f(x)\|_2^2$. Therefore we have

$$f(x^+) = \tilde{f}(t_{\text{exact}}) \leq f(x) - \frac{1}{2M}\|\nabla(f(x))\|_2^2.$$

Subtracting $p^\star$ from both sides, we get

$$f(x^+) - p^\star \leq f(x) - p^\star - \frac{1}{2M}\|\nabla f(x)\|_2^2.$$

We combine this with $\|\nabla f(x)\|_2^2 \geq 2m(f(x) - p^\star)$ (which follows from (9.9)) to conclude

$$f(x^+) - p^\star \leq (1 - m/M)(f(x) - p^\star).$$

Applying this inequality recursively, we find that

$$f(x^{(k)}) - p^\star \leq c^k(f(x^{(0)}) - p^\star) \tag{9.18}$$

where $c = 1 - m/M < 1$, which shows that $f(x^{(k)})$ converges to $p^\star$ as $k \to \infty$. In particular, we must have $f(x^{(k)}) - p^\star \leq \epsilon$ after at most

$$\frac{\log((f(x^{(0)}) - p^\star)/\epsilon)}{\log(1/c)} \tag{9.19}$$

iterations of the gradient method with exact line search.

This bound on the number of iterations required, even though crude, can give some insight into the gradient method. The numerator,

$$\log((f(x^{(0)}) - p^\star)/\epsilon)$$

can be interpreted as the log of the ratio of the initial suboptimality (*i.e.*, gap between $f(x^{(0)})$ and $p^\star$), to the final suboptimality (*i.e.*, less than ϵ). This term suggests that the number of iterations depends on how good the initial point is, and what the final required accuracy is.

The denominator appearing in the bound (9.19), $\log(1/c)$, is a function of M/m, which we have seen is a bound on the condition number of $\nabla^2 f(x)$ over S, or the condition number of the sublevel sets $\{z \mid f(z) \leq \alpha\}$. For large condition number bound M/m, we have

$$\log(1/c) = -\log(1 - m/M) \approx m/M,$$

so our bound on the number of iterations required increases approximately linearly with increasing M/m.

We will see that the gradient method does in fact require a large number of iterations when the Hessian of f, near $x^\star$, has a large condition number. Conversely, when the sublevel sets of f are relatively isotropic, so that the condition number bound M/m can be chosen to be relatively small, the bound (9.18) shows that convergence is rapid, since c is small, or at least not too close to one.

The bound (9.18) shows that the error $f(x^{(k)}) - p^\star$ converges to zero at least as fast as a geometric series. In the context of iterative numerical methods, this is called *linear convergence*, since the error lies below a line on a log-linear plot of error versus iteration number.

Analysis for backtracking line search

Now we consider the case where a backtracking line search is used in the gradient descent method. We will show that the backtracking exit condition,

$$\tilde{f}(t) \leq f(x) - \alpha t \|\nabla f(x)\|_2^2,$$

is satisfied whenever $0 \leq t \leq 1/M$. First note that

$$0 \leq t \leq 1/M \implies -t + \frac{Mt^2}{2} \leq -t/2$$

(which follows from convexity of $-t+Mt^2/2$). Using this result and the bound (9.17), we have, for $0 \leq t \leq 1/M$,

$$\begin{aligned} \tilde{f}(t) &\leq f(x) - t\|\nabla f(x)\|_2^2 + \frac{Mt^2}{2}\|\nabla(f(x))\|_2^2 \\ &\leq f(x) - (t/2)\|\nabla f(x)\|_2^2 \\ &\leq f(x) - \alpha t\|\nabla f(x)\|_2^2, \end{aligned}$$

since $\alpha < 1/2$. Therefore the backtracking line search terminates either with $t = 1$ or with a value $t \geq \beta/M$. This provides a lower bound on the decrease in the objective function. In the first case we have

$$f(x^+) \leq f(x) - \alpha\|\nabla f(x)\|_2^2,$$

and in the second case we have

$$f(x^+) \leq f(x) - (\beta\alpha/M)\|\nabla f(x)\|_2^2.$$

Putting these together, we always have

$$f(x^+) \leq f(x) - \min\{\alpha, \beta\alpha/M\}\|\nabla f(x)\|_2^2.$$

Now we can proceed exactly as in the case of exact line search. We subtract $p^\star$ from both sides to get

$$f(x^+) - p^\star \leq f(x) - p^\star - \min\{\alpha, \beta\alpha/M\}\|\nabla f(x)\|_2^2,$$

and combine this with $\|\nabla f(x)\|_2^2 \geq 2m(f(x) - p^\star)$ to obtain

$$f(x^+) - p^\star \leq (1 - \min\{2m\alpha, 2\beta\alpha m/M\})(f(x) - p^\star).$$

From this we conclude

$$f(x^{(k)}) - p^\star \leq c^k(f(x^{(0)}) - p^\star)$$

where

$$c = 1 - \min\{2m\alpha, 2\beta\alpha m/M\} < 1.$$

In particular, $f(x^{(k)})$ converges to $p^\star$ at least as fast as a geometric series with an exponent that depends (at least in part) on the condition number bound M/m. In the terminology of iterative methods, the convergence is at least linear.

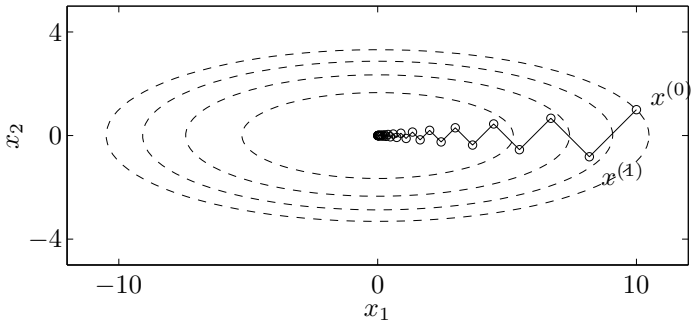

Figure 9.2 Some contour lines of the function $f(x) = (1/2)(x_1^2 + 10x_2^2)$. The condition number of the sublevel sets, which are ellipsoids, is exactly 10. The figure shows the iterates of the gradient method with exact line search, started at $x^{(0)} = (10, 1)$.

9.3.2 Examples

A quadratic problem in $\mathbf{R}^2$

Our first example is very simple. We consider the quadratic objective function on $\mathbf{R}^2$

$$f(x) = \frac{1}{2}(x_1^2 + \gamma x_2^2),$$

where $\gamma > 0$. Clearly, the optimal point is $x^\star = 0$, and the optimal value is 0. The Hessian of f is constant, and has eigenvalues 1 and γ, so the condition numbers of the sublevel sets of f are all exactly

$$\frac{\max\{1, \gamma\}}{\min\{1, \gamma\}} = \max\{\gamma, 1/\gamma\}.$$

The tightest choices for the strong convexity constants m and M are

$$m = \min\{1, \gamma\}, \qquad M = \max\{1, \gamma\}.$$

We apply the gradient descent method with exact line search, starting at the point $x^{(0)} = (\gamma, 1)$. In this case we can derive the following closed-form expressions for the iterates $x^{(k)}$ and their function values (exercise 9.6):

$$x_1^{(k)} = \gamma \left(\frac{\gamma - 1}{\gamma + 1}\right)^k, \qquad x_2^{(k)} = \left(-\frac{\gamma - 1}{\gamma + 1}\right)^k,$$

and

$$f(x^{(k)}) = \frac{\gamma(\gamma + 1)}{2} \left(\frac{\gamma - 1}{\gamma + 1}\right)^{2k} = \left(\frac{\gamma - 1}{\gamma + 1}\right)^{2k} f(x^{(0)}).$$

This is illustrated in figure 9.2, for $\gamma = 10$.

For this simple example, convergence is exactly linear, *i.e.*, the error is exactly a geometric series, reduced by the factor $|(\gamma - 1)/(\gamma + 1)|^2$ at each iteration. For

$\gamma = 1$, the exact solution is found in one iteration; for γ not far from one (say, between 1/3 and 3) convergence is rapid. The convergence is very slow for $\gamma \gg 1$ or $\gamma \ll 1$.

We can compare the convergence with the bound derived above in §9.3.1. Using the least conservative values $m = \min\{1, \gamma\}$ and $M = \max\{1, \gamma\}$, the bound (9.18) guarantees that the error in each iteration is reduced at least by the factor $c = (1 - m/M)$. We have seen that the error is in fact reduced exactly by the factor

$$\left(\frac{1 - m/M}{1 + m/M}\right)^2$$

in each iteration. For small m/M, which corresponds to large condition number, the upper bound (9.19) implies that the number of iterations required to obtain a given level of accuracy grows at most like M/m. For this example, the exact number of iterations required grows approximately like $(M/m)/4$, *i.e.*, one quarter of the value of the bound. This shows that for this simple example, the bound on the number of iterations derived in our simple analysis is only about a factor of four conservative (using the least conservative values for m and M). In particular, the convergence rate (as well as its upper bound) is very dependent on the condition number of the sublevel sets.

A nonquadratic problem in $\mathbf{R}^2$

We now consider a nonquadratic example in $\mathbf{R}^2$, with

$$f(x_1, x_2) = e^{x_1 + 3x_2 - 0.1} + e^{x_1 - 3x_2 - 0.1} + e^{-x_1 - 0.1}. \tag{9.20}$$

We apply the gradient method with a backtracking line search, with $\alpha = 0.1$, $\beta = 0.7$. Figure 9.3 shows some level curves of f, and the iterates $x^{(k)}$ generated by the gradient method (shown as small circles). The lines connecting successive iterates show the scaled steps,

$$x^{(k+1)} - x^{(k)} = -t^{(k)} \nabla f(x^{(k)}).$$

Figure 9.4 shows the error $f(x^{(k)}) - p^\star$ versus iteration k. The plot reveals that the error converges to zero approximately as a geometric series, *i.e.*, the convergence is approximately linear. In this example, the error is reduced from about 10 to about 10^{-7} in 20 iterations, so the error is reduced by a factor of approximately $10^{-8/20} \approx 0.4$ each iteration. This reasonably rapid convergence is predicted by our convergence analysis, since the sublevel sets of f are not too badly conditioned, which in turn means that M/m can be chosen as not too large.

To compare backtracking line search with an exact line search, we use the gradient method with an exact line search, on the same problem, and with the same starting point. The results are given in figures 9.5 and 9.4. Here too the convergence is approximately linear, about twice as fast as the gradient method with backtracking line search. With exact line search, the error is reduced by about 10^{-11} in 15 iterations, *i.e.*, a reduction by a factor of about $10^{-11/15} \approx 0.2$ per iteration.

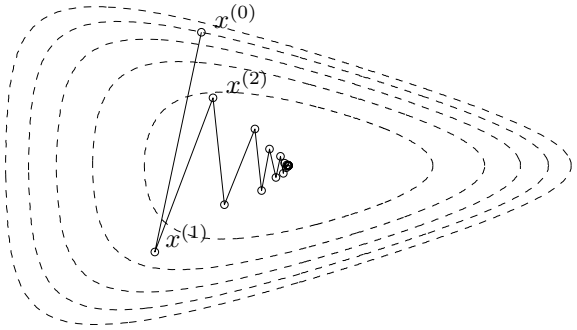

Figure 9.3 Iterates of the gradient method with backtracking line search, for the problem in $\mathbf{R}^2$ with objective f given in (9.20). The dashed curves are level curves of f, and the small circles are the iterates of the gradient method. The solid lines, which connect successive iterates, show the scaled steps $t^{(k)}\Delta x^{(k)}$.

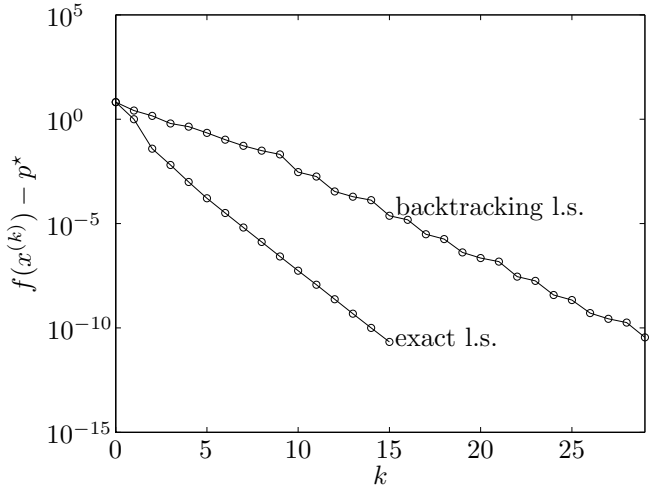

Figure 9.4 Error $f(x^{(k)}) - p^\star$ versus iteration k of the gradient method with backtracking and exact line search, for the problem in $\mathbf{R}^2$ with objective f given in (9.20). The plot shows nearly linear convergence, with the error reduced approximately by the factor 0.4 in each iteration of the gradient method with backtracking line search, and by the factor 0.2 in each iteration of the gradient method with exact line search.

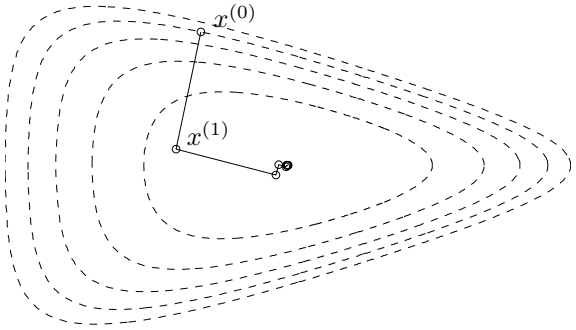

Figure 9.5 Iterates of the gradient method with exact line search for the problem in $\mathbf{R}^2$ with objective f given in (9.20).

A problem in $\mathbf{R}^{100}$

We next consider a larger example, of the form

$$f(x) = c^T x - \sum_{i=1}^{m} \log(b_i - a_i^T x), \tag{9.21}$$

with $m = 500$ terms and $n = 100$ variables.

The progress of the gradient method with backtracking line search, with parameters $\alpha = 0.1$, $\beta = 0.5$, is shown in figure 9.6. In this example we see an initial approximately linear and fairly rapid convergence for about 20 iterations, followed by a slower linear convergence. Overall, the error is reduced by a factor of around 10^6 in around 175 iterations, which gives an average error reduction by a factor of around $10^{-6/175} \approx 0.92$ per iteration. The initial convergence rate, for the first 20 iterations, is around a factor of 0.8 per iteration; the slower final convergence rate, after the first 20 iterations, is around a factor of 0.94 per iteration.

Figure 9.6 shows the convergence of the gradient method with exact line search. The convergence is again approximately linear, with an overall error reduction by approximately a factor $10^{-6/140} \approx 0.91$ per iteration. This is only a bit faster than the gradient method with backtracking line search.

Finally, we examine the influence of the backtracking line search parameters α and β on the convergence rate, by determining the number of iterations required to obtain $f(x^{(k)}) - p^\star \leq 10^{-5}$. In the first experiment, we fix $\beta = 0.5$, and vary α from 0.05 to 0.5. The number of iterations required varies from about 80, for larger values of α, in the range 0.2–0.5, to about 170 for smaller values of α. This, and other experiments, suggest that the gradient method works better with fairly large α, in the range 0.2–0.5.

Similarly, we can study the effect of the choice of β by fixing $\alpha = 0.1$ and varying β from 0.05 to 0.95. Again the variation in the total number of iterations is not large, ranging from around 80 (when $\beta \approx 0.5$) to around 200 (for β small, or near 1). This experiment, and others, suggest that $\beta \approx 0.5$ is a good choice.

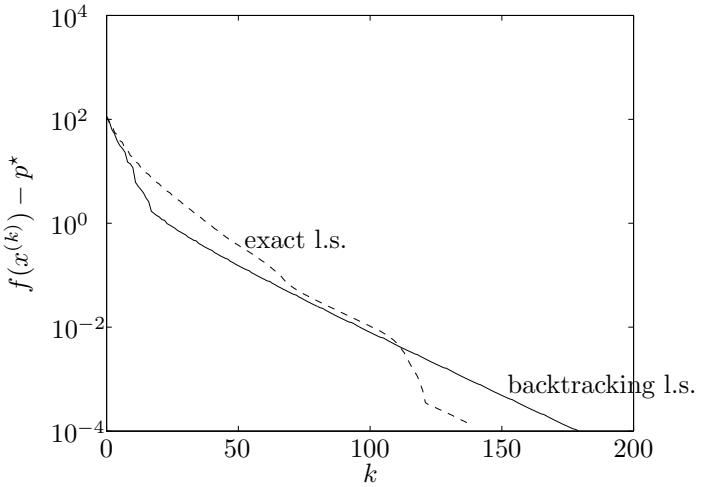

Figure 9.6 Error $f(x^{(k)}) - p^\star$ versus iteration k for the gradient method with backtracking and exact line search, for a problem in $\mathbf{R}^{100}$.

These experiments suggest that the effect of the backtracking parameters on the convergence is not large, no more than a factor of two or so.

Gradient method and condition number

Our last experiment will illustrate the importance of the condition number of $\nabla^2 f(x)$ (or the sublevel sets) on the rate of convergence of the gradient method. We start with the function given by (9.21), but replace the variable x by $x = T\bar{x}$, where

$$T = \mathbf{diag}((1, \gamma^{1/n}, \gamma^{2/n}, \ldots, \gamma^{(n-1)/n})),$$

i.e., we minimize

$$\bar{f}(\bar{x}) = c^T T\bar{x} - \sum_{i=1}^{m} \log(b_i - a_i^T T\bar{x}). \tag{9.22}$$

This gives us a family of optimization problems, indexed by γ, which affects the problem condition number.

Figure 9.7 shows the number of iterations required to achieve $\bar{f}(\bar{x}^{(k)}) - \bar{p}^\star < 10^{-5}$ as a function of γ, using a backtracking line search with $\alpha = 0.3$ and $\beta = 0.7$. This plot shows that for diagonal scaling as small as $10:1$ (*i.e.*, $\gamma = 10$), the number of iterations grows to more than a thousand; for a diagonal scaling of 20 or more, the gradient method slows to essentially useless.

The condition number of the Hessian $\nabla^2 \bar{f}(\bar{x}^\star)$ at the optimum is shown in figure 9.8. For large and small γ, the condition number increases roughly as $\max\{\gamma^2, 1/\gamma^2\}$, in a very similar way as the number of iterations depends on γ. This shows again that the relation between conditioning and convergence speed is a real phenomenon, and not just an artifact of our analysis.

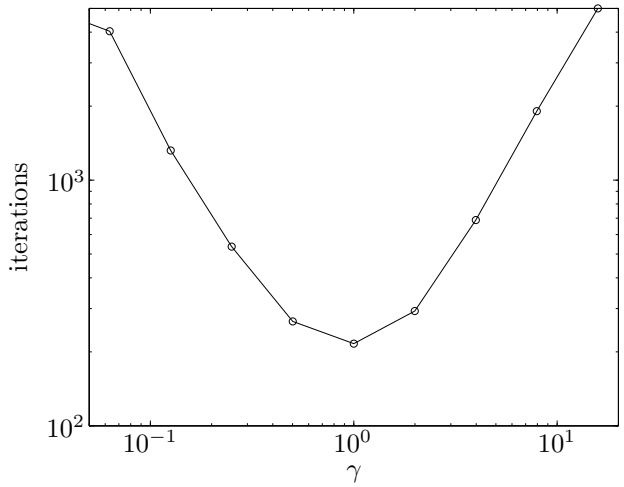

Figure 9.7 Number of iterations of the gradient method applied to problem (9.22). The vertical axis shows the number of iterations required to obtain $\bar{f}(\bar{x}^{(k)}) - \bar{p}^\star < 10^{-5}$. The horizontal axis shows γ, which is a parameter that controls the amount of diagonal scaling. We use a backtracking line search with $\alpha = 0.3$, $\beta = 0.7$.

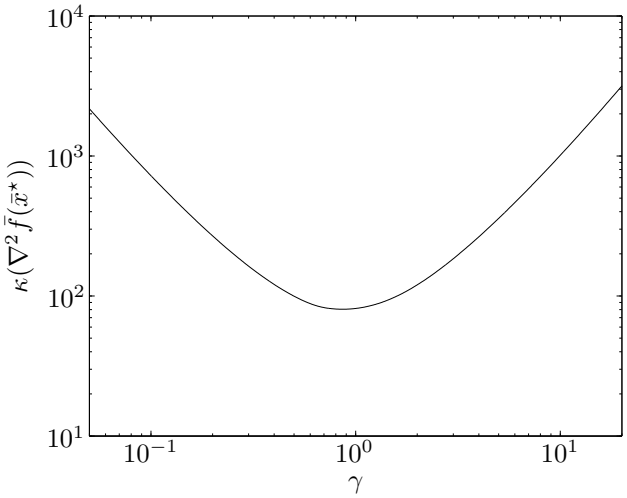

Figure 9.8 Condition number of the Hessian of the function at its minimum, as a function of γ. By comparing this plot with the one in figure 9.7, we see that the condition number has a very strong influence on convergence rate.

Conclusions

From the numerical examples shown, and others, we can make the conclusions summarized below.

- The gradient method often exhibits approximately linear convergence, *i.e.*, the error $f(x^{(k)}) - p^\star$ converges to zero approximately as a geometric series.
- The choice of backtracking parameters α, β has a noticeable but not dramatic effect on the convergence. An exact line search sometimes improves the convergence of the gradient method, but the effect is not large (and probably not worth the trouble of implementing the exact line search).
- The convergence rate depends greatly on the condition number of the Hessian, or the sublevel sets. Convergence can be very slow, even for problems that are moderately well conditioned (say, with condition number in the 100s). When the condition number is larger (say, 1000 or more) the gradient method is so slow that it is useless in practice.

The main advantage of the gradient method is its simplicity. Its main disadvantage is that its convergence rate depends so critically on the condition number of the Hessian or sublevel sets.

9.4 Steepest descent method

The first-order Taylor approximation of $f(x+v)$ around x is

$$f(x+v) \approx \widehat{f}(x+v) = f(x) + \nabla f(x)^T v.$$

The second term on the righthand side, $\nabla f(x)^T v$, is the *directional derivative* of f at x in the direction v. It gives the approximate change in f for a small step v. The step v is a descent direction if the directional derivative is negative.

We now address the question of how to choose v to make the directional derivative as negative as possible. Since the directional derivative $\nabla f(x)^T v$ is linear in v, it can be made as negative as we like by taking v large (provided v is a descent direction, *i.e.*, $\nabla f(x)^T v < 0$). To make the question sensible we have to limit the size of v, or normalize by the length of v.

Let $\|\cdot\|$ be any norm on $\mathbf{R}^n$. We define a *normalized steepest descent direction* (with respect to the norm $\|\cdot\|$) as

$$\Delta x_{\rm nsd} = \operatorname{argmin}\{\nabla f(x)^T v \mid \|v\| = 1\}. \tag{9.23}$$

(We say 'a' steepest descent direction because there can be multiple minimizers.) A normalized steepest descent direction $\Delta x_{\rm nsd}$ is a step of unit norm that gives the largest decrease in the linear approximation of f.

A normalized steepest descent direction can be interpreted geometrically as follows. We can just as well define $\Delta x_{\rm nsd}$ as

$$\Delta x_{\rm nsd} = \operatorname{argmin}\{\nabla f(x)^T v \mid \|v\| \leq 1\},$$