Lecture Notes 7: Convex Optimization

1 Convex functions

Convex functions are of crucial importance in optimization-based data analysis because they can be efficiently minimized. In this section we introduce the concept of convexity and then discuss norms, which are convex functions that are often used to design convex cost functions when fitting models to data.

1.1 Convexity

A function is convex if and only if its curve lies below any chord joining two of its points.

Definition 1.1 (Convex function)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if for any $\vec{x}, \vec{y} \in \mathbb{R}^n$ and any $\theta \in (0, 1)$,*

$$\theta f(\vec{x}) + (1-\theta) f(\vec{y}) \geq f(\theta \vec{x} + (1-\theta)\vec{y}). \tag{1}$$

The function is strictly convex if the inequality is always strict, i.e. if $\vec{x} \neq \vec{y}$ implies that

$$\theta f(\vec{x}) + (1-\theta) f(\vec{y}) > f(\theta \vec{x} + (1-\theta)\vec{y}). \tag{2}$$

A concave function is a function f such that $-f$ is convex.

Linear functions are convex, but not strictly convex.

Lemma 1.2. *Linear functions are convex but not strictly convex.*

Proof. If f is linear, for any $\vec{x}, \vec{y} \in \mathbb{R}^n$ and any $\theta \in (0, 1)$,

$$f(\theta \vec{x} + (1-\theta)\vec{y}) = \theta f(\vec{x}) + (1-\theta) f(\vec{y}). \tag{3}$$

□

Condition (1) is illustrated in Figure 1. The following lemma shows that when determining whether a function is convex we can restrict our attention to its behavior along lines in $\mathbb{R}^n$.

Lemma 1.3 (Proof in Section 4.1)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if for any two points $\vec{x}, \vec{y} \in \mathbb{R}^n$ the univariate function $g_{\vec{x},\vec{y}} : [0, 1] \to \mathbb{R}$ defined by*

$$g_{\vec{x},\vec{y}}(\alpha) := f(\alpha \vec{x} + (1-\alpha)\vec{y}) \tag{4}$$

is convex. Similarly, f is strictly convex if and only if $g_{\vec{x},\vec{y}}$ is strictly convex for any $\vec{x} \neq \vec{y}$.

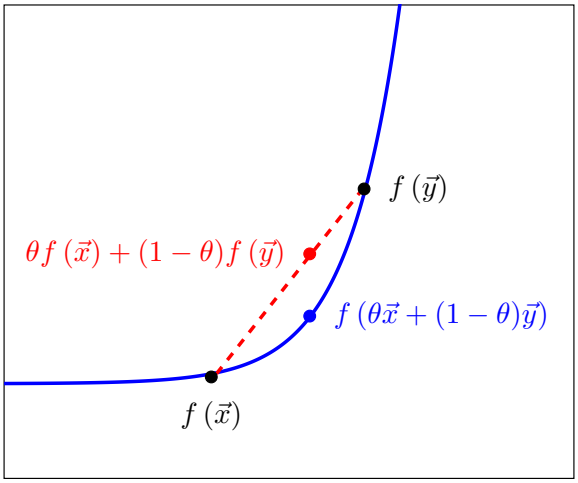

Figure 1: Illustration of condition (1) in Definition 1.1. The curve corresponding to the function must lie below any chord joining two of its points.

Convex functions are easier to optimize than nonconvex functions because once we find a local minimum of the function we are done: every local minimum is guaranteed to be a global minimum.

Theorem 1.4 (Local minima are global)**.** *Any local minimum of a convex function is also a global minimum.*

Proof. We prove the result by contradiction. Let $\vec{x}_{\text{loc}}$ be a local minimum and $\vec{x}_{\text{glob}}$ a global minimum such that $f(\vec{x}_{\text{glob}}) < f(\vec{x}_{\text{loc}})$. Since $\vec{x}_{\text{loc}}$ is a local minimum, there exists $\gamma > 0$ for which $f(\vec{x}_{\text{loc}}) \leq f(\vec{x})$ for all $\vec{x} \in \mathbb{R}^n$ such that $||\vec{x} - \vec{x}_{\text{loc}}||_2 \leq \gamma$. If we choose $\theta \in (0, 1)$ small enough, $\vec{x}_\theta := \theta\vec{x}_{\text{loc}} + (1-\theta)\vec{x}_{\text{glob}}$ satisfies $||\vec{x}_\theta - \vec{x}_{\text{loc}}||_2 \leq \gamma$ and therefore

$$
\begin{aligned}
f(\vec{x}_{\text{loc}}) &\leq f(\vec{x}_\theta) && (5)\\
&\leq \theta f(\vec{x}_{\text{loc}}) + (1-\theta) f(\vec{x}_{\text{glob}}) \quad \text{by convexity of } f && (6)\\
&< f(\vec{x}_{\text{loc}}) \quad \text{because } f(\vec{x}_{\text{glob}}) < f(\vec{x}_{\text{loc}}). && (7)
\end{aligned}
$$

□

1.2 Norms

Many of the cost functions that we consider in data analysis involve norms. Conveniently, all norms are convex.

Lemma 1.5 (Norms are convex)**.** *Any valid norm $||\cdot||$ is a convex function.*

Proof. By the triangle inequality inequality and homogeneity of the norm, for any $\vec{x}, \vec{y} \in \mathbb{R}^n$ and any $\theta \in (0, 1)$

$$
||\theta\vec{x} + (1-\theta)\vec{y}|| \leq ||\theta\vec{x}|| + ||(1-\theta)\vec{y}|| = \theta ||\vec{x}|| + (1-\theta) ||\vec{y}|| . \tag{8}
$$

□

The following lemma establishes that the composition between a convex function and an affine function is convex. In particular, this means that any function of the form

$$f(\vec{x}) := \left|\left|A\vec{x} + \vec{b}\right|\right| \tag{9}$$

is convex for any fixed matrix A and vector $\vec{b}$ with suitable dimensions.

Lemma 1.6 (Composition of convex and affine function)**.** *If $f : \mathbb{R}^n \to \mathbb{R}$ is convex, then for any $A \in \mathbb{R}^{n\times m}$ and any $\vec{b} \in \mathbb{R}^n$, the function*

$$h(\vec{x}) := f\left(A\vec{x} + \vec{b}\right) \tag{10}$$

is convex.

Proof. By convexity of f, for any $\vec{x}, \vec{y} \in \mathbb{R}^m$ and any $\theta \in (0, 1)$

$$\begin{aligned} h(\theta\vec{x} + (1-\theta)\vec{y}) &= f\left(\theta\left(A\vec{x} + \vec{b}\right) + (1-\theta)\left(A\vec{y} + \vec{b}\right)\right) && (11)\\ &\leq \theta f\left(A\vec{x} + \vec{b}\right) + (1-\theta) f\left(A\vec{y} + \vec{b}\right) && (12)\\ &= \theta\, h(\vec{x}) + (1-\theta)\, h(\vec{y})\,. && (13) \end{aligned}$$

□

The number of nonzero entries in a vector is often called the ℓ_0 "norm" of the vector. Despite its name, it is not a valid norm (it is not homogeneous: for any $\vec{x}$ $||2\vec{x}||_0 = ||\vec{x}||_0 \neq ||\vec{x}||_0$). In fact, the ℓ_0 "norm" is not even convex.

Lemma 1.7 (ℓ_0 "norm" is not convex)**.** *The ℓ_0 "norm" defined as the number of nonzero entries in a vector is not convex.*

Proof. We provide a simple counterexample with vectors in $\mathbb{R}^2$ that can be easily extended to vectors in $\mathbb{R}^n$. Let $\vec{x} := \left(\begin{smallmatrix}1\\0\end{smallmatrix}\right)$ and $\vec{y} := \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$, then for any $\theta \in (0, 1)$

$$||\theta\vec{x} + (1-\theta)\vec{y}||_0 = 2 > 1 = \theta\,||\vec{x}||_0 + (1-\theta)\,||\vec{y}||_0\,. \tag{14}$$

□

Example 1.8 (Promoting sparsity)**.** Finding sparse vectors that are consistent with observed data is often very useful in data analysis. Let us consider a toy problem where the entries of a vector are constrained to be of the form

$$\vec{v}_t := \begin{bmatrix} t \\ t-1 \\ t-1 \end{bmatrix}. \tag{15}$$

Our objective is to fit t so that $\vec{v}_t$ is as sparse as possible or, in other words, minimize $||\vec{v}_t||_0$. Unfortunately this function is nonconvex. The graph of the function is depicted in Figure 2. In contrast, if we consider

$$f(t) := ||\vec{v}_t|| \tag{16}$$

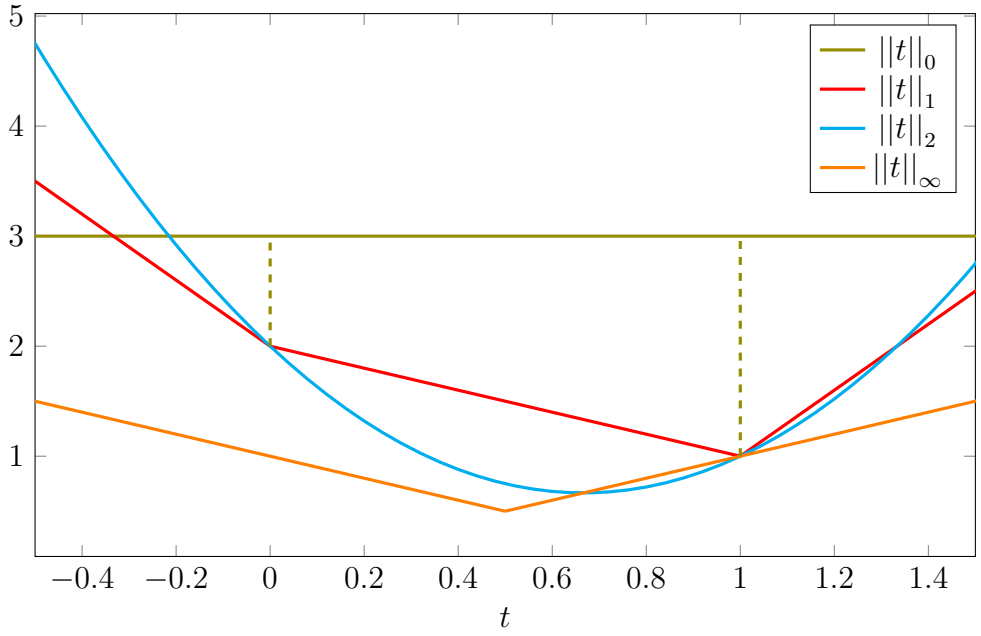

Figure 2: Graph of the function (16) for different norms and for the nonconvex ℓ_0 "norm".

where $||\cdot||$ is a valid norm, then we can exploit local information to find the global minimum (we will discuss how to do this in more detail later on) because the function is convex in t by Lemma 1.6. This is impossible to do for $||\vec{v}_t||_0$ because it is constant except at two isolated points. Figure 2 shows f for different norms.

The ℓ_1 norm is the best choice for our purposes: it is convex and its global minimum is at the same location as the minimum ℓ_0 "norm" solution. This is not a coincidence: minimizing the ℓ_1 norm tends to promote sparsity. When compared to the ℓ_2 norm, it penalizes small entries much more (ϵ^2 is much smaller than $|\epsilon|$ for small ϵ), as a result it tends to produce solutions that contain a small number of larger nonzero entries. △

The rank of a matrix interpreted as a function of its entries is also not convex.

Lemma 1.9 (The rank is not convex)**.** *The rank of matrices in $\mathbb{R}^{n\times n}$ interpreted as a function from $\mathbb{R}^{n\times n}$ to $\mathbb{R}$ is not convex.*

Proof. We provide a counterexample that is very similar to the one in the proof of Lemma 1.7. Let

$$X := \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad Y := \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{17}$$

For any $\theta \in (0, 1)$

$$\operatorname{rank}(\theta X + (1-\theta) Y) = 2 > 1 = \theta \operatorname{rank}(X) + (1-\theta) \operatorname{rank}(Y). \tag{18}$$

□

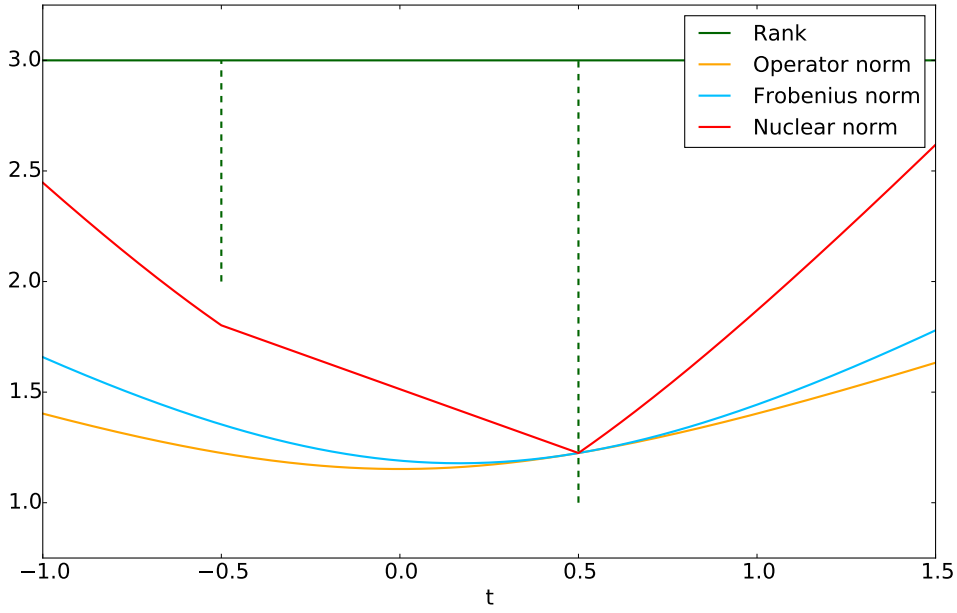

Figure 3: Values of different norms for the matrix $M(t)$ defined by (19). The rank of the matrix for each t is marked in green.

Example 1.10 (Promoting low-rank structure)**.** Finding low-rank matrices that are consistent with data is useful in applications of PCA where data may be corrupted or missing. Let us consider a toy problem where our goal is to find t so that

$$M(t) := \begin{bmatrix} 0.5+t & 1 & 1 \\ 0.5 & 0.5 & t \\ 0.5 & 1-t & 0.5 \end{bmatrix}, \tag{19}$$

is as low rank as possible. In Figure 3 we compare the rank, the operator norm, the Frobenius norm and the nuclear norm of $M(t)$ for different values of t. As expected, the rank is highly nonconvex, whereas the norms are all convex, which follows from Lemma 1.6. The value of t that minimizes the rank is the same as the one that minimizes the nuclear norm. In contrast, the values of t that minimize the operator and Frobenius norms are different. Just like the ℓ_1 norm promotes sparsity, the nuclear norm, which the ℓ_1 norm of the singular values, promotes solutions with low rank, which is the ℓ_0 "norm" of the singular values. △

2 Differentiable convex functions

2.1 First-order conditions

The gradient is the generalization of the concept of derivative, which captures the local rate of change in the value of a function, in multiple directions.

Definition 2.1 (Gradient)**.** *The gradient of a function $f : \mathbb{R}^n \to \mathbb{R}$ at a point $\vec{x} \in \mathbb{R}^n$ is defined to be the unique vector $\nabla f(\vec{x}) \in \mathbb{R}^n$ satisfying*

$$\lim_{\vec{p} \to 0} \frac{f(x + \vec{p}) - f(x) - \nabla f(\vec{x})^T \vec{p}}{\|\vec{p}\|_2} = 0,$$

assuming such a vector $\nabla f(\vec{x})$ exists. If $\nabla f(\vec{x})$ exists then it is given by the vector of partial derivatives:

$$\nabla f\left(\vec{x}\right) = \begin{bmatrix} \frac{\partial f(\vec{x})}{\partial \vec{x}[1]} \\ \frac{\partial f(\vec{x})}{\partial \vec{x}[2]} \\ \cdots \\ \frac{\partial f(\vec{x})}{\partial \vec{x}[n]} \end{bmatrix}. \tag{20}$$

If the gradient exists at every point, the function is said to be differentiable.

The gradient encodes the variation of the function *in every direction.*

Lemma 2.2. *If a function $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable, the directional derivative $f'_{\vec{u}}$ of f at $\vec{x}$ equals*

$$f'_{\vec{u}}\left(\vec{x}\right) := \lim_{h \to 0} \frac{f\left(\vec{x} + h\vec{u}\right) - f\left(\vec{x}\right)}{h} \tag{21}$$

$$= \left\langle \nabla f\left(\vec{x}\right), \vec{u} \right\rangle \tag{22}$$

for any unit-norm vector $\vec{u} \in \mathbb{R}^n$.

We omit the proof of the lemma, which is a basic result from multivariable calculus. An important corollary is that the gradient provides the direction of maximum positive and negative variation of the function.

Corollary 2.3. *The direction of the gradient ∇f of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is the direction of maximum increase of the function. The opposite direction is the direction of maximum decrease.*

Proof. By the Cauchy-Schwarz inequality

$$\left|f'_{\vec{u}}\left(\vec{x}\right)\right| = \left|\nabla f\left(\vec{x}\right)^T \vec{u}\right| \tag{23}$$

$$\leq \left|\left|\nabla f\left(\vec{x}\right)\right|\right|_2 \left|\left|\vec{u}\right|\right|_2 \tag{24}$$

$$= \left|\left|\nabla f\left(\vec{x}\right)\right|\right|_2 \tag{25}$$

with equality if and only if $\vec{u} = \pm \frac{\nabla f(\vec{x})}{||\nabla f(\vec{x})||_2}$. □

Figure 4 shows the gradient of a function $f : \mathbb{R}^2 \to \mathbb{R}$ at different locations. The gradient is orthogonal to the contour lines of the function. The reason is that by definition the function does not change along the contour lines, so the directional derivatives in those directions are zero.

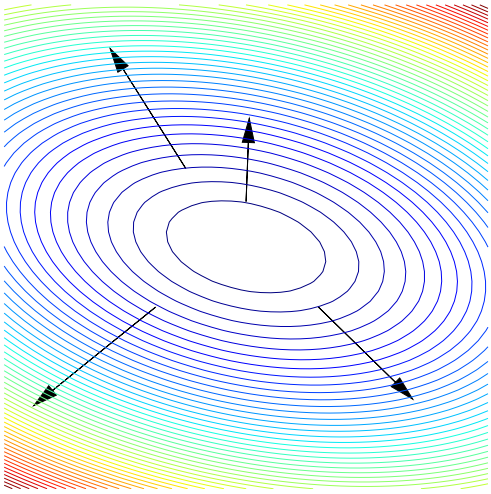

Figure 4: Contour lines of a function $f : \mathbb{R}^2 \to \mathbb{R}$. The gradients at different points are represented by black arrows, which are orthogonal to the contour lines.

The first-order Taylor expansion of a differentiable function is a linear function that approximates the function around a certain point. Geometrically, in one dimension this linear approximation is a line that is tangent to the curve $(x, f(x))$. In multiple dimensions, it is a hyperplane that is tangent to the hypersurface $(\vec{x}, f(\vec{x}))$ at that point.

Definition 2.4 (First-order approximation)**.** *The first-order or linear approximation of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ at $\vec{x}$ is*

$$f_{\vec{x}}^1(\vec{y}) := f(\vec{x}) + \nabla f(\vec{x})^T (\vec{y} - \vec{x}). \tag{26}$$

By construction, the first-order approximation of a function at a given point is a linear function that has exactly the same directional derivatives at that point. The following theorem establishes that a function f is convex if and only if the linear approximation $f_{\vec{x}}^1$ is a lower bound of f for any $\vec{x} \in \mathbb{R}^n$. Figure 5 illustrates the condition.

Theorem 2.5 (Proof in Section 4.2)**.** *A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if for every $\vec{x}, \vec{y} \in \mathbb{R}^n$*

$$f(\vec{y}) \geq f(\vec{x}) + \nabla f(\vec{x})^T (\vec{y} - \vec{x}). \tag{27}$$

It is strictly convex if and only if

$$f(\vec{y}) > f(\vec{x}) + \nabla f(\vec{x})^T (\vec{y} - \vec{x}). \tag{28}$$

An immediate corollary is that for a convex function, any point at which the gradient is zero is a global minimum. If the function is strictly convex, the minimum is unique. This is very useful for minimizing such functions, once we find a point where the gradient is zero we are done!

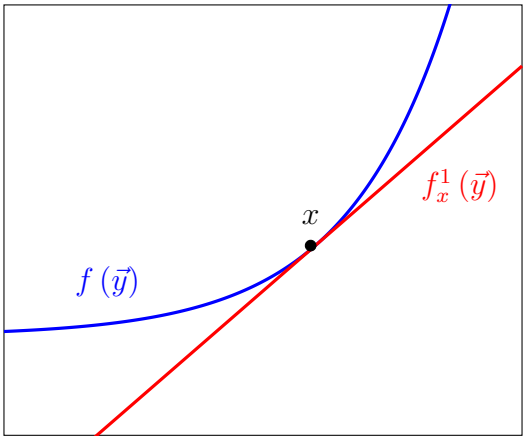

Figure 5: An example of the first-order condition for convexity. The first-order approximation at any point is a lower bound of the function.

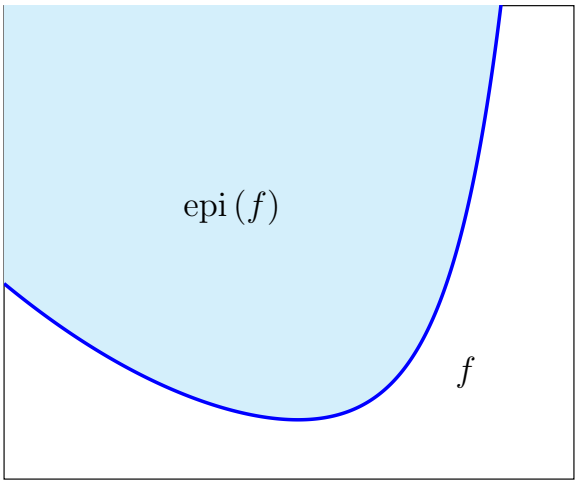

Figure 6: Epigraph of a function.

Corollary 2.6. *If a differentiable function f is convex and $\nabla f(\vec{x}) = 0$, then for any $\vec{y} \in \mathbb{R}$*

$$f(\vec{y}) \geq f(\vec{x}). \tag{29}$$

If f is strictly convex then for any $\vec{y} \neq \vec{x}$

$$f(\vec{y}) > f(\vec{x}). \tag{30}$$

For any differentiable function f and any $\vec{x} \in \mathbb{R}^n$ let us define the hyperplane $\mathcal{H}_{f,\vec{x}} \subset \mathbb{R}^{n+1}$ that corresponds to the first-order approximation of f at $\vec{x}$,

$$\mathcal{H}_{f,\vec{x}} := \left\{ \vec{y} \mid \vec{y}[n+1] = f_{\vec{x}}^1 \left(\begin{bmatrix} \vec{y}[1] \\ \cdots \\ \vec{y}[n] \end{bmatrix} \right) \right\}. \tag{31}$$

The epigraph is the subset of $\mathbb{R}^{n+1}$ that lies above the graph of the function. Recall that the graph is the set of vectors in $\mathbb{R}^{n+1}$ obtained by concatenating $\vec{x} \in \mathbb{R}^n$ and $f(\vec{x})$ for every $\vec{x} \in \mathbb{R}^n$. Figure 6 shows the epigraph of a convex function.

Definition 2.7 (Epigraph)**.** *The epigraph of a function $f : \mathbb{R}^n \to \mathbb{R}$ is the set*

$$\text{epi}(f) := \left\{ \vec{x} \mid f\left(\begin{bmatrix} \vec{x}[1] \\ \cdots \\ \vec{x}[n] \end{bmatrix} \right) \leq \vec{x}[n+1] \right\}. \tag{32}$$

Geometrically, Theorem 2.5 establishes that the epigraph of a convex function always lies above $\mathcal{H}_{f,\vec{x}}$. By construction, $\mathcal{H}_{f,\vec{x}}$ and $\text{epi}(f)$ intersect at $\vec{x}$. This implies that $\mathcal{H}_{f,\vec{x}}$ is a supporting hyperplane of $\text{epi}(f)$ at $\vec{x}$.

Definition 2.8 (Supporting hyperplane)**.** *A hyperplane $\mathcal{H}$ is a supporting hyperplane of a set $\mathcal{S}$ at $\vec{x}$ if*

- *$\mathcal{H}$ and $\mathcal{S}$ intersect at $\vec{x}$,*
- *$\mathcal{S}$ is contained in one of the half-spaces bounded by $\mathcal{H}$.*

The optimality condition in Corollary 2.6 has a very intuitive geometric interpretation in terms of the supporting hyperplane $\mathcal{H}_{f,\vec{x}}$. $\nabla f = 0$ implies that $\mathcal{H}_{f,\vec{x}}$ is horizontal if the vertical dimension corresponds to the $n+1$th coordinate. Since the epigraph lies above hyperplane, the point at which they intersect must be a minimum of the function.

2.2 Second-order conditions

The Hessian matrix of a function contains its second-order partial derivatives.

Definition 2.9 (Hessian matrix)**.** *A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable at $\vec{x} \in \mathbb{R}^n$ if there is a matrix $\nabla^2 f(\vec{x}) \in \mathbb{R}^{n \times n}$ such that*

$$\lim_{\vec{p} \to 0} \frac{\|\nabla f(x+h) - \nabla f(x) - \nabla^2 f(x)\vec{p}\|_2}{\|p\|_2} = 0.$$

If $\nabla^2 f(\vec{x})$ exists then it is given by

$$\nabla^2 f(\vec{x}) = \begin{bmatrix} \frac{\partial^2 f(\vec{x})}{\partial \vec{x}[1]^2} & \frac{\partial^2 f(\vec{x})}{\partial \vec{x}[1] \partial \vec{x}[2]} & \cdots & \frac{\partial^2 f(\vec{x})}{\partial \vec{x}[1] \partial \vec{x}[n]} \\ \frac{\partial^2 f(\vec{x})}{\partial \vec{x}[1] \partial \vec{x}[2]} & \frac{\partial^2 f(\vec{x})}{\partial \vec{x}[1]^2} & \cdots & \frac{\partial^2 f(\vec{x})}{\partial \vec{x}[2] \partial \vec{x}[n]} \\ & & \cdots & \\ \frac{\partial^2 f(\vec{x})}{\partial \vec{x}[1] \partial \vec{x}[n]} & \frac{\partial^2 f(\vec{x})}{\partial \vec{x}[2] \partial \vec{x}[n]} & \cdots & \frac{\partial^2 f(\vec{x})}{\partial \vec{x}[n]^2} \end{bmatrix}. \tag{33}$$

If a function has a Hessian matrix at every point, we say that the function is twice differentiable. If each entry of the Hessian is continuous, we say f is twice continuously differentiable.

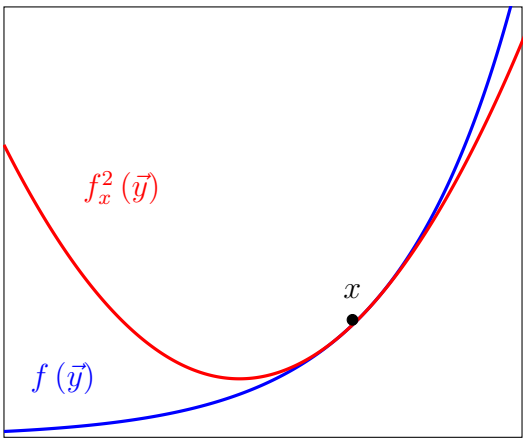

Figure 7: Second-order approximation of a function.

Note that by (33) if $f : \mathbb{R} \to \mathbb{R}^n$ is differentiable everywhere and twice differentiable at $\vec{x} \in \mathbb{R}^n$ then the Hessian $\nabla^2 f(\vec{x})$ is always a symmetric matrix.

As you might recall from basic calculus, curvature is the rate of change of the slope of the function and is consequently given by its second derivative. The Hessian matrix encodes the curvature of the function in every direction, another basic result from multivariable calculus.

Lemma 2.10. *If a function $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable, the second directional derivative $f''_{\vec{u}}$ of f at $\vec{x}$ equals*

$$f''_{\vec{u}}(\vec{x}) = \vec{u}^T \nabla^2 f(\vec{x})\, \vec{u}, \tag{34}$$

for any unit-norm vector $\vec{u} \in \mathbb{R}^n$.

The Hessian and the gradient of a twice-differentiable function can be used to build a quadratic approximation of the function. This approximation is depicted in Figure 7 for a one-dimensional function.

Definition 2.11 (Second-order approximation)**.** *The second-order or quadratic approximation of f at $\vec{x}$ is*

$$f^2_{\vec{x}}(\vec{y}) := f(\vec{x}) + \nabla f(\vec{x})(\vec{y} - \vec{x}) + \frac{1}{2}(\vec{y} - \vec{x})^T \nabla^2 f(\vec{x})(\vec{y} - \vec{x}) \tag{35}$$

By construction, the second-order approximation of a function at a given point is a quadratic form that has exactly the same directional derivatives and curvature at that point. This second-order approximation is a quadratic form.

Definition 2.12 (Quadratic functions/forms)**.** *A quadratic function $q : \mathbb{R}^n \to \mathbb{R}$ is a second-order polynomial in several dimensions. Such polynomials can be written in terms of a symmetric matrix $A \in \mathbb{R}^{n \times n}$, a vector $\vec{b} \in \mathbb{R}^n$ and a constant c*

$$q(\vec{x}) := \vec{x}^T A \vec{x} + \vec{b}^T \vec{x} + c. \tag{36}$$

A quadratic form is a (pure) quadratic function where $\vec{b} = 0$ and $c = 0$.

The quadratic form $f_{\vec{x}}^2(\vec{y})$ becomes an arbitrarily good approximation of f as we approach $\vec{x}$, even if we divide the error by the squared distance between $\vec{x}$ and $\vec{y}$. We omit the proof that follows from multivariable calculus.

Lemma 2.13. *The quadratic approximation $f_{\vec{x}}^2 : \mathbb{R}^n \to \mathbb{R}$ at $\vec{x} \in \mathbb{R}^n$ of a twice differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ satisfies*

$$\lim_{\vec{y} \to \vec{x}} \frac{f(\vec{y}) - f_{\vec{x}}^2(\vec{y})}{||\vec{y} - \vec{x}||_2^2} = 0 \tag{37}$$

To find the maximum curvature of a function at a given point, we can compute an eigendecomposition of its Hessian.

Theorem 2.14. *Let $A = U \Lambda U^T$ be the eigendecomposition of a symmetric matrix A, where $\lambda_1 \geq \cdots \geq \lambda_n$ (which can be negative) are the eigenvalues and $\vec{u}_1, \ldots, \vec{u}_n$ the corresponding eigenvectors. Then*

$$\lambda_1 = \max_{\left\{||\vec{x}||_2 = 1 \mid \vec{x} \in \mathbb{R}^n\right\}} \vec{x}^T A \vec{x}, \tag{38}$$

$$\vec{u}_1 = \arg\max_{\left\{||\vec{x}||_2 = 1 \mid \vec{x} \in \mathbb{R}^n\right\}} \vec{x}^T A \vec{x}, \tag{39}$$

$$\lambda_n = \min_{\left\{||\vec{x}||_2 = 1 \mid \vec{x} \in \mathbb{R}^n\right\}} \vec{x}^T A \vec{x}, \tag{40}$$

$$\vec{u}_n = \arg\min_{\left\{||\vec{x}||_2 = 1 \mid \vec{x} \in \mathbb{R}^n\right\}} \vec{x}^T A \vec{x}. \tag{41}$$

Proof. By Theorem 4.3 in Lecture Notes 2 the eigendecomposition of A is the same as its SVD, except that some of the eigenvalues may be negative (which flips the direction of the corresponding eigenvectors with respect to the singular vectors). The result then follows from Theorem 2.7 in the same lecture notes. □

Corollary 2.15. *Consider the eigendecomposition of the Hessian matrix of a twice-differentiable function f at a point $\vec{x}$. The maximum curvature of f at $\vec{x}$ is given by the largest eigenvalue of $\nabla^2 f(\vec{x})$ and is in the direction of the corresponding eigenvector. The smallest curvature, or the largest negative curvature, of f at $\vec{x}$ is given by the smallest eigenvalue of $\nabla^2 f(\vec{x})$ and is in the direction of the corresponding eigenvector.*

If all the eigenvalues of a symmetric matrix $A \in \mathbb{R}^{n \times n}$ are nonnegative, the matrix is said to be *positive semidefinite*. The pure quadratic form corresponding to such matrices is always nonnegative.

Lemma 2.16 (Positive semidefinite matrices)**.** *The eigenvalues of a symmetric matrix $A \in \mathbb{R}^{n \times n}$ are all nonnegative if and only if*

$$\vec{x}^T A \vec{x} \geq 0 \tag{42}$$

for all $\vec{x} \in \mathbb{R}^n$. Such matrices are called positive semidefinite.

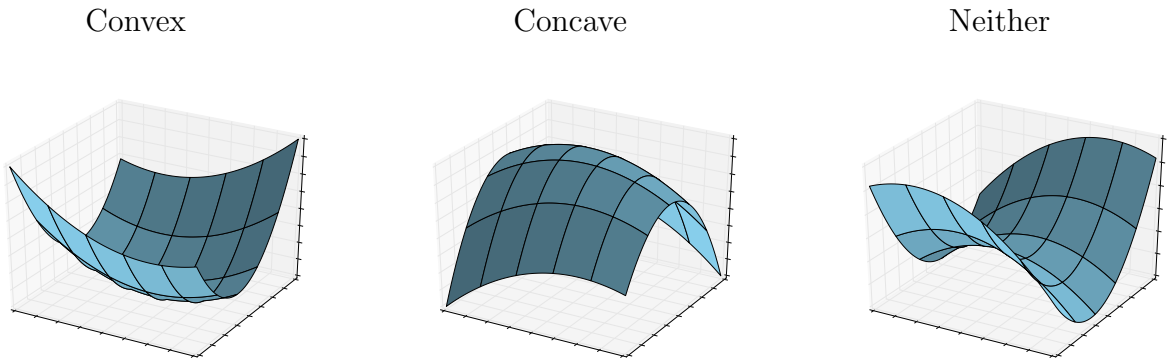

Figure 8: Quadratic forms for which the Hessian is positive definite (left), negative definite (center) and neither positive nor negative definite (right).

Proof. By Theorem 2.14, the matrix has an eigendecomposition $A = U\Lambda U^T$ where the eigenvectors $\vec{u}_1, \ldots, \vec{u}_n$ form an orthonormal basis so that

$$\vec{x}^T A\vec{x} = \vec{x}^T U\Lambda U^T \vec{x} \tag{43}$$

$$= \sum_{i=1}^{n} \lambda_i \langle \vec{u}_i, \vec{x} \rangle^2. \tag{44}$$

□

If the eigenvalues are positive the matrix is *positive definite*. If the eigenvalues are all nonpositive, the matrix is negative semidefinite. If they are negative, the matrix is negative definite. By Corollary 2.15 a twice-differentiable function has positive (resp. nonnegative) curvature in *every* direction if its Hessian is positive definite (resp. semidefinite) and it has negative (resp. nonpositive) curvature in every direction if the Hessian is negative definite (resp. semidefinite). Figure 8 illustrates this in the case of quadratic forms in two dimensions.

For univariate functions that are twice differentiable, convexity is dictated by the curvature. The following lemma establishes that univariate functions are convex if and only if their curvature is always nonnegative.

Lemma 2.17 (Proof in Section 4.4)**.** *A twice-differentiable function $g : \mathbb{R} \to \mathbb{R}$ is convex if and only if $g''(x) \geq 0$ for all $x \in \mathbb{R}$.*

A corollary of this result is that twice-differentiable functions in $\mathbb{R}^n$ are convex if and only if their Hessian is positive semidefinite at every point.

Corollary 2.18. *A twice-differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if for every $\vec{x} \in \mathbb{R}^n$, the Hessian matrix $\nabla^2 f(\vec{x})$ is positive semidefinite.*

Proof. By Lemma 1.3 we just need to show that the univariate function $g_{\vec{a},\vec{b}}$ defined by (4) is convex for all $\vec{a}, \vec{b} \in \mathbb{R}^n$. By Lemma 2.17 this holds if and only if the second derivative of $g_{\vec{a},\vec{b}}$ is nonnegative. This quantity is nonnegative for all $\vec{a}, \vec{b} \in \mathbb{R}^n$ if and only if $\nabla^2 f(\vec{x})$ is positive semidefinite for any $\vec{x} \in \mathbb{R}^n$. □

Remark 2.19 (Strict convexity)**.** *If the Hessian is positive definite, then the function is strictly convex (the proof is essentially the same). However, there are functions that are strictly convex for which the Hessian may equal zero at some points. An example is the univariate function* $f(x) = x^4$*, for which* $f''(0) = 0$*.*

We can interpret Corollary 2.18 in terms of the second-order Taylor expansion of $f : \mathbb{R}^n \to \mathbb{R}$ at $\vec{x}$: f is convex if and only if this quadratic approximation is always convex.

3 Minimizing differentiable convex functions

In this section we describe different techniques for solving the optimization problem

$$\min_{\vec{x} \in \mathbb{R}^n} \quad f(\vec{x}), \tag{45}$$

when f is differentiable and convex. By Theorem 1.4 any local minimum of the function is also a global minimum. This motivates trying to make progress towards a solution by exploiting local first and second order information.

3.1 Gradient descent

Gradient descent exploits first-order local information encoded in the gradient to iteratively approach the point at which f achieves its minimum value. The idea is to take steps in the direction of steepest descent, which is $-\nabla f(\vec{x})$ by Corollary 2.3.

Algorithm 3.1 (Gradient descent, aka steepest descent)**.** *Set the initial point* $\vec{x}^{(0)}$ *to an arbitrary value in* $\mathbb{R}^n$*. Update by setting*

$$\vec{x}^{(k+1)} := \vec{x}^{(k)} - \alpha_k \nabla f\left(\vec{x}^{(k)}\right), \tag{46}$$

where $\alpha_k > 0$ *is a nonnegative real number which we call the step size, until a stopping criterion is met.*

Examples of stopping criteria include checking whether the relative progress

$$\frac{\left|\left|\vec{x}^{(k+1)} - \vec{x}^{(k)}\right|\right|_2}{\left|\left|\vec{x}^{(k)}\right|\right|_2} \tag{47}$$

or the norm of the gradient are below a predefined tolerance. Figure 9 shows two examples in which gradient descent is applied in one and two dimensions. In both cases the method converges to the minimum.

In the examples of Figure 9 the step size is constant. In practice, determining a constant step that is adequate for a particular function can be challenging. Figure 10 shows two examples to illustrate this. In the first, the step size is too small and as a result convergence is extremely slow. In the second the step size is too large which causes the algorithm to repeatedly overshoot the minimum and eventually diverge.

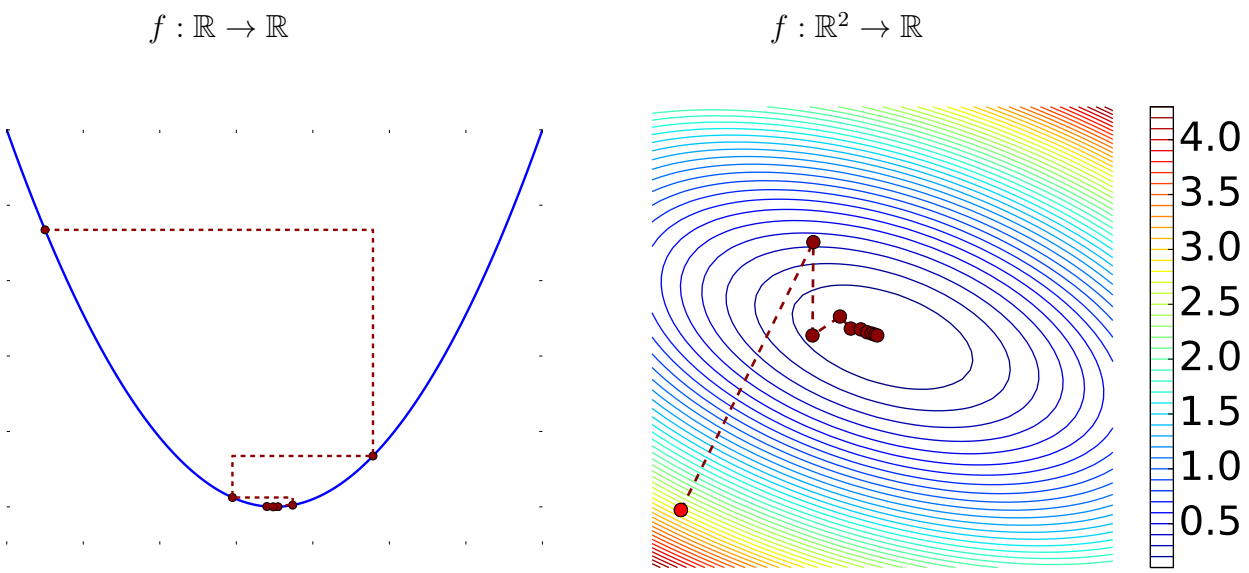

Figure 9: Iterations of gradient descent applied to a univariate (left) and a bivariate (right) function. The algorithm converges to the minimum in both cases.

Ideally, we would like to adapt the step size automatically as the iterations progress. A possibility is to search for the minimum of the function along the direction of the gradient,

$$\alpha_k := \arg\min_{\alpha} h(\alpha) \tag{48}$$

$$= \arg\min_{\alpha \in \mathbb{R}} f\left(\vec{x}^{(k)} - \alpha_k \nabla f\left(\vec{x}^{(k)}\right)\right). \tag{49}$$

This is called a line search. Recall that the restriction of an n-dimensional convex function to a line in its domain is also convex. As a result the line-search problem is a one-dimensional convex problem. However, it may still be costly to solve. The backtracking line search is an alternative heuristic that produces very similar results in practice at less cost. The idea is to ensure that we make some progress in each iteration, without worrying about actually minimizing the univariate function.

Algorithm 3.2 (Backtracking line search with Armijo rule)**.** *Given $\alpha^0 \geq 0$ and $\beta, \eta \in (0, 1)$, set $\alpha_k := \alpha^0 \, \beta^i$ for the smallest integer i such that $\vec{x}^{(k+1)} := \vec{x}^{(k)} - \alpha_k \nabla f\left(\vec{x}^{(k)}\right)$ satisfies*

$$f\left(\vec{x}^{(k+1)}\right) \leq f\left(\vec{x}^{(k)}\right) - \frac{1}{2}\alpha_k \left|\left|\nabla f\left(\vec{x}^{(k)}\right)\right|\right|_2^2, \tag{50}$$

a condition known as Armijo rule

Figure 11 shows the result of applying gradient descent with a backtracking line search to the same example as in Figure 10. In this case, the line search manages to adjust the step size so that the method converges.

Example 3.3 (Gradient descent for least squares)**.** Gradient descent can be used to minimize the least-squares cost function

$$\text{minimize}_{\vec{\beta} \in \mathbb{R}^p} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2, \tag{51}$$

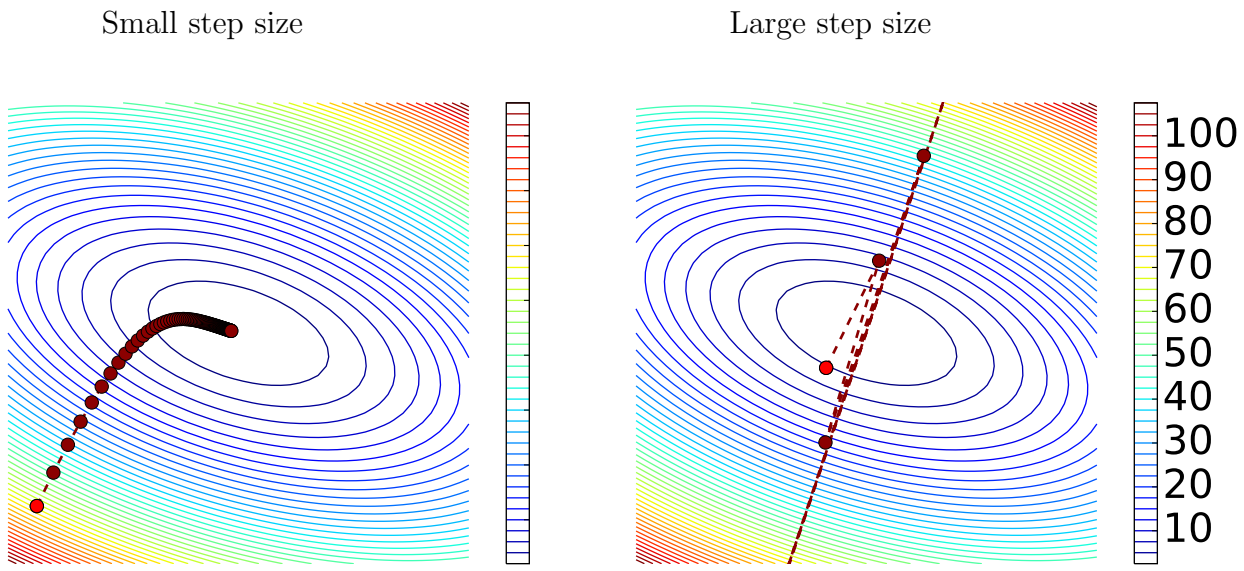

Figure 10: Iterations of gradient descent when the step size is small (left) and large (right). In the first case the convergence is very small, whereas in the second the algorithm diverges away from the minimum. The initial point is bright red.

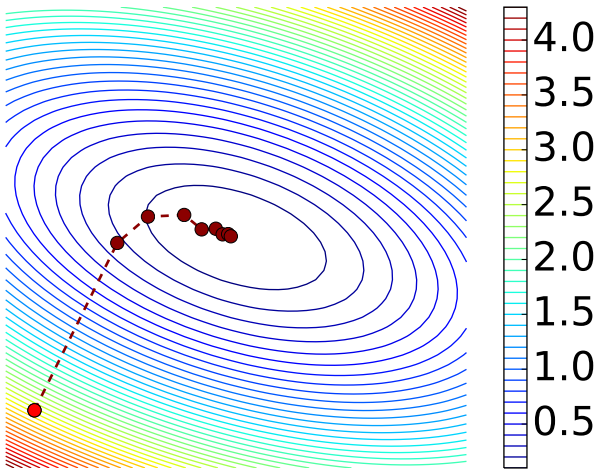

Figure 11: Gradient descent using a backtracking line search based on the Armijo rule. The function is the same as in Figure 10.

described in Section 2 of Lecture Notes 6 to fit a linear regression model from n examples of the form,

$$\left(y^{(1)}, \vec{x}^{\,(1)}\right), \left(y^{(2)}, \vec{x}^{\,(2)}\right), \ldots, \left(y^{(n)}, \vec{x}^{\,(n)}\right). \tag{52}$$

The cost function is convex by Lemma 1.6 (also its Hessian $X^T X$ is positive semidefinite). The gradient of the quadratic function

$$f(\vec{\beta}) := \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 \tag{53}$$

$$= \vec{\beta}^T X^T X \vec{\beta} - 2\vec{\beta}^T X^T \vec{y} + \vec{y}^T \vec{y} \tag{54}$$

equals

$$\nabla f(\vec{\beta}) = 2X^T X\vec{\beta} - 2X^T \vec{y} \tag{55}$$

so the gradient descent updates are

$$\vec{\beta}^{(k+1)} = \vec{\beta}^{(k)} + 2\alpha_k X^T \left(\vec{y} - X\vec{\beta}^{(k)}\right) \tag{56}$$

$$= \vec{\beta}^{(k)} + 2\alpha_k \sum_{i=1}^{n} \left(\vec{y}^{(i)} - \langle x^{(i)}, \vec{\beta}^{(k)}\rangle\right) x^{(i)}. \tag{57}$$

This has a very intuitive interpretation in terms of the examples: if $\vec{y}^{(i)}$ is larger than $\langle x^{(i)}, \vec{\beta}^{(k)}\rangle$ we add a small multiple of $x^{(i)}$ in order to reduce the difference, if it is smaller we subtract it.

Gradient descent is not the best first-order iterative optimization method for least-squares minimization. The conjugate-gradients method is better suited for this problem. We refer to [5] for an excellent tutorial on this method. △

Example 3.4 (Gradient ascent for logistic regression)**.** Since gradient descent minimizes convex functions, gradient ascent (where we climb in the direction of the gradient) can be used to maximize concave functions. In particular, let us consider the logistic regression log-likelihood cost function

$$f(\vec{\beta}) := \sum_{i=1}^{n} y^{(i)} \log g\left(\langle \vec{x}^{\,(i)}, \vec{\beta}\rangle\right) + \left(1 - y^{(i)}\right) \log\left(1 - g\left(\langle \vec{x}^{\,(i)}, \vec{\beta}\rangle\right)\right) \tag{58}$$

from Definition 4.2 of Lecture Notes 6, where $g(t) = (1 - \exp -t)^- 1$ the labels $y^{(i)}$, $1 \leq i \leq n$, are equal to 0 or 1, and the features $\vec{x}^{\,(i)}$ are vectors in $\mathbb{R}^n$. We establish that this cost function is concave in Corollary 3.23. The gradient of this cost function is given in the following lemma.

Lemma 3.5. *The gradient of the function f in equation (58) equals*

$$\nabla f(\vec{\beta}) = \sum_{i=1}^{n} y^{(i)} \left(1 - g(\langle \vec{x}^{\,(i)}, \vec{\beta}\rangle)\right) \vec{x}^{\,(i)} - \left(1 - y^{(i)}\right) g(\langle \vec{x}^{\,(i)}, \vec{\beta}\rangle) \vec{x}^{\,(i)}. \tag{59}$$

Proof. The result follows from the identities

$$g'(t) = g(t)(1 - g(t)), \tag{60}$$

$$(1 - g(t))' = -g(t)(1 - g(t)). \tag{61}$$

and the chain rule. △

The gradient ascent updates

$$\vec{\beta}^{(k+1)} := \vec{\beta}^{(k)} + \alpha_k \sum_{i=1}^{n} y^{(i)} \left(1 - g(\langle \vec{x}^{(i)}, \vec{\beta}^{(k)} \rangle)\right) \vec{x}^{(i)} - \left(1 - y^{(i)}\right) g(\langle \vec{x}^{(i)}, \vec{\beta}^{(k)} \rangle) \vec{x}^{(i)}. \tag{62}$$

have an intuitive interpretation. If $\vec{y}^{(i)}$ equals 1, we add $x^{(i)}$ scaled by the error $1 - g(\langle \vec{x}^{(i)}, \vec{\beta} \rangle)$ to push $g(\langle \vec{x}^{(i)}, \vec{\beta} \rangle)$ up towards $\vec{y}^{(i)}$. Similarly, if $\vec{y}^{(i)}$ equals 0, we subtract $x^{(i)}$ scaled by the error $g(\langle \vec{x}^{(i)}, \vec{\beta} \rangle)$ to push $g(\langle \vec{x}^{(i)}, \vec{\beta} \rangle)$ down towards $\vec{y}^{(i)}$. △

3.2 Convergence of gradient descent

In this section we analyze the convergence of gradient descent. We begin by introducing a notion of continuity for functions from $\mathbb{R}^n$ to $\mathbb{R}^m$.

Definition 3.6 (Lipschitz continuity)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is Lipschitz continuous with Lipschitz constant L if for any $\vec{x}, \vec{y} \in \mathbb{R}^n$*

$$||f(\vec{y}) - f(\vec{x})||_2 \leq L ||\vec{y} - \vec{x}||_2 . \tag{63}$$

We focus on functions that have Lipschitz-continuous gradients. The following theorem shows that these functions are upper bounded by a quadratic function.

Theorem 3.7 (Proof in Section 4.5)**.** *If the gradient of a function $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz continuous with Lipschitz constant L,*

$$||\nabla f(\vec{y}) - \nabla f(\vec{x})||_2 \leq L ||\vec{y} - \vec{x}||_2 \tag{64}$$

then for any $\vec{x}, \vec{y} \in \mathbb{R}^n$

$$f(\vec{y}) \leq f(\vec{x}) + \nabla f(\vec{x})^T (\vec{y} - \vec{x}) + \frac{L}{2} ||\vec{y} - \vec{x}||_2^2 . \tag{65}$$

The quadratic upper bound immediately implies a bound on the value of the cost function after k iterations of gradient descent.

Corollary 3.8. *Let $\vec{x}^{(i)}$ be the ith iteration of gradient descent and $\alpha_i \geq 0$ the ith step size, if ∇f is L-Lipschitz continuous,*

$$f\left(\vec{x}^{(k+1)}\right) \leq f\left(\vec{x}^{(k)}\right) - \alpha_k \left(1 - \frac{\alpha_k L}{2}\right) \left|\left|\nabla f\left(\vec{x}^{(k)}\right)\right|\right|_2^2 . \tag{66}$$

Proof. Applying the quadratic upper bound we obtain

$$f\left(\vec{x}^{(k+1)}\right) \leq f\left(\vec{x}^{(k)}\right) + \nabla f\left(\vec{x}^{(k)}\right)^T \left(\vec{x}^{(k+1)} - \vec{x}^{(k)}\right) + \frac{L}{2} \left|\left|\vec{x}^{(k+1)} - \vec{x}^{(k)}\right|\right|_2^2 . \tag{67}$$

The result follows because $\vec{x}^{(k+1)} - \vec{x}^{(k)} = -\alpha_k \nabla f\left(\vec{x}^{(k)}\right)$. □

We can now establish that if the step size is small enough, the value of the cost function at each iteration will decrease (unless we are at the minimum, where the gradient is zero).

Corollary 3.9 (Gradient descent is a descent method)**.** *If* $\alpha_k \leq \frac{1}{L}$

$$f\left(\vec{x}^{(k+1)}\right) \leq f\left(\vec{x}^{(k)}\right) - \frac{\alpha_k}{2}\left|\left|\nabla f\left(\vec{x}^{(k)}\right)\right|\right|_2^2. \tag{68}$$

Note that up to now we are *not* assuming that the function we are minimizing is convex. Gradient descent will make local progress even for nonconvex functions if the step size is sufficiently small. We now establish global convergence for gradient descent applied to convex functions with Lipschitz-continuous gradients.

Theorem 3.10. *We assume that* f *is convex,* ∇f *is* L*-Lipschitz continuous and there exists a point* $\vec{x}^*$ *at which* f *achieves a finite minimum. If we set the step size of gradient descent to* $\alpha_k = \alpha \leq 1/L$ *for every iteration,*

$$f\left(\vec{x}^{(k)}\right) - f\left(\vec{x}^*\right) \leq \frac{\left|\left|\vec{x}^{(0)} - \vec{x}^*\right|\right|_2^2}{2\,\alpha\,k} \tag{69}$$

Proof. By the first-order characterization of convexity

$$f\left(\vec{x}^{(k-1)}\right) + \nabla f\left(\vec{x}^{(k-1)}\right)^T\left(\vec{x}^* - \vec{x}^{(k-1)}\right) \leq f\left(\vec{x}^*\right), \tag{70}$$

which together with Corollary 3.9 yields

$$\begin{aligned} f\left(\vec{x}^{(k)}\right) - f\left(\vec{x}^*\right) &\leq \nabla f\left(\vec{x}^{(k-1)}\right)^T\left(\vec{x}^{(k-1)} - \vec{x}^*\right) - \frac{\alpha}{2}\left|\left|\nabla f\left(\vec{x}^{(k-1)}\right)\right|\right|_2^2 && (71)\\ &= \frac{1}{2\,\alpha}\left(\left|\left|\vec{x}^{(k-1)} - \vec{x}^*\right|\right|_2^2 - \left|\left|\vec{x}^{(k-1)} - \vec{x}^* - \alpha\nabla f\left(\vec{x}^{(k-1)}\right)\right|\right|_2^2\right) && (72)\\ &= \frac{1}{2\,\alpha}\left(\left|\left|\vec{x}^{(k-1)} - \vec{x}^*\right|\right|_2^2 - \left|\left|\vec{x}^{(k)} - \vec{x}^*\right|\right|_2\right) && (73) \end{aligned}$$

Using the fact that by Corollary 3.9 the value of f never increases, we have

$$\begin{aligned} f\left(\vec{x}^{(k)}\right) - f\left(\vec{x}^*\right) &\leq \frac{1}{k}\sum_{i=1}^{k} f\left(\vec{x}^{(k)}\right) - f\left(\vec{x}^*\right) && (74)\\ &\leq \frac{1}{2\,\alpha\,k}\left(\left|\left|\vec{x}^{(0)} - \vec{x}^*\right|\right|_2^2 - \left|\left|\vec{x}^{(k)} - \vec{x}^*\right|\right|_2^2\right) && (75)\\ &\leq \frac{\left|\left|\vec{x}^{(0)} - \vec{x}^*\right|\right|_2^2}{2\,\alpha\,k}. && (76) \end{aligned}$$

□

The theorem assumes that we know the Lipschitz constant of the gradient beforehand. However, the following lemma establishes that a backtracking line search with the Armijo rule is capable of adjusting the step size adequately.

Lemma 3.11 (Backtracking line search). *If the gradient of a function $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz continuous with Lipschitz constant L the step size obtained by applying a backtracking line search using the Armijo rule with $\eta = 0.5$ satisfies*

$$\alpha_k \geq \alpha_{\min} := \min \left\{ \alpha^0, \frac{\beta}{L} \right\}. \tag{77}$$

Proof. By Corollary 3.8 the Armijo rule with $\eta = 0.5$ is satisfied if $\alpha_k \leq 1/L$. Since there must exist an integer i for which $\beta/L \leq \alpha^0 \beta^i \leq 1/L$ this establishes the result. □

We can now adapt the proof of Theorem 3.10 to establish convergence when we apply a backtracking line search.

Theorem 3.12 (Convergence with backtracking line search). *If f is convex and ∇f is L-Lipschitz continuous. Gradient descent with a backtracking line search produces a sequence of points that satisfy*

$$f\left(\vec{x}^{(k)}\right) - f\left(\vec{x}^*\right) \leq \frac{\left|\left|\vec{x}^{(0)} - \vec{x}^*\right|\right|_2^2}{2\,\alpha_{\min}\,k}, \tag{78}$$

where $\alpha_{\min} := \min\left\{\alpha^0, \frac{\beta}{L}\right\}$.

Proof. Following the reasoning in the proof of Theorem 3.10 up until equation (73) we have

$$f\left(\vec{x}^{(k)}\right) - f\left(\vec{x}^*\right) \leq \frac{1}{2\,\alpha_i}\left(\left|\left|x^{(k-1)} - x^*\right|\right|_2^2 - \left|\left|x^{(k)} - x^*\right|\right|_2\right). \tag{79}$$

By Lemma 3.11 $\alpha_i \geq \alpha_{\min}$, so we just mimic the steps at the end of the proof of Theorem 3.10 to obtain

$$\begin{aligned} f\left(x^{(k)}\right) - f\left(x^*\right) &\leq \frac{1}{k}\sum_{i=1}^{k} f\left(x^{(k)}\right) - f\left(x^*\right) && (80)\\ &= \frac{1}{2\,\alpha_{\min}\,k}\left(\left|\left|x^{(0)} - x^*\right|\right|_2^2 - \left|\left|x^{(k)} - x^*\right|\right|_2^2\right) && (81)\\ &\leq \frac{\left|\left|x^{(0)} - x^*\right|\right|_2^2}{2\,\alpha_{\min}\,k}. && (82) \end{aligned}$$

□

The results that we have proved imply that we need $\mathcal{O}\left(1/\epsilon\right)$ to compute a point at which the cost function has a value that is ϵ close to the minimum. However, in practice gradient descent and related methods often converge much faster.

3.3 Accelerated gradient descent

The following theorem by Nesterov shows that no algorithm that uses first-order information can converge faster than $\mathcal{O}\left(1/\sqrt{\epsilon}\right)$ for the class of functions with Lipschitz-continuous gradients. The proof is constructive, see Section 2.1.2 of [4] for the details.

Theorem 3.13 (Lower bound on rate of convergence)**.** *There exist convex functions with L-Lipschitz-continuous gradients such that for any algorithm that selects $x^{(k)}$ from*

$$x^{(0)} + \text{span}\left\{\nabla f\left(x^{(0)}\right), \nabla f\left(x^{(1)}\right), \ldots, \nabla f\left(x^{(k-1)}\right)\right\} \tag{83}$$

we have

$$f\left(x^{(k)}\right) - f\left(x^*\right) \geq \frac{3L\left|\left|x^{(0)} - x^*\right|\right|_2^2}{32\left(k+1\right)^2}. \tag{84}$$

This rate is in fact optimal. The convergence of $\mathcal{O}\left(1/\sqrt{\epsilon}\right)$ can be achieved if we modify gradient descent by adding a momentum term.

Algorithm 3.14 (Nesterov's accelerated gradient descent)**.** *Set the initial point $\vec{x}^{(0)}$ to an arbitrary value in $\mathbb{R}^n$. Update by setting*

$$y^{(k+1)} = x^{(k)} - \alpha_k \nabla f\left(x^{(k)}\right), \tag{85}$$

$$x^{(k+1)} = \beta_k\, y^{(k+1)} + \gamma_k\, y^{(k)}, \tag{86}$$

where α_k is the step size and β_k and γ_k are nonnegative real parameters, until a stopping criterion is met.

Intuitively, the momentum term $y^{(k)}$ prevents the algorithm from overreacting to changes in the local slope of the function. We refer the interested reader to [3, 4] for more details.

Example 3.15 (Digit classification)**.** In this example we apply both gradient descent and accelerated gradient descent to train a logistic-regression model on the MNIST data set[1]. We consider the task of determining whether a digit is a 5 or not. The feature vector $\vec{x}_i$ contains the pixel values of an image of a 5 ($\vec{y}_i = 1$) or another number ($\vec{y}_i = 0$). We use different numbers of training examples to fit a logistic regression model. The cost function is maximized by running gradient descent and accelerated gradient descent until the gradient is smaller than a certain value. Figure 12 shows the time taken by both algorithms for different training-set sizes. △

3.4 Stochastic gradient descent

Cost functions used to fit models from data are often additive, in the sense that we can write them as a sum of m terms, each of which often depends on just one measurement,

$$f\left(\vec{x}\right) = \frac{1}{m}\sum_{i=1}^{m} f_i\left(\vec{x}\right). \tag{87}$$

[1] Available at http://yann.lecun.com/exdb/mnist/

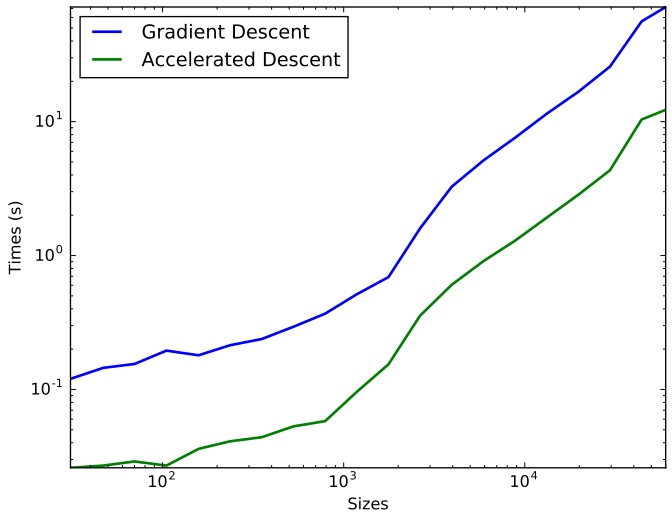

Figure 12: Time taken by gradient descent and accelerated gradient descent to train a logistic-regression model on the MNIST data set for different training-set sizes.

Two important examples are the least-squares and logistic-regression log-likelihood functions discussed in Examples 3.3 and 3.4. In those cases each term f_i corresponds to a different example in the training set. If the training set is extremely large, then computing the whole gradient may be computationally infeasible. Stochastic gradient descent circumvents this issue by using the gradient of individual components instead.

Algorithm 3.16 (Stochastic gradient descent)**.** *Set the initial point $\vec{x}^{(0)}$ to an arbitrary value in $\mathbb{R}^n$. Until a stopping criterion is met, update by:*

1. *Choosing a random subset of b indices $\boldsymbol{\mathcal{B}}$, where $b \ll m$ is the batch size.*
2. *Setting*

$$\vec{x}^{(k+1)} := \vec{x}^{(k)} - \alpha_k \sum_{i \in \boldsymbol{\mathcal{B}}} \nabla f_i \left(\vec{x}^{(k)} \right) \tag{88}$$

where $\alpha_k > 0$ is the step size.

Apart from its computational efficiency, an advantage of stochastic gradient descent is that it can be applied in *online* settings, where we need to optimize a function that depends on a large data set but only have access to portions of the data set at a time.

Intuitively, stochastic gradient descent replaces the gradient of the whole function by

$$\sum_{i=1}^{m} 1_{i \in \boldsymbol{\mathcal{B}}} \nabla f_i \left(\vec{x}^{(k+1)} \right). \tag{89}$$

If $\boldsymbol{\mathcal{B}}$ is generated so that every index has the same probability p of belonging to it, then this is an

unbiased estimate of ∇f

$$\mathrm{E}\left(\sum_{i=1}^{m} 1_{i\in\boldsymbol{\mathcal{B}}}\nabla f_i\left(\vec{x}^{(k)}\right)\right) = \sum_{i=1}^{m}\mathrm{E}\left(1_{i\in\boldsymbol{\mathcal{B}}}\right)\nabla f_i\left(\vec{x}^{(k)}\right) \tag{90}$$

$$= \sum_{i=1}^{m}\mathrm{P}\left(i\in\boldsymbol{\mathcal{B}}\right)\nabla f_i\left(\vec{x}^{(k)}\right) \tag{91}$$

$$= mp\nabla f\left(\vec{x}^{(k)}\right). \tag{92}$$

Intuitively, the direction of the stochastic-gradient descent is the one of steepest descent *on average*. However, the variation in the estimate can make stochastic gradient descent diverge unless the step size is diminishing. We refer to [1] for more details on this algorithm.

Example 3.17 (Stochastic gradient descent for least squares and logistic regression)**.** By the derivation in Example 3.3, in the case of least squares the stochastic gradient descent update is

$$\vec{\beta}^{(k+1)} := \vec{\beta}^{(k)} + 2\alpha_k \sum_{i\in\boldsymbol{\mathcal{B}}}\left(\vec{y}^{(i)} - \langle x^{(i)}, \vec{\beta}^{(k)}\rangle\right)x^{(i)}, \tag{93}$$

Similarly, by the derivation in Example 3.4, the update for stochastic gradient ascent applied to logistic regression is

$$\vec{\beta}^{(k+1)} := \vec{\beta}^{(k)} + \alpha_k \sum_{i\in\boldsymbol{\mathcal{B}}} y^{(i)}\left(1 - g(\langle\vec{x}^{(i)}, \vec{\beta}^{(k)}\rangle)\right)\vec{x}^{(i)} - \left(1-y^{(i)}\right)g(\langle\vec{x}^{(i)}, \vec{\beta}^{(k)}\rangle)\vec{x}^{(i)}. \tag{94}$$

In both cases the algorithm is very intuitive: instead of adjusting $\vec{\beta}$ using all of the examples, we just use the ones in the batch. △

Example 3.18 (Digit classification)**.** In this example we apply stochastic gradient descent for to train a logistic-regression model on the MNIST data set for the same task as Example 3.18. Figure 13 shows the convergence of the algorithm for different batch sizes. △

3.5 Newton's method

Newton's method minimizes a convex function by iteratively minimizing its quadratic approximation. The following simple lemma derives a closed form for the minimum of the quadratic approximation at a given point.

Lemma 3.19. *The minimum of the second-order approximation of a convex function f at $\vec{x}\in\mathbb{R}^n$*

$$f^2_{\vec{x}}\left(\vec{y}\right) := f\left(\vec{x}\right) + \nabla f\left(\vec{x}\right)\left(\vec{y}-\vec{x}\right) + \frac{1}{2}\left(\vec{y}-\vec{x}\right)^T\nabla^2 f\left(\vec{x}\right)\left(\vec{y}-\vec{x}\right), \tag{95}$$

which has a positive definite Hessian at $\vec{x}$, is equal to

$$\underset{\vec{y}\in\mathbb{R}^n}{\arg\min}\, f^2_{\vec{x}}\left(\vec{y}\right) = \vec{x} - \nabla^2 f\left(\vec{x}\right)^{-1}\nabla f\left(\vec{x}\right). \tag{96}$$

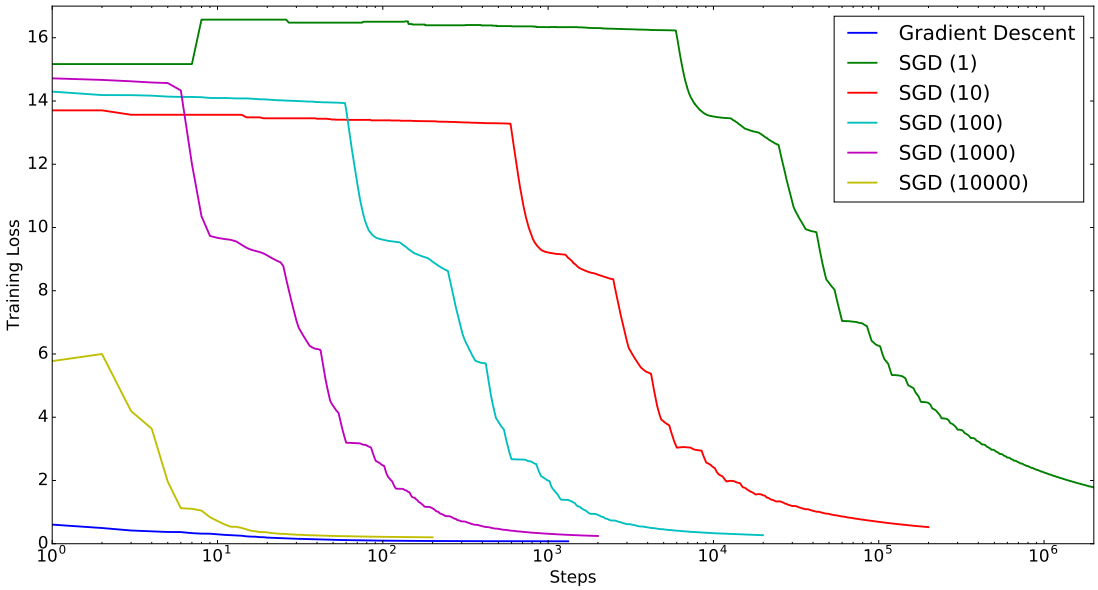

Figure 13: Convergence of stochastic gradient descent when fitting a logistic-regression model on the MNIST data set. The plot shows the value of the cost function on the training set for different batch sizes. Note that the updates for smaller batch sizes are much faster so the horizontal axis does not reflect running time.

Proof. If the Hessian is positive definite, then $f_{\vec{x}}^2$ is strictly convex and it has a unique global minimum. Its gradient equals

$$\nabla f_{\vec{x}}^2 \left(y \right) = \nabla f \left(\vec{x} \right) + \nabla^2 f \left(\vec{x} \right) \left(\vec{y} - \vec{x} \right) \tag{97}$$

so it is equal to zero if

$$\nabla^2 f \left(\vec{x} \right) \left(\vec{y} - \vec{x} \right) = -\nabla f \left(\vec{x} \right). \tag{98}$$

If the Hessian is positive definite, then it is also full rank. The only solution to this system of equations is consequently $\vec{y} = \vec{x} - \nabla^2 f \left(\vec{x} \right)^{-1} \nabla f \left(\vec{x} \right)$, which must be the minimum of $f_{\vec{x}}^2$ by Corollary 2.6 because the gradient vanishes. □

The idea behind Newton's method is that convex functions are often well approximated by quadratic functions, especially close to their minimum.

Algorithm 3.20 (Newton's method)**.** *Set the initial point $\vec{x}^{\,(0)}$ to an arbitrary value in $\mathbb{R}^n$. Update by setting*

$$\vec{x}^{\,(k+1)} := \vec{x}^{\,(k)} - \nabla^2 f \left(\vec{x}^{\,(k)} \right)^{-1} \nabla f \left(\vec{x}^{\,(k)} \right) \tag{99}$$

until a stopping criterion is met.

Figure 14 illustrates Newton's method in a one-dimensional setting. When applied to a quadratic function, the algorithm converges in one step: if we start at the origin it is equivalent to computing

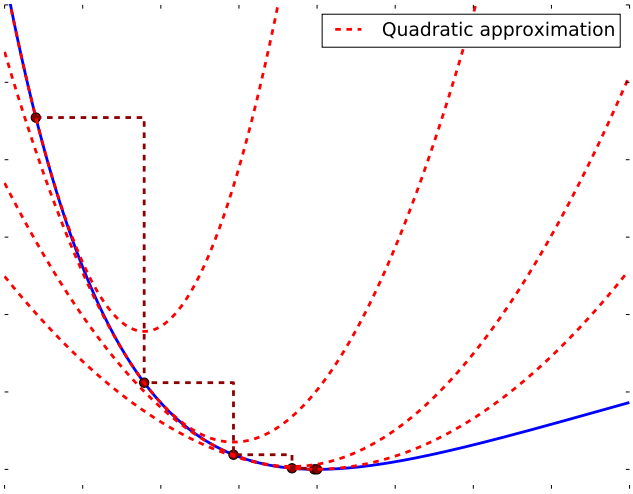

Figure 14: Newton's method applied to a one-dimensional convex function. The quadratic approximations to the function at each iteration are depicted in red.

the closed-form solution for least squares derived in Theorem 2.1 of Lectures Notes 6. This is illustrated in Figure 15, along with another example where the function is convex but not quadratic. Newton's method can provide significant acceleration for problems of moderate sizes where the quadratic approximation is accurate, but often inverting the Hessian may be computationally expensive.

Example 3.21 (Newton's method for logistic regression)**.** The following lemma derives the Hessian of the logistic regression log-likelihood cost function (58).

Lemma 3.22. *The Hessian of the function f in equation (58) equals*

$$\nabla^2 f(\vec{\beta}) = -X^T G(\vec{\beta}) X, \tag{100}$$

where the rows of $X \in \mathbb{R}^{n \times p}$ contain the feature vectors $\vec{x}^{\,(1)}, \ldots \vec{x}^{\,(n)}$ and G is a diagonal matrix such that

$$G(\vec{\beta})_{ii} := g(\langle \vec{x}^{\,(i)}, \vec{\beta} \rangle) \left(1 - g(\langle \vec{x}^{\,(i)}, \vec{\beta} \rangle)\right), \qquad 1 \leq i \leq n. \tag{101}$$

Proof. By the identities (60) and (61) and the chain rule we have

$$\frac{\partial^2 f(\vec{x})}{\partial \vec{x}[j] \partial \vec{x}[l]} = -\sum_{i=1}^{n} g(\langle \vec{x}^{\,(i)}, \vec{\beta} \rangle) \left(1 - g(\langle \vec{x}^{\,(i)}, \vec{\beta} \rangle)\right) \vec{x}^{\,(i)}[j] \vec{x}^{\,(i)}[l] \tag{102}$$

for $1 \leq j, l \leq p$. △

Corollary 3.23. *The logistic regression log-likelihood cost function (58) is concave.*

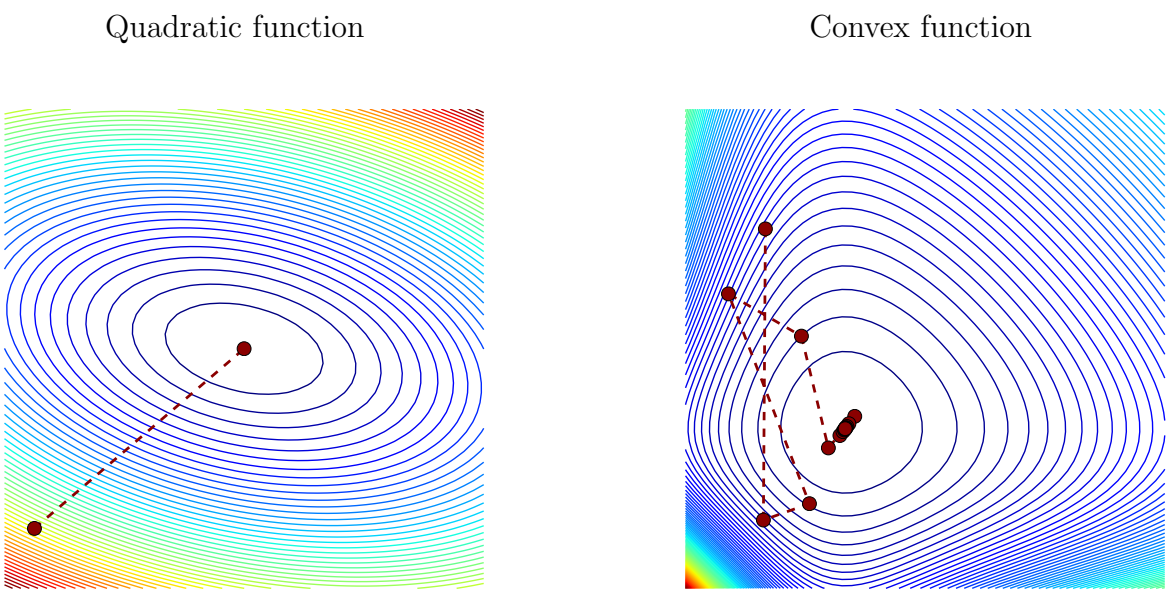

Figure 15: Newton's method applied to a quadratic function (left) and to a convex function that is not quadratic.

Proof. The Hessian is negative semidefinite, for any arbitrary $\vec{\beta}, \vec{v} \in \mathbb{R}^p$

$$\vec{v}^T \nabla^2 f(\vec{\beta}) \vec{v} = -\sum_{i=1}^{n} G(\vec{\beta})_{ii} \left(X\vec{v}\right)[i]^2 \leq 0 \tag{103}$$

since the entries of $G(\vec{\beta})$ are nonnegative. △

The Newton updates are consequently of the form

$$\vec{\beta}^{(k+1)} := \vec{\beta}^{(k)} - \left(X^T G(\vec{\beta}^{(k)}) X\right)^{-1} \nabla f(\vec{\beta}^{(k)}). \tag{104}$$

A potentially problematic feature of this application of Newton's method is that the Hessian $X^T G(\vec{\beta}) X$ may become ill conditioned if most of the examples are classified correctly, since in that case the matrix G mostly contains zeros. Intuitively, in this case the cost function is very *flat* in certain directions. △

4 Proofs

4.1 Proof of Lemma 1.3

The proof for strict convexity is exactly the same, replacing the inequalities by strict inequalities.

f being convex implies that $g_{\vec{x},\vec{y}}$ is convex for any $\vec{x}, \vec{y} \in \mathbb{R}^n$

For any $\alpha, \beta, \theta \in (0,1)$

$$
\begin{aligned}
g_{\vec{x},\vec{y}}\left(\theta\alpha + (1-\theta)\,\beta\right) &= f\left(\left(\theta\alpha + (1-\theta)\,\beta\right)\vec{x} + \left(1 - \theta\alpha - (1-\theta)\,\beta\right)\vec{y}\right) && (105)\\
&= f\left(\theta\left(\alpha\vec{x} + (1-\alpha)\,\vec{y}\right) + (1-\theta)\left(\beta\vec{x} + (1-\beta)\,\vec{y}\right)\right) && (106)\\
&\leq \theta f\left(\alpha\vec{x} + (1-\alpha)\,\vec{y}\right) + (1-\theta)\, f\left(\beta\vec{x} + (1-\beta)\,\vec{y}\right) \quad \text{by convexity of } f\\
&= \theta g_{\vec{x},\vec{y}}\left(\alpha\right) + (1-\theta)\, g_{\vec{x},\vec{y}}\left(\beta\right). && (107)
\end{aligned}
$$

$g_{\vec{x},\vec{y}}$ being convex for any $\vec{x}, \vec{y} \in \mathbb{R}^n$ implies that f is convex

For any $\alpha, \beta, \theta \in (0,1)$

$$
\begin{aligned}
f\left(\theta\vec{x} + (1-\theta)\,\vec{y}\right) &= g_{\vec{x},\vec{y}}\left(\theta\right) && (108)\\
&\leq \theta g_{\vec{x},\vec{y}}\left(1\right) + (1-\theta)\, g_{\vec{x},\vec{y}}\left(0\right) \quad \text{by convexity of } g_{\vec{x},\vec{y}} && (109)\\
&= \theta f\left(\vec{x}\right) + (1-\theta)\, f\left(\vec{y}\right). && (110)
\end{aligned}
$$

4.2 Proof of Theorem 2.5

The proof for strict convexity is almost exactly the same; we omit the details.

The following lemma, proved in Section 4.3 below establishes that the result holds for univariate functions

Lemma 4.1. *A univariate differentiable function $g : \mathbb{R} \to \mathbb{R}$ is convex if and only if for all $x, y \in \mathbb{R}$*

$$
g\left(y\right) \geq g'\left(x\right)\left(y - x\right) + g\left(x\right) \tag{111}
$$

and strictly convex if and only if for all $x, y \in \mathbb{R}$

$$
g\left(y\right) > g'\left(x\right)\left(y - x\right) + g\left(x\right). \tag{112}
$$

To complete the proof we extend the result to the multivariable case using Lemma 1.3.

If $f\left(\vec{y}\right) \geq f\left(\vec{x}\right) + \nabla f\left(\vec{x}\right)^T\left(\vec{y} - \vec{x}\right)$ for any $\vec{x}, \vec{y} \in \mathbb{R}^n$ then f is convex

By Lemma 1.3 we just need to show that the univariate function $g_{\vec{a},\vec{b}}$ defined by (4) is convex for all $\vec{a}, \vec{b} \in \mathbb{R}^n$. Applying some basic multivariate calculus yields

$$
g'_{\vec{a},\vec{b}}\left(\alpha\right) = \nabla f\left(\alpha\,\vec{a} + (1-\alpha)\,\vec{b}\right)^T\left(\vec{a} - \vec{b}\right). \tag{113}
$$

Let $\alpha, \beta \in \mathbb{R}$. Setting $\vec{x} := \alpha\,\vec{a} + (1-\alpha)\,\vec{b}$ and $\vec{y} := \beta\,\vec{a} + (1-\beta)\,\vec{b}$ we have

$$
\begin{aligned}
g_{\vec{a},\vec{b}}\left(\beta\right) &= f\left(\vec{y}\right) && (114)\\
&\geq f\left(\vec{x}\right) + \nabla f\left(\vec{x}\right)^T\left(y - \vec{x}\right) && (115)\\
&= f\left(\alpha\,\vec{a} + (1-\alpha)\,\vec{b}\right) + \nabla f\left(\alpha\,\vec{a} + (1-\alpha)\,\vec{b}\right)^T\left(\vec{a} - \vec{b}\right)(\beta - \alpha) && (116)\\
&= g_{\vec{a},\vec{b}}\left(\alpha\right) + g'_{\vec{a},\vec{b}}\left(\alpha\right)(\beta - \alpha) \quad \text{by (113)}, && (117)
\end{aligned}
$$

which establishes that $g_{\vec{a},\vec{b}}$ is convex by Lemma 4.1 above.

If f is convex then $f(\vec{y}) \geq f(\vec{x}) + \nabla f(\vec{x})^T (\vec{y} - \vec{x})$ for any $\vec{x}, \vec{y} \in \mathbb{R}^n$

By Lemma 1.3, $g_{\vec{x},\vec{y}}$ is convex for any $\vec{x}, y \in \mathbb{R}^n$.

$$
\begin{aligned}
f(\vec{y}) &= g_{\vec{x},\vec{y}}(1) && (118)\\
&\geq g_{\vec{x},\vec{y}}(0) + g'_{\vec{x},\vec{y}}(0) \quad \text{by convexity of } g_{\vec{x},\vec{y}} \text{ and Lemma 4.1} && (119)\\
&= f(\vec{x}) + \nabla f(\vec{x})^T (y - \vec{x}) \quad \text{by (113)}. && (120)
\end{aligned}
$$

4.3 Proof of Lemma 4.1

g being convex implies $g(y) \geq g'(x)(y - x) + g(x)$ for all $x, y \in \mathbb{R}$

If g is convex then for any $x, y \in \mathbb{R}$ and any $0 \leq \theta \leq 1$

$$\theta (g(y) - g(x)) + g(x) \geq g(x + \theta(y - x)). \tag{121}$$

Rearranging the terms we have

$$g(y) \geq \frac{g(x + \theta(y - x)) - g(x)}{\theta} + g(x). \tag{122}$$

Setting $h = \theta(y - x)$, this implies

$$g(y) \geq \frac{g(x + h) - g(x)}{h}(y - x) + g(x). \tag{123}$$

Taking the limit when $h \to 0$ yields

$$g(y) \geq g'(x)(y - x) + g(x). \tag{124}$$

If $g(y) \geq g'(x)(y - x) + g(x)$ for all $x, y \in \mathbb{R}$ then g is convex

Let $z = \theta x + (1 - \theta) y$, then by if $g(y) \geq g'(x)(y - x) + g(x)$

$$
\begin{aligned}
g(x) &\geq g'(z)(x - z) + g(z) && (125)\\
&= g'(z)(1 - \theta)(x - y) + g(z) && (126)\\
g(y) &\geq g'(z)(y - z) + g(z) && (127)\\
&= g'(z)\theta(y - x) + g(z) && (128)
\end{aligned}
$$

Multiplying (126) by θ, then (128) by $1 - \theta$ and summing the inequalities, we obtain

$$\theta g(x) + (1 - \theta) g(y) \geq g(\theta x + (1 - \theta) y). \tag{129}$$

4.4 Proof of Lemma 2.17

The second derivative of g is nonnegative anywhere if and only if the first derivative is nondecreasing, because g'' is the derivative of g'.

If g is convex g' is nondecreasing

By Lemma 4.1, if the function is convex then for any $x, y \in \mathbb{R}$ such that $y > x$

$$g(x) \geq g'(y)(x-y) + g(y), \tag{130}$$
$$g(y) \geq g'(x)(y-x) + g(x). \tag{131}$$

Rearranging, we obtain

$$g'(y)(y-x) \geq g(y) - g(x) \geq g'(x)(y-x). \tag{132}$$

Since $y - x > 0$, we have $g'(y) \geq g'(x)$.

If g' is nondecreasing, g is convex

For arbitrary $x, y, \theta \in \mathbb{R}$, such that $y > x$ and $0 < \theta < 1$, let $\eta = \theta y + (1-\theta)x$. Since $y > \eta > x$, by the mean-value theorem there exist $\gamma_1 \in [x, \eta]$ and $\gamma_2 \in [\eta, y]$ such that

$$g'(\gamma_1) = \frac{g(\eta) - g(x)}{\eta - x}, \tag{133}$$
$$g'(\gamma_2) = \frac{g(y) - g(\eta)}{y - \eta}. \tag{134}$$

Since $\gamma_1 < \gamma_2$, if g' is nondecreasing

$$\frac{g(y) - g(\eta)}{y - \eta} \geq \frac{g(\eta) - g(x)}{\eta - x}, \tag{135}$$

which implies

$$\frac{\eta - x}{y - x} g(y) + \frac{y - \eta}{y - x} g(x) \geq g(\eta). \tag{136}$$

Recall that $\eta = \theta y + (1-\theta)x$, so that $\theta = (\eta - x)/(y - x)$ and $\theta = (\eta - x)/(y - x)$ and $1 - \theta = (y - \eta)/(y - x)$. (136) is consequently equivalent to

$$\theta g(y) + (1-\theta) g(x) \geq g(\theta y + (1-\theta)x). \tag{137}$$

4.5 Proof of Proposition 3.7

Consider the function

$$g(\vec{x}) := \frac{L}{2} \vec{x}^T \vec{x} - f(\vec{x}). \tag{138}$$

We first establish that g is convex using the following lemma, proved in Section 4.6 below.

Lemma 4.2 (Monotonicity of gradient)**.** *A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if*

$$\left(\nabla f\left(\vec{y}\right) - \nabla f\left(\vec{x}\right)\right)^T \left(\vec{y} - \vec{x}\right) \geq 0. \tag{139}$$

By the Cauchy-Schwarz inequality, Lipschitz continuity of the gradient of f implies

$$\left(\nabla f\left(\vec{y}\right) - \nabla f\left(\vec{x}\right)\right)^T \left(\vec{y} - \vec{x}\right) \leq L \left|\left|\vec{y} - \vec{x}\right|\right|_2^2, \tag{140}$$

for any $\vec{x}, \vec{y} \in \mathbb{R}^n$. This directly implies

$$\begin{aligned}
\left(\nabla g\left(\vec{y}\right) - \nabla g\left(\vec{x}\right)\right)^T \left(\vec{y} - \vec{x}\right) &= \left(L\vec{y} - L\vec{x} + \nabla f\left(\vec{x}\right) - \nabla f\left(\vec{y}\right)\right)^T \left(\vec{y} - \vec{x}\right) && (141)\\
&= L \left|\left|\vec{y} - \vec{x}\right|\right|_2^2 - \left(\nabla f\left(\vec{y}\right) - \nabla f\left(\vec{x}\right)\right)^T \left(\vec{y} - \vec{x}\right) && (142)\\
&\geq 0 && (143)
\end{aligned}$$

and hence that g is convex. By the first-order condition for convexity,

$$\begin{aligned}
\frac{L}{2}\vec{y}^T\vec{y} - f\left(\vec{y}\right) &= g\left(\vec{y}\right) && (144)\\
&\geq g\left(\vec{x}\right) + \nabla g\left(\vec{x}\right)^T \left(\vec{y} - \vec{x}\right) && (145)\\
&= \frac{L}{2}\vec{x}^T\vec{x} - f\left(\vec{x}\right) + \left(L\vec{x} - \nabla f\left(\vec{x}\right)\right)^T \left(\vec{y} - \vec{x}\right). && (146)
\end{aligned}$$

Rearranging the inequality we conclude that

$$f\left(\vec{y}\right) \leq f\left(\vec{x}\right) + \nabla f\left(\vec{x}\right)^T \left(\vec{y} - \vec{x}\right) + \frac{L}{2} \left|\left|\vec{y} - \vec{x}\right|\right|_2^2. \tag{147}$$

4.6 Proof of Lemma 4.2

Convexity implies $\left(\nabla f\left(\vec{y}\right) - \nabla f\left(\vec{x}\right)\right)^T \left(\vec{y} - \vec{x}\right) \geq 0$ for all $\vec{x}, \vec{y} \in R^n$

If f is convex, by the first-order condition for convexity

$$\begin{aligned}
f\left(\vec{y}\right) &\geq f\left(\vec{x}\right) + \nabla f\left(\vec{x}\right)^T \left(\vec{y} - \vec{x}\right), && (148)\\
f\left(\vec{x}\right) &\geq f\left(\vec{y}\right) + \nabla f\left(\vec{y}\right)^T \left(\vec{x} - \vec{y}\right), && (149)\\
& && (150)
\end{aligned}$$

Adding the two inequalities directly implies the result.

$\left(\nabla f\left(\vec{y}\right) - \nabla f\left(\vec{x}\right)\right)^T \left(\vec{y} - \vec{x}\right) \geq 0$ for all $\vec{x}, \vec{y} \in R^n$ implies convexity

Recall the univariate function $g_{a,b} : [0, 1] \to \mathbb{R}$ defined by

$$g_{a,b}\left(\alpha\right) := f\left(\alpha a + \left(1 - \alpha\right) b\right), \tag{151}$$

for any $a, b \in \mathbb{R}^n$. By multivariate calculus, $g'_{a,b}\left(\alpha\right) = \nabla f\left(\alpha\, a + \left(1 - \alpha\right) b\right)^T \left(a - b\right)$. For any $\alpha \in (0, 1)$ we have

$$\begin{aligned}
g'_{a,b}\left(\alpha\right) - g'_{a,b}\left(0\right) &= \left(\nabla f\left(\alpha\, a + \left(1 - \alpha\right) b\right) - \nabla f\left(b\right)\right)^T \left(a - b\right) && (152)\\
&= \frac{1}{\alpha}\left(\nabla f\left(\alpha\, a + \left(1 - \alpha\right) b\right) - \nabla f\left(b\right)\right)^T \left(\alpha\, a + \left(1 - \alpha\right) b - b\right) && (153)\\
&\geq 0 \quad \text{because } \left(\nabla f\left(y\right) - \nabla f\left(x\right)\right)^T \left(y - x\right) \geq 0 \text{ for any } x, y. && (154)
\end{aligned}$$

This allows us to prove that the first-order condition for convexity holds. For any $\vec{x}, \vec{y}$

$$
\begin{aligned}
f(\vec{x}) &= g_{\vec{x},\vec{y}}(1) && (155)\\
&= g_{\vec{x},\vec{y}}(0) + \int_0^1 g'_{\vec{x},\vec{y}}(\alpha)\ \mathrm{d}\alpha && (156)\\
&\geq g_{\vec{x},\vec{y}}(0) + g'_{\vec{x},\vec{y}}(0) && (157)\\
&= f(\vec{y}) + \nabla f(\vec{y})(\vec{x} - \vec{y}). && (158)
\end{aligned}
$$

References

A very readable and exhaustive reference on convex optimization is Boyd and Vandenberghe's seminal book [2].

[1] L. Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[3] S. Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.

[4] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Publishing Company, Incorporated, 1 edition, 2014.

[5] J. R. Shewchuk et al. An introduction to the conjugate gradient method without the agonizing pain, 1994.