Lecture notes 3 February 8, 2016

Optimization methods

1 Introduction

In these notes we provide an overview of a selection of optimization methods. We focus on methods which rely on first-order information, i.e. gradients and subgradients, to make local progress towards a solution. In practice, these algorithms tend to converge to medium-precision solutions very rapidly and scale reasonably well with the problem dimension. As a result, they are widely used in machine-learning and signal-processing applications.

2 Differentiable functions

We begin by presenting gradient descent, a very simple algorithm that provably converges to the minimum of a convex differentiable function.

2.1 Gradient descent

Consider the optimization problem

$$\text{minimize} \quad f(x), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable and convex. Gradient descent exploits first-order local information encoded in the gradient to iteratively approach the point at which f achieves its minimum value. By multivariable calculus, at any point $x \in \mathbb{R}^n$ $-\nabla f(x)$ is the direction in which f decreases the most, see Figure 1. If we have no additional information about f, it makes sense to move in this direction at each iteration.

Algorithm 2.1 (Gradient descent, aka steepest descent)**.** *We set the initial point $x^{(0)}$ to an arbitrary value in $\mathbb{R}^n$. Then we apply*

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

$\alpha_k > 0$ *is a nonnegative real number which we call the step size.*

Gradient descent can be run for a certain number of iterations, which might depend on computational constraints, or until a stopping criterion is met. An example of a stopping rule is checking whether the relative progress $\left|\left|x^{(k+1)} - x^{(k)}\right|\right|_2 / \left|\left|x^{(k)}\right|\right|_2$ is below a certain

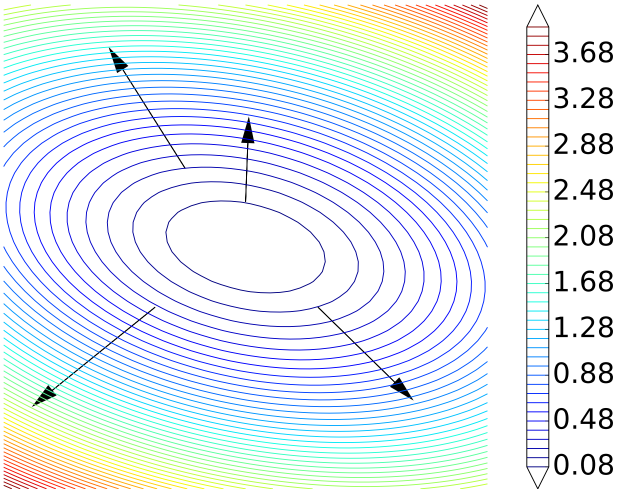

Figure 1: Contour lines of a function $f : \mathbb{R}^2 \to \mathbb{R}$. The gradients at different points are represented by black arrows, which are orthogonal to the contour lines.

value. Figure 2 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 2 the step size is constant. In practice, determining a constant step that is adequate for a particular function can be challenging. Figure 3 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.

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,

$$\alpha_k := \arg\min_{\alpha} h(\alpha) \tag{3}$$

$$= \arg\min_{\alpha \in \mathbb{R}} f\left(x^{(k)} - \alpha_k \nabla f\left(x^{(k)}\right)\right). \tag{4}$$

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

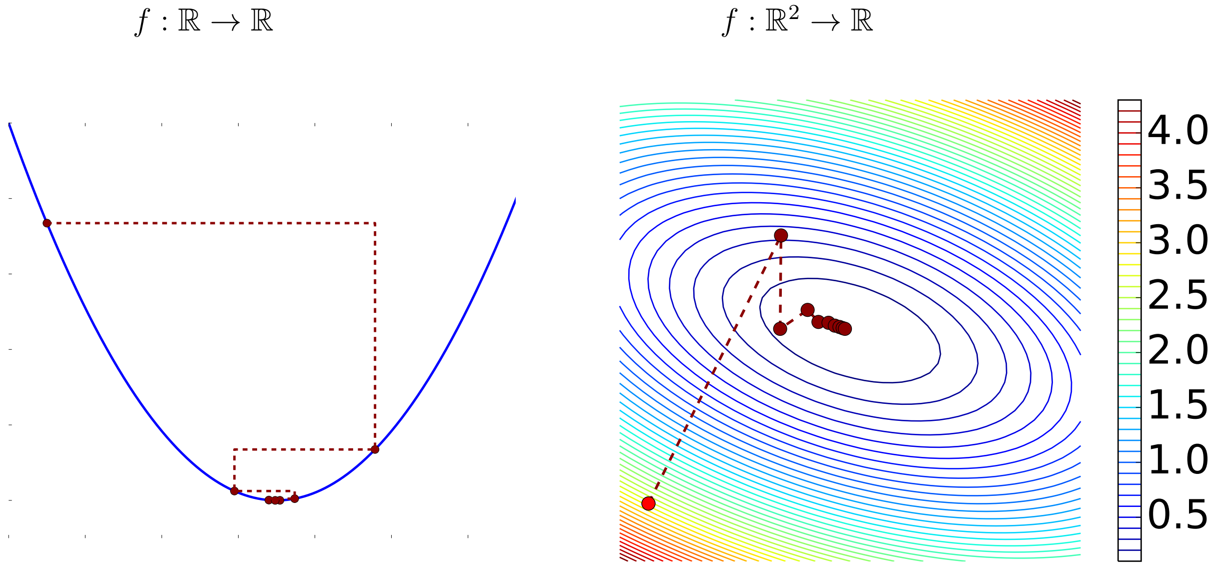

Figure 2: Iterations of gradient descent applied to a univariate (left) and a bivariate (right) function. The algorithm converges to the minimum in both cases.

minimizing the univariate function. By the first-order characterization of convexity we have

$$\begin{aligned} h(\alpha) &= f(x - \alpha \nabla f(x)) && (5)\\ &\geq f(x) - \nabla f(x)^T (x - (x - \alpha \nabla f(x))) && (6)\\ &= f(x) - \alpha \left|\left|\nabla f(x)\right|\right|_2^2 && (7)\\ &= h(0) - \alpha \left|\left|\nabla f(x)\right|\right|_2^2 . && (8)\end{aligned}$$

The backtracking line search starts at a large value of α and decreases it until the function is below $f(x) - \frac{1}{2}\left|\left|\nabla f(x)\right|\right|_2^2$, a condition known as Armijo rule. Note that the Armijo rule will be satisfied eventually. The reason is that the line $h(0) - \alpha \left|\left|\nabla f(x)\right|\right|_2^2$ is the only supporting line of h at zero because h is differentiable and convex (so the only subgradient at a point is the gradient). Consequently $h(\alpha)$ must be below the line $h(0) - \frac{\alpha}{2}\left|\left|f(x)\right|\right|_2^2$ as $\alpha \to 0$, because otherwise this other line would also support h at zero.

Algorithm 2.2 (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*

$$f\left(x^{(k+1)}\right) \leq f\left(x^{(k)}\right) - \frac{1}{2}\,\alpha_k \left|\left|\nabla f\left(x^{(k)}\right)\right|\right|_2^2 . \qquad (9)$$

Figure 4 shows the result of applying gradient descent with a backtracking line search to the same example as in Figure 3. In this case, the line search manages to adjust the step size so that the method converges.

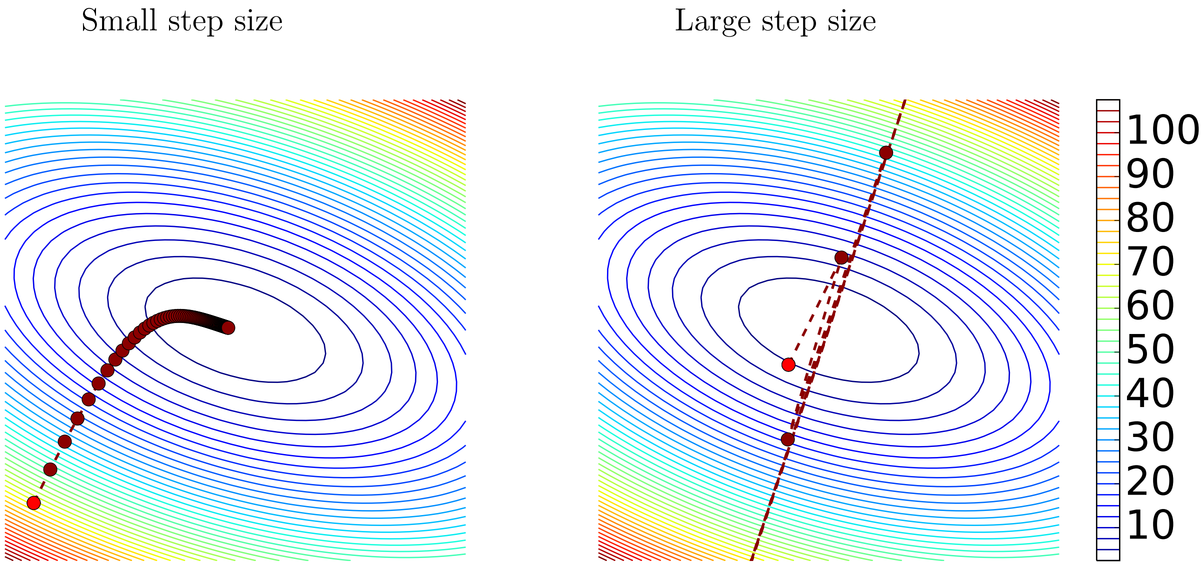

Figure 3: Iterations of gradient descent when the step size is small (left) and large (right). In the first case the convergence is very small, whereas in the second the algorithm diverges away from the minimum. The initial point is bright red.

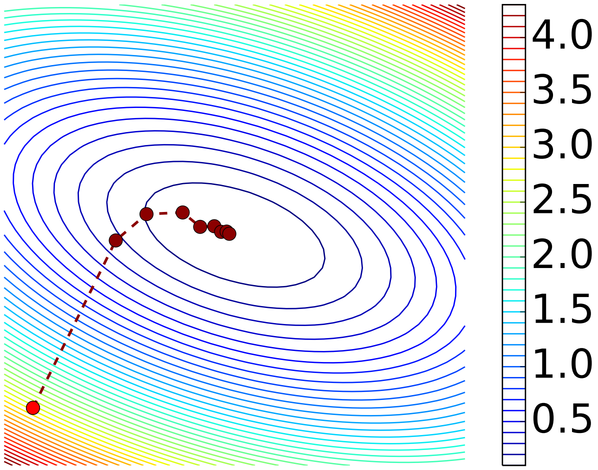

Figure 4: Gradient descent using a backtracking line search based on the Armijo rule. The function is the same as in Figure 3.

2.2 Convergence analysis

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

Definition 2.3 (Lipschitz continuity)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is Lipschitz continuous with Lipschitz constant L if for any $x, y \in \mathbb{R}^n$*

$$||f(y) - f(x)||_2 \leq L\, ||y - x||_2 \,. \tag{10}$$

We will focus on functions that have Lipschitz-continuous gradients. The following proposition, proved in Section A.1 of the appendix, shows that we are essentially considering functions that are upper bounded by a quadratic function.

Proposition 2.4 (Quadratic upper bound)**.** *If the gradient of a function $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz continuous with Lipschitz constant L,*

$$||\nabla f(y) - \nabla f(x)||_2 \leq L\, ||y - x||_2 \tag{11}$$

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

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{L}{2} ||y - x||_2^2 \,. \tag{12}$$

The quadratic upper bound immediately implies a bound on the value of the cost function after k iterations of gradient descent that will be very useful.

Corollary 2.5. *Let $x^{(i)}$ be the ith iteration of gradient descent and $\alpha_i \geq 0$ the ith step size, if ∇f is L-Lipschitz continuous,*

$$f\left(x^{(k+1)}\right) \leq f\left(x^{(k)}\right) - \alpha_k \left(1 - \frac{\alpha_k L}{2}\right) \left|\left|\nabla f\left(x^{(k)}\right)\right|\right|_2^2 . \tag{13}$$

Proof. Applying the quadratic upper bound we obtain

$$f\left(x^{(k+1)}\right) \leq f\left(x^{(k)}\right) + \nabla f\left(x^{(k)}\right)^T \left(x^{(k+1)} - x^{(k)}\right) + \frac{L}{2} \left|\left|x^{(k+1)} - x^{(k)}\right|\right|_2^2 . \tag{14}$$

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

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

$$f\left(x^{(k+1)}\right) \leq f\left(x^{(k)}\right) - \frac{\alpha_k}{2} \left|\left|\nabla f\left(x^{(k)}\right)\right|\right|_2^2 . \tag{15}$$

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 2.7. *We assume that f is convex, ∇f is L-Lipschitz continuous and there exists a point 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,*

$$f\left(x^{(k)}\right) - f\left(x^*\right) \leq \frac{\left|\left|x^{(0)} - x^*\right|\right|_2^2}{2\,\alpha\,k} \tag{16}$$

Proof. By the first-order characterization of convexity

$$f\left(x^{(i-1)}\right) + \nabla f\left(x^{(i-1)}\right)^T\left(x^* - x^{(i-1)}\right) \leq f\left(x^*\right), \tag{17}$$

which together with Corollary 2.6 yields

$$\begin{aligned} f\left(x^{(i)}\right) - f\left(x^*\right) &\leq \nabla f\left(x^{(i-1)}\right)^T\left(x^{(i-1)} - x^*\right) - \frac{\alpha}{2}\left|\left|\nabla f\left(x^{(i-1)}\right)\right|\right|_2^2 && (18)\\ &= \frac{1}{2\,\alpha}\left(\left|\left|x^{(i-1)} - x^*\right|\right|_2^2 - \left|\left|x^{(i-1)} - x^* - \alpha\nabla f\left(x^{(i-1)}\right)\right|\right|_2^2\right) && (19)\\ &= \frac{1}{2\,\alpha}\left(\left|\left|x^{(i-1)} - x^*\right|\right|_2^2 - \left|\left|x^{(i)} - x^*\right|\right|_2\right) && (20)\end{aligned}$$

Using the fact that by Corollary 2.6 the value of f never increases, we have

$$\begin{aligned} f\left(x^{(k)}\right) - f\left(x^*\right) &\leq \frac{1}{k}\sum_{i=1}^{k} f\left(x^{(i)}\right) - f\left(x^*\right) && (21)\\ &= \frac{1}{2\,\alpha\,k}\left(\left|\left|x^{(0)} - x^*\right|\right|_2^2 - \left|\left|x^{(k)} - x^*\right|\right|_2^2\right) && (22)\\ &\leq \frac{\left|\left|x^{(0)} - x^*\right|\right|_2^2}{2\,\alpha\,k}. && (23)\end{aligned}$$

□

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 2.8 (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*

$$\alpha_k \geq \alpha_{\min} := \min\left\{\alpha^0, \frac{\beta}{L}\right\}. \tag{24}$$

Proof. By Corollary 2.5 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 2.7 to establish convergence when we apply a backtracking line search.

Theorem 2.9 (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*

$$f\left(x^{(k)}\right) - f\left(x^*\right) \leq \frac{\left|\left|x^{(0)} - x^*\right|\right|_2^2}{2\,\alpha_{\min}\,k}, \tag{25}$$

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

Proof. Following the reasoning in the proof of Theorem 2.7 up until equation (20) we have

$$f\left(x^{(i)}\right) - f\left(x^*\right) \leq \frac{1}{2\,\alpha_i}\left(\left|\left|x^{(i-1)} - x^*\right|\right|_2^2 - \left|\left|x^{(i)} - x^*\right|\right|_2\right). \tag{26}$$

By Lemma 2.8 $\alpha_i \geq \alpha_{\min}$, so we just mimic the steps at the end of the proof of Theorem 2.7 to obtain

$$\begin{aligned} f\left(x^{(k)}\right) - f\left(x^*\right) &\leq \frac{1}{k}\sum_{i=1}^{k} f\left(x^{(i)}\right) - f\left(x^*\right) && (27)\\ &= \frac{1}{2\,\alpha_{\min}\,k}\left(\left|\left|x^{(0)} - x^*\right|\right|_2^2 - \left|\left|x^{(k)} - x^*\right|\right|_2^2\right) && (28)\\ &\leq \frac{\left|\left|x^{(0)} - x^*\right|\right|_2^2}{2\,\alpha_{\min}\,k}. && (29)\end{aligned}$$

□

The results that we have proved imply that we need $\mathcal{O}(1/\epsilon)$ 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. If we restrict our class of functions of interest further this can often be made theoretically rigorous. To illustrate this we introduce strong convexity.

Definition 2.10 (Strong convexity). *A function $f : \mathbb{R}^n$ is S-strongly convex if for any $x, y \in \mathbb{R}^n$*

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + S\,||y - x||^2. \tag{30}$$

Strong convexity means that the function is lower bounded by a quadratic with a fixed curvature at any point. Under this condition, gradient descent converges much faster. The following result establishes that we only need $\mathcal{O}\left(\log \frac{1}{\epsilon}\right)$ iterations to get an ϵ-optimal solution. For the proof see [6].

Theorem 2.11. *If f is S-strongly convex and ∇f is L-Lipschitz continuous*

$$f\left(x^{(k)}\right) - f\left(x^{*}\right) \leq \frac{c^{k} L \left|\left|x^{(k)} - x_0\right|\right|_2^2}{2}, \qquad c := \frac{\frac{L}{S} - 1}{\frac{L}{S} + 1}. \tag{31}$$

2.3 Accelerated gradient descent

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

Theorem 2.12 (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*

$$x^{(0)} + \operatorname{span}\left\{\nabla f\left(x^{(0)}\right), \nabla f\left(x^{(1)}\right), \ldots, \nabla f\left(x^{(k-1)}\right)\right\} \tag{32}$$

we have

$$f\left(x^{(k)}\right) - f\left(x^{*}\right) \geq \frac{3L \left|\left|x^{(0)} - x^{*}\right|\right|_2^2}{32\,(k+1)^2} \tag{33}$$

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:

$$y^{(k+1)} = x^{(k)} - \alpha_k \nabla f\left(x^{(k)}\right), \tag{34}$$

$$x^{(k+1)} = \beta_k\, y^{(k+1)} + \gamma_k\, y^{(k)}, \tag{35}$$

where β_k and γ_k may depend on k. This version of gradient descent is usually known as accelerated gradient descent or Nesterov's method. In Section 3.2 we will illustrate the application of this idea to the proximal gradient method. Intuitively, a momentum term prevents descent methods from overreacting to changes in the local slope of the function. We refer the interested reader to [2, 4] for more details.

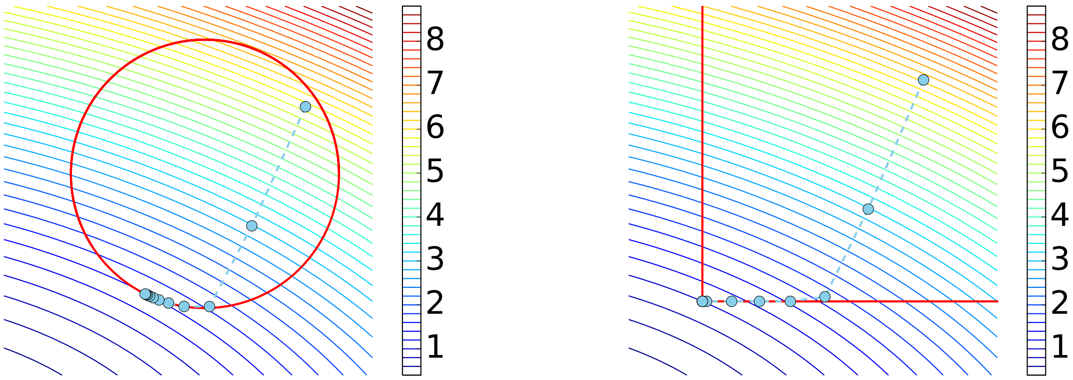

Figure 5: Iterations of projected gradient descent applied to a convex function with feasibility sets equal to the unit ℓ_2 norm (left) and the positive quadrant(right).

2.4 Projected gradient descent

In this section we explain how to adapt gradient descent to minimize a function within a convex feasibility set, i.e. to solve

$$\begin{aligned} \text{minimize} \quad & f(x) & (36) \\ \text{subject to} \quad & x \in \mathcal{S}, & (37) \end{aligned}$$

where f is differentiable and $\mathcal{S}$ is convex. The method is very simple. At each iteration we take a gradient-descent step and project on the feasibility set.

Algorithm 2.13 (Projected gradient descent)**.** *We set the initial point $x^{(0)}$ to an arbitrary value in $\mathbb{R}^n$. Then we compute*

$$x^{(k+1)} = \mathcal{P}_{\mathcal{S}}\left(x^{(k)} - \alpha_k \nabla f\left(x^{(k)}\right)\right), \tag{38}$$

until a convergence criterion is satisfied.

Figure 5 shows the results of applying projected gradient descent to minimize a convex function with two different feasibility sets in $\mathbb{R}^2$: the unit ℓ_2 norm and the positive quadrant.

3 Nondifferentiable functions

In this section we describe several methods to minimize convex nondifferentiable functions.

3.1 Subgradient method

Consider the optimization problem

$$\text{minimize} \quad f(x) \tag{39}$$

where f is convex but nondifferentiable. This implies that we cannot compute a gradient and advance in the steepest descent direction as in gradient descent. However, we can generalize the idea by using subgradients, which exist because f is convex. This will be useful as long as it is efficient to compute the subgradient of the function.

Algorithm 3.1 (Subgradient method)**.** *We set the initial point $x^{(0)}$ to an arbitrary value in $\mathbb{R}^n$. Then we compute*

$$x^{(k+1)} = x^{(k)} - \alpha_k \, q^{(k)}, \tag{40}$$

where $q^{(k)}$ is a subgradient of f at $x^{(k)}$, until a convergence criterion is satisfied.

Interestingly, the subgradient method is not a descent method. The value of the cost function can actually increase as the iterations progress. However, the method can be shown to converge at a rate of order $\mathcal{O}\left(1/\epsilon^2\right)$ as long as the step size decreases along iterations, see [6].

We now illustrate the performance of the subgradient method applied to least-squares regression with ℓ_1-norm regularization. The cost function in the optimization problem,

$$\text{minimize} \quad \frac{1}{2} \left|\left|Ax - y\right|\right|_2^2 + \lambda \left|\left|x\right|\right|_1, \tag{41}$$

is convex but not differentiable. To compute a subgradient of the function at $x^{(k)}$ we use the fact that $\operatorname{sign}(x)$ is a subgradient of the ℓ_1 norm at x, as we established in the previous lecture,

$$q^{(k)} = A^T \left(Ax^{(k)} - y\right) + \lambda \operatorname{sign}\left(x^{(k)}\right). \tag{42}$$

The subgradient-method iteration for this problem is consequently of the form

$$x^{(k+1)} = x^{(k)} - \alpha_k \left(A^T \left(Ax^{(k)} - y\right) + \lambda \operatorname{sign}\left(x^{(k)}\right)\right). \tag{43}$$

Figure 6 shows the result of applying this algorithm to an example in which $A \in \mathbb{R}^{2000 \times 1000}$, $y = Ax_0 + z$ where x_0 is 100-sparse and z is iid Gaussian. The example illustrates that decreasing the step size at each iteration achieves faster convergence.

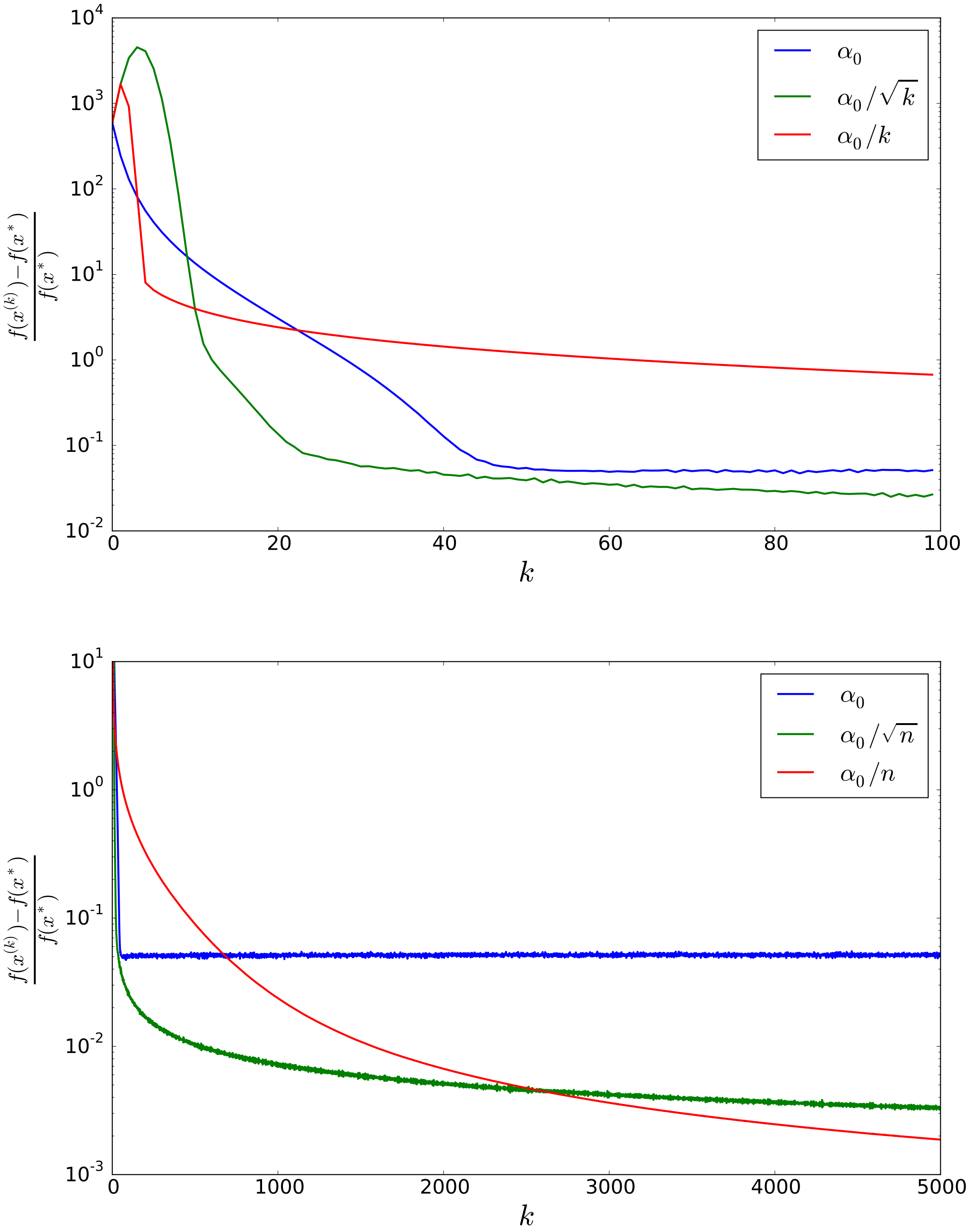

Figure 6: Subgradient method applied to least-squares regression with ℓ_1-norm regularization for different choices of step size (α_0 is a constant).

3.2 Proximal gradient method

As we saw in the previous section, convergence of subgradient method is slow, both in terms of theoretical guarantees and in the example of Figure 6. In this section we introduce an alternative method that can be applied to a class of functions which is very useful for optimization-based data analysis.

Definition 3.2 (Composite function)**.** *A composite function is a function that can be written as the sum*

$$f(x) + g(x) \tag{44}$$

where f convex and differentiable and g is convex but not differentiable.

Clearly, the least-squares regression cost function with ℓ_1-norm regularization is of this form.

In order to motivate proximal methods, let us begin by interpreting the gradient-descent iteration as the solution to a *local* linearization of the function.

Lemma 3.3. *The minimum of the function*

$$h(x) := f\left(x^{(k)}\right) + \nabla f\left(x^{(k)}\right)^T\left(x - x^{(k)}\right) + \frac{1}{2\,\alpha}\left|\left|x - x^{(k)}\right|\right|_2^2 \tag{45}$$

is $x^{(k)} - \alpha \nabla f\left(x^{(k)}\right)$.

Proof.

$$\begin{aligned}
x^{(k+1)} &:= x^{(k)} - \alpha_k\,\nabla f\left(x^{(k)}\right) && (46)\\
&= \arg\min_x \left|\left|x - \left(x^{(k)} - \alpha_k\,\nabla f\left(x^{(k)}\right)\right)\right|\right|_2^2 && (47)\\
&= \arg\min_x f\left(x^{(k)}\right) + \nabla f\left(x^{(k)}\right)^T\left(x - x^{(k)}\right) + \frac{1}{2\,\alpha_k}\left|\left|x - x^{(k)}\right|\right|_2^2. && (48)
\end{aligned}$$

□

A natural generalization of gradient descent is to minimize the sum of g and the local first-order approximation of f.

$$\begin{aligned}
x^{(k+1)} &= \arg\min_x f\left(x^{(k)}\right) + \nabla f\left(x^{(k)}\right)^T\left(x - x^{(k)}\right) + \frac{1}{2\,\alpha_k}\left|\left|x - x^{(k)}\right|\right|_2^2 + g(x) && (49)\\
&= \arg\min_x \frac{1}{2}\left|\left|x - \left(x^{(k)} - \alpha_k\,\nabla f\left(x^{(k)}\right)\right)\right|\right|_2^2 + \alpha_k\,g(x) && (50)\\
&= \operatorname{prox}_{\alpha_k\,g}\left(x^{(k)} - \alpha_k\,\nabla f\left(x^{(k)}\right)\right). && (51)
\end{aligned}$$

We have written the iteration in terms of the proximal operator of the function g.

Definition 3.4 (Proximal operator). *The proximal operator of a function $g : \mathbb{R}^n \to \mathbb{R}$ is*

$$\operatorname{prox}_g(y) := \arg\min_x g(x) + \frac{1}{2} ||x - y||_2^2 . \tag{52}$$

Solving the modified local first-order approximation of the composite function iteratively yields the proximal-gradient method, which will be useful if the proximal operator of g can be computed efficiently.

Algorithm 3.5 (Proximal-gradient method). *We set the initial point $x^{(0)}$ to an arbitrary value in $\mathbb{R}^n$. Then we compute*

$$x^{(k+1)} = \operatorname{prox}_{\alpha_k g}\left(x^{(k)} - \alpha_k \nabla f\left(x^{(k)}\right)\right), \tag{53}$$

until a convergence criterion is satisfied.

This algorithm may be interpreted as a fixed-point method. Indeed, vector is a fixed point of the proximal-gradient iteration if and only if it is a minimum of the composite function. This suggests that it is a good idea to apply the iteration repeatedly but that does not prove convergence (for this we would need to prove that the operator is contractive, see [6]).

Proposition 3.6 (Fixed point of proximal operator). *A vector $\hat{x}$ is a solution to*

$$\text{minimize} \quad f(x) + g(x), \tag{54}$$

if and only if it is a fixed point of the proximal-gradient iteration

$$\hat{x} = \operatorname{prox}_{\alpha g}(\hat{x} - \alpha \nabla f(\hat{x})) \tag{55}$$

for any $\alpha > 0$.

Proof. $\hat{x}$ is a solution to the optimization problem if and only if there exists a subgradient q of g at $\hat{x}$ such that $\nabla f(\hat{x}) + q = 0$. $\hat{x}$ is the solution to

$$\text{minimize} \quad \alpha\, g(x) + \frac{1}{2} ||\hat{x} - \alpha \nabla f(\hat{x}) - x||_2^2, \tag{56}$$

which is the case if and only if there exists a subgradient q of g at $\hat{x}$ such that $\alpha \nabla f(\hat{x}) + \alpha q = 0$. As long as $\alpha > 0$ the two conditions are equivalent. □

In the case of the indicator function of a set,

$$\mathcal{I}_{\mathcal{S}}(x) := \begin{cases} 0 & \text{if } x \in \mathcal{S}, \\ \infty & \text{if } x \notin \mathcal{S}, \end{cases} \tag{57}$$

the proximal operator is just the projection onto the set.

Lemma 3.7 (Proximal operator of indicator function)**.** *The proximal operator of the indicator function of a convex set $\mathcal{S} \subseteq \mathbb{R}^n$ is projection onto $\mathcal{S}$.*

Proof. The lemma follows directly from the definitions of proximal operator, projection and indicator function. □

An immediate consequence of this is that projected-gradient descent can be interpreted as a special case of the proximal-gradient method.

Proximal methods are very useful for fitting sparse models because the proximal operator of the ℓ_1 norm is very tractable.

Proposition 3.8 (Proximal operator of ℓ_1 norm)**.** *The proximal operator of the ℓ_1 norm weighted by a constant $\lambda > 0$ is the soft-thresholding operator*

$$\operatorname{prox}_{\lambda \, ||\cdot||_1}(y) = \mathcal{S}_\lambda(y) \tag{58}$$

where

$$\mathcal{S}_\lambda(y)_i := \begin{cases} y_i - \operatorname{sign}(y_i)\,\lambda & \text{if } |y_i| \geq \lambda, \\ 0 & \text{otherwise.} \end{cases} \tag{59}$$

The proposition, proved in Section A.3 of the appendix, allows us to derive the following algorithm for least-squares regression with ℓ_1-norm regularization.

Algorithm 3.9 (Iterative Shrinkage-Thresholding Algorithm (ISTA))**.** *We set the initial point $x^{(0)}$ to an arbitrary value in $\mathbb{R}^n$. Then we compute*

$$x^{(k+1)} = \mathcal{S}_{\alpha_k \lambda}\left(x^{(k)} - \alpha_k \, A^T \left(Ax^{(k)} - y\right)\right), \tag{60}$$

until a convergence criterion is satisfied.

ISTA can be accelerated using a momentum term as in Nesterov's accelerated gradient method. This yields a fast version of the algorithm called FISTA.

Algorithm 3.10 (Fast Iterative Shrinkage-Thresholding Algorithm (FISTA))**.** *We set the initial point $x^{(0)}$ to an arbitrary value in $\mathbb{R}^n$. Then we compute*

$$z^{(0)} = x^{(0)} \tag{61}$$

$$x^{(k+1)} = \mathcal{S}_{\alpha_k \lambda}\left(z^{(k)} - \alpha_k \, A^T \left(Az^{(k)} - y\right)\right), \tag{62}$$

$$z^{(k+1)} = x^{(k+1)} + \frac{k}{k+3}\left(x^{(k+1)} - x^{(k)}\right), \tag{63}$$

until a convergence criterion is satisfied.

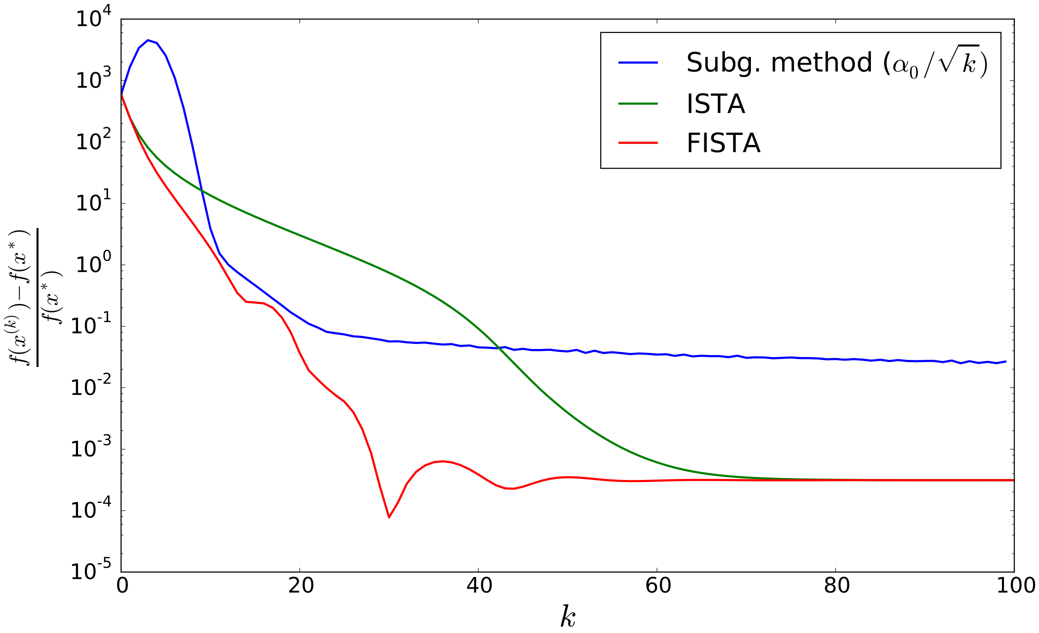

Figure 7: ISTA and FISTA applied to least-squares regression with ℓ_1-norm regularization.

ISTA and FISTA were proposed by Beck and Teboulle in [1]. ISTA is a descent method. It has the same convergence rate as gradient descent $\mathcal{O}(1/\epsilon)$ both with a constant step size and with a backtracking line search, under the condition that ∇f be L-Lipschitz continuous. FISTA in contrast is not a descent method, but it can be shown to converge in $\mathcal{O}(1/\sqrt{\epsilon})$ to an ϵ-optimal solution.

To illustrate the performance of ISTA and FISTA, we apply them to the same example used in Figure 6. Even without applying a backtracking line search both methods converge to a solution of middle precision (around 10^{-3} or 10^{-4}) much more rapidly than the subgradient method. The results are shown in Figure 7.

3.3 Coordinate descent

Coordinate descent is an optimization method that decouples n-dimensional problems of the form

$$\text{minimize} \quad h(x) \tag{64}$$

by solving a sequence of 1D problems. The algorithm is very simple; we just fix all the entries of the variable vector except one and optimize over it. This procedure is called coordinate descent because it is equivalent to iteratively minimizing the function in the direction of the axes.

Algorithm 3.11 (Coordinate descent)**.** *We set the initial point* $x^{(0)}$ *to an arbitrary value in* $\mathbb{R}^n$*. At each iteration we choose an arbitrary position* $1 \leq i \leq n$ *and set*

$$x_i^{(k+1)} = \arg\min_{x_i \in \mathbb{R}} h\left(x_1^{(k)}, \ldots, x_i, \ldots, x_n^{(k)}\right), \tag{65}$$

until a convergence criterion is satisfied. The order of the entries that we optimize over can be fixed or random.

The method is guaranteed to converge for composite functions where f is differentiable and convex and g has an additive decomposition,

$$h(x) := f(x) + g(x) = f(x) + \sum_{i=1}^{n} g_i(x_i),$$

where the functions $g_1, g_2, \ldots, g_n : \mathbb{R} \to \mathbb{R}$ are convex and nondifferentiable [5].

In order to apply coordinate descent it is necessary for the one-dimensional optimization problems to be easy to solve. This is the case for least-squares regression with ℓ_1-norm regularization as demonstrated by the following proposition, which is proved in Section A.4.

Proposition 3.12 (Coordinate-descent subproblems for least-squares regression with ℓ_1-norm regularization)**.** *Let*

$$h(x) = \frac{1}{2}\left|\left|Ax - y\right|\right|_2^2 + \lambda\left|\left|x\right|\right|_1. \tag{66}$$

The solution to the subproblem $\min_{x_i} h(x_1, \ldots, x_i, \ldots, x_n)$ *is*

$$\hat{x}_i = \frac{\mathcal{S}_\lambda(\gamma_i)}{\left|\left|A_i\right|\right|_2^2} \tag{67}$$

where A_i *is the* i*th column of* A *and*

$$\gamma_i := \sum_{l=1}^{m} A_{li}\left(y_l - \sum_{j \neq i} A_{lj} x_j\right). \tag{68}$$

For more information on coordinate descent and its application to sparse regression, see Chapter 5 of [3].

References

For further reader on the convergence of first-order methods we recommend Nesterov's book [4] and Bubeck's notes [2]. Chapter 5 of [3] in Hastie, Tibshirani and Wainwright is a great description of proximal-gradient and coordinate-descent methods and their application to sparse regression.

[1] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

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

[3] T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical learning with sparsity: the lasso and generalizations.* CRC Press, 2015.

[4] Y. Nesterov. *Introductory lectures on convex optimization: A basic course.*

[5] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001.

[6] L. Vandenberghe. Notes on optimization methods for large-scale systems.

A Proofs

A.1 Proof of Proposition 2.4

Consider the function

$$g(x) := \frac{L}{2} x^T x - f(x). \tag{69}$$

We first establish that g is convex using the following lemma, proved in Section A.2 below.

Lemma A.1 (Monotonicity of gradient)**.** *A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if*

$$(\nabla f(y) - \nabla f(x))^T (y - x) \geq 0. \tag{70}$$

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

$$(\nabla f(y) - \nabla f(x))^T (y - x) \leq L \, ||y - x||_2^2, \tag{71}$$

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

$$(\nabla g(y) - \nabla g(x))^T (y - x) = (Ly - Lx + \nabla f(x) - \nabla f(y))^T (y - x) \tag{72}$$

$$= L \, ||y - x||_2^2 - (\nabla f(y) - \nabla f(x))^T (y - x) \tag{73}$$

$$\geq 0 \tag{74}$$

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

$$\frac{L}{2} y^T y - f(y) = g(y) \tag{75}$$

$$\geq g(x) + \nabla g(x)^T (y - x) \tag{76}$$

$$= \frac{L}{2} x^T x - f(x) + (Lx - \nabla f(x))^T (y - x) . \tag{77}$$

Rearranging the inequality we conclude that

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{L}{2} ||y - x||_2^2 . \tag{78}$$

A.2 Proof of Lemma A.1

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

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

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) , \tag{79}$$

$$f(x) \geq f(y) + \nabla f(y)^T (x - y) , \tag{80}$$

$$\tag{81}$$

Adding the two inequalities directly implies the result.

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

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

$$g_{a,b}(\alpha) := f(\alpha a + (1 - \alpha) b) , \tag{82}$$

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

$$g'_{a,b}(\alpha) - g'_{a,b}(0) = (\nabla f(\alpha \, a + (1 - \alpha) \, b) - \nabla f(b))^T (a - b) \tag{83}$$

$$= \frac{1}{\alpha} (\nabla f(\alpha \, a + (1 - \alpha) \, b) - \nabla f(b))^T (\alpha \, a + (1 - \alpha) \, b - b) \tag{84}$$

$$\geq 0 \quad \text{because } (\nabla f(y) - \nabla f(x))^T (y - x) \geq 0 \text{ for any } x, y. \tag{85}$$

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

$$\begin{aligned} f(x) &= g_{x,y}(1) && (86)\\ &= g_{x,y}(0) + \int_0^1 g'_{x,y}(\alpha)\, \mathrm{d}\,\alpha && (87)\\ &\geq g_{x,y}(0) + g'_{x,y}(0) && (88)\\ &= f(y) + \nabla f(y)(x-y). && (89)\end{aligned}$$

A.3 Proof of Proposition 3.8

Writing the function as a sum,

$$\lambda \left|\left|x\right|\right|_1 + \frac{1}{2}\left|\left|y-x\right|\right|_2^2 = \sum_{i=1}^n |x_i| + \frac{1}{2}(y_i - x_i)^2 \tag{90}$$

reveals that it decomposes into independent nonnegative terms. The univariate function

$$h(\beta) := \lambda |\beta| + \frac{1}{2}(y_i - \beta)^2 \tag{91}$$

is strictly convex and consequently has a unique global minimum. It is also differentiable everywhere except at zero. If $\beta \geq 0$ the derivative is $\lambda + \beta - y_i$, so if $y_i \geq \lambda$, the minimum is achieved at $y_i - \lambda$. If $y_i < \lambda$ the function is increasing for $\beta \geq 0$, so the minimizer must be smaller or equal to zero. The derivative for $\beta < 0$ is $-\lambda + \beta - y_i$ so the minimum is achieved at $y_i + \lambda$ if $y_i \leq \lambda$. Otherwise the function is decreasing for all $\beta < 0$. As a result, if $-\lambda < y_i < \lambda$ the minimum must be at zero.

A.4 Proof of Proposition 3.12

Note that

$$\begin{aligned} \min_{x_i} h(x) &= \min_{x_i} \sum_{l=1}^n \frac{1}{2}\left(\sum_{j=1}^m A_{lj}x_j - y_l\right)^2 + \lambda \sum_{j=l}^n |x_l| && (92)\\ &= \min_{x_i} \sum_{l=1}^n \frac{1}{2}A_{li}^2 x_i^2 + \left(\sum_{j\neq i}^m A_{lj}x_j - y_l\right) A_{li}x_i + \lambda\, |x_l| && (93)\\ &= \min_{x_i} \frac{1}{2}\left|\left|A_i\right|\right|_2^2 x_i^2 - \gamma_i x_i + \lambda\, |x_i|. && (94)\end{aligned}$$

The univariate function

$$g(\beta) := \frac{1}{2}\left|\left|A_i\right|\right|_2^2 \beta^2 - \gamma_i \beta + \lambda\, |\beta| \tag{95}$$

is strictly convex and consequently has a unique global minimum. It is also differentiable everywhere except at zero. If $\beta \geq 0$ the derivative is $||A_i||_2^2\,\beta - \gamma_i + \lambda$, so if $\gamma_i \geq \lambda$, the minimum is achieved at $(\gamma_i - \lambda) \,/\, ||A_i||_2^2$. If $\gamma_i < \lambda$ the function is increasing for $\beta \geq 0$, so the minimizer must be smaller or equal to zero. The derivative for $\beta < 0$ is $||A_i||_2^2\,\beta - \gamma_i - \lambda$ so the minimum is achieved at $(\gamma_i + \lambda) \,/\, ||A_i||_2^2$ if $\gamma_i \leq \lambda$. Otherwise the function is decreasing for all $\beta < 0$. As a result, if $-\lambda < \gamma_i < \lambda$ the minimum must be at zero.