Convex Optimization

1 Motivation

1.1 Sparse regression

In our description of linear regression in the notes on the SVD, we observed that the performance of linear regression degrades when the number of features is close to the number of training data. This makes sense: the number of parameters of a model should be significantly smaller than the number of measurements used to fit it. However, when the number of features is very large, it is often possible to achieve accurate prediction using only a subset of them. Selecting a useful subset of features is a crucial problem in statistics, known as model selection. Consider a linear regression problem with p features associated to a coefficient vector $\vec{\beta} \in \mathbb{R}^p$. The goal of model selection is to find a set of indices $\mathcal{I} \subset \{1, \ldots, p\}$, such that the response $y \in \mathbb{R}$ is well approximated by the corresponding features,

$$y \approx \sum_{i \in \mathcal{I}} \vec{\beta}[i] \vec{x}[i] + \beta_0, \tag{1}$$

where $\beta_0 \in \mathbb{R}$ is the intercept. Equivalently, we would like to find a sparse coefficient vector $\vec{\beta} \in \mathbb{R}^p$ such that

$$y \approx \left\langle \vec{x}, \vec{\beta} \right\rangle + \beta_0. \tag{2}$$

The problem of finding sparse coefficients that achieve a good fit to the data is called sparse regression. In these notes, we study the lasso, a popular sparse-regression method based on regularization.

1.2 Matrix completion for collaborative filtering

The aim of collaborative filtering is to the value of a certain quantity $y[i, j]$ that depends on two indices i and j. We can think of $y[i, j]$ as the rating assigned to a movie i by a user j. In the notes on the SVD, we described a low-rank bilinear model to tackle this problem. The idea is to fit a rank-r model

$$y[i, j] \approx \sum_{l=1}^{r} a_l[i] b_l[j], \tag{3}$$

where the factors a_1, $\ldots$, a_r, b_1, $\ldots$, b_r capture the correlations between the movies and users respectively. We showed that the truncated SVD yields a rank-r approximation that is optimal with respect to Frobenius-norm error when all entries are observed. However, this is no longer the case when the data are incomplete, which is the actual problem we want to solve! In these notes we will show how to tackle the low-rank matrix completion problem using regularization.

2 Convex functions

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

2.1 Convexity

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

Definition 2.1 (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 $\theta \in (0, 1)$,*

$$\theta f\left(\vec{x}\right) + \left(1 - \theta\right) f\left(\vec{y}\right) \geq f\left(\theta \vec{x} + \left(1 - \theta\right) \vec{y}\right). \tag{4}$$

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

$$\theta f\left(\vec{x}\right) + \left(1 - \theta\right) f\left(\vec{y}\right) > f\left(\theta \vec{x} + \left(1 - \theta\right) \vec{y}\right). \tag{5}$$

Lemma 2.2. *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 $\theta \in (0, 1)$,

$$f\left(\theta \vec{x} + \left(1 - \theta\right) \vec{y}\right) = \theta f\left(\vec{x}\right) + \left(1 - \theta\right) f\left(\vec{y}\right). \tag{6}$$

□

Figure 1 shows a 1D convex function: the line between any two points on the curve defined by the function must lie above the curve. In higher-dimensions, we can restrict our attention to lines in $\mathbb{R}^n$. A function is convex if and only if it is convex when restricted to every line.

Lemma 2.3 (Proof in Section 6.1)**.** *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*

$$g_{\vec{x},\vec{y}}\left(\alpha\right) := f\left(\alpha \vec{x} + \left(1 - \alpha\right) \vec{y}\right) \tag{7}$$

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

A crucial property of convex functions is that they cannot have suboptimal local minima.

Theorem 2.4 (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 $f\left(\vec{x}_{\text{glob}}\right) < f\left(\vec{x}_{\text{loc}}\right)$. Since $\vec{x}_{\text{loc}}$ is a local minimum, there exists $\gamma > 0$ for which

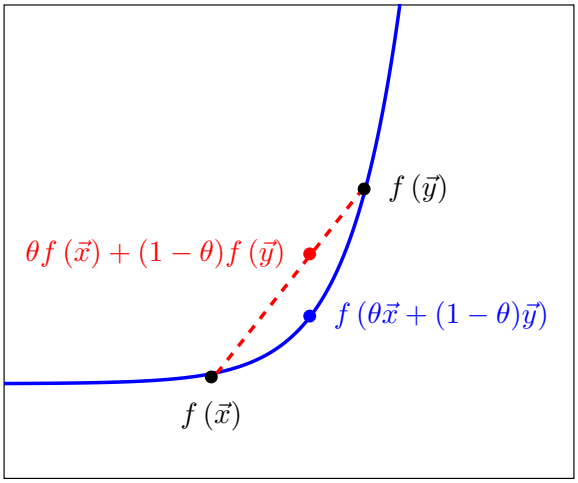

Figure 1: Illustration of condition (4) in Definition 2.1. The curve corresponding to the function must lie below any chord joining two of its points.

$f(\vec{x}_{\text{loc}}) \leq f(\vec{x})$ 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

$$
\begin{aligned}
f(\vec{x}_{\text{loc}}) &\leq f(\vec{x}_\theta) && (8)\\
&\leq \theta f(\vec{x}_{\text{loc}}) + (1-\theta) f(\vec{x}_{\text{glob}}) \quad \text{by convexity of } f && (9)\\
&< f(\vec{x}_{\text{loc}}) \quad \text{because } f(\vec{x}_{\text{glob}}) < f(\vec{x}_{\text{loc}}). && (10)
\end{aligned}
$$

□

If a function is strictly convex then any local minimum is both global and unique: every other point is guaranteed to yield a larger value.

Corollary 2.5. *Strictly convex functions have at most one global minimum, and no other local minima.*

Proof. By Theorem 2.4 all local minima of the function are global minima and hence have the same value $v_{\min} := f(\vec{x}) = f(\vec{y})$. Let $\vec{x}$ and $\vec{y}$ be two such minima. By strict convexity

$$
\begin{aligned}
f(0.5\vec{x} + 0.5\vec{y}) &< 0.5 f(\vec{x}) + 0.5 f(\vec{y}) && (11)\\
&= v_{\min}, && (12)
\end{aligned}
$$

which contradicts the assumption that $\vec{x}$ and $\vec{y}$ are global minima. □

2.2 Norms

Norms are functions that can be used to measure the length of vectors in a vector space. The most popular is the Euclidean ℓ_2 norm, but there are many other functions that are valid norms.

Definition 2.6 (Norm)**.** *Let $\mathcal{V}$ be a vector space, a norm is a function $||\cdot||$ from $\mathcal{V}$ to $\mathbb{R}$ that satisfies the following conditions.*

- *It is homogeneous. For any scalar α and any $\vec{x} \in \mathcal{V}$*

$$||\alpha\, \vec{x}|| = |\alpha|\, ||\vec{x}||\,. \tag{13}$$

- *It satisfies the triangle inequality*

$$||\vec{x} + \vec{y}|| \leq ||\vec{x}|| + ||\vec{y}||\,. \tag{14}$$

  In particular, it is nonnegative (set $\vec{y} = -\vec{x}$).

- $||\vec{x}|| = 0$ *implies that $\vec{x}$ is the zero vector $\vec{0}$.*

Conveniently, all norms are convex.

Lemma 2.7 (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 $\theta \in (0, 1)$

$$\begin{aligned} ||\theta\vec{x} + (1-\theta)\,\vec{y}|| &\leq ||\theta\vec{x}|| + ||(1-\theta)\,\vec{y}|| && (15)\\ &= \theta\, ||\vec{x}|| + (1-\theta)\, ||\vec{y}||\,. && (16) \end{aligned}$$

□

The following lemma establishes that the composition between a convex function and an affine function is convex.

Lemma 2.8 (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*

$$h\,(\vec{x}) := f(A\vec{x} + \vec{b}) \tag{17}$$

is convex.

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

$$\begin{aligned} h\,(\theta\vec{x} + (1-\theta)\,\vec{y}) &= f\left(\theta(A\vec{x} + \vec{b}) + (1-\theta)(A\vec{y} + \vec{b})\right) && (18)\\ &\leq \theta f(A\vec{x} + \vec{b}) + (1-\theta)\, f(A\vec{y} + \vec{b}) && (19)\\ &= \theta\, h\,(\vec{x}) + (1-\theta)\, h\,(\vec{y})\,. && (20) \end{aligned}$$

□

Consequently any function of the form

$$f\,(\vec{x}) := \|A\vec{x} + \vec{b}\| \tag{21}$$

is convex for any fixed matrix A and vector $\vec{b}$ with suitable dimensions. In particular, the least-squares cost function is convex.

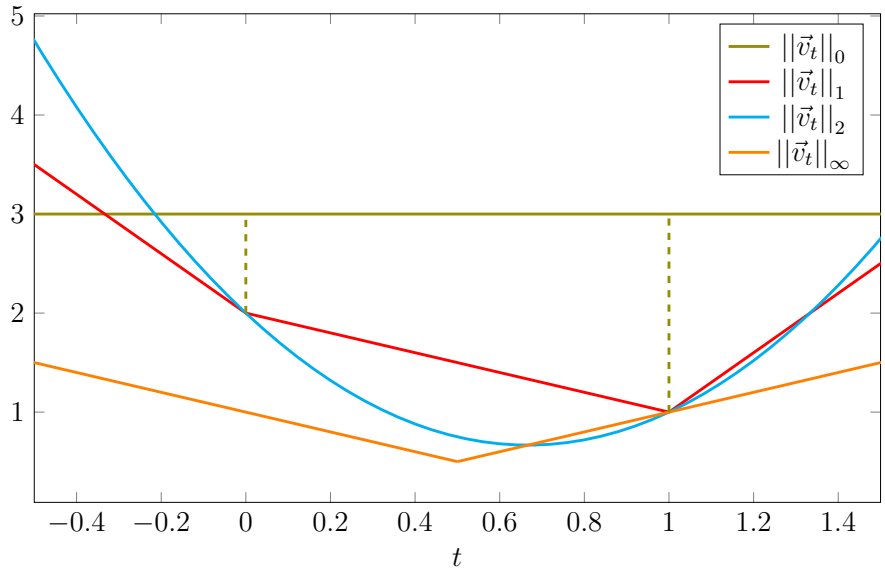

Figure 2: Graph of the function in Eq. (24) for different norms and for the nonconvex ℓ_0 "norm".

2.3 Promoting sparsity

In sparse regression, our goal is to find a sparse vector of coefficients that fits the observed data. As we established in the case of ridge regression, regularization makes it possible to constrain the coefficients learned by minimizing the resulting cost function. The question is: what regularization term should we use to promote sparsity? Ideally, the regularization term should reflect the number of nonzero entries in the coefficient vector. This quantity is often called the ℓ_0 "norm" of the vector. Unfortunately, it is not a valid norm because it is not homogeneous: for any $\vec{x}$ $||2\vec{x}||_0 = ||\vec{x}||_0 \neq 2\,||\vec{x}||_0$. In fact, the ℓ_0 "norm" is not even convex.

Lemma 2.9. *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 $\theta \in (0,1)$

$$||\theta\vec{x} + (1-\theta)\,\vec{y}||_0 = 2 > 1 = \theta\,||\vec{x}||_0 + (1-\theta)\,||\vec{y}||_0\,. \tag{22}$$

□

In order to illustrate why using a nonconvex function to promote sparsity is not practical, we build a toy problem. Consider a vector, parametrized by a constant $t \in \mathbb{R}$,

$$\vec{v}_t := \begin{bmatrix} t \\ t-1 \\ t-1 \end{bmatrix}. \tag{23}$$

Our objective is to fit t so that $\vec{v}_t$ is as sparse as possible or, in other words, minimize $||\vec{v}_t||_0$. The graph of the function is depicted in Figure 2. It is constant except at two isolated points,

so we essentially have to evaluate the function everywhere to find the global minimum, which is obviously computationally infeasible. In contrast, if we consider

$$f(t) := ||\vec{v}_t|| \tag{24}$$

where $||\cdot||$ is a valid norm, then the function is convex in t by Lemma 2.8. This means that it has no local minima and, as we will see later on, that we can exploit local information to find the global minimum. Figure 2 shows f for different norms.

It turns out that 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 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 nonzero entries.

2.4 Promoting low-rank structure

In low-rank matrix completion, the goal is to find a low-rank matrix that approximates a set of data. Unfortunately, just like the ℓ_0 "norm", the rank of a matrix interpreted as a function of its entries is not convex.

Lemma 2.10 (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 2.9. Let

$$X := \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad Y := \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{25}$$

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

$$\operatorname{rank}(\theta X + (1-\theta) Y) = 2 > 1 = \theta \operatorname{rank}(X) + (1-\theta) \operatorname{rank}(Y). \tag{26}$$

□

We again consider a toy problem to illustrate the low-rank estimation problem. Consider the following matrix, parametrized by a constant $t \in \mathbb{R}$,

$$M(t) := \begin{bmatrix} 0.5+t & 1 & 1 \\ 0.5 & 0.5 & t \\ 0.5 & 1-t & 0.5 \end{bmatrix}. \tag{27}$$

Figure 3 shows the rank of the matrix as a function of t. The function is again mostly constant, and therefore impossible to optimize unless we evaluate it everywhere. To enable tractable rank minimization, we need a convex function that plays the role of the ℓ_1 norm with regards to sparsity. The rank can be seen as the ℓ_0 "norm" of the singular values of a matrix. This suggests using the ℓ_1 norm of the singular values (i.e. their sum since they are nonnegative).

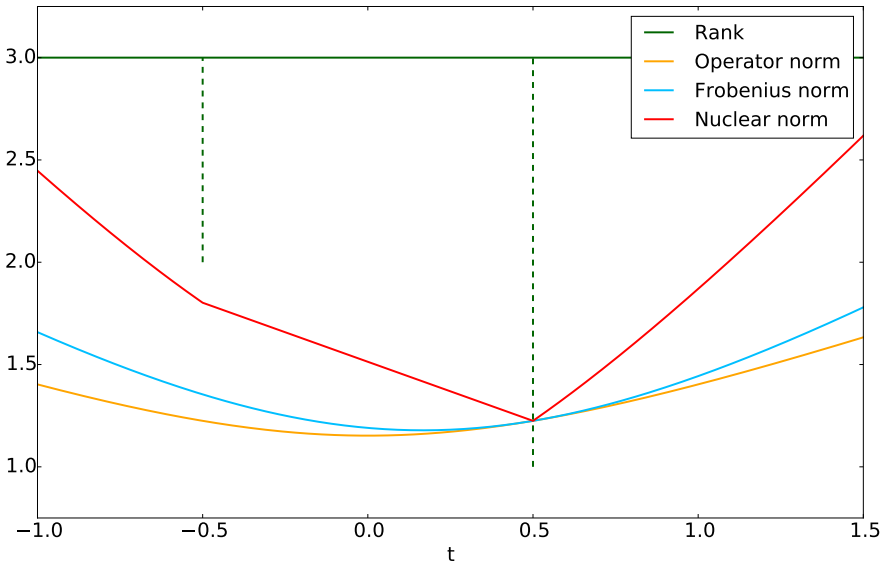

Figure 3: Values of different norms for the matrix $M(t)$ defined by Eq. (27) as a function of t. The rank of the matrix for each t is marked in green.

Definition 2.11 (Nuclear norm)**.** *The nuclear norm of a matrix $A \in \mathbb{R}^{m\times n}$ is equal to the sum of its singular values $s_1, \ldots, s_{\min\{m,n\}}$*

$$||A||_* := \sum_{i=1}^{\min\{m,n\}} s_i. \tag{28}$$

The following theorem is analogous to Hölder's inequality for vectors, since the operator norm can be interpreted as the ℓ_∞ norm of the singular values.

Theorem 2.12 (Proof in Section 6.2)**.** *For any matrix $A \in \mathbb{R}^{m\times n}$,*

$$||A||_* = \sup_{\{||B||\leq 1 \mid B\in\mathbb{R}^{m\times n}\}} \langle A, B\rangle . \tag{29}$$

A direct consequence of the result is that the nuclear norm satisfies the triangle inequality. This implies that it is a norm, since it clearly satisfies the remaining properties in Definition 2.6.

Corollary 2.13. *For any $m \times n$ matrices A and B*

$$||A+B||_* \leq ||A||_* + ||B||_* . \tag{30}$$

Proof.

$$\begin{aligned} ||A+B||_* &= \sup_{\{||C||\leq 1 \mid C\in\mathbb{R}^{m\times n}\}} \langle A+B, C\rangle && (31)\\ &\leq \sup_{\{||C||\leq 1 \mid C\in\mathbb{R}^{m\times n}\}} \langle A, C\rangle + \sup_{\{||D||\leq 1 \mid D\in\mathbb{R}^{m\times n}\}} \langle B, D\rangle && (32)\\ &= ||A||_* + ||B||_* . && (33) \end{aligned}$$

□

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. The norms are all convex, which follows from Lemma 2.8. 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 promotes low-rank structure.

3 Optimality conditions

3.1 Gradients

Recall that the derivative of a 1D function captures the local rate of change in the value of a function. The gradient is a generalization of the derivative to multiple dimensions.

Definition 3.1 (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*

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

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:

$$\nabla f(\vec{x}) = \begin{bmatrix} \frac{\partial f(\vec{x})}{\partial \vec{x}[1]} \\ \frac{\partial f(\vec{x})}{\partial \vec{x}[2]} \\ \cdots \\ \frac{\partial f(\vec{x})}{\partial \vec{x}[n]} \end{bmatrix}. \tag{34}$$

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

$$f'_{\vec{u}}(\vec{x}) := \lim_{h \to 0} \frac{f(\vec{x} + h\vec{u}) - f(\vec{x})}{h} \tag{35}$$

$$= \langle \nabla f(\vec{x}), \vec{u} \rangle \tag{36}$$

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.

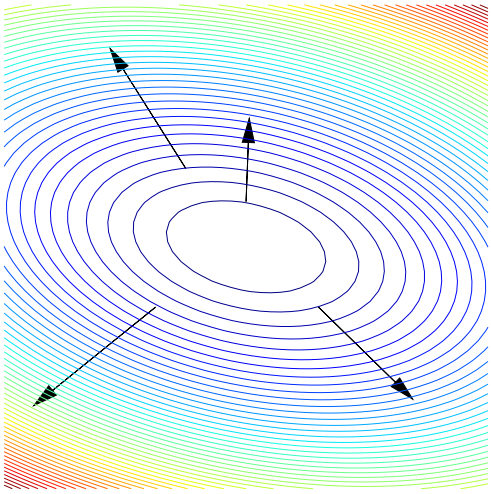

Figure 4: Contour lines of a function $f : \mathbb{R}^2 \to \mathbb{R}$. The gradients at different points are represented by black arrows, which are orthogonal to the contour lines.

Corollary 3.3. *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

$$\begin{aligned} |f'_{\vec{u}}(\vec{x})| &= \left|\nabla f(\vec{x})^T \vec{u}\right| && (37)\\ &\leq ||\nabla f(\vec{x})||_2 \, ||\vec{u}||_2 && (38)\\ &= ||\nabla f(\vec{x})||_2 && (39) \end{aligned}$$

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 derivative in those directions equals zero.

The next lemma shows that the gradient of constant, linear and quadratic functions are almost direct generalizations from the 1D case. We omit the proof, which just relies on straightforward derivative calculations.

Lemma 3.4 (Gradients of simple functions)**.** *Let $A \in \mathbb{R}^{a\times b}$, $M \in \mathbb{R}^{b\times b}$ and symmetric, $\vec{x} \in \mathbb{R}^b$,$\vec{c} \in \mathbb{R}^b$, $d \in \mathbb{R}$. The gradients of the functions*

$$\begin{aligned} f_1(\vec{x}) &:= d, && (40)\\ f_2(\vec{x}) &:= \vec{c}^T \vec{x}, && (41)\\ f_3(\vec{x}) &:= \vec{x}^T M \vec{x}, && (42) \end{aligned}$$

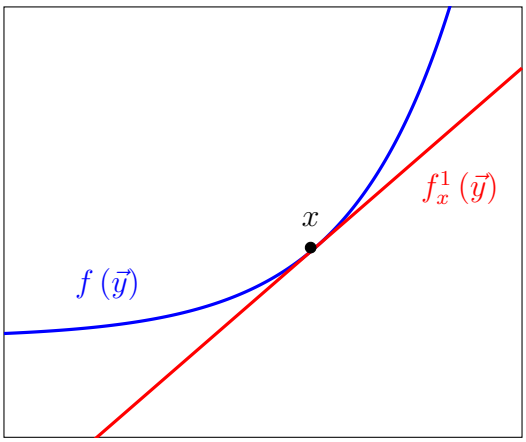

Figure 5: An example of the first-order condition for convexity. The first-order approximation at any point is a lower bound of the function.

equal

$$\nabla f_1(\vec{x}) = 0, \tag{43}$$
$$\nabla f_2(\vec{x}) = \vec{c}, \tag{44}$$
$$\nabla f_3(\vec{x}) = 2M\vec{x}. \tag{45}$$

We can now easily compute the gradient of the linear regression least-squares cost function.

Corollary 3.5 (Least squares)**.** *Let $\vec{y} \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times p}$, $\vec{\beta} \in \mathbb{R}^p$. The gradient of the least-squares cost function*

$$f(\vec{\beta}) := \frac{1}{2}\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 = \frac{1}{2}\vec{y}^T\vec{y} + \frac{1}{2}\vec{\beta}^T X^T X\vec{\beta} - \vec{y}^T X\vec{\beta} \tag{46}$$

equals

$$\nabla f(\vec{\beta}) = X^T(X\vec{\beta} - \vec{y}). \tag{47}$$

3.2 First-order optimality conditions for differentiable functions

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

Definition 3.6 (First-order approximation)**.** *The first-order or linear approximation of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ at $\vec{x}$ is*

$$f_{\vec{x}}^1(\vec{y}) := f(\vec{x}) + \nabla f(\vec{x})^T(\vec{y} - \vec{x}). \tag{48}$$

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

Theorem 3.7 (Proof in Section 6.3)**.** *A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if for every $\vec{x}, \vec{y} \in \mathbb{R}^n$*

$$f(\vec{y}) \geq f(\vec{x}) + \nabla f(\vec{x})^T (\vec{y} - \vec{x}). \tag{49}$$

It is strictly convex if and only if

$$f(\vec{y}) > f(\vec{x}) + \nabla f(\vec{x})^T (\vec{y} - \vec{x}). \tag{50}$$

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

$$\mathcal{H}_{f,\vec{x}} := \left\{ \vec{y} \in \mathbb{R}^{n+1} \mid \vec{y}[n+1] = f^1_{\vec{x}} \left(\begin{bmatrix} \vec{y}[1] \\ \cdots \\ \vec{y}[n] \end{bmatrix} \right) \right\}. \tag{51}$$

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

$$\text{epi}(f) := \left\{ \vec{x} \mid f \left(\begin{bmatrix} \vec{x}[1] \\ \cdots \\ \vec{x}[n] \end{bmatrix} \right) \leq \vec{x}[n+1] \right\}. \tag{52}$$

Geometrically, Theorem 3.7 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 3.9 (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 $\mathcal{H}$.*

An important corollary to Theorem 3.7 is that for a convex function, any point at which the gradient is zero is a global minimum.

Corollary 3.10. *Let f be a convex differentiable function and let $\vec{0}$ denote a vector with zero entries. A point $\vec{x}$ satisfies $\nabla f(\vec{x}) = \vec{0}$ if and only if it is a global minimum, i.e. for any $\vec{y} \in \mathbb{R}^n$*

$$f(\vec{y}) \geq f(\vec{x}). \tag{53}$$

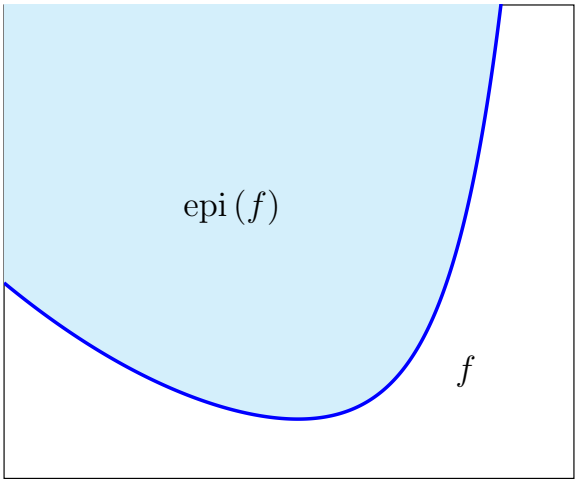

Figure 6: Epigraph of a function.

This optimality condition has a very intuitive geometric interpretation in terms of the supporting hyperplane $\mathcal{H}_{f,\vec{x}}$. $\nabla f(\vec{x}) = \vec{0}$ 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.

The corollary allows us to rederive the closed-form solution for the least-squares cost function

$$f(\vec{\beta}) := \frac{1}{2}\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2, \tag{54}$$

where $\vec{y} \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times p}$, $\vec{\beta} \in \mathbb{R}^p$. By Corollary 3.5, setting the gradient to zero yields the so-called normal equations

$$X^T X\vec{\beta} = X^T\vec{y}, \tag{55}$$

which has the unique solution

$$\vec{\beta}_{\text{LS}} := (X^T X)^{-1} X^T\vec{y} \tag{56}$$

as long as the matrix $X^T X$ is invertible.

3.3 Subgradients

A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only at any point $\vec{x} \in \mathbb{R}^n$ there exists a vector $\vec{g}$ such that

$$f(\vec{y}) \geq f(\vec{x}) + \langle \vec{g}, \vec{y} - \vec{x} \rangle \tag{57}$$

for every $\vec{y} \in \mathbb{R}^n$. Geometrically, the epigraph must have a supporting hyperplane at every point. The vector $\vec{g}$ is the gradient of f at $\vec{x}$. Nondifferentiable functions do not have gradients, but the existence of a supporting hyperplane still characterizes convexity. The vector $\vec{g}$ that determines such a hyperplane is called a subgradient.

Definition 3.11 (Subgradient)**.** *The subgradient of a function $f : \mathbb{R}^n \to \mathbb{R}$ at $\vec{x} \in \mathbb{R}^n$ is a vector $\vec{g} \in \mathbb{R}^n$ such that*

$$f(\vec{y}) \geq f(\vec{x}) + \vec{g}^T(\vec{y} - \vec{x}), \quad \text{for all } \vec{y} \in \mathbb{R}^n. \tag{58}$$

The set of all subgradients is called the subdifferential of the function at $\vec{x}$.

If a function is differentiable at a given point, then the gradient is the only subgradient at that point.

Theorem 3.12 (Subdifferential of differentiable functions)**.** *If a convex function $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable at $\vec{x} \in \mathbb{R}^n$, then its subdifferential at $\vec{x}$ only contains $\nabla f(\vec{x})$.*

Proof. By Theorem 3.7 $\nabla f(\vec{x})$ is a subgradient at $\vec{x}$. Now, let $\vec{g}$ be an arbitrary subgradient at $\vec{x}$. By the definition of subgradient, for any $1 \leq i \leq n$

$$\begin{aligned} f(\vec{x} + \alpha \vec{e}_i) &\geq f(\vec{x}) + \vec{g}^T \alpha \vec{e}_i && (59)\\ &= f(\vec{x}) + \vec{g}[i]\,\alpha, && (60)\\ f(\vec{x}) &\leq f(\vec{x} - \alpha \vec{e}_i) + \vec{g}^T \alpha \vec{e}_i && (61)\\ &= f(\vec{x} - \alpha \vec{e}_i) + \vec{g}[i]\,\alpha, && (62) \end{aligned}$$

where $\vec{e}_i$ is the ith vector in the standard basis (all its entries are equal to zero, except the ith entry which is equal to one). Combining both inequalities

$$\frac{f(\vec{x}) - f(\vec{x} - \alpha \vec{e}_i)}{\alpha} \leq \vec{g}[i] \leq \frac{f(\vec{x} + \alpha \vec{e}_i) - f(\vec{x})}{\alpha}. \tag{63}$$

If we let $\alpha \to 0$, this implies $\vec{g}[i] = \frac{\partial f(\vec{x})}{\partial \vec{x}[i]}$. Consequently, $\vec{g} = \nabla f(\vec{x})$. □

Figure 7 shows a one-dimensional nondifferentiable convex function, along with some of the hyperplanes that support its epigraph. The following theorem establishes that a function is convex if and only if a subgradient exists at every point.

Theorem 3.13. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if it has a non-empty subdifferential at any $\vec{x} \in \mathbb{R}^n$. It is strictly convex if and only for all $\vec{x} \in \mathbb{R}^n$ there exists a subgradient $\vec{g} \in \mathbb{R}^n$ such that*

$$f(\vec{y}) \geq f(\vec{x}) + \vec{g}^T(\vec{y} - \vec{x}), \quad \text{for all } \vec{y} \in \mathbb{R}^n. \tag{64}$$

Proof. One can prove that the subdifferential of any convex function is not empty by applying the Supporting-Hyperplane Theorem, but this is beyond the scope of the course. We refer the interested reader to Section 3.1.5 in [5].

To prove the converse statement, assume that for any $\vec{x}, \vec{y} \in \mathbb{R}^n$ and $\theta \in (0, 1)$ there exists a subgradient $\vec{g}$ of f at $\theta\vec{x} + (1-\theta)\vec{y}$. This implies

$$\begin{aligned} f(\vec{y}) &\geq f(\theta\vec{x} + (1-\theta)\vec{y}) + \vec{g}^T(\vec{y} - \theta\vec{x} - (1-\theta)\vec{y}) && (65)\\ &= f(\theta\vec{x} + (1-\theta)\vec{y}) + \theta\,\vec{g}^T(\vec{y} - \vec{x}), && (66)\\ f(\vec{x}) &\geq f(\theta\vec{x} + (1-\theta)\vec{y}) + \vec{g}^T(\vec{x} - \theta\vec{x} - (1-\theta)\vec{y}) && (67)\\ &= f(\theta\vec{x} + (1-\theta)\vec{y}) + (1-\theta)\,\vec{g}^T(y - \vec{x}). && (68) \end{aligned}$$

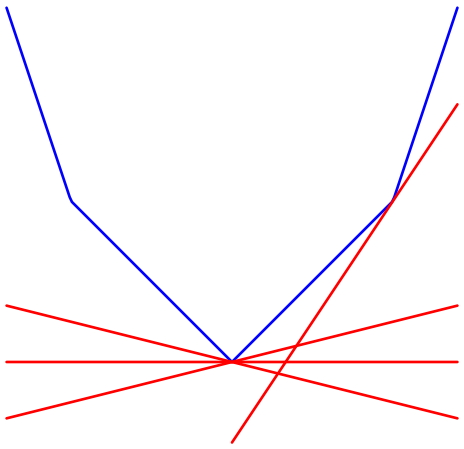

Figure 7: A nondifferentiable convex function (blue). The red supporting lines are specified by subgradients that determine their slope.

Multiplying equation (66) by $1-\theta$ and equation (68) by θ and adding them together yields

$$\theta f(\vec{x}) + (1-\theta) f(\vec{y}) \geq f(\theta\vec{x} + (1-\theta)\vec{y}). \tag{69}$$

□

Subgradients are a useful tool to characterize the minima of nondifferentiable convex functions.

Theorem 3.14 (Optimality condition)**.** *A convex function attains its minimum value at a vector $\vec{x}$ if and only if the zero vector is a subgradient of f at $\vec{x}$. If the function is strictly convex, then the minimum is unique.*

Proof. By the definition of subgradient, if $\vec{g} := \vec{0}$ is a subgradient at $\vec{x}$, then for any $\vec{y} \in \mathbb{R}^n$

$$f(\vec{y}) \geq f(\vec{x}) + \vec{g}^T(\vec{y} - \vec{x}) = f(\vec{x}), \tag{70}$$

which is equivalent to $\vec{x}$ being a global minimum of the function. If the function is strictly convex, then the inequality is strict for all $\vec{y} \neq \vec{x}$. □

A useful property is that the sum of subgradients of two or more functions is a subgradient of their sum.

Lemma 3.15 (Sum of subgradients)**.** *Let $\vec{g}_1$ and $\vec{g}_2$ be subgradients at $\vec{x} \in \mathbb{R}^n$ of $f_1 : \mathbb{R}^n \to \mathbb{R}$ and $f_2 : \mathbb{R}^n \to \mathbb{R}$ respectively. Then $\vec{g} := \vec{g}_1 + \vec{g}_2$ is a subgradient of $f := f_1 + f_2$ at $\vec{x}$.*

Proof. For any $\vec{y} \in \mathbb{R}^n$

$$\begin{aligned} f(\vec{y}) &= f_1(\vec{y}) + f_2(\vec{y}) && (71)\\ &\geq f_1(\vec{x}) + \vec{g}_1^T(\vec{y} - \vec{x}) + f_2(\vec{y}) + \vec{g}_2^T(\vec{y} - \vec{x}) && (72)\\ &\geq f(\vec{x}) + \vec{g}^T(\vec{y} - \vec{x}). && (73) \end{aligned}$$

□

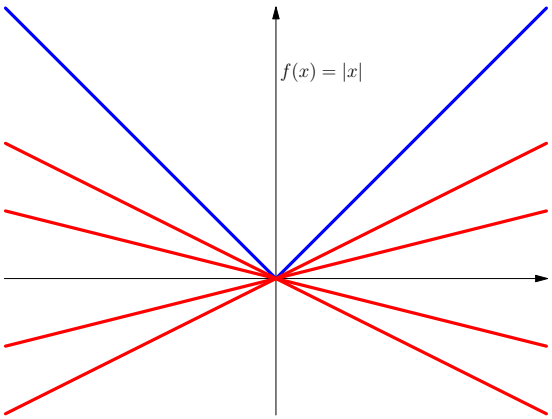

Figure 8: Examples of supporting lines of the absolute value function at the origin. The subgradients at the origin determine the slope of the lines.

Another useful property is that the subgradient of a function scaled by a constant can be obtained by scaling the subgradient.

Lemma 3.16 (Subgradient of scaled function)**.** *Let $\vec{g}_1$ be a subgradient at $\vec{x} \in \mathbb{R}^n$ of $f_1 : \mathbb{R}^n \to \mathbb{R}$. Then for any nonnegative $\alpha \in \mathbb{R}$ $\vec{g}_2 := \alpha \vec{g}_1$ is a subgradient of $f_2 := \alpha f_1$ at $\vec{x}$.*

Proof. For any $\vec{y} \in \mathbb{R}^n$

$$\begin{aligned} f_2(\vec{y}) &= \alpha f_1(\vec{y}) && (74)\\ &\geq \alpha \left(f_1(\vec{x}) + \vec{g}_1^T (\vec{y} - \vec{x}) \right) && (75)\\ &\geq f_2(\vec{x}) + \vec{g}_2^T (\vec{y} - \vec{x}) . && (76) \end{aligned}$$

□

3.4 The ℓ_1 norm

In this section we derive the subdifferential of the ℓ_1 norm. This is important to analyze the effect of ℓ_1-norm regularization. We begin by characterizing the subdifferential in one dimension, where the ℓ_1-norm is just the absolute-value function. Figure 8 shows the supporting hyperplanes (in this case 1D lines) corresponding to a few subgradients.

Lemma 3.17 (Subdifferential of absolute value)**.** *The subdifferential of the absolute value function $|\cdot| : \mathbb{R} \to \mathbb{R}$ at x is equal to $\{\text{sign}(x)\}$ if $x \neq 0$ and to $\{g \in \mathbb{R} \mid |g| \leq 1\}$ if $x = 0$.*

Proof. If $x \neq 0$ the function is differentiable and the only subgradient is equal to the derivative by Theorem 3.12. At $x = 0$, we need

$$\begin{aligned} |y| &= f(0 + y) && (77)\\ &\geq f(0) + g(y - 0) && (78)\\ &\geq gy && (79) \end{aligned}$$

for all $y \in \mathbb{R}$, which holds if and only if $|g| \leq 1$. □

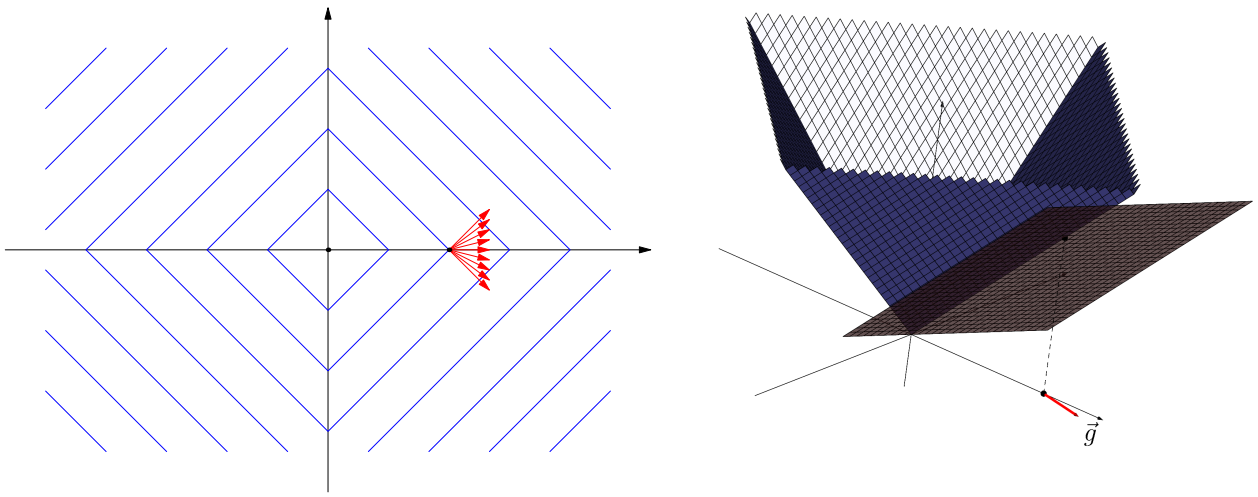

Figure 9: On the left the blue lines are contour lines of the ℓ_1 norm in $\mathbb{R}^2$. The red arrows correspond to subgradients at a point where the function is nondifferentiable. On the right, the graph of the function is shown in blue, and the supporting hyperplane corresponding to one of the subgradients (plotted as a red line with the label $\vec{g}$) is shown in brown.

The following theorem characterizes the subdifferential of the ℓ_1 norm. Figure 9 shows several examples, as well as a supporting hyperplane corresponding to a specific subgradient.

Theorem 3.18 (Subdifferential of ℓ_1 norm)**.** *The subdifferential of the ℓ_1 norm at $\vec{x} \in \mathbb{R}^n$ is the set of vectors $\vec{g} \in \mathbb{R}^n$ that satisfy*

$$\vec{g}[i] = \text{sign}\left(\vec{x}[i]\right) \quad \text{if } \vec{x}[i] \neq 0, \tag{80}$$

$$|\vec{g}[i]| \leq 1 \quad \text{if } \vec{x}[i] = 0. \tag{81}$$

By Hölder's inequality $||g||_\infty \leq 1$ implies $||\vec{y}||_1 \geq \langle g, \vec{y} \rangle$. The fact that $\vec{g}$ is a subgradient follows almost automatically,

$$\begin{aligned} ||\vec{y}||_1 &\geq \langle g, \vec{y} \rangle && (82)\\ &= \langle g, \vec{x} \rangle + \langle g, \vec{y} - \vec{x} \rangle && (83)\\ &= \sum_{i=1}^{n} \vec{x}[i] \,\text{sign}\left(\vec{x}[i]\right) + \langle g, \vec{y} - \vec{x} \rangle && (84)\\ &= ||\vec{x}||_1 + \langle g, \vec{y} - \vec{x} \rangle \,. && (85) \end{aligned}$$

The converse (all subgradients are of that form) is a direct consequence of Lemma 3.17 and the following result.

Lemma 3.19 (Proof in Section 6.5)**.** *The vector $\vec{g} \in \mathbb{R}^n$ is a subgradient of $||\cdot||_1 : \mathbb{R}^n \to \mathbb{R}$ at $\vec{x}$ if and only if $\vec{g}[i]$ is a subgradient of $|\cdot| : \mathbb{R} \to \mathbb{R}$ at $\vec{x}[i]$ for all $1 \leq i \leq n$.*

3.5 The nuclear norm

In this section we derive the subdifferential of the nuclear norm, with the motivation of analyzing methods based on nuclear-norm regularization.

Theorem 3.20 (Subdifferential of the nuclear norm)**.** *Let $X \in \mathbb{R}^{m\times n}$ be a rank-r matrix with SVD USV^T, where $U \in \mathbb{R}^{m\times r}$, $V \in \mathbb{R}^{n\times r}$ and $S \in \mathbb{R}^{r\times r}$ contains the nonzero singular values of X. The subdifferential of the nuclear norm at X is the set of matrices of the form*

$$G := UV^T + W \tag{86}$$

where W satisfies

$$||W|| \leq 1, \tag{87}$$
$$U^T W = 0, \tag{88}$$
$$W\, V = 0. \tag{89}$$

Proof. We only prove that a matrix of the form (86) is a valid subgradient. For the converse (all subgradients are of this form) see [9]. We mimic the proof for the ℓ_1 norm, replacing the ℓ_1 norm by the nuclear norm and the ℓ_∞ norm by the operator norm. First we establish that G has operator norm bounded by one. By Pythagoras' Theorem, for any $\vec{x} \in \mathbb{R}^m$ with unit ℓ_2 norm we have

$$\left|\left|\mathcal{P}_{\text{row}(X)}\, \vec{x}\right|\right|_2^2 + \|\mathcal{P}_{\text{row}(X)^\perp}\, \vec{x}\|_2^2 = ||\vec{x}||_2^2 = 1. \tag{90}$$

By Condition (89), the rows of UV^T are all in row (X) and the rows of W are in row $(X)^\perp$, which implies

$$\begin{aligned}
||G||^2 &:= \max_{\left\{||\vec{x}||_2=1 \mid \vec{x}\in\mathbb{R}^n\right\}} ||G\,\vec{x}||_2^2 && (91)\\
&= \max_{\left\{||\vec{x}||_2=1 \mid \vec{x}\in\mathbb{R}^n\right\}} \left|\left|UV^T\,\vec{x}\right|\right|_2^2 + ||W\,\vec{x}||_2^2 && (92)\\
&= \max_{\left\{||\vec{x}||_2=1 \mid \vec{x}\in\mathbb{R}^n\right\}} \left|\left|UV^T\,\mathcal{P}_{\text{row}(X)}\,\vec{x}\right|\right|_2^2 + \|W\,\mathcal{P}_{\text{row}(X)^\perp}\,\vec{x}\|_2^2 && (93)\\
&\leq \left|\left|UV^T\right|\right|^2 \left|\left|\mathcal{P}_{\text{row}(X)}\,\vec{x}\right|\right|_2^2 + ||W||^2\, \|\mathcal{P}_{\text{row}(X)^\perp}\,\vec{x}\|_2^2 && (94)\\
&\leq 1 \quad \text{by condition (87).} && (95)
\end{aligned}$$

Equality (92) follows from Pythagoras' Theorem because the column spaces of U and W are orthogonal by condition (88), which also implies

$$\langle W, X\rangle = 0. \tag{96}$$

One can think of the matrix UV^T as analogous to the sign of a vector. The sign of a vector $\vec{x}$ is a vector $\text{sign}(\vec{x})$ with ℓ_∞ norm bounded by one that satisfies $\langle \vec{x}, \text{sign}(\vec{x})\rangle = ||\vec{x}||_1$. UV^T has bounded

operator norm and satisfies

$$\begin{aligned}
\left\langle X, UV^T \right\rangle &= \operatorname{tr}\left(X^T U V^T\right) && (97)\\
&= \operatorname{tr}\left(V\, S\, U^T U V^T\right) && (98)\\
&= \operatorname{tr}\left(V^T V\, S\right) && (99)\\
&= \operatorname{tr}\left(S\right) && (100)\\
&= ||X||_* . && (101)
\end{aligned}$$

Finally, for any matrix $Y \in \mathbb{R}^{m \times n}$

$$\begin{aligned}
||Y||_* &\geq \langle G, Y \rangle \qquad \text{by (95) and Theorem 2.12} && (102)\\
&= \langle G, X \rangle + \langle G, Y - X \rangle && (103)\\
&= \left\langle UV^T, X \right\rangle + \langle G, Y - X \rangle \qquad \text{by (96)} && (104)\\
&= ||X||_* + \langle G, Y - X \rangle \qquad \text{by (101).} && (105)
\end{aligned}$$

□

4 Convex Regularization

4.1 Sparse regression

As discussed in Section 1.1, sparse linear regression is the problem of finding a linear model that only uses a small number of features to fit a set of training data. When fitting a sparse linear model we have two objectives:

- Achieving a good fit to the data; i.e. minimizing $\|X\vec{\beta} - \vec{y}\|_2^2$.
- Using a small number of features; i.e. making $\vec{\beta}$ as sparse as possible.

This suggests minimizing a cost function that simultaneously minimizes the fitting error and maximizes the sparsity of the coefficients. In the lecture notes on the SVD we describe ridge regression, where an ℓ_2-norm regularization term penalizes coefficients that are too large to reduce overfitting. Here we would like to penalize the number of nonzeros in the support of the coefficient, i.e. its ℓ_0 "norm" instead. As explained in Section 2.3, this "norm" is intractable to minimize, whereas the ℓ_1 norm is convex and promotes sparsity. This suggests leveraging as a regularizer. In statistics, the solution to an ℓ_1-norm-regularized least-squares problem is called the *lasso* estimator, introduced in [7] (see also [4]).

Definition 4.1 (The lasso)**.** *For $X \in \mathbb{R}^{n \times p}$ and $\vec{y} \in \mathbb{R}^p$, the lasso estimate is the minimizer of the optimization problem*

$$\vec{\beta}_{\text{lasso}} := \arg\min_{\vec{\beta}} \frac{1}{2}\|\vec{y} - X\vec{\beta}\|_2^2 + \lambda\|\vec{\beta}\|_1, \tag{106}$$

where $\lambda > 0$ is a regularization parameter.

The following lemma shows that sums of convex functions are convex, so the lasso cost function is indeed convex.

Lemma 4.2 (Nonnegative weighted sums)**.** *The weighted sum of m convex functions $f_1, \ldots, f_m$*

$$f := \sum_{i=1}^{m} \alpha_i \, f_i \tag{107}$$

is convex as long as the weights $\alpha_1, \ldots, \alpha \in \mathbb{R}$ are nonnegative.

Proof. By convexity of $f_1, \ldots, f_m$, for any $\vec{x}, \vec{y} \in \mathbb{R}^m$ and any $\theta \in (0, 1)$

$$f\left(\theta \vec{x} + (1-\theta)\, \vec{y}\right) = \sum_{i=1}^{m} \alpha_i \, f_i \left(\theta \vec{x} + (1-\theta)\, \vec{y}\right) \tag{108}$$

$$\leq \sum_{i=1}^{m} \alpha_i \left(\theta f_i\left(\vec{x}\right) + (1-\theta)\, f_i\left(\vec{y}\right)\right) \tag{109}$$

$$= \theta \, f\left(\vec{x}\right) + (1-\theta)\, f\left(\vec{y}\right). \tag{110}$$

□

Corollary 4.3 (Regularized least squares)**.** *Regularized least-squares cost functions of the form*

$$||A\vec{x} - \vec{y}||_2^2 + ||\vec{x}||\,, \tag{111}$$

where $||\cdot||$ is an arbitrary norm, are convex.

In the following example, we apply the lasso to a real-world dataset.

Example 4.4 (Temperature prediction via sparse regression)**.** We consider the same dataset of hourly temperatures measured at weather stations all over the United States[1] that we discussed in the lecture notes on the SVD. Our goal is to design a model that can be used to estimate the temperature in Jamestown (North Dakota) from the temperatures of 133 other stations. We perform estimation by fitting a linear model where the response is the temperature in Jamestown and the features are the rest of the temperatures ($p = 133$). We use 10^3 measurements from 2015 as a test set, and train a ridge-regression and a lasso model using a variable number of training data also from 2015 but disjoint from the test data. In addition, we test both models on data from 2016.

Figure 10 compares the coefficients obtained by the lasso and ridge regression for different values of the regularization parameter λ when the number of data n equals 135. Since the number of features is just 133, the least-squares estimator severely overfits the training data. This is evident in the large magnitude of the coefficients for small values of λ. As λ increases the coefficients of the ridge-regression and lasso estimators shrink, limiting the overfitting effect. However, there is a striking difference: the ridge-regression coefficients all shrink simultaneously, whereas the lasso coefficients shrink sequentially yielding increasingly sparse models. The left image in Figure 11 shows that this shrinkage indeed controls overfitting and results in an improved validation error

[1]The data are available at http://www1.ncdc.noaa.gov/pub/data/uscrn/products

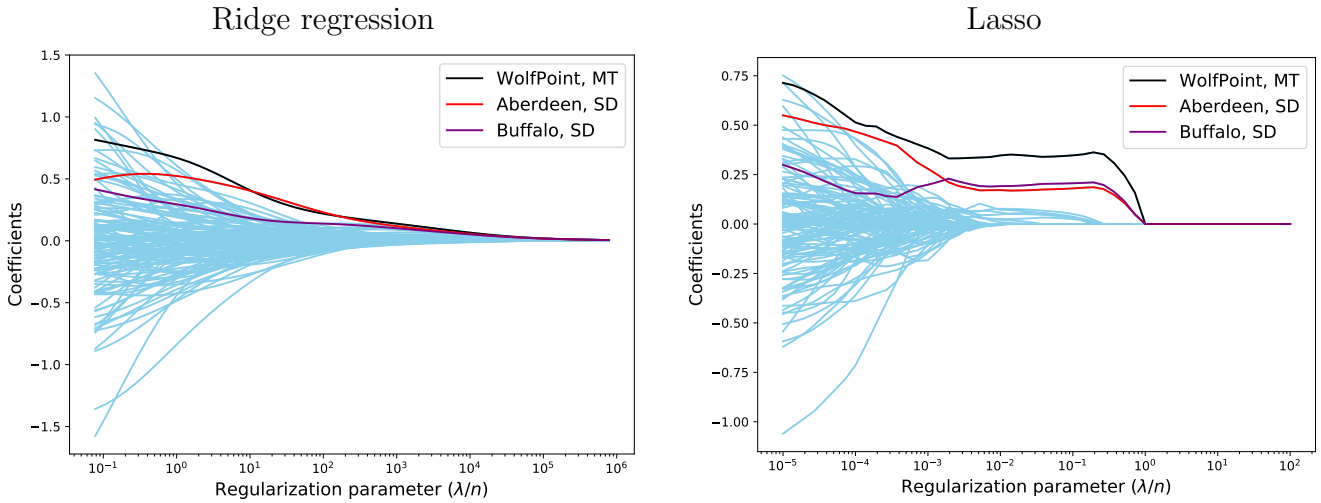

Figure 10: Coefficients of the ridge-regression (left) and lasso (right) estimators computed from the data described in Example 4.4 for different values of the regularization parameter λ. The number of training data is fixed to $n := 135$ training data. All coefficients are depicted in light blue except the three that have the largest magnitudes for large n.

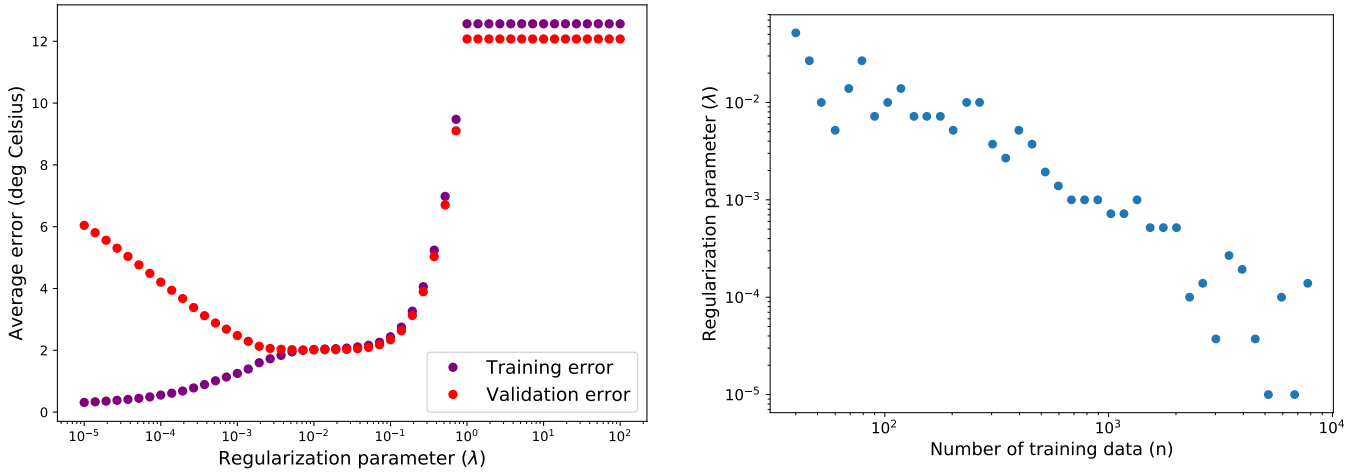

Figure 11: The left image shows the training and validation RMSE of the lasso estimator on the temperature data described in Example 4.4 when the number of training data is fixed to $n := 135$ training data. The right graph shows the values of λ selected from a validation dataset of size 100 for different values of n.

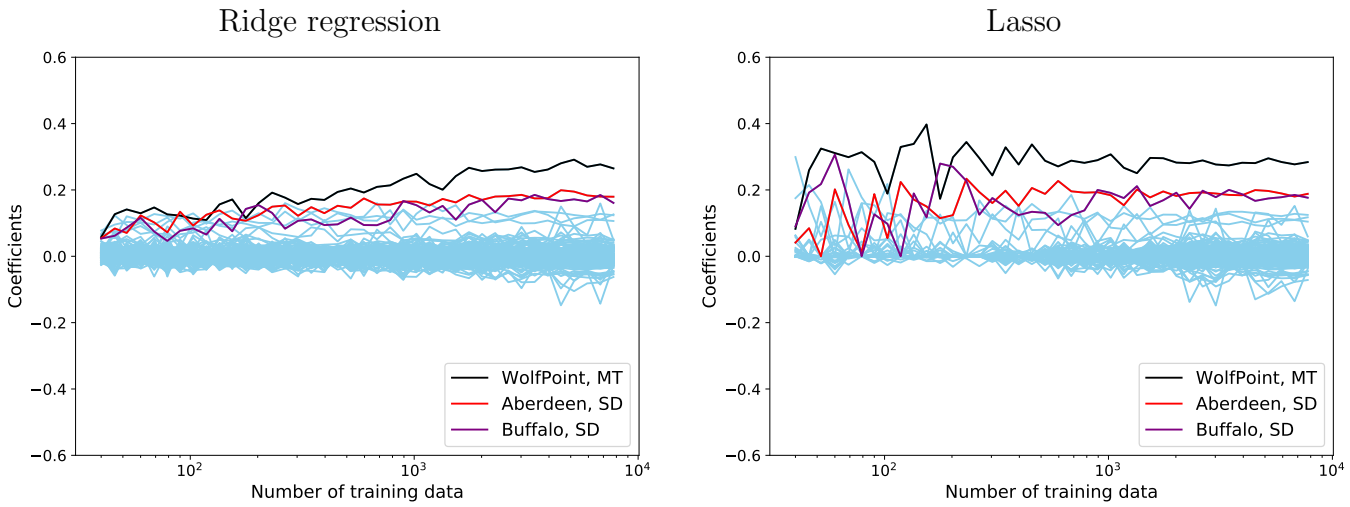

Figure 12: Coefficients of the ridge-regression (left) and lasso (right) estimators computed from the data described in Example 4.4 for different sizes of the training dataset. The coefficients to a value of λ chosen via cross validation. All coefficients are depicted in light blue except the three that have the largest magnitudes for large n.

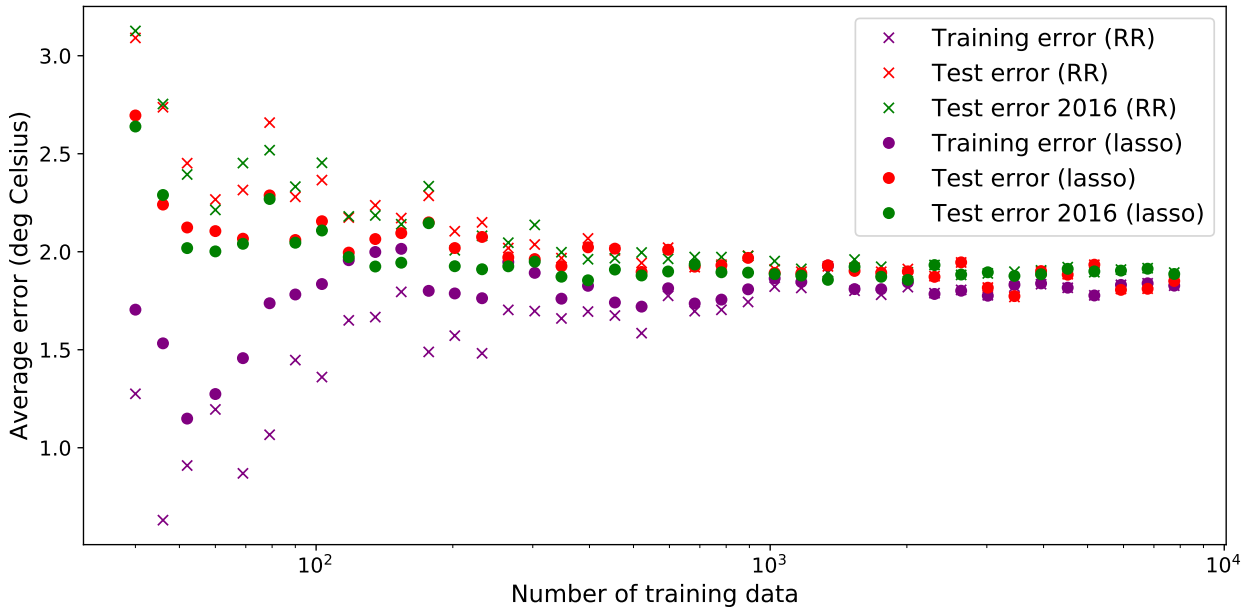

Figure 13: Performance of the lasso estimator on the temperature data described in Example 4.4. The graph shows the RMSE achieved by the models on the training and test sets, and on the 2016 data, for different number of training data and compares it to the RMSE achieved by ridge regression.

for the lasso. The right image shows the values of λ that minimize validation error for different values of n. As expected, larger values of λ are more useful for smaller values of n, where we need regularization to avoid overfitting. Figure 12 shows the ridge-regression and lasso coefficients corresponding to the optimal λ for values of n. The lasso results in a much sparser linear model, except for very large values of n where both estimators approach the least-squares solution. Finally, Figure 13 shows the performance of the lasso and ridge-regression estimators on a held-out test set, as well as on data from another year. The lasso achieves better prediction for all values of n, up until the point where the estimators approach the least-squares solution. △

4.2 Soft thresholding

In order to gain intuition as to how minimizing the ℓ_1 norm promotes sparsity we study the corresponding proximal operator. The proximal operator of a function f maps vectors to nearby points corresponding to small values of f. This is achieved by minimizing a cost function that combines f with a least-squares term that penalizes distant points.

Definition 4.5 (Proximal operator)**.** *The proximal operator of a function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as*

$$\text{prox}_f\left(\vec{y}\right) := \arg\min_{\vec{x}} f\left(\vec{x}\right) + \frac{1}{2}\left|\left|\vec{x} - \vec{y}\right|\right|_2^2. \tag{112}$$

The proximal operator of the squared ℓ_2 norm just scales vectors by a constant.

Lemma 4.6. *The proximal operator of the squared ℓ_2 norm weighted by a constant $\alpha > 0$ is the scaling operator*

$$\text{prox}_{\alpha\left|\left|\cdot\right|\right|_2^2}\left(\vec{y}\right) = \frac{\vec{y}}{1+2\alpha}. \tag{113}$$

Proof. By Lemma 3.4 the gradient of the function

$$f(\vec{x}) := \alpha\left|\left|\vec{x}\right|\right|_2^2 + \frac{1}{2}\left|\left|\vec{x} - \vec{y}\right|\right|_2^2 \tag{114}$$

$$= \left(\frac{1}{2} + \alpha\right)\vec{x}^T\vec{x} + \frac{1}{2}\vec{y}^T\vec{y} - \vec{y}^T\vec{x} \tag{115}$$

equals

$$\nabla f(\vec{x}) = \left(1+2\alpha\right)\vec{x} - \vec{y}. \tag{116}$$

Setting it to zero yields the result. □

In contrast to the ℓ_2 norm, the proximal operator associated to the ℓ_1 norm shrinks each entry individually towards zero. The operator is known as soft thresholding because it applies a continuous thresholding function, as opposed to the hard-thresholding operator that we discussed in the notes on signal representations. Figure 14 shows both thresholding functions.

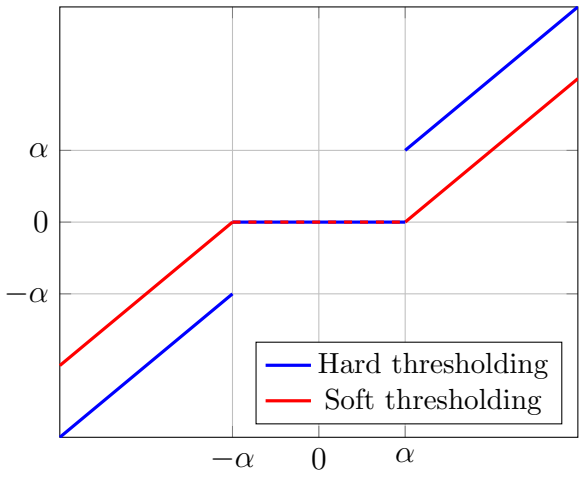

Figure 14: Hard-thresholding and soft-thresholding functions.

Theorem 4.7 (Proximal operator of ℓ_1 norm)**.** *The proximal operator of the ℓ_1 norm weighted by a constant $\alpha > 0$ is the soft-thresholding operator*

$$\text{prox}_{\alpha ||\cdot||_1}(y) = \mathcal{S}_\alpha(\vec{y}) \tag{117}$$

where

$$\mathcal{S}_\alpha(\vec{y})[i] := \begin{cases} \vec{y}[i] - \text{sign}(\vec{y}[i])\,\alpha & \text{if } |\vec{y}[i]| \geq \alpha, \\ 0 & \text{otherwise.} \end{cases} \tag{118}$$

Proof. By Lemmas 3.15 and 3.16 the subgradients of the function

$$f(\vec{x}) := \alpha ||\vec{x}||_1 + \frac{1}{2} ||\vec{x} - \vec{y}||_2^2 \tag{119}$$

at $\vec{x}$ equal

$$\vec{g}_{\text{prox}} = \alpha \vec{g}_{\ell_1}(\vec{x}) + \vec{x} - \vec{y} \tag{120}$$

where $\vec{g}_{\ell_1}(\vec{x})$ is any subgradient of the ℓ_1 norm at $\vec{x}$. To find the minimum of the function we set $\vec{g}_{\text{prox}}$ to zero, which yields the system of equations

$$\vec{x}_{\text{min}} + \alpha \vec{g}_{\ell_1}(\vec{x}_{\text{min}}) = \vec{y}. \tag{121}$$

By Theorem 3.18, if the jth entry of $\vec{x}_{\text{min}}$ is nonzero, then $\vec{g}_{\ell_1}(\vec{x}_{\text{min}})[j] = \text{sign}(\vec{x}_{\text{min}}[j])$ so that

$$\vec{x}_{\text{min}}[j] = \vec{y}[j] - \alpha\,\text{sign}(\vec{x}[j]). \tag{122}$$

This is fine if $|\vec{y}[j]| \geq \alpha$. Otherwise it generates a contradiction. Let $0 < \vec{y}[j] < \alpha$. If $\vec{x}_{\text{min}}[j] > 0$ then $\vec{x}_{\text{min}}[j] = \vec{y}[j] - \alpha\,\text{sign}(\vec{x}[j]) < 0$. Similarly, if $\vec{x}_{\text{min}}[j] < 0$ then $\vec{x}_{\text{min}}[j] = \vec{y}[j] - \alpha\,\text{sign}(\vec{x}[j]) > 0$. The same happens if $-\alpha < \vec{y}[j] < 0$. We conclude that if $|\vec{y}[j]| < \alpha$ then $\vec{x}_{\text{min}}[j] = 0$. Note that this is consistent with the system of equations (121) because then $|\vec{g}_{\ell_1}(\vec{x}_{\text{min}})[j]| = \vec{y}[j]/\alpha \leq 1$. □

The following lemma provides a connection between regularized linear regression and proximal operators in a simplified case where the feature matrix has orthonormal columns.

Lemma 4.8. *Let $U \in \mathbb{R}^{n \times p}$ be a matrix with orthonormal columns and $\vec{y} \in \mathbb{R}^n$, then for any function f*

$$\arg\min_{\vec{\beta}} \frac{1}{2} \left|\left|\vec{y} - U\vec{\beta}\right|\right|_2^2 + f(\vec{\beta}) = \arg\min_{\vec{\beta}} \frac{1}{2} \left|\left|U^T\vec{y} - \vec{\beta}\right|\right|_2^2 + f(\vec{\beta}). \tag{123}$$

Proof. We have

$$\begin{aligned}
\frac{1}{2} \left|\left|\vec{y} - U\vec{\beta}\right|\right|_2^2 + f(\vec{\beta}) &= \frac{1}{2}\vec{\beta}^T U^T U\vec{\beta} + \frac{1}{2}\vec{y}^T\vec{y} - \vec{y}^T U\vec{\beta} + f(\vec{\beta}) && (124)\\
&= \frac{1}{2}\vec{\beta}^T\vec{\beta} + \frac{1}{2}\vec{y}^T\vec{y} - (U^T\vec{y})^T\vec{\beta} + f(\vec{\beta}) && (125)\\
&= \frac{1}{2} \left|\left|U^T\vec{y} - \vec{\beta}\right|\right|_2^2 + f(\vec{\beta}) + \frac{1}{2}\vec{y}^T\vec{y} - \frac{1}{2}\vec{y}^T UU^T\vec{y}. && (126)
\end{aligned}$$

□

Note that for orthogonal columns, the least-squares estimator equals exactly $U^T\vec{y}$. Lemmas 4.8 and 4.6 imply that ridge regression just scales all least-squares coefficients by the same factor. By Lemma 4.7 the lasso does something very different: it soft-thresholds the least-squares coefficients. Even though this only holds for matrices with orthogonal columns, it is consistent with the coefficient trajectories that we observe in Figure 11 (where the columns are not orthogonal).

4.3 Matrix completion

In this section we consider the problem of completing a low-rank matrix from a subset of its entries, motivated by the collaborative-filtering problem. The goal is to achieve a tradeoff between approximating the observed entries well, and obtaining a low-rank solution. In the notes on the SVD we established that is all entries are observed then truncating the SVD achieves the optimal low-rank approximation in Frobenius norm. However, when there are missing entries minimizing the rank directly is intractable, as discussed in Section 2.4. Instead, the nuclear norm is an attractive alternative, because it is convex and promotes low-rank structure. Combining it with a least-squares term yields a popular approach for matrix completion.

Definition 4.9 (Nuclear-norm regularized least squares)**.** *Let $\vec{y} \in \mathbb{R}^m$ contain the m observed entries in a $n_1 \times n_2$ matrix, and let Ω denote the indices of the observations, so that for any matrix $M \in \mathbb{R}^{n_1 \times n_2}$ the vector $M_\Omega \in \mathbb{R}^m$ contains the corresponding entries. The nuclear-norm regularized least squares estimator given Ω and $\vec{y}$ is the solution of the optimization problem*

$$\min_{X \in \mathbb{R}^{n_1 \times n_2}} \frac{1}{2} ||X_\Omega - \vec{y}||_2^2 + \lambda ||X||_*, \tag{127}$$

where $\lambda > 0$ is a regularization parameter.

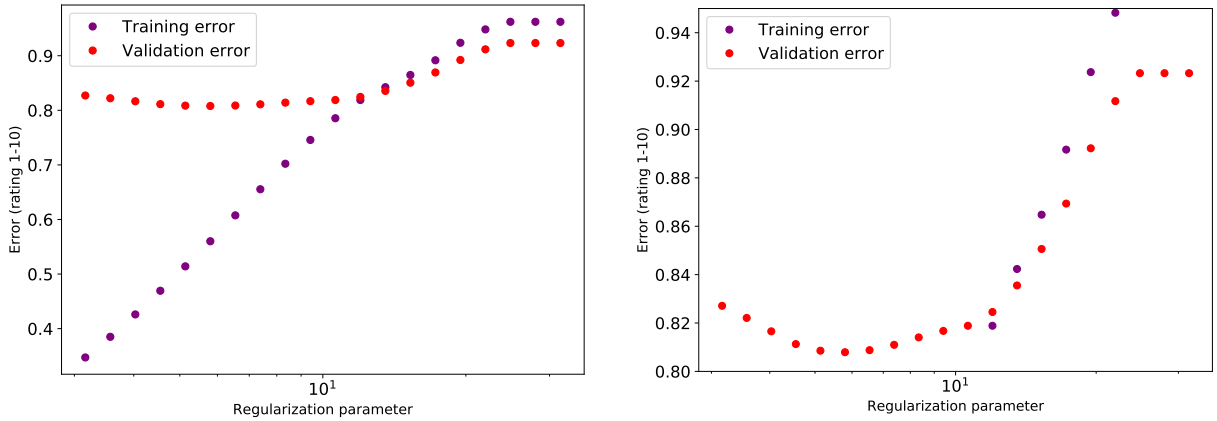

Figure 15: Training and validation error for the experiment described in Example 4.10 when $n_{\text{train}} := 4{,}600$ for different values of the regularization parameter λ.

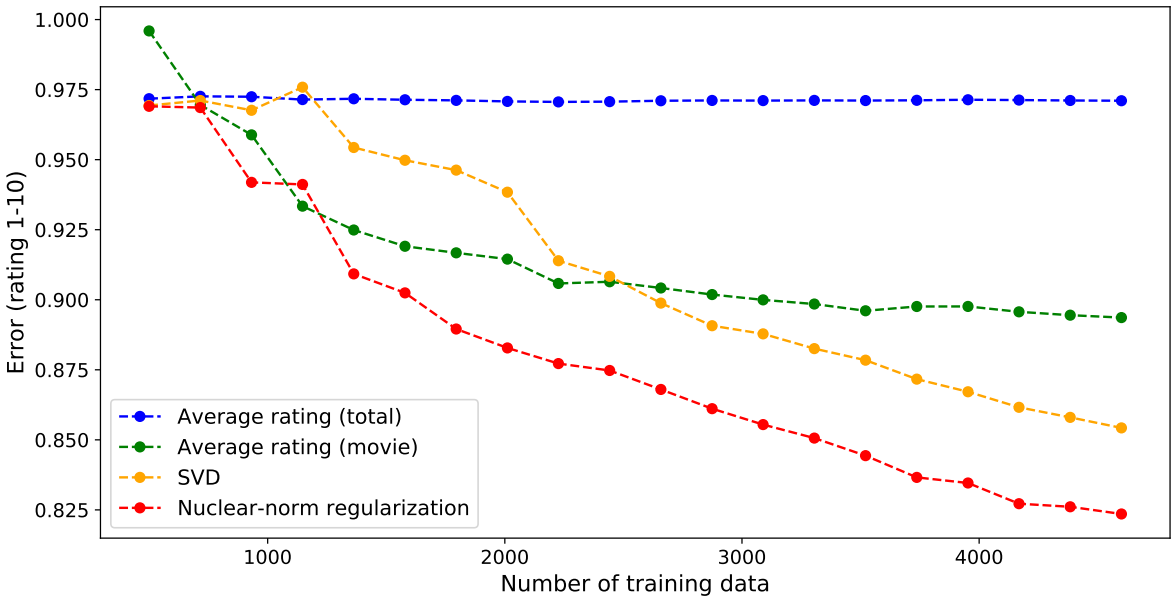

Figure 16: Test error of nuclear-norm regularized least-squares prediction for the data described in Example 4.10, compared to the average rating, the average rating per movie, and truncating the SVD of the data.

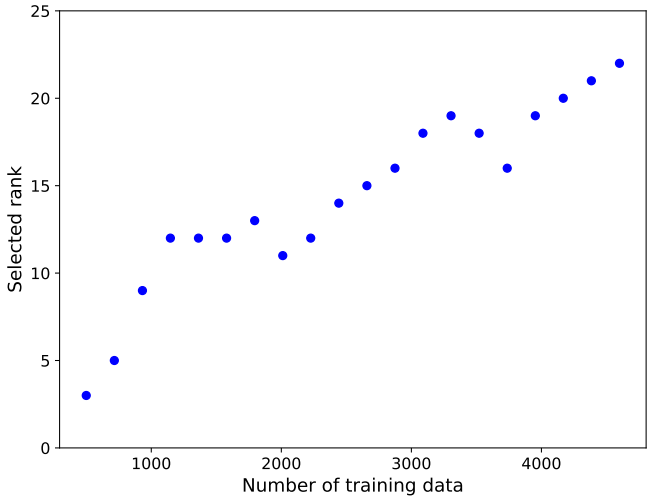

Figure 17: Rank selected using cross validation for different amounts of training data in Example 4.10.

Example 4.10 (Collaborative filtering)**.** In this example, we use the same dataset as the one in the notes on the SVD: the Movielens dataset[2], which contains ratings given by a group of users to popular movies. The ratings are integers between 1 and 10. We select the 100 users and movies with most ratings to form the matrix of ratings. From the 10^4 possible ratings, 6,031 are observed. The test set consists of 10^3 ratings. We separate the remaining observed ratings into a training set of size n_{train} and a validation set of size $n_{\text{val}} := \max\{n_{\text{train}}, 400\}$.

In order to perform collaborative filtering, we solve the nuclear-norm regularized least-squares problem in Eq. (127), setting $\vec{y}$ to equal the centered observed ratings (i.e. after subtracting their average). The left image of Figure (15) shows the training and validation errors when $n_{\text{train}} :=$ 4,600 for different values of the regularization parameter λ. For small λ the solution overfits the observed entries. Increasing λ reduces overfitting, which results in a larger training error and a smaller test error. When λ becomes too large the test error increases again and eventually plateaus once the solution becomes the zero matrix. Figure 16 shows the test error achieved by the method, compared to predictions based on the average rating, on the average rating per movie, and on a truncated SVD of the data. The value of λ is set via cross validation. Nuclear-norm regularized least squares clearly outperforms the other approaches. Figure 17 shows the rank of the solution obtained by the method for different amounts of training data. The method favors solutions with increasing rank as more data becomes available. △

In order to gain intuition about the effect of nuclear-norm regularization, we derive its proximal operator in the following theorem. Applying the operator to a matrix is equivalent to soft-thresholding its singular values. Note that the proximal operator only solves the problem in Eq. (127) when all entries are observed, so this provides some intuition, but does not completely characterize why nuclear-norm regularization succeeds in performing matrix completion.

[2]The data are available at https://grouplens.org/datasets/movielens.

Theorem 4.11 (Proximal operator of the nuclear norm)**.** *The solution to the problem*

$$\min_{X \in \mathbb{R}^{m \times n}} \frac{1}{2} ||Y - X||_{\mathrm{F}}^2 + \alpha ||X||_* \tag{128}$$

is $\mathcal{D}_\alpha(Y)$*, obtained by soft-thresholding the singular values of* $Y = USV^T$

$$\mathcal{D}_\alpha(Y) := U \mathcal{S}_\alpha(S) V^T, \tag{129}$$

$$\mathcal{S}_\alpha(S)_{ii} := \begin{cases} S_{ii} - \alpha & \text{if } S_{ii} > \alpha, \\ 0 & \text{otherwise.} \end{cases} \tag{130}$$

Proof. Due to the Frobenius norm term, the cost function is strictly convex. This implies that any point at which there exists a subgradient that is equal to zero is the solution to the optimization problem. The subgradients of the cost function at X are of the form,

$$X - Y + \alpha G, \tag{131}$$

where G is a subgradient of the nuclear norm at X. If we can show that

$$\frac{1}{\alpha}(Y - D_\alpha(Y)) \tag{132}$$

is a subgradient of the nuclear norm at $D_\alpha(Y)$ then $D_\alpha(Y)$ is the solution.

Let us separate the singular-value decomposition of Y into the singular vectors corresponding to singular values greater than α– denoted by U_0 and V_0– and the rest

$$Y = U SV^T \tag{133}$$

$$= \begin{bmatrix} U_0 & U_1 \end{bmatrix} \begin{bmatrix} S_0 & 0 \\ 0 & S_1 \end{bmatrix} \begin{bmatrix} V_0 & V_1 \end{bmatrix}^T. \tag{134}$$

Note that $D_\alpha(Y) = U_0(S_0 - \alpha I) V_0^T$, so that

$$\frac{1}{\alpha}(Y - D_\alpha(Y)) = U_0 V_0^T + \frac{1}{\alpha} U_1 S_1 V_1^T. \tag{135}$$

By construction all the singular values of $U_1 S_1 V_1^T$ are smaller than α, so

$$\left|\left| \frac{1}{\alpha} U_1 S_1 V_1^T \right|\right| \leq 1. \tag{136}$$

In addition, by definition of the singular-value decomposition $U_0^T U_1 = 0$ and $V_0^T V_1 = 0$. As a result, (135) is a subgradient of the nuclear norm at $D_\alpha(Y)$ and the proof is complete. □

5 Optimization algorithms

5.1 Gradient descent

In this section we describe a simple method to solve the optimization problem

$$\min_{\vec{x} \in \mathbb{R}^n} \quad f(\vec{x}), \tag{137}$$

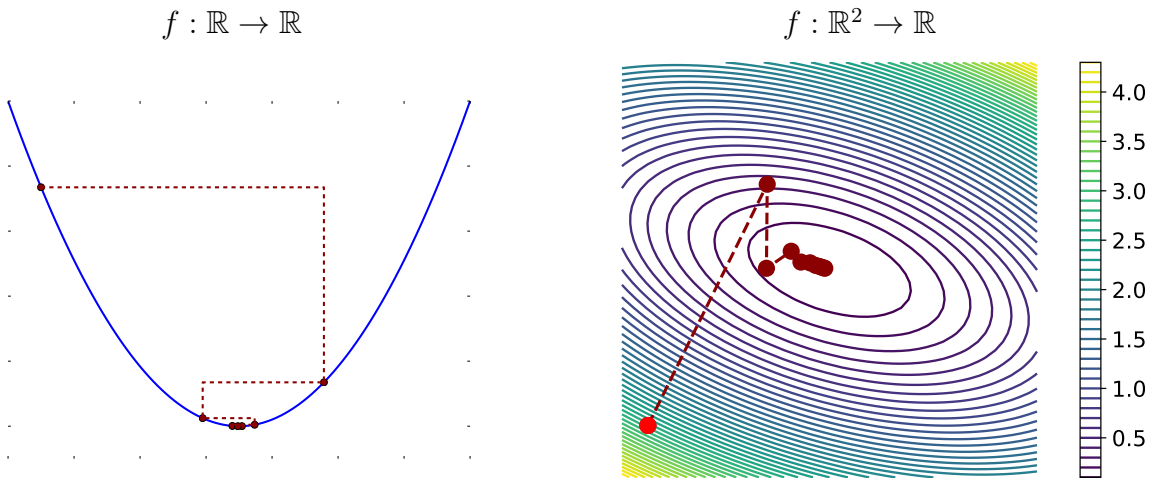

Figure 18: Iterations of gradient descent applied to a univariate (left) and a bivariate (right) function. The algorithm converges to the minimum in both cases.

when f is differentiable and convex. By Theorem 2.4 any local minimum of the function is also a global minimum. This motivates making progress towards a solution by exploiting local first-order information. Gradient descent uses information encoded in the gradient. The idea is to take steps in the direction of steepest descent, which is $-\nabla f\left(\vec{x}\right)$ by Corollary 3.3.

Algorithm 5.1 (Gradient descent)**.** *Given an initial point $\vec{x}^{(0)} \in \mathbb{R}^n$, update by setting*

$$\vec{x}^{(k+1)} := \vec{x}^{(k)} - \alpha_k \nabla f\left(\vec{x}^{(k)}\right), \tag{138}$$

where the step size α_k is a nonnegative real number, until a stopping criterion is met.

The stopping criterion should quantify how close we are to the minimum, or how satisfied we are with the current solution. A popular criterion is to check when the relative progress

$$\frac{\left|\left|\vec{x}^{(k+1)} - \vec{x}^{(k)}\right|\right|_2}{\left|\left|\vec{x}^{(k)}\right|\right|_2} \tag{139}$$

or the norm of the gradient are below a predefined tolerance. Figure 18 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 18 the step size is constant. In practice, determining a constant step that is adequate for a particular function can be challenging. Figure 19 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. Consider the 1D function obtained by restricting f to the line that intersects $f\left(\vec{x}^{(k)}\right)$ and lies in the direction of the gradient

$$h\left(\alpha\right) := f\left(\vec{x}^{(k)} - \alpha \nabla f\left(\vec{x}^{(k)}\right)\right) \tag{140}$$

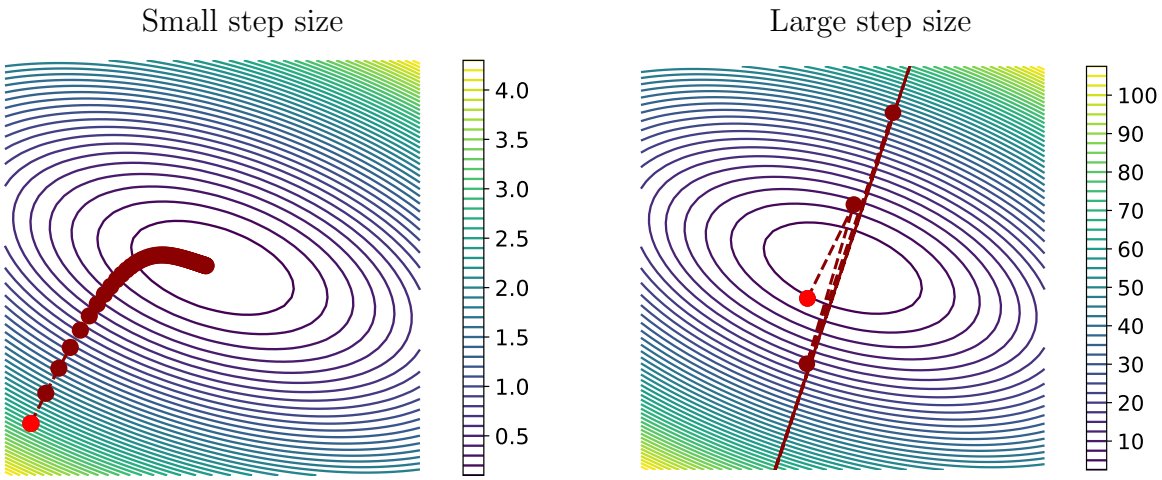

Figure 19: Iterations of gradient descent when the step size is small (left) and large (right). In the first case the convergence is very small, whereas in the second the algorithm diverges away from the minimum. The initial point is bright red.

The function is differentiable, and also convex by Lemma 2.3. This suggests setting the step size by minimizing h, i.e. $\alpha_k := \arg\min_\alpha h(\alpha)$. However, the 1D minimization problem may be costly to solve. The backtracking line search is an alternative heuristic that produces a similar result at less cost. By convexity of h we have

$$h(\alpha) \geq h(0) + \alpha h'(0). \tag{141}$$

By differentiability, this is the only hyperplane (in this case a line) that is tangent to h at 0 and is a lower bound. Consequently for α small enough

$$\begin{aligned} h(\alpha) &\leq h(0) + \frac{\alpha}{2} h'(0) && (142)\\ &= f\left(\vec{x}^{(k)}\right) + \frac{\alpha}{2} \nabla f\left(\vec{x}^{(k)}\right)^T \left(-\nabla f\left(\vec{x}^{(k)}\right)\right) \quad \text{by the chain rule} && (143)\\ &= f\left(\vec{x}^{(k)}\right) - \frac{\alpha}{2} \left|\left|\nabla f\left(\vec{x}^{(k)}\right)\right|\right|_2^2 && (144) \end{aligned}$$

since otherwise this would also be a lower-bounding tangent line. Reformulating the inequality in terms of f yields

$$f\left(\vec{x}^{(k)} - \alpha \nabla f\left(\vec{x}^{(k)}\right)\right) \leq f\left(\vec{x}^{(k)}\right) - \frac{\alpha}{2} \left|\left|\nabla f\left(\vec{x}^{(k)}\right)\right|\right|_2^2, \tag{145}$$

which is known as the Armijo rule. The following line-search scheme exploits this insight to set the step size. The idea is to start with a large step size and reduce it geometrically until the condition is met.

Algorithm 5.2 (Backtracking line search with Armijo rule)**.** *Set $\alpha_k := 0.5^i$ where i is the smallest integer $i \geq 0$ such that*

$$f\left(\vec{x}^{(k)} - \alpha_k \nabla f\left(\vec{x}^{(k)}\right)\right) \leq f\left(\vec{x}^{(k)}\right) - \frac{\alpha_k}{2} \left|\left|\nabla f\left(\vec{x}^{(k)}\right)\right|\right|_2^2. \tag{146}$$

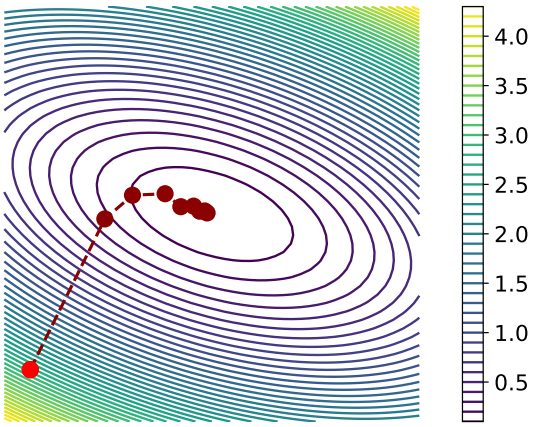

Figure 20: Gradient descent using a backtracking line search based on the Armijo rule. The function is the same as in Figure 19.

Figure 20 shows the result of applying gradient descent with a backtracking line search to the same example as in Figure 19. In this case, the line search manages to adjust the step size so that the method converges.

5.2 Gradient descent for least squares

In this section we study gradient descent applied to the least-squares cost function

$$\text{minimize}_{\vec{\beta}\in\mathbb{R}^p} \frac{1}{2} \left|\left| \vec{y} - X\vec{\beta} \right|\right|_2^2, \tag{147}$$

where $X \in \mathbb{R}^{n\times p}$ and $\vec{y} \in \mathbb{R}^n$. By Corollary 3.5 the gradient equals

$$\nabla f(\vec{\beta}) = X^T X\vec{\beta} - X^T\vec{y}, \tag{148}$$

so the gradient-descent updates are

$$\vec{\beta}^{(k+1)} = \vec{\beta}^{(k)} + \alpha_k X^T \left(\vec{y} - X\vec{\beta}^{(k)}\right). \tag{149}$$

This has a very intuitive interpretation in terms of the linear-regression problem where each row of X is a feature vector $\vec{x}^{\,(i)} \in \mathbb{R}^p$, $1 \leq i \leq n$, corresponding to an entry $y^{(i)}$ of $\vec{y}$. In that case the updates boil down to

$$\vec{\beta}^{(k+1)} = \vec{\beta}^{(k)} + \alpha_k \sum_{i=1}^{n} \left(y^{(i)} - \langle \vec{x}^{(i)}, \vec{\beta}^{(k)} \rangle\right) \vec{x}^{(i)}. \tag{150}$$

Gradient descent iteratively corrects the coefficient vector. If $y^{(i)}$ is larger than $\langle \vec{x}^{(i)}, \vec{\beta}^{(k)} \rangle$ we add a small multiple of $\vec{x}^{(i)}$ in order to reduce the difference. If it is smaller we subtract it.

The following theorem provides a closed-form solution for the iterations of gradient descent applied to least squares.

Theorem 5.3. *Let $X^{n\times p}$, $n \geq p$, be full rank. The $k+1$th iteration of gradient descent with a constant step size $\alpha > 0$ applied to the least-squares cost function in Eq. (147) equals*

$$\vec{\beta}^{(k+1)} = V \operatorname{diag}_{j=1}^{p}\left(\left(1-\alpha s_j^2\right)^{k+1}\right) V^T\vec{\beta}^{(0)} + V \operatorname{diag}_{j=1}^{p}\left(\frac{1-\left(1-\alpha s_j^2\right)^{k+1}}{s_j}\right) U^T\vec{y}, \quad k = 1, 2, 3\ldots,$$

where $X = USV^T$ is the SVD of X, $\vec{\beta}^{(0)} \in \mathbb{R}^p$ is the initial coefficient vector, and $\operatorname{diag}_{j=1}^{p}(d_i)$ denotes a diagonal matrix with entries $d_1, \ldots, d_p$.

Proof. We reformulate Eq. (149) as

$$\vec{\beta}^{(k+1)} = \left(I - \alpha X^T X\right)\vec{\beta}^{(k)} + \alpha X^T\vec{y}, \tag{151}$$

which yields

$$\vec{\beta}^{(k+1)} = \left(I - \alpha X^T X\right)^{k+1}\vec{\beta}^{(0)} + \sum_{i=0}^{k}\left(I - \alpha X^T X\right)^{i}\alpha X^T\vec{y}. \tag{152}$$

Let $X = USV^T$ be the SVD of X. Since $p \leq n$ and X is full rank, we have $VV^T = V^TV = I$, so that

$$\vec{\beta}^{(k+1)} = \left(VV^T - \alpha VS^2V^T\right)^{k+1}\vec{\beta}^{(0)} + \alpha\sum_{i=0}^{k}\left(VV^T - \alpha VS^2V^T\right)^{i} VSU^T\vec{y} \tag{153}$$

$$= V\left(I - \alpha S^2\right)^{k+1}V^T\vec{\beta}^{(0)} + \alpha V\sum_{i=0}^{k}\left(I - \alpha S^2\right)^{i} SU^T\vec{y} \tag{154}$$

$$= V \operatorname{diag}_{j=1}^{p}\left(\left(1-\alpha s_j^2\right)^{k+1}\right) V^T\vec{\beta}^{(0)} + \alpha V \operatorname{diag}_{j=1}^{p}\left(\sum_{i=0}^{k}\left(1-\alpha s_j^2\right)^{i}\right) SU^T\vec{y}. \tag{155}$$

By the geometric-sum formula we conclude:

$$\vec{\beta}^{(k+1)} = V \operatorname{diag}_{j=1}^{p}\left(\left(1-\alpha s_j^2\right)^{k+1}\right) V^T\vec{\beta}^{(0)} + \alpha V \operatorname{diag}_{j=1}^{p}\left(\frac{1-\left(1-\alpha s_j^2\right)^{k+1}}{\alpha s_j^2}\right) SU^T\vec{y}. \tag{156}$$

□

An immediate consequence is that gradient descent converges to the optimal solution if the step size is small enough.

Corollary 5.4. *Let $0 < \alpha < 2/s_1^2$, where s_1 is the largest singular value of X. If X is full rank, gradient descent with step size α converges to the minimum of the least-squares cost function.*

Proof. If $0 < \alpha < 2/s_1^2 \leq 2/s_j^2$ for $1 \leq j \leq p$ then $\left|1-\alpha s_j^2\right| < 1$ so $\lim_{k\to\infty}\left(1-\alpha s_j^2\right)^k = 0$. This implies

$$\lim_{k\to\infty}\vec{\beta}^{(k)} = \lim_{k\to\infty} V \operatorname{diag}_{j=1}^{p}\left(\left(1-\alpha s_j^2\right)^{k}\right) V^T\vec{\beta}^{(0)} + V \operatorname{diag}_{j=1}^{p}\left(\frac{1-\left(1-\alpha s_j^2\right)^{k}}{s_j}\right) U^T\vec{y} \tag{157}$$

$$= VS^{-1}U^T\vec{y}, \tag{158}$$

which is the solution to the least-squares problem (see Eq. (56) or the notes on the SVD). □

The rate of convergence of the gradient-descent iterations to the least-squares solution is governed by the condition number of the feature matrix.

Corollary 5.5. *Let $\vec{y}_{\text{LS}} := X\vec{\beta}_{\text{LS}}$, where $\vec{\beta}_{\text{LS}}$ is the solution to the least-squares problem, and $\vec{y}^{(k)} := X\vec{\beta}^{(k)}$, where $\vec{\beta}^{(k)}$ is the kth iteration of gradient descent initialized with zero coefficients. If the step size is set to $\alpha := 1/s_1^2$ then*

$$\frac{\left|\left|\vec{y}_{\text{LS}} - \vec{y}^{(k)}\right|\right|_2}{||\vec{y}||_2} \leq \left(1 - \frac{s_p^2}{s_1^2}\right)^k, \tag{159}$$

where s_1 is the largest singular value of X and s_p is the smallest.

Proof. By the theorem

$$\vec{y}^{(k)} = X\vec{\beta}^{(k)} \tag{160}$$

$$= USV^TV \operatorname{diag}_{j=1}^{p}\left(\frac{1 - \left(1 - \alpha s_j^2\right)^k}{s_j}\right) U^T\vec{y} \tag{161}$$

$$= UU^T\vec{y} - U \operatorname{diag}_{j=1}^{p}\left(\left(1 - \alpha s_j^2\right)^k\right) U^T\vec{y}. \tag{162}$$

Since $\vec{y}_{\text{LS}} = UU^T\vec{y}$ by Eq. (56) (or the notes on the SVD),

$$\left|\left|\vec{y}_{\text{LS}} - \vec{y}^{(k)}\right|\right|_2 = \left|\left|U \operatorname{diag}_{j=1}^{p}\left(\left(1 - \alpha s_j^2\right)^k\right) U^T\vec{y}\right|\right|_2 \tag{163}$$

$$\leq ||U|| \left|\left|\operatorname{diag}_{j=1}^{p}\left(\left(1 - \alpha s_j^2\right)^k\right)\right|\right| \left|\left|U^T\vec{y}\right|\right|_2 \tag{164}$$

$$\leq \left|1 - \frac{s_p^2}{s_1^2}\right|^k ||\vec{y}||_2 \tag{165}$$

because $\left(1 - \alpha s_p^2\right)^k$ is the largest singular value of the diagonal matrix, and U has orthonormal columns. □

If the matrix is well conditioned, convergence is extremely fast. If there are singular values that are much smaller than the rest, gradient descent can take very long to converge. Unfortunately, large condition numbers are common in practical applications: the linear-regression matrix in the temperature-prediction example of the SVD notes has condition number around 10^3 (see Figure 16 in those notes). If one cares about finding the least-squares solution fast, the method of choice should instead be conjugate gradients method, an optimization technique designed to achieve fast convergence (see [6] for an excellent tutorial). However, as we explain in the following section for certain tasks we may not want to reach the minimum.

5.3 Early stopping

Let us consider a linear-regression model where the training data $\vec{y}_{\text{train}} \in \mathbb{R}^n$ are given by

$$\vec{y}_{\text{train}} := X_{\text{train}}\vec{\beta}_{\text{true}} + \vec{z}_{\text{train}}. \tag{166}$$

$X_{\text{train}} \in \mathbb{R}^{n \times p}$ contains the corresponding n p-dimensional feature vectors and $\vec{z}_{\text{train}} \in \mathbb{R}^n$ is noise. The following lemma characterizes the coefficient error incurred by gradient descent as the iterations progress.

Lemma 5.6. *Let $X^{n \times p}$, $n \geq p$, be full rank. Let $\vec{\beta}^{(k+1)}$ be the $k+1$th iteration of gradient descent with a constant step size $\alpha > 0$ initialized with zero coefficients applied to the least-squares cost function. If the data are generated according to the generative model in Eq. (166) the coefficient error is of the form*

$$\vec{\beta}_{\text{true}} - \vec{\beta}^{(k+1)} = V \operatorname{diag}_{j=1}^{p} \left(\left(1 - \alpha s_j^2\right)^{k+1} \right) V^T \vec{\beta}_{\text{true}} - V \operatorname{diag}_{j=1}^{p} \left(\frac{1 - \left(1 - \alpha s_j^2\right)^{k+1}}{s_j} \right) U^T \vec{z}_{\text{train}}.$$

Proof. By Theorem 5.3

$$\begin{aligned}
\vec{\beta}^{(k+1)} &= V \operatorname{diag}_{j=1}^{p} \left(\frac{1 - \left(1 - \alpha s_j^2\right)^{k+1}}{s_j} \right) U^T \left(X_{\text{train}} \vec{\beta}_{\text{true}} + \vec{z}_{\text{train}} \right) && (167) \\
&= V \operatorname{diag}_{j=1}^{p} \left(\frac{1 - \left(1 - \alpha s_j^2\right)^{k+1}}{s_j} \right) U^T \left(U S V^T \vec{\beta}_{\text{true}} + \vec{z}_{\text{train}} \right) && (168) \\
&= \vec{\beta}_{\text{true}} - V \operatorname{diag}_{j=1}^{p} \left(\left(1 - \alpha s_j^2\right)^{k+1} \right) V^T \vec{\beta}_{\text{true}} + V \operatorname{diag}_{j=1}^{p} \left(\frac{1 - \left(1 - \alpha s_j^2\right)^{k+1}}{s_j} \right) U^T \vec{z}_{\text{train}}.
\end{aligned}$$

□

There are two terms in the error, the first is related to the value of the true coefficients and decreases monotonically with k. The second depends on the noise term. We can decouple this error by considering the projection of the training noise onto each of the left singular vectors of X $\langle \vec{u}_j, \vec{z}_{\text{train}} \rangle$, $1 \leq j \leq p$. The jth term is scaled by

$$\frac{1 - \left(1 - \alpha s_j^2\right)^{k+1}}{s_j}. \tag{169}$$

The scaling factor is increasing in k and eventually converges to $1/s_j$. This can produce noise amplification for the smallest singular values if the sample covariance matrix is not a good estimate for the test covariance matrix, as explained in the notes on the SVD. This suggests terminating the iterations at a small value of k, a technique commonly known as *early stopping*. Of course, a small value of k increases the first error term, which is decreasing in k. The k that achieves the best tradeoff between the two terms can be chosen by minimizing the validation error on held-out data. When the number of training data is small (which results in a worse estimate of the sample covariance), early stopping can produce dramatic improvement, as illustrated in the following example. The effect is similar to ridge regression, which can also be interpreted as controlling the effect of small singular values in the training matrix (see the notes on the SVD).

Example 5.7 (Temperature prediction via gradient descent)**.** We consider the same dataset of hourly temperatures measured at weather stations all over the United States[3] that we discussed

[3]The data are available at http://www1.ncdc.noaa.gov/pub/data/uscrn/products

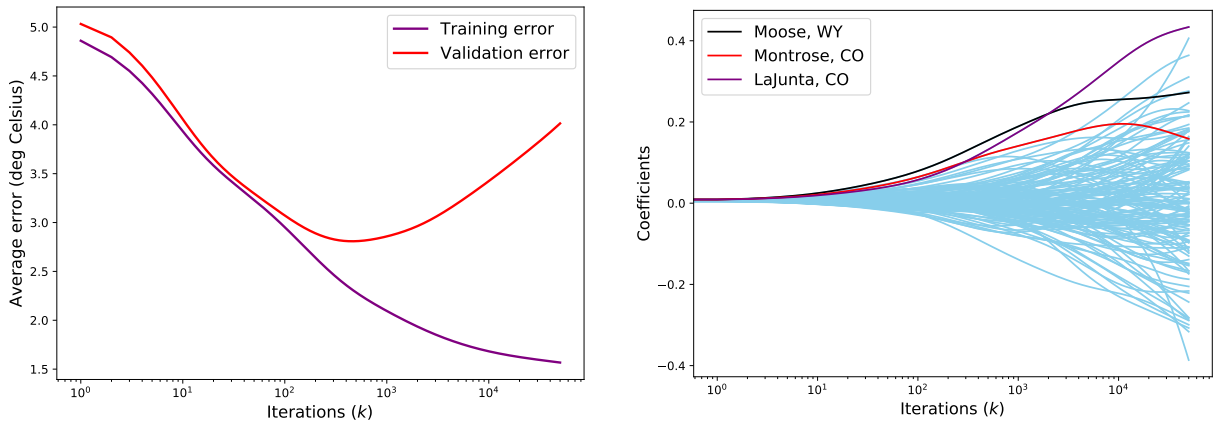

Figure 21: The left graph shows the training and validation errors of the gradient-descent estimator applied to the temperature-prediction task as the iterations progress. The number of training data is fixed to $n = 200$ training data. The right figure shows the values of the corresponding model coefficients. All coefficients are depicted in light blue except the three that have the largest magnitudes for large n, which correspond to the stations of Moose in Wyoming, and Montrose and La Junta in Colorado.

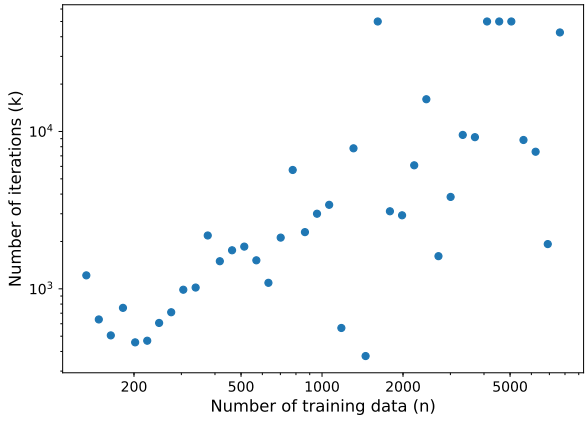

Figure 22: Results of selecting the number of iterations via cross-validation for the experiment described in Example 5.7. The image shows the number of iterations at which the gradient-descent estimator achieves minimum validation error for different numbers of training data. The maximum number of iterations was limited to 10^5.

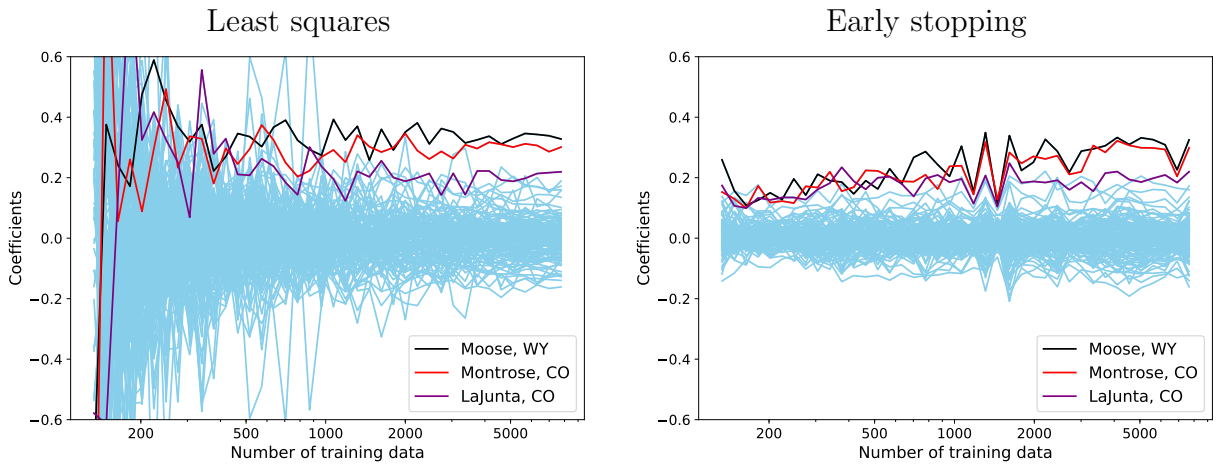

Figure 23: Coefficients of the least-squares (left) and gradient-descent (right) estimators for the experiment described in Example 5.7 for different values of training data. All coefficients are depicted in light blue except the three that have the largest magnitudes for large n, which correspond to the stations of Moose in Wyoming, and Montrose and La Junta in Colorado.

in the lecture notes on the SVD. Our goal is to design a model that can be used to estimate the temperature in Yosemite (California) from the temperatures of 133 other stations. We perform estimation by fitting a linear model where the response is the temperature in Yosemite and the features are the rest of the temperatures ($p = 133$). 10^3 measurements from 2015 are used as a test set. We apply gradient descent to minimize the least-squares cost function using a variable number of training data also from 2015 but disjoint from the test data. The coefficients are initialized to be zero. The number of iterations of gradient descent are chosen by minimizing the error over a separate validation set. In addition, we test the model on data from 2016.

The left image in Figure 21 shows training and validation errors of the gradient-descent estimator for $n = 200$ training data as the iterations progress. Both initially decrease, but at one point the validation error starts increasing due to overfitting. The right image shows that the coefficients amplitudes increase until they reach the value of the least-squares estimator. The minimum validation error is reached when the coefficients are still not too large. Figure 22 shows the number of iterations selected for different numbers of training data based on validation error. Figure 23 shows the corresponding coefficients and compares them the least-squares coefficients. The effect achieved by early stopping is very similar to that of ridge regression. Figure Figure 24 compares the error obtained by the estimator on training and test data compared to least squares and ridge regression. The method avoids the overfitting issues of least squares when the number of training data is small, and achieves very similar results to ridge regression. △

5.4 Convergence of gradient descent

In this section we analyze the convergence of gradient descent for convex functions. Our analysis focuses on a general class of functions that can be locally upper bounded by a quadratic function with fixed curvature. This precludes strange functions with unbounded curvature. We begin by introducing a notion of continuity for functions from $\mathbb{R}^n$ to $\mathbb{R}^m$.

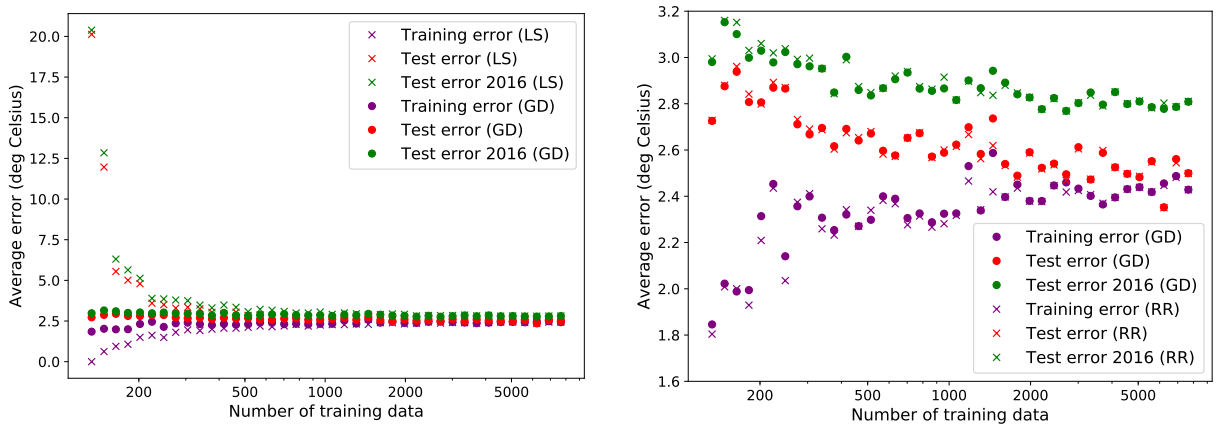

Figure 24: Performance of the gradient-descent estimator for the experiment described in Example 5.7. The left image compares the method to the least-squares estimator on the training and test sets, and on the 2016 data, for different number of training data. The right image shows the same comparison to the ridge-regression estimator.

Definition 5.8 (Lipschitz continuity)**.** *A function $g : \mathbb{R}^n \to \mathbb{R}^m$ is Lipschitz continuous with Lipschitz constant L if for any $\vec{x}, \vec{y} \in \mathbb{R}^n$*

$$||g(\vec{y}) - g(\vec{x})||_2 \leq L ||\vec{y} - \vec{x}||_2 . \tag{170}$$

Functions that have Lipschitz-continuous gradients with constant L are upper bounded locally by a quadratic function with curvature bounded by L. Figure 25 shows an example of a quadratic upper bound.

Theorem 5.9 (Proof in Section 6.6)**.** *If the gradient of a function $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz continuous with Lipschitz constant L,*

$$||\nabla f(\vec{y}) - \nabla f(\vec{x})||_2 \leq L ||\vec{y} - \vec{x}||_2 \tag{171}$$

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

$$f(\vec{y}) \leq f_{\vec{x}}^2(\vec{y}) := f(\vec{x}) + \nabla f(\vec{x})^T (\vec{y} - \vec{x}) + \frac{L}{2} ||\vec{y} - \vec{x}||_2^2 . \tag{172}$$

For example, the Lipschitz constant of the gradient of the least-squares cost function equals the square of the largest singular value s_1 of the feature matrix $X \in \mathbb{R}^{n \times p}$, since for any $\vec{\beta}_1$, $\vec{\beta}_2 \in \mathbb{R}^p$,

$$\left|\left|\nabla f\left(\vec{\beta}_2\right) - \nabla f\left(\vec{\beta}_1\right)\right|\right|_2 = \left|\left|X^T X\left(\vec{\beta}_2 - \vec{\beta}_1\right)\right|\right|_2 \tag{173}$$

$$\leq s_1^2 \left|\left|\vec{\beta}_2 - \vec{\beta}_1\right|\right|_2 \tag{174}$$

The quadratic upper bound in Theorem 5.9 immediately implies a bound on the value of the cost function after k iterations of gradient descent.

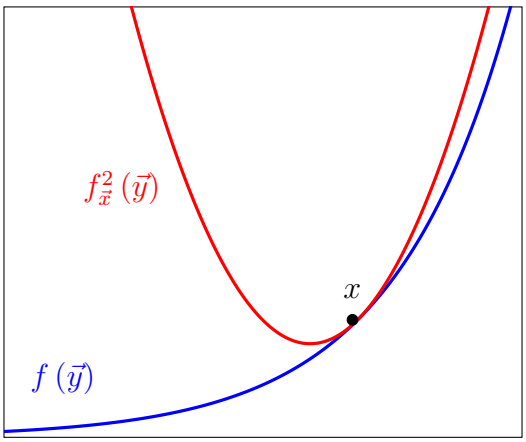

Figure 25: Quadratic upper bound on a convex function.

Corollary 5.10. *Let $\vec{x}^{(k)}$ be the kth iteration of gradient descent and $\alpha_k \geq 0$ the kth step size, if ∇f is L-Lipschitz continuous,*

$$f\left(\vec{x}^{(k+1)}\right) \leq f\left(\vec{x}^{(k)}\right) - \alpha_k \left(1 - \frac{\alpha_k L}{2}\right) \left|\left|\nabla f\left(\vec{x}^{(k)}\right)\right|\right|_2^2. \tag{175}$$

Proof. Applying the quadratic upper bound we obtain

$$f\left(\vec{x}^{(k+1)}\right) \leq f\left(\vec{x}^{(k)}\right) + \nabla f\left(\vec{x}^{(k)}\right)^T \left(\vec{x}^{(k+1)} - \vec{x}^{(k)}\right) + \frac{L}{2} \left|\left|\vec{x}^{(k+1)} - \vec{x}^{(k)}\right|\right|_2^2. \tag{176}$$

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

$$f\left(\vec{x}^{(k+1)}\right) \leq f\left(\vec{x}^{(k)}\right) - \frac{\alpha_k}{2} \left|\left|\nabla f\left(\vec{x}^{(k)}\right)\right|\right|_2^2. \tag{177}$$

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 5.12. *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,*

$$f\left(\vec{x}^{(k)}\right) - f\left(\vec{x}^*\right) \leq \frac{\left|\left|\vec{x}^{(0)} - \vec{x}^*\right|\right|_2^2}{2\,\alpha\,k} \tag{178}$$

Proof. By the first-order characterization of convexity

$$f\left(\vec{x}^{*}\right) \geq f\left(\vec{x}^{(k-1)}\right)+\nabla f\left(\vec{x}^{(k-1)}\right)^{T}\left(\vec{x}^{*}-\vec{x}^{(k-1)}\right), \tag{179}$$

which together with Corollary 5.11 yields

$$\begin{aligned} f\left(\vec{x}^{(k)}\right)-f\left(\vec{x}^{*}\right) &\leq \nabla f\left(\vec{x}^{(k-1)}\right)^{T}\left(\vec{x}^{(k-1)}-\vec{x}^{*}\right)-\frac{\alpha}{2}\left|\left|\nabla f\left(\vec{x}^{(k-1)}\right)\right|\right|_{2}^{2} && (180)\\ &= \frac{1}{2\,\alpha}\left(\left|\left|\vec{x}^{(k-1)}-\vec{x}^{*}\right|\right|_{2}^{2}-\left|\left|\vec{x}^{(k-1)}-\vec{x}^{*}-\alpha\nabla f\left(\vec{x}^{(k-1)}\right)\right|\right|_{2}^{2}\right) && (181)\\ &= \frac{1}{2\,\alpha}\left(\left|\left|\vec{x}^{(k-1)}-\vec{x}^{*}\right|\right|_{2}^{2}-\left|\left|\vec{x}^{(k)}-\vec{x}^{*}\right|\right|_{2}\right) && (182) \end{aligned}$$

Using the fact that by Corollary 5.11 the value of f never increases, we have

$$\begin{aligned} f\left(\vec{x}^{(k)}\right)-f\left(\vec{x}^{*}\right) &\leq \frac{1}{k}\sum_{i=1}^{k} f\left(\vec{x}^{(k)}\right)-f\left(\vec{x}^{*}\right) && (183)\\ &\leq \frac{1}{2\,\alpha\,k}\left(\left|\left|\vec{x}^{(0)}-\vec{x}^{*}\right|\right|_{2}^{2}-\left|\left|\vec{x}^{(k)}-\vec{x}^{*}\right|\right|_{2}^{2}\right) && (184)\\ &\leq \frac{\left|\left|\vec{x}^{(0)}-\vec{x}^{*}\right|\right|_{2}^{2}}{2\,\alpha\,k}. && (185) \end{aligned}$$

□

The theorem assumes that we know the Lipschitz constant of the gradient beforehand. However, the argument can be adapted to incorporate a backtracking line search with the Armijo rule. The result implies 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. This convergence rate can be improved by incorporating a momentum term to gradient descent, yielding Nesterov's accelerated gradient method. We refer to [3, 5] for more information on this technique.

5.5 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,

$$f\left(\vec{x}\right)=\frac{1}{m}\sum_{i=1}^{m} f_{i}\left(\vec{x}\right). \tag{186}$$

An important example is the least-squares cost function where each term $f_i := (y^{(i)} - \langle \vec{x}^{(i)}, \vec{\beta} \rangle)$ corresponds to a feature-vector and response pair $\left\{\vec{x}^{(i)}, \vec{y}^{(i)}\right\}$ 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 sampling a subset of components and using the gradient of their sum individual instead.

Algorithm 5.13 (Stochastic gradient descent)**.** *Set the initial point $\vec{x}^{(0)}$ to a predetermined value. 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*

$$\vec{\mathbf{x}}^{(k+1)} := \vec{\mathbf{x}}^{(k)} - \alpha_k \sum_{i \in \boldsymbol{\mathcal{B}}} \nabla f_i \left(\vec{\mathbf{x}}^{(k)} \right) \tag{187}$$

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 dataset but only have access to portions of it at a time.

Intuitively, for a fixed value of $\vec{x}^{(k)}$, stochastic gradient descent replaces the gradient of the whole function at $\vec{x}^{(k)}$ by

$$\sum_{i=1}^{m} 1_{i \in \boldsymbol{\mathcal{B}}} \nabla f_i \left(\vec{x}^{(k)} \right). \tag{188}$$

If $\boldsymbol{\mathcal{B}}$ is generated by selecting components uniformly at random with replacement, then the approximation is aligned with the true gradient ∇f in expectation,

$$\mathrm{E}\left(\sum_{i=1}^{m} 1_{i \in \boldsymbol{\mathcal{B}}} \nabla f_i \left(\vec{x}^{(k)} \right) \right) = \sum_{i=1}^{m} \mathrm{E}\left(1_{i \in \boldsymbol{\mathcal{B}}} \right) \nabla f_i \left(\vec{x}^{(k)} \right) \tag{189}$$

$$= \sum_{i=1}^{m} \mathrm{P}\left(i \in \boldsymbol{\mathcal{B}} \right) \nabla f_i \left(\vec{x}^{(k)} \right) \tag{190}$$

$$= \frac{m}{b} \nabla f \left(\vec{x}^{(k)} \right). \tag{191}$$

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 [2] for more details. In Figure 26 we show the training and validation errors for stochastic gradient descent using a step size that diminishes proportionally to $\sqrt{k}$ for the data described in Example 5.7. The curves look very similar to the ones for regular gradient descent, only noisier.

5.6 Proximal gradient method

In this section we consider the problem of minimizing nondifferentiable convex functions. Motivated by convex regularizers, we focus on functions that can be expressed as the sum of a differentiable and a nondifferentiable term.

Definition 5.14 (Composite function)**.** *A composite function is a function that can be written as the sum*

$$f\left(\vec{x} \right) + h\left(\vec{x} \right) \tag{192}$$

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

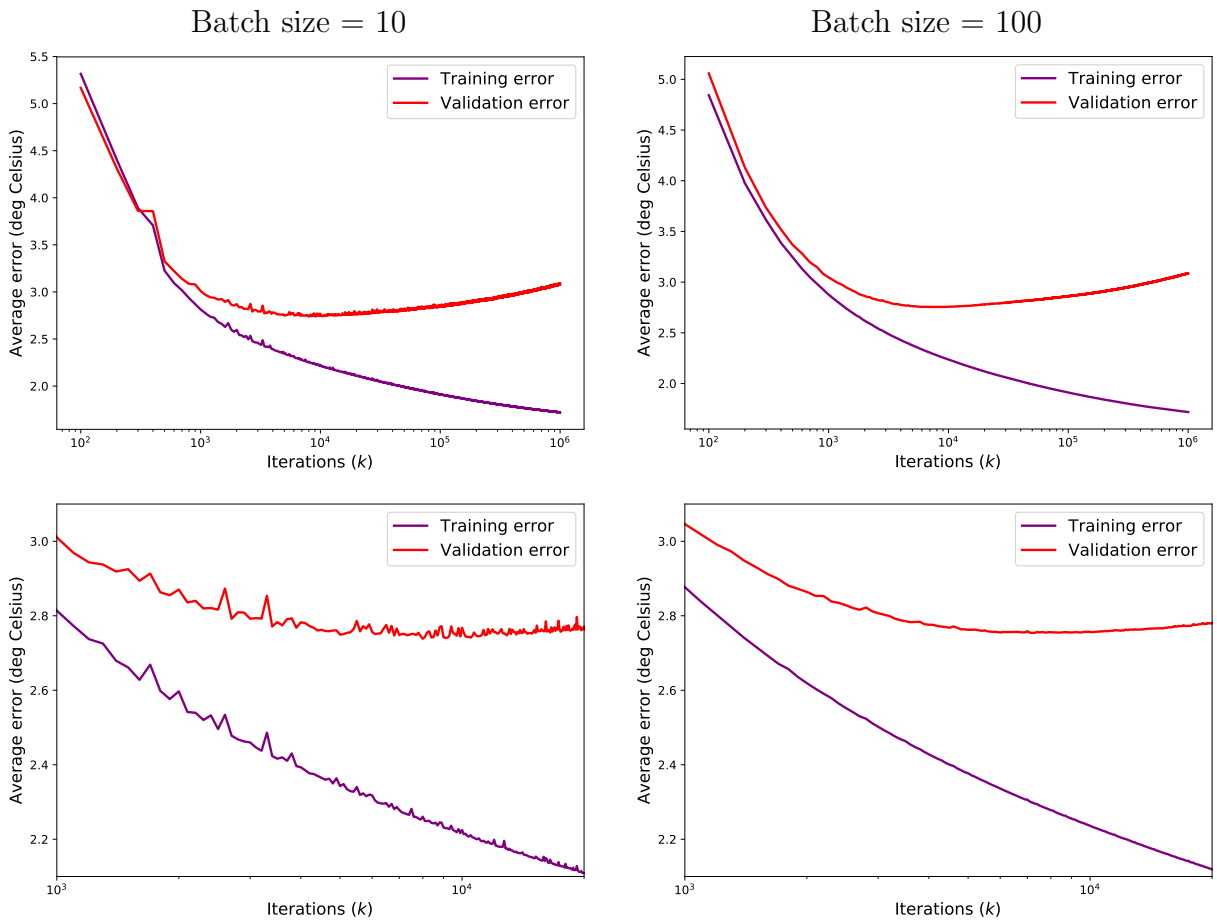

Figure 26: Training and validation errors for stochastic gradient descent using a step size that diminishes proportionally to $\sqrt{k}$ for the data described in Example 5.7. The number of training data is fixed to $n = 200$ training data. The left figure shows the results for a batch size equal to 10, the right column for a batch size equal to 100.

Clearly, the least-squares cost function with ℓ_1-norm or nuclear-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 5.15. *The minimum of the function*

$$h\left(\vec{x}\right) := f\left(\vec{x}^{(k)}\right) + \nabla f\left(\vec{x}^{(k)}\right)^T \left(\vec{x} - \vec{x}^{(k)}\right) + \frac{1}{2\,\alpha} \left|\left|\vec{x} - \vec{x}^{(k)}\right|\right|_2^2 \tag{193}$$

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

Proof.

$$\begin{aligned}
\vec{x}^{(k+1)} &:= \vec{x}^{(k)} - \alpha_k \nabla f\left(\vec{x}^{(k)}\right) && (194)\\
&= \arg\min_{\vec{x}} \left|\left|\vec{x} - \left(\vec{x}^{(k)} - \alpha_k \nabla f\left(\vec{x}^{(k)}\right)\right)\right|\right|_2^2 && (195)\\
&= \arg\min_{\vec{x}} f\left(\vec{x}^{(k)}\right) + \nabla f\left(\vec{x}^{(k)}\right)^T \left(\vec{x} - \vec{x}^{(k)}\right) + \frac{1}{2\,\alpha_k} \left|\left|\vec{x} - \vec{x}^{(k)}\right|\right|_2^2. && (196)
\end{aligned}$$

□

A natural generalization of gradient descent is to minimize the sum of h and the local first-order approximation of f, which can be expressed in terms of the proximal operator of h.

$$\begin{aligned}
\vec{x}^{(k+1)} &= \arg\min_{\vec{x}} f\left(\vec{x}^{(k)}\right) + \nabla f\left(\vec{x}^{(k)}\right)^T \left(\vec{x} - \vec{x}^{(k)}\right) + \frac{1}{2\,\alpha_k} \left|\left|\vec{x} - \vec{x}^{(k)}\right|\right|_2^2 + h\left(\vec{x}\right) && (197)\\
&= \arg\min_{\vec{x}} \frac{1}{2} \left|\left|x - \left(\vec{x}^{(k)} - \alpha_k \nabla f\left(\vec{x}^{(k)}\right)\right)\right|\right|_2^2 + \alpha_k\, h\left(\vec{x}\right) && (198)\\
&= \mathrm{prox}_{\alpha_k\, h}\left(\vec{x}^{(k)} - \alpha_k \nabla f\left(\vec{x}^{(k)}\right)\right). && (199)
\end{aligned}$$

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

Algorithm 5.16 (Proximal-gradient method)**.** *Set the initial point* $\vec{x}^{(0)}$ *to a predetermined value. Update by:*

$$\vec{x}^{(k+1)} := \mathrm{prox}_{\alpha_k\, h}\left(\vec{x}^{(k)} - \alpha_k \nabla f\left(\vec{x}^{(k)}\right)\right), \tag{200}$$

until a convergence criterion is satisfied.

This algorithm may be interpreted as a fixed-point method. Indeed, fixed points of the proximal-gradient iteration are minima of the composite function and vice versa. This suggests applying the iteration repeatedly to minimize the function, although it does not prove convergence (for this we would need to prove that the operator is contractive, see [8]).

Theorem 5.17 (Fixed point of proximal operator). *A vector $\vec{x}^*$ is a solution to*

$$\text{minimize} \quad f\left(\vec{x}\right) + h\left(\vec{x}\right), \tag{201}$$

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

$$\vec{x}^* = \text{prox}_{\alpha\, h}\left(\vec{x}^* - \alpha\, \nabla f\left(\vec{x}^*\right)\right) \tag{202}$$

for any $\alpha > 0$.

Proof. $\vec{x}^*$ is a solution to the optimization problem if and only if there exists a subgradient $\vec{g}$ of h at $\vec{x}^*$ such that $\nabla f\left(\vec{x}^*\right) + \vec{g} = \vec{0}$. $\vec{x}^*$ being a fixed point of the proximal-gradient method is equivalent to $\vec{x}^*$ being the solution to

$$\text{minimize} \quad \alpha\, h\left(\vec{x}\right) + \frac{1}{2}\left|\left|\vec{x}^* - \alpha\, \nabla f\left(\vec{x}^*\right) - x\right|\right|_2^2, \tag{203}$$

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

By Theorem 4.7, the proximal gradient method applied to ℓ_1-norm regularized least squares yields a very intuitive algorithm: alternating between taking gradient descent steps on the least-squares cost function and soft-thresholding.

Algorithm 5.18 (Iterative Shrinkage-Thresholding Algorithm (ISTA)). *Set the initial point $\vec{x}^{\,(0)}$ to a predetermined value. Update by:*

$$\vec{x}^{\,(k+1)} = \mathcal{S}_{\alpha_k \lambda}\left(\vec{x}^{\,(k)} - \alpha_k\, A^T\left(A\vec{x}^{\,(k)} - \vec{y}\right)\right), \tag{204}$$

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. 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 $\mathcal{O}\left(1/\epsilon\right)$ 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 $\mathcal{O}\left(1/\sqrt{\epsilon}\right)$ to an ϵ-optimal solution.

By Theorem 4.11, the proximal gradient method for nuclear-norm regularized least squares alternates between gradient steps on the least-squares cost function and soft-thresholding the singular values. This is the method used to perform matrix completion in Example 4.10.

Algorithm 5.19 (Proximal-gradient method for nuclear-norm regularization). *Let Y be a matrix such that $Y_\Omega = y$ and let us abuse notation by interpreting $X_\Omega^{(k)}$ as a matrix which is zero on Ω^c. We set the initial point $X^{(0)}$ to Y. Then we iterate the update*

$$X^{(k+1)} = \mathcal{D}_{\alpha_k \lambda}\left(X^{(k)} - \alpha_k\left(X_\Omega^{(k)} - Y\right)\right), \tag{205}$$

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

6 Proofs

6.1 Proof of Lemma 2.3

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

$$\begin{aligned} g_{\vec{x},\vec{y}}\left(\theta\alpha+(1-\theta)\beta\right) &= f\left(\left(\theta\alpha+(1-\theta)\beta\right)\vec{x}+\left(1-\theta\alpha-(1-\theta)\beta\right)\vec{y}\right) && (206)\\ &= f\left(\theta\left(\alpha\vec{x}+(1-\alpha)\vec{y}\right)+(1-\theta)\left(\beta\vec{x}+(1-\beta)\vec{y}\right)\right) && (207)\\ &\leq \theta f\left(\alpha\vec{x}+(1-\alpha)\vec{y}\right)+(1-\theta)f\left(\beta\vec{x}+(1-\beta)\vec{y}\right) \quad \text{by convexity of } f\\ &= \theta g_{\vec{x},\vec{y}}(\alpha)+(1-\theta)g_{\vec{x},\vec{y}}(\beta). && (208)\end{aligned}$$

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

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

$$\begin{aligned} f\left(\theta\vec{x}+(1-\theta)\vec{y}\right) &= g_{\vec{x},\vec{y}}(\theta) && (209)\\ &\leq \theta g_{\vec{x},\vec{y}}(1)+(1-\theta)g_{\vec{x},\vec{y}}(0) \quad \text{by convexity of } g_{\vec{x},\vec{y}} && (210)\\ &= \theta f(\vec{x})+(1-\theta)f(\vec{y}). && (211)\end{aligned}$$

6.2 Proof of Theorem 2.12

The proof relies on the following lemma.

Lemma 6.1. *For any $Q\in\mathbb{R}^{n\times n}$*

$$\max_{1\leq i\leq n}|Q_{ii}|\leq||Q||. \tag{212}$$

Proof. Since $||\vec{e}_i||_2=1$,

$$\begin{aligned} \max_{1\leq i\leq n}|Q_{ii}| &\leq \max_{1\leq i\leq n}\sqrt{\sum_{j=1}^{n}Q_{ji}^2} && (213)\\ &= \max_{1\leq i\leq n}||Q\,\vec{e}_i||_2 && (214)\\ &\leq ||Q||. && (215)\end{aligned}$$

□

We denote the SVD of A by USV^T,

$$\sup_{\{||B||\leq 1 \mid B\in\mathbb{R}^{m\times n}\}} \operatorname{tr}\left(A^T B\right) = \sup_{\{||B||\leq 1 \mid B\in\mathbb{R}^{m\times n}\}} \operatorname{tr}\left(V\,S\,U^T B\right) \tag{216}$$

$$= \sup_{\{||B||\leq 1 \mid B\in\mathbb{R}^{m\times n}\}} \operatorname{tr}\left(S\,U^T B V\right) \tag{217}$$

$$\leq \sup_{\{||M||\leq 1 \mid M\in\mathbb{R}^{m\times n}\}} \operatorname{tr}\left(S\,M\right) \qquad \text{because } ||B|| = \left|\left|U^T B V\right|\right|$$

$$\leq \sup_{\left\{\max_{1\leq i\leq n}|M_{ii}|\leq 1 \mid M\in\mathbb{R}^{m\times n}\right\}} \operatorname{tr}\left(S\,M\right) \qquad \text{by Lemma 6.1}$$

$$\leq \sup_{\left\{\max_{1\leq i\leq n}|M_{ii}|\leq 1 \mid M\in\mathbb{R}^{m\times n}\right\}} \sum_{i=1}^{n} M_{ii}\,s_i \tag{218}$$

$$\leq \sum_{i=1}^{n} s_i \tag{219}$$

$$= ||A||_* . \tag{220}$$

To complete the proof, we need to show that the equality holds. Note that UV^T has operator norm equal to one because its r singular values (recall that r is the rank of A) are equal to one. We have

$$\left\langle A, UV^T\right\rangle = \operatorname{tr}\left(A^T U V^T\right) \tag{221}$$

$$= \operatorname{tr}\left(V\,S\,U^T U V^T\right) \tag{222}$$

$$= \operatorname{tr}\left(V^T V\,S\right) \tag{223}$$

$$= \operatorname{tr}\left(S\right) \tag{224}$$

$$= ||A||_* . \tag{225}$$

6.3 Proof of Theorem 3.7

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

The following lemma, proved in Section 6.4 below establishes that the result holds for univariate functions

Lemma 6.2. *A univariate differentiable function $g : \mathbb{R} \to \mathbb{R}$ is convex if and only if for all $x, y \in \mathbb{R}$*

$$g\left(y\right) \geq g'\left(x\right)\left(y - x\right) + g\left(x\right) \tag{226}$$

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

$$g\left(y\right) > g'\left(x\right)\left(y - x\right) + g\left(x\right). \tag{227}$$

To complete the proof we extend the result to the multivariable case using Lemma 2.3.

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 2.3 we just need to show that the univariate function $g_{\vec{a},\vec{b}}$ defined by (7) is convex for all $\vec{a}, \vec{b} \in \mathbb{R}^n$. Applying some basic multivariate calculus yields

$$g'_{\vec{a},\vec{b}}(\alpha) = \nabla f\left(\alpha\,\vec{a} + (1-\alpha)\,\vec{b}\right)^T \left(\vec{a} - \vec{b}\right). \tag{228}$$

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

$$\begin{aligned}
g_{\vec{a},\vec{b}}(\beta) &= f(\vec{y}) && (229)\\
&\geq f(\vec{x}) + \nabla f(\vec{x})^T (y - \vec{x}) && (230)\\
&= f\left(\alpha\,\vec{a} + (1-\alpha)\,\vec{b}\right) + \nabla f\left(\alpha\,\vec{a} + (1-\alpha)\,\vec{b}\right)^T \left(\vec{a} - \vec{b}\right)(\beta - \alpha) && (231)\\
&= g_{\vec{a},\vec{b}}(\alpha) + g'_{\vec{a},\vec{b}}(\alpha)\,(\beta - \alpha) \quad \text{by (228)}, && (232)
\end{aligned}$$

which establishes that $g_{\vec{a},\vec{b}}$ is convex by Lemma 6.2 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 2.3, $g_{\vec{x},\vec{y}}$ is convex for any $\vec{x}, y \in \mathbb{R}^n$.

$$\begin{aligned}
f(\vec{y}) &= g_{\vec{x},\vec{y}}(1) && (233)\\
&\geq g_{\vec{x},\vec{y}}(0) + g'_{\vec{x},\vec{y}}(0) \quad \text{by convexity of } g_{\vec{x},\vec{y}} \text{ and Lemma 6.2} && (234)\\
&= f(\vec{x}) + \nabla f(\vec{x})^T (y - \vec{x}) \quad \text{by (228)}. && (235)
\end{aligned}$$

6.4 Proof of Lemma 6.2

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 $0 \leq \theta \leq 1$

$$\theta\,(g(y) - g(x)) + g(x) \geq g(x + \theta\,(y - x)). \tag{236}$$

Rearranging the terms we have

$$g(y) \geq \frac{g(x + \theta\,(y - x)) - g(x)}{\theta} + g(x). \tag{237}$$

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

$$g(y) \geq \frac{g(x + h) - g(x)}{h}(y - x) + g(x). \tag{238}$$

Taking the limit when $h \to 0$ yields

$$g(y) \geq g'(x)(y - x) + g(x). \tag{239}$$

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 if $g(y) \geq g'(x)(y-x) + g(x)$

$$\begin{aligned} g(x) &\geq g'(z)(x-z) + g(z) && (240)\\ &= g'(z)(1-\theta)(x-y) + g(z) && (241)\\ g(y) &\geq g'(z)(y-z) + g(z) && (242)\\ &= g'(z)\theta(y-x) + g(z) && (243) \end{aligned}$$

Multiplying (241) by θ, then (243) by $1-\theta$ and summing the inequalities, we obtain

$$\theta g(x) + (1-\theta) g(y) \geq g(\theta x + (1-\theta) y). \tag{244}$$

6.5 Proof of Lemma 3.19

If $\vec{g}$ is a subgradient of $||\cdot||_1$ at $\vec{x}$ then for any $y \in \mathbb{R}$

$$\begin{aligned} |y| &= ||y\vec{e}_i||_1 && (245)\\ &= |\vec{x}[i]| + ||\vec{x} + (y - \vec{x}[i])\vec{e}_i||_1 - ||\vec{x}||_1 && (246)\\ &\geq |\vec{x}[i]| + ||\vec{x}||_1 + \vec{g}^T(y - \vec{x}[i])\vec{e}_i - ||\vec{x}||_1 && (247)\\ &= |\vec{x}[i]| + \vec{g}[i](y - \vec{x}[i]), && (248) \end{aligned}$$

so $\vec{g}[i]$ is a subgradient of $|\cdot|$ at $|\vec{x}[i]|$ for any $1 \leq i \leq n$.

If $\vec{g}[i]$ is a subgradient of $|\cdot|$ at $|\vec{x}[i]|$ for $1 \leq i \leq n$ then for any $\vec{y} \in \mathbb{R}^n$

$$\begin{aligned} ||\vec{y}||_1 &= \sum_{i=1}^{n} |\vec{y}[i]| && (249)\\ &\geq \sum_{i=1}^{n} |\vec{x}[i]| + \vec{g}[i](\vec{y}[i] - \vec{x}[i]) && (250)\\ &= ||\vec{x}||_1 + \vec{g}^T(\vec{y} - \vec{x}) && (251) \end{aligned}$$

so $\vec{g}$ is a subgradient of $||\cdot||_1$ at $\vec{x}$.

6.6 Proof of Proposition 5.9

Consider the function

$$g(\vec{x}) := \frac{L}{2}\vec{x}^T\vec{x} - f(\vec{x}). \tag{252}$$

We first establish that g is convex using the following lemma, proved in Section 6.7 below.

Lemma 6.3 (Monotonicity of gradient)**.** *A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if*

$$(\nabla f(\vec{y}) - \nabla f(\vec{x}))^T(\vec{y} - \vec{x}) \geq 0. \tag{253}$$

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

$$\left(\nabla f\left(\vec{y}\right)-\nabla f\left(\vec{x}\right)\right)^{T}\left(\vec{y}-\vec{x}\right)\leq L\left|\left|\vec{y}-\vec{x}\right|\right|_{2}^{2}, \tag{254}$$

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

$$\begin{aligned}\left(\nabla g\left(\vec{y}\right)-\nabla g\left(\vec{x}\right)\right)^{T}\left(\vec{y}-\vec{x}\right) &= \left(L\vec{y}-L\vec{x}+\nabla f\left(\vec{x}\right)-\nabla f\left(\vec{y}\right)\right)^{T}\left(\vec{y}-\vec{x}\right) &(255)\\ &= L\left|\left|\vec{y}-\vec{x}\right|\right|_{2}^{2}-\left(\nabla f\left(\vec{y}\right)-\nabla f\left(\vec{x}\right)\right)^{T}\left(\vec{y}-\vec{x}\right) &(256)\\ &\geq 0 &(257)\end{aligned}$$

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

$$\begin{aligned}\frac{L}{2}\vec{y}^{T}\vec{y}-f\left(\vec{y}\right) &= g\left(\vec{y}\right) &(258)\\ &\geq g\left(\vec{x}\right)+\nabla g\left(\vec{x}\right)^{T}\left(\vec{y}-\vec{x}\right) &(259)\\ &= \frac{L}{2}\vec{x}^{T}\vec{x}-f\left(\vec{x}\right)+\left(L\vec{x}-\nabla f\left(\vec{x}\right)\right)^{T}\left(\vec{y}-\vec{x}\right). &(260)\end{aligned}$$

Rearranging the inequality we conclude that

$$f\left(\vec{y}\right)\leq f\left(\vec{x}\right)+\nabla f\left(\vec{x}\right)^{T}\left(\vec{y}-\vec{x}\right)+\frac{L}{2}\left|\left|\vec{y}-\vec{x}\right|\right|_{2}^{2}. \tag{261}$$

6.7 Proof of Lemma 6.3

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

$$\begin{aligned}f\left(\vec{y}\right) &\geq f\left(\vec{x}\right)+\nabla f\left(\vec{x}\right)^{T}\left(\vec{y}-\vec{x}\right), &(262)\\ f\left(\vec{x}\right) &\geq f\left(\vec{y}\right)+\nabla f\left(\vec{y}\right)^{T}\left(\vec{x}-\vec{y}\right), &(263)\\ & &(264)\end{aligned}$$

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 $g_{a,b}:[0,1]\rightarrow\mathbb{R}$ defined by

$$g_{a,b}\left(\alpha\right):=f\left(\alpha a+\left(1-\alpha\right)b\right), \tag{265}$$

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 $\alpha\in\left(0,1\right)$ we have

$$\begin{aligned}g'_{a,b}\left(\alpha\right)-g'_{a,b}\left(0\right) &= \left(\nabla f\left(\alpha\, a+\left(1-\alpha\right)b\right)-\nabla f\left(b\right)\right)^{T}\left(a-b\right) &(266)\\ &= \frac{1}{\alpha}\left(\nabla f\left(\alpha\, a+\left(1-\alpha\right)b\right)-\nabla f\left(b\right)\right)^{T}\left(\alpha\, a+\left(1-\alpha\right)b-b\right) &(267)\\ &\geq 0 \quad \text{because } \left(\nabla f\left(y\right)-\nabla f\left(x\right)\right)^{T}\left(y-x\right)\geq 0 \text{ for any } x,y. &(268)\end{aligned}$$

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

$$f(\vec{x}) = g_{\vec{x},\vec{y}}(1) \tag{269}$$

$$= g_{\vec{x},\vec{y}}(0) + \int_0^1 g'_{\vec{x},\vec{y}}(\alpha)\ \mathrm{d}\alpha \tag{270}$$

$$\geq g_{\vec{x},\vec{y}}(0) + g'_{\vec{x},\vec{y}}(0) \tag{271}$$

$$= f(\vec{y}) + \nabla f(\vec{y})(\vec{x} - \vec{y}). \tag{272}$$

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

[3] G. Goh. Why momentum really works. *Distill*, 2(4):e6, 2017.

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

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

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

[7] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

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

[9] G. A. Watson. Characterization of the subdifferential of some matrix norms. *Linear algebra and its applications*, 170:33–45, 1992.