Derivation of Principal Components Analysis

Given a set of data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$, we want to find the linear subspace (plus an affine offset) that is the best fit in the least-squares sense. Mathematically, we want to solve

$$\underset{\boldsymbol{\mu}, \boldsymbol{A}, \{\boldsymbol{z}_i\}}{\text{minimize}} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{\mu} - \boldsymbol{A}\boldsymbol{z}_i\|_2^2, \quad \text{subject to} \quad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{I},$$

where $\boldsymbol{\mu} \in \mathbb{R}^d, \boldsymbol{z}_i \in \mathbb{R}^k$, and $\boldsymbol{A}$ is a $d \times k$ matrix; the constraint $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{I}$ means that we will consider $\boldsymbol{A}$ with orthonormal columns.

Minimizing the expression above over the $\{\boldsymbol{z}_i\}$ and $\boldsymbol{\mu}$ is straightforward. For the $\{\boldsymbol{z}_i\}$, suppose that $\boldsymbol{A}$ and $\boldsymbol{\mu}$ are fixed. Then we have a series of n decoupled least-squares problems: for $i = 1, \ldots, n$, we solve

$$\underset{\boldsymbol{z}_i}{\text{minimize}} \; \|\boldsymbol{x}_i - \boldsymbol{\mu} - \boldsymbol{A}\boldsymbol{z}_i\|_2^2$$

This is a standard unconstrained least-squares problem that has solution

$$\widehat{\boldsymbol{z}}_i = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{\mu}) = \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{\mu}),$$

where the second equality follows from $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{I}$. With $\boldsymbol{A}$ still fixed, we solve for $\boldsymbol{\mu}$ by plugging in our expression for the $\boldsymbol{z}_i$:

$$\begin{aligned}
\underset{\boldsymbol{\mu}}{\text{minimize}} \; & \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{\mu} - \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{\mu})\|_2^2, \\
= & \sum_{i=1}^{n} \|(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})(\boldsymbol{x}_i - \boldsymbol{\mu})\|_2^2, \\
= & \sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu})^{\mathrm{T}}(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})(\boldsymbol{x}_i - \boldsymbol{\mu}),
\end{aligned}$$

where the last step comes from expanding out the norm squared as an inner product, and using the fact that $(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})$ is a projector; it is symmetric, and $(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})^2 = (\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})$. Taking the gradient of the expression above and setting it to zero means that $\widehat{\boldsymbol{\mu}}$ will obey

$$\begin{aligned}\boldsymbol{0} &= -2\sum_{i=1}^{n}(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}) \\ &= -2(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})\left(\sum_{i=1}^{n}\boldsymbol{x}_i - n\widehat{\boldsymbol{\mu}}\right).\end{aligned}$$

This can be satisfied by taking

$$\widehat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i.$$

Note that this is not the only choice for $\boldsymbol{\mu}$ — any choice that puts $\sum_i \boldsymbol{x}_i - n\boldsymbol{\mu}$ into the column space of $\boldsymbol{A}$ will work. But the choice above is intuitive, so we will go with it.

With $\{\widehat{\boldsymbol{z}}_i\}$ and $\widehat{\boldsymbol{\mu}}$ solved for, we now optimize over $\boldsymbol{A}$. We want to solve

$$\underset{\boldsymbol{A}\in\mathbb{R}^{n\times k}}{\text{minimize}} \ \sum_{i=1}^{n}\|\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}} - \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})\|_2^2 \quad \text{subject to} \ \ \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{I}.$$

We will assume, without loss of generality, that $\widehat{\boldsymbol{\mu}} = \boldsymbol{0}$, as we could simply use the variable substitution $\widetilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}$ above. The program becomes

$$\underset{\boldsymbol{A}\in\mathbb{R}^{n\times k}}{\text{minimize}} \ \sum_{i=1}^{n}\|(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})\boldsymbol{x}_i\|_2^2 \quad \text{subject to} \ \ \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{I}.$$

Expanding the functional, and again using the fact that $(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})$ is a projector,

$$\begin{aligned}\sum_{i=1}^{n} \|(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})\boldsymbol{x}_i\|_2^2 &= \sum_{i=1}^{n} \boldsymbol{x}_i^{\mathrm{T}}(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})\boldsymbol{x}_i \\ &= \sum_{i=1}^{n} \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_i - \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x}_i.\end{aligned}$$

The first term does not depend on $\boldsymbol{A}$, and the second term is always negative, so our problem is equivalent to

$$\underset{\boldsymbol{A}\in\mathbb{R}^{n\times k}}{\text{maximize}} \ \sum_{i=1}^{n} \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x}_i \quad \text{subject to} \quad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{I}.$$

For any vector $\boldsymbol{v}$, it is easy to see that $\|\boldsymbol{v}\|_2^2 = \operatorname{trace}(\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}})$. Thus, the objective function above can also be written as

$$\begin{aligned}\sum_{i=1}^{n} \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x}_i &= \sum_{i=1}^{n} \|\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x}_i\|_2^2 \\ &= \sum_{i=1}^{n} \operatorname{trace}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{A}) \\ &= \operatorname{trace}\left(\boldsymbol{A}^{\mathrm{T}}\left(\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}\right)\boldsymbol{A}\right) \\ &= \operatorname{trace}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{A}),\end{aligned}$$

where $\boldsymbol{S} = \sum_{i=1}^{n} \boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}$ is a scaled version of the sample covariance matrix.

By construction, $\boldsymbol{S}$ is symmetric positive semi-definite, so it has eigenvalue decomposition

$$\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{T}},$$

where $\boldsymbol{U}$ is a $d \times d$ orthonormal matrix, $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^{\mathrm{T}} = \boldsymbol{I}$, and $\boldsymbol{\Lambda} = \operatorname{diag}(\{\lambda_i\})$, with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$. Then

$$\operatorname{trace}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{A}) = \operatorname{trace}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{A}) = \operatorname{trace}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{W}),$$

where $\boldsymbol{W} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{A}$. Notice that $\boldsymbol{W}$ also has orthonormal columns, as $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{I}$. So we can solve the program

$$\underset{\boldsymbol{W} \in \mathbb{R}^{n \times k}}{\text{maximize}} \ \operatorname{trace}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{W}) \quad \text{subject to} \ \ \boldsymbol{W}^{\mathrm{T}}\boldsymbol{W} = \boldsymbol{I},$$

and then take $\widehat{\boldsymbol{A}} = \boldsymbol{U}\widehat{\boldsymbol{W}}$.

We can show that the last maximization program above is equivalent to a simple linear program that we can solve by inspection. Let $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k$ be the columns of $\boldsymbol{W}$. Then

$$\begin{aligned}
\operatorname{trace}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{W}) &= \sum_{i=1}^{k} \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{w}_i \\
&= \sum_{i=1}^{k}\sum_{j=1}^{d} w_i(j)^2\lambda_j \\
&= \sum_{j=1}^{d} h_j\lambda_j, \quad h_j = \sum_{i=1}^{k} w_i(j)^2 = \sum_{i=1}^{k} W(i,j)^2.
\end{aligned}$$

The $h_j,\ j = 1, \ldots, d$ above are the sums of the squares of the *rows* of $\boldsymbol{W}$. It is clear that $h_j \geq 0$. It is also true that

$$\sum_{j=1}^{d} h_j = k,$$

as the fact that the norm of each columns of $\boldsymbol{W}$ is 1 means that

$$\sum_{j=1}^{d}\sum_{i=1}^{k} W(i,j)^2 = \sum_{i=1}^{k}\left(\sum_{j=1}^{d} W(i,j)^2\right) = \sum_{i=1}^{k} 1 = k.$$

Finally, it is also true that $h_j \leq 1$. Here is why: since the columns of $\boldsymbol{W}$ are orthonormal, they can be considered as part of an orthonormal basis for all of $\mathbb{R}^d$. That is, there is a (and actually there are many) $d \times (d-k)$ matrix $\boldsymbol{W}_0$ such that the columns of

$$\boldsymbol{W}' = \begin{bmatrix} \boldsymbol{W} & \boldsymbol{W}_0 \end{bmatrix}$$

form an orthonormal basis for $\mathbb{R}^d$. Since $\boldsymbol{W}'$ is square, $\boldsymbol{W}'\boldsymbol{W}'^{\mathrm{T}} = \boldsymbol{I}$, meaning the sum of the squares of each row are equal to 1. Thus

$$h_j = \sum_{i=1}^{k} W(i,j)^2 \leq \sum_{i=1}^{d} W'(i,j)^2 = 1.$$

With these constraints on the h_j, let's see how large we can make the quantity of interest:

$$\underset{\boldsymbol{h} \in \mathbb{R}^d}{\text{maximize}} \ \sum_{j=1}^{d} h_j \lambda_j \quad \text{subject to} \quad \sum_{j=1}^{d} h_j = k, \ \ 0 \leq h_j \leq 1.$$

This is a linear program, but we can intuit the answer. Since all of the λ_j are positive, we want to have their weights (i.e., the h_j) as large as possible for the largest entries. Since the weights are constrained to be less than 1, and their sum is k, this simply means we assign a weight of 1 to the k largest terms, and 0 to the others:

$$\widehat{h}_j = \begin{cases} 1, & j = 1, \ldots, k, \\ 0, & \text{otherwise.} \end{cases}$$

This means that the sum of the squares of the entries in the rows of the corresponding $\widehat{\boldsymbol{W}}$ are 1 for the first k, and zero below — there are many matrices with orthonormal columns which fit the bill, but a specific one which does is

$$\widehat{\boldsymbol{W}} = \begin{bmatrix} \boldsymbol{I}_{k\times k} \\ \mathbf{0}_{(d-k)\times k} \end{bmatrix}. \tag{1}$$

Taking $\widehat{\boldsymbol{A}} = \boldsymbol{U}\widehat{\boldsymbol{W}}$, this results in

$$\widehat{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_k \end{bmatrix},$$

where the $\boldsymbol{u}_i$ above are the first k columns of $\boldsymbol{U}$.

PCA Theorem

$$\underset{\boldsymbol{\mu},\boldsymbol{A},\{\boldsymbol{z}_i\}}{\text{minimize}} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{\mu} - \boldsymbol{A}\boldsymbol{z}_i\|_2^2, \quad \text{subject to } \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{I},$$

has solution

$$\widehat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i, \quad \widehat{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_k \end{bmatrix}, \quad \widehat{\boldsymbol{z}}_i = \widehat{\boldsymbol{A}}^{\mathrm{T}}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}),$$

where $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ are the eigenvectors corresponding to the k largest eigenvalues of

$$\boldsymbol{S} = \sum_{i=1}^{n} \boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}.$$

Note that our analysis above shows that the choice of $\boldsymbol{A}$ is not unique — we are really choosing the subspace spanned by the columns of $\boldsymbol{A}$, and do not care which orthobasis we use to span it. In the end, taking $\widehat{\boldsymbol{A}}' = \widehat{\boldsymbol{A}}\boldsymbol{Q}$, for any $k \times k$ orthonormal matrix $\boldsymbol{Q}$ would also work, as

$$\widehat{\boldsymbol{A}}'\widehat{\boldsymbol{A}}'^{\mathrm{T}} = \widehat{\boldsymbol{A}}\boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}}\widehat{\boldsymbol{A}}^{\mathrm{T}} = \widehat{\boldsymbol{A}}\widehat{\boldsymbol{A}}^{\mathrm{T}}.$$

In our choice for $\widehat{\boldsymbol{W}}$ in (1) above, we would take

$$\widehat{\boldsymbol{W}} = \begin{bmatrix} \boldsymbol{Q} \\ \mathbf{0}_{(d-k)\times k} \end{bmatrix},$$

which also meets the constraints dictated by the $\widehat{h}_j$ — the sum of the squares of the entries in the rows if 1 for the first k, zero for the last $d-k$, and the columns are orthonormal.

The LASSO for feature selection

Dimensionality reduction (using something like PCA) is one common approach to mitigate the risk of overfitting. In the context of regression, another popular approach to mitigating this risk is called the **LASSO**. In this setting, we will use the squared-error loss along with $\mathcal{F}$ as the set of linear (plus offset) functions on $\mathbb{R}^d$:

$$L(f(\boldsymbol{x}_i), y_i) = (y_i - \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}_i - \beta_0)^2 = (y_i - \boldsymbol{\theta}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i)^2,$$

where

$$\boldsymbol{\theta} = \begin{bmatrix} \beta_0 \\ \boldsymbol{\beta} \end{bmatrix}, \quad \tilde{\boldsymbol{x}}_i = \begin{bmatrix} 1 \\ \boldsymbol{x}_i \end{bmatrix}.$$

Stacking the $\tilde{\boldsymbol{x}}_i^{\mathrm{T}}$ up as rows of a $n \times (d+1)$ matrix $\boldsymbol{A}$ and collecting the y_i into a single vector,

$$\boldsymbol{A} = \begin{bmatrix} 1 & \boldsymbol{x}_1^{\mathrm{T}} \\ 1 & \boldsymbol{x}_2^{\mathrm{T}} \\ \vdots & \vdots \\ 1 & \boldsymbol{x}_n^{\mathrm{T}} \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

we can write

$$\sum_{i=1}^{n} L(f(\boldsymbol{x}_i), y_i) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2.$$

For the penalty, we use the sum of the absolute values of the entries in $\boldsymbol{\theta}$:

$$r(f) = \sum_{i=1}^{d+1} |\theta(i)| = \|\boldsymbol{\theta}\|_1.$$

Our optimization program is now[1]

$$\text{(LASSO)} \qquad \underset{\boldsymbol{\theta}}{\text{minimize}}\ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_1.$$

This regression technique is commonly referred to as the **LASSO**.

The motivation for using the ℓ_1 norm as a regularizer is that doing so tends to produce $\boldsymbol{\theta}$ which have a small number of non-zero terms. This is certainly true in practice, and we can do a little bit of analysis that explains why.

Specifically, we can show that there is a solution to (LASSO) above that has at most n non-zero entries using a relatively simple argument. Let $\boldsymbol{\theta}$ be any vector with more than n non-zero terms in it:

$$\text{nnz}(\boldsymbol{\theta}) := \#\{i\ :\ \theta(i) \neq 0\} \geq n + 1.$$

Then at least one of the following is true:

1. There is another $\boldsymbol{\theta}' \in \mathbb{R}^{d+1}$ such that $\text{nnz}(\boldsymbol{\theta}') \leq \text{nnz}(\boldsymbol{\theta})$ and

   $$\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}'\|_2^2 + \lambda\|\boldsymbol{\theta}'\|_1 \ < \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_1,$$

   or

2. there is another $\boldsymbol{\theta}' \in \mathbb{R}^{d+1}$ such that $\text{nnz}(\boldsymbol{\theta}') < \text{nnz}(\boldsymbol{\theta})$ and

   $$\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}'\|_2^2 + \lambda\|\boldsymbol{\theta}'\|_1 \ = \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_1.$$

Given $\boldsymbol{\theta}$ as above, let Γ be the locations of the non-zero terms in $\boldsymbol{\theta}$:

$$\Gamma = \{i\ :\ \theta(i) \neq 0\}.$$

[1] We have introduced the factor of 1/2 in front of the loss simply for convenience and consistency with the literature.

We are supposing that $|\Gamma| \geq n+1$. Since $|\Gamma|$ is greater than the number of rows in $\boldsymbol{A}$, there is at least one vector $\boldsymbol{z}$ that is also supported on Γ,

$$z(i) = 0, \quad \text{for} \;\; i \notin \Gamma,$$

such that $\alpha \boldsymbol{z} \in \text{Null}(\boldsymbol{A})$ for all $\alpha \in \mathbb{R}$. We will show that by adding a little bit of $\boldsymbol{z}$ to $\boldsymbol{\theta}$, we can hold the ℓ_2 loss term constant while either decreasing the ℓ_1 regularization term or driving one of the non-zero terms to zero. Notice that

$$\|\boldsymbol{y} - \boldsymbol{A}(\boldsymbol{\theta} + \epsilon \boldsymbol{z})\|_2^2 = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2$$

for all $\epsilon > 0$, since $\boldsymbol{z} \in \text{Null}(\boldsymbol{A})$.

Now let $\boldsymbol{s}_\theta$ be the vector supported on Γ that contains the signs of the non-zero entries of $\boldsymbol{\theta}$:

$$s_\theta(i) = \begin{cases} \text{sign}(\theta(i)), & i \in \Gamma \\ 0, & i \notin \Gamma \end{cases}.$$

We consider two cases: $\boldsymbol{s}_\theta^{\mathrm{T}} \boldsymbol{z} \neq 0$ and $\boldsymbol{s}_\theta^{\mathrm{T}} \boldsymbol{z} = 0$.

First, suppose that $\boldsymbol{s}_\theta^{\mathrm{T}} \boldsymbol{z} \neq 0$. Without loss of generality, we can assume that $\boldsymbol{s}_\theta^{\mathrm{T}} \boldsymbol{z} < 0$, as otherwise we can just replace $\boldsymbol{z}$ with $-\boldsymbol{z}$ (since both of these are supported on Γ and are in the nullspace of $\boldsymbol{A}$). Notice that we can write

$$\|\boldsymbol{\theta}\|_1 = \sum_{i=1}^{d+1} |\theta(i)| = \sum_{i=1}^{d+1} \text{sign}(\theta(i))\theta(i)$$

For $\epsilon > 0$ small enough, it is a fact that

$$\text{sign}(\theta(i) + \epsilon z(i)) = \text{sign}(\theta(i)), \quad \text{for all} \quad i \in \Gamma,$$

and so

$$\begin{aligned}\|\boldsymbol{\theta} + \epsilon \boldsymbol{z}\|_1 &= \sum_{i=1}^{d+1} \operatorname{sign}(\theta(i) + \epsilon z(i))(\theta(i) + \epsilon z(i)) \\ &= \sum_{i=1}^{d+1} \operatorname{sign}(\theta(i))(\theta(i) + \epsilon z(i)) \\ &= \|\boldsymbol{\theta}\|_1 + \epsilon \boldsymbol{s}_\theta^{\mathrm{T}} \boldsymbol{z} \\ &< \|\boldsymbol{\theta}\|_1.\end{aligned}$$

Thus, taking $\boldsymbol{\theta}' = \boldsymbol{\theta} + \epsilon \boldsymbol{z}$ for a small enough value of $\epsilon > 0$ gives a vector with a smaller functional value, so $\boldsymbol{\theta}$ cannot be a solution to (LASSO).

Now suppose that $\boldsymbol{s}_\theta^{\mathrm{T}} \boldsymbol{z} = 0$. Then

$$\|\boldsymbol{\theta}\|_1 = \|\boldsymbol{\theta} + \epsilon \boldsymbol{z}\|_1$$

as long as $\boldsymbol{\theta} + \epsilon \boldsymbol{z}$ remains non-zero on Γ. But there must be at least one $i \in \Gamma$ such that

$$\operatorname{sign}(\theta(i)) \neq \operatorname{sign}(z(i)),$$

otherwise the inner product could not be equal to zero. Let i' be the index that obeys the condition above such that $\theta(i)$ is smallest relative to $z(i)$:

$$i' = \arg\min_{i \in \Gamma} \left\{ \frac{|\theta(i)|}{|z(i)|} \ : \ \operatorname{sign}(\theta(i)) \neq \operatorname{sign}(z(i)) \right\}.$$

Then

$$\boldsymbol{\theta}' = \boldsymbol{\theta} + \frac{|\theta(i')|}{|z(i')|} \boldsymbol{z},$$

will be exactly equal to zero at i' and will still be non-zero outside of Γ. Thus

$$\operatorname{nnz}(\boldsymbol{\theta}') < \operatorname{nnz}(\boldsymbol{\theta}),$$

while

$$\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}'\|_2^2 + \lambda\|\boldsymbol{\theta}'\|_1 \; = \; \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_1.$$

The LASSO as a quadratic program

Unlike standard least-squares, or least-squares with Tikhonov regularization, the solution to the LASSO does not have a closed form. We can, however, write it as a convex quadratic program with linear inequality constraints. This puts it in the same class of optimization program as SVMs.

The main idea is to introduce slack variables that allow us to re-write the ℓ_1 norm, which is piecewise linear, as a linear function subject to linear constraints. In particular, the solution to the LASSO is exactly the same as the solution to

$$\begin{aligned} \underset{\boldsymbol{\theta},\boldsymbol{u}}{\text{minimize}} \; & \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 + \lambda\sum_{i=1}^{d+1} u(i) \\ & \text{subject to} \quad -u(i) \le \theta(i) \le u(i), \; i = 1, \ldots, d+1. \end{aligned}$$

This is the same as solving

$$\begin{aligned} \underset{\boldsymbol{\theta},\boldsymbol{u}}{\text{minimize}} \; & \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 + \lambda\sum_{i=1}^{d+1} u(i) \\ \text{subject to} \quad & \boldsymbol{\theta} - \boldsymbol{u} \le \boldsymbol{0} \\ & -\boldsymbol{\theta} - \boldsymbol{u} \le \boldsymbol{0}, \end{aligned}$$

which is the same as solving

$$\operatorname*{minimize}_{\boldsymbol{z}\in\mathbb{R}^{2d+2}} \ \frac{1}{2}\boldsymbol{z}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{z} + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{z} \quad \text{subject to} \quad \boldsymbol{R}\boldsymbol{z} \leq \boldsymbol{0},$$

where

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{u} \end{bmatrix}, \quad \boldsymbol{P} = \begin{bmatrix} \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} -\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} \\ \lambda\boldsymbol{1} \end{bmatrix}, \quad \boldsymbol{R} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{I} \\ -\boldsymbol{I} & -\boldsymbol{I} \end{bmatrix}.$$

General approach to regression

We now describe a more general framework for thinking about regression. Recall that we observe $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$, with $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, and our goal is to estimate a function $h(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}$. We can fit this function by trading off two factors:

1. **Data Fidelity.** Our solution should satisfy

$$h(\boldsymbol{x}_i) \approx y_i, \quad \text{for} \;\; i = 1, \ldots, n$$

   This is typically quantified using a **loss function**, which penalizes the deviations of $h(\boldsymbol{x}_i)$ from y_i. This typically has the form of a single scalar function that is applied to each data point and then added up:

$$\text{Loss}(h, \{\boldsymbol{x}_i\}_{i=1}^n, \{y_i\}_{i=1}^n) = \sum_{i=1}^n L(h(\boldsymbol{x}_i), y_i)$$

   So far we have focused exclusively on the *squared-error* loss function:

$$L(h(\boldsymbol{x}_i), y_i) = (y_i - h(\boldsymbol{x}_i))^2.$$

2. **Modeling and Regularization.** We can temper the regression function in one of two ways. The first is to simply restrict it to lying in some function class $\mathcal{H}$. We then solve

$$\underset{h \in \mathcal{H}}{\text{minimize}} \sum_{i=1}^n L(h(\boldsymbol{x}_i), y_i).$$

   For example, we might consider the set of *linear functions*:

$$\mathcal{F} = \{h \;:\; h(\boldsymbol{x}) = \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{x} + \beta_0 \;\text{ for some }\; \boldsymbol{\beta} \in \mathbb{R}^d, \; \beta_0 \in \mathbb{R}\}.$$

The problem of estimating the function $h : \mathbb{R}^d \to \mathbb{R}$ is now distilled down to the problem of estimating $\boldsymbol{\beta}$ and β_0.

There are of course trade-offs in choosing $\mathcal{H}$. A larger $\mathcal{H}$ gives us a richer class of functions to choose from, but we run the risk of overfitting — the empirical risk $\frac{1}{n}\sum_{i=1}^{n} L(h_{\mathcal{D}}(\boldsymbol{x}_i, y_i))$ might be very different than the true risk $\mathbb{E}[L(h_{\mathcal{D}}(X), Y)]$ of our choice $h_{\mathcal{D}}$.

The second way to mitigate the danger of overfitting is to use a large, rich set $\mathcal{H}$, but then have some penalty on the complexity of the choices of h inside of this class. This "complexity" is quantified using a regularization function $r(h)$ — there are many choices of r we might consider. We then solve

$$\underset{h\in\mathcal{H}}{\text{minimize}} \sum_{i=1}^{n} L(h(\boldsymbol{x}_i), y_i) + \lambda\, r(h),$$

where $\lambda \geq 0$ is a user-specified parameter that controls the balance between these two terms.

We have seen two main variants of regularization (based on the ℓ_2 and ℓ_1 norms), but have only considered the squared error loss. There are many other possible choices of loss functions. Three that are particularly noteworthy in the context of regression are:

- the mean absolute error $L_{\mathrm{AE}}(r) = |r|$;
- the Huber loss $L_{\mathrm{H}}(r) = \begin{cases} \frac{1}{2}r^2 \text{ if } |r| \leq c \\ c\,|r| - \frac{c^2}{2} \text{ else} \end{cases}$;
- the ϵ-insensitive loss $L_{\epsilon}(r) = \begin{cases} 0 \text{ if } |r| \leq \epsilon \\ |r| - \epsilon \text{ else} \end{cases}$.

These losses are illustrated in Fig. 1. All of these loss functions are somewhat more robust to "outliers", in that they place less of a penalty on having a few large prediction errors, provided that there are not too many of them. This is often a very useful property.

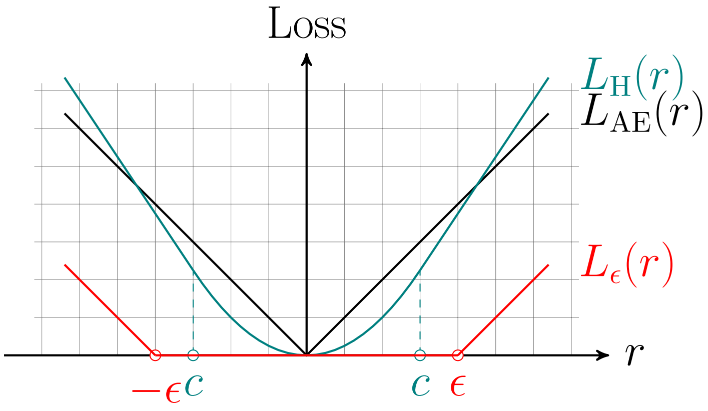

Figure 1: Illustration of loss functions