Regression and Regularization

For the regression problem, 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 try to estimate a function $f(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}$. We fit this function by trading off two factors:

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

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

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

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

   One commonly used loss function is *squared-error*,

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

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 $\mathcal{F}$. We then solve

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

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

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

Georgia Tech ECE 6254 Spring 2017; Notes by M. A. Davenport and J. Romberg. Last updated 14:09, February 23, 2017

The problem of estimating the function $f : \mathbb{R}^d \to \mathbb{R}$ is now distilled down to the problem of estimating a vector $\boldsymbol{\theta} = \begin{bmatrix} \beta_0 \\ \boldsymbol{\beta} \end{bmatrix} \in \mathbb{R}^{d+1}$.

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

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

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

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

Example: Linear regression using the LASSO

Let's look carefully at one particular (and popular!) technique for regression. 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]

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

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

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 $\boldsymbol{\theta}$:

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

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

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

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

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

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 $z(i)$:

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

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

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

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

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

This is the same as solving

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

which is the same as solving

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

where

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