Why mean squared error and ℓ_2 regularization? A probabilistic justification.[*]

Avital Oliver

March 2017

When you solve a regression problem with gradient descent, you're minimizing some differentiable loss function. The most commonly used loss function is mean squared error (aka MSE, ℓ_2 loss). Why? Here is a simple probabilistic justification, which can also be used to explain ℓ_1 loss, as well as ℓ_1 and ℓ_2 regularization.

1 What is regression?

What is a regression problem? In simplest form, we have a dataset $\mathcal{D} = \{(x_i \in \mathbb{R}^n, y_i \in \mathbb{R})\}$ and want a function f that approximately maps x_i to y_i without overfitting. We typically choose a function (from some family Θ) parametrized by θ. A simple parametrization is $f_\theta : x \mapsto x \cdot \theta$ where $\theta \in \Theta = \mathbb{R}^n$ – this is linear regression. Neural networks are another kind of parametrization.

Now we use some optimization scheme to find a function in that family that minimizes some loss function on our data. Which loss function should we use? People commonly use mean squared error (aka ℓ_2 loss): $\frac{1}{|\mathcal{D}|}\sum(y_i - f_\theta(x_i))^2$. Why?

2 Two assumptions: (1) Data is noisy; (2) We want the most likely model

Let's start with a few assumptions:

1. The data is generated by a function in our family, parametrized by θ_{true}, plus noise, which can be modeled by a zero-mean Gaussian random variable:
$$f_{\text{data}}(x) = f_{\theta_{\text{true}}}(x) + \epsilon \tag{1}$$
$$\epsilon \sim \mathcal{N}(0, \sigma^2) \tag{2}$$
(Why Gaussian? We'll get back to this question later.)

[*] Read and comment on the latest version of this note at `http://aoliver.org/why-mse`

2. Given the data, we'd like to find the most probable model within our family. Formally, we're looking for parameters θ with the highest probability:

$$\arg\max_{\theta}(P(\theta \mid \mathcal{D})) \tag{3}$$

With these assumptions, we can derive ℓ_2 loss as the principled error metric to optimize. Let's see how.

3 Probability of data given parameters

First, observe that with these two assumptions, we can derive the probability of a particular datapoint (x, y):

$$\begin{align}
P((x,y) \in \mathcal{D} \mid \theta) &= P(y = f_\theta(x) + \epsilon \mid \epsilon \sim \mathcal{N}(0, \sigma^2))) \tag{4}\\
&= \mathcal{N}(y - f_\theta(x); 0, \sigma^2) \tag{5}\\
&= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y - f_\theta(x)^2}{2\sigma^2}} \tag{6}
\end{align}$$

The math will be less complicated if we use log probability, so let's switch to that here:

$$\begin{align}
\log P((x,y) \in \mathcal{D} \mid \theta) &= \log \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y - f_\theta(x))^2}{2\sigma^2}} \tag{7}\\
&= -\frac{(y - f_\theta(x))^2}{2\sigma^2} + const. \tag{8}
\end{align}$$

Notice the $(y - f_\theta(x))^2$ term above – that's how we're going to get the ℓ_2 loss. (Where did it come from? Could we have gotten something else there?)

Now we can extend this from the log probability of a data point to the log probability of the entire dataset. This requires us to assume that each data point is independently sampled, commonly called the *i.i.d. assumption.*

$$\begin{align}
\log P(\mathcal{D} \mid \theta) &= \sum \log P(y_i = f_\theta(x_i) + \epsilon \mid \epsilon \sim \mathcal{N}(0, \sigma^2))) \tag{9}\\
&= -\frac{1}{2\sigma^2} \sum_{x,y \in \mathcal{D}} (y - f_\theta(x))^2 + const. \tag{10}
\end{align}$$

That's a simple formula for the probability of our data given our parameters. However, what we really want is to maximize the probability of the parameters given the data, i.e. $P(\theta \mid \mathcal{D})$.

4 Minimizing MSE *is* maximizing probability

We turn to Bayes' rule, $P(\theta \mid \mathcal{D}) \propto P(\mathcal{D} \mid \theta)P(\theta)$, and find that:

$$\begin{aligned}
\log P(\theta \mid \mathcal{D}) &= \log P(\mathcal{D} \mid \theta) + \log P(\theta) + \mathit{const.} \qquad (11)\\
&= \left[-\frac{1}{2\sigma^2} \sum_{x,y \in \mathcal{D}} (y - f_\theta(x))^2\right] + \log P(\theta) + \mathit{const.} \qquad (12)
\end{aligned}$$

The term in the left-hand side logarithm, $P(\theta \mid \mathcal{D})$, is called the *posterior distribution*. The two non-constant right-hand side terms also have names: $P(\mathcal{D} \mid \theta)$ is the *likelihood*, and $P(\theta)$ is the *prior distribution* (the likelihood does not integrate to 1, so it's not a distribution). The prior is a distribution we have to choose based on assumptions outside of our data. Let's start with the simplest – the so-called *uninformative prior* $P(\theta) \propto 1$, which doesn't describe a real probability distribution but still lets us compute the posterior. Choosing an uninformative prior corresponds to making no judgement about which parameters are more likely. If we choose the uninformative prior, we get:

$$\begin{aligned}
\log P(\theta \mid \mathcal{D}) &= \log P(\mathcal{D} \mid \theta) + \log P(\theta) + \mathit{const.} \qquad (13)\\
&= -\frac{1}{2\sigma^2} \sum_{x,y \in \mathcal{D}} (y - f_\theta(x))^2 + \mathit{const.} \qquad (14)
\end{aligned}$$

Ok woah. We're there. Maximizing $P(\theta \mid \mathcal{D})$ is the same as minimizing $\sum(y_i - f_\theta(x_i))^2$. The formal way of saying this is that minimizing mean squared error maximizes the *likelihood* of the parameters. In short, we've found the *maximum likelihood estimator* (MLE).

5 If we change our assumptions, though...

We can also change our assumptions and see what happens:

1. What if we change the variance on the noise? The log posterior which we're maximizing changes by a constant factor, so the same model is most likely. We only needed to assume that the noise is drawn from *some* zero-mean Gaussian. (The variance matters if we place a prior as in (3) below)

2. If we assume a different type of noise distribution, we'd derive a different loss function. For example, if we model the noise as being drawn from a Laplace distribution, we'd end up with ℓ_1 error instead.

3. If we actually place a prior on our parameters we'd get a regularization term added to the log posterior that we're maximizing. For example,

if the prior is a zero-mean Gaussian, we'd get ℓ_2 regularization. And if the prior is a zero-mean Laplacian, we'd get ℓ_1 regularization. When we set a prior, we call the most likely parameters the *maximum a posteriori estimate* (MAP).

4. What if our models have different types of parameters, such as the layers in a neural network? We would still want to place a prior on them to avoid overfitting, but we'd want a different prior for different layers. This corresponds to choosing different regularization hyperparameters for each layer.

But don't believe me – derive these yourself!