Linear Regression via Maximization of the Likelihood

Ryan P. Adams
COS 324 – Elements of Machine Learning
Princeton University

In least squares regression, we presented the common viewpoint that our approach to supervised learning be framed in terms of a loss function that scores our predictions relative to the ground truth as determined by the training data. That is, we introduced the idea of a function $\ell(\hat{y}, y)$ that is bigger when our machine learning model produces an estimate $\hat{y}$ that is worse relative to y. The loss function is a critical piece for turning the model-fitting problem into an optimization problem. In the case of least-squares regression, we used a squared loss:

$$\ell(\hat{y}, y) = (\hat{y} - y)^2 \,. \tag{1}$$

In this note we'll discuss a probabilistic view on constructing optimization problems that fit parameters to data by turning our loss function into a *likelihood*.

Maximizing the Likelihood

An alternative view on fitting a model is to think about a probabilistic procedure that might've given rise to the data. This probabilistic procedure would have parameters and then we can take the approach of trying to identify which parameters would assign the highest probability to the data that was observed. When we talk about probabilistic procedures that generate data given some parameters or covariates, we are really talking about *conditional probability distributions*.

Let's step away from regression for a minute and just talk about how we might think about a probabilistic procedure that generates data from a Gaussian distribution with a known variance but an unknown mean. Consider a set of data $\{y_n\}_{n=1}^N$ where $y_n \in \mathbb{R}$ and we assume that they are all independently and identically distributed according to a Gaussian distribution with unknown mean μ and variance σ^2. We would write this as a conditional probability as

$$y_n \mid \mu, \sigma^2 \sim \mathcal{N}(y_n \mid \mu, \sigma^2) \,. \tag{2}$$

The probability density function associate with this conditional distribution is the familiar univariate Gaussian:

$$\Pr(y_n \mid \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(y_n - \mu)^2\right\} \,. \tag{3}$$

We have N i.i.d. data, however, and so we write the conditional distribution for all of them as a product:

$$\Pr(\{y_n\}_{n=1}^N \mid \mu, \sigma^2) = \prod_{n=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(y_n - \mu)^2\right\} . \tag{4}$$

This function, which we are here thinking of as *being parameterized by* μ is what we call the *likelihood*. When we maximize this with respect to μ, we are asking "what μ would assign the highest probability to the data we've seen?" This inductive criterion of selecting model parameters based on their ability to probabilistically explain the data is what we refer to as *maximum likelihood estimation* (MLE). Maximum likelihood estimation is a cornerstone of statistics and it has many wonderful properties that are out of scope for this course. At the end of the day, however, we can think of this as being a different (negative) loss function:

$$\mu^\star = \mu^{\mathsf{MLE}} = \arg\max_\mu \Pr(\{y_n\}_{n=1}^N \mid \mu, \sigma^2) = \arg\max_\mu \prod_{n=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(y_n - \mu)^2\right\} . \tag{5}$$

In practice, this isn't exactly the problem that we like to solve. Rather, we actually prefer to maximize the *log* likelihood because it turns all of our products into sums, which are easier to manipulate and differentiate, while preserving the location of the maximum. Also, when we take the product of many things that may be less than 1, the floating point numbers on our computer may become very close to zero and the maximization may not be numerically stable; taking the log makes those small positive numbers into better behaved negative numbers. As such, our (negative) loss function becomes

$$L(\mu) = \log \Pr(\{x_n\}_{n=1}^N \mid \mu, \sigma^2) = \sum_{n=1}^{N} \log\left(\frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(y_n - \mu)^2\right\}\right) \tag{6}$$

$$= -N \log \sigma - \frac{N}{2} \log 2\pi - \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mu)^2 . \tag{7}$$

Figure 1 shows the likelihood function $L(\mu)$ that arises from a small set of data. Note in particular how the vertical scale of the likelihood is very small; this is one reason we transform it with the natural logarithm. We can go on and find the maximum likelihood estimate of μ by following the same kind of procedure that we used for least squares regression: differentiate, set to zero, and solve for μ:

$$\frac{d}{d\mu} L(\mu) = \frac{1}{\sigma^2} \sum_{n=1}^{N} (y_n - \mu) = 0 \tag{8}$$

$$\frac{1}{\sigma^2} \sum_{n=1}^{N} y_n - \frac{N}{\sigma^2}\mu = 0 \tag{9}$$

$$\mu = \frac{1}{N} \sum_{n=1}^{N} y_n . \tag{10}$$

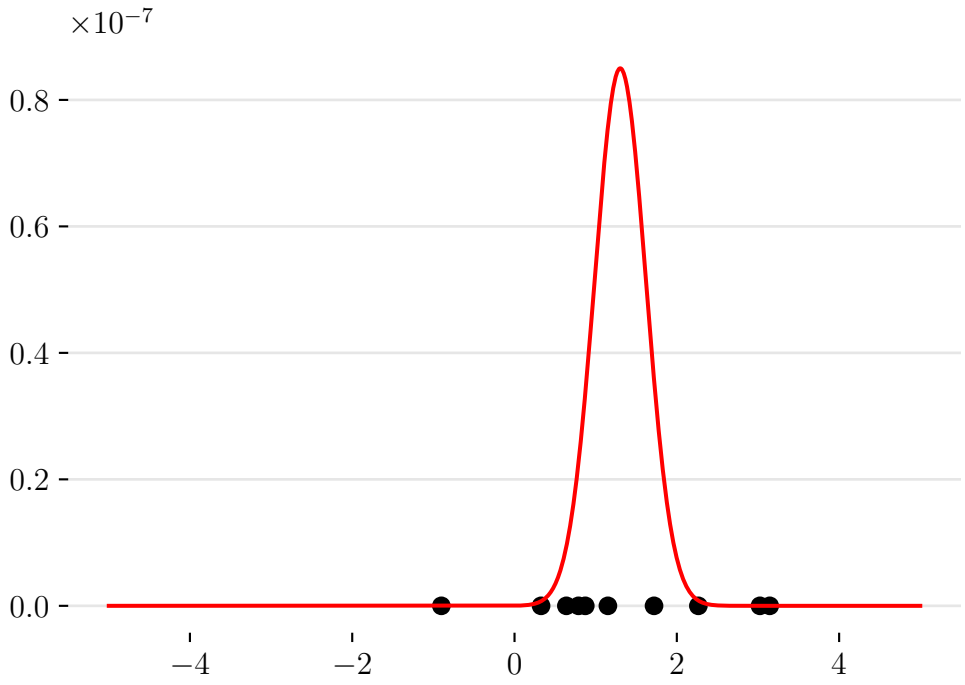

Figure 1: The black dots are ten ($N = 10$) data from a Gaussian distribution with $\sigma^2 = 1$ and $\mu = 1.4$. The red line is the likelihood as a function of μ. The maximum likelihood estimate is the peak of the red line. The red line is proportional to a Gaussian distribution but it is not generally true that likelihoods will have the same shape as the data distribution; this is a property that happens to arise when the location parameter is the quantity being estimated. Note also that the scale of the likelihood is tiny.

Unsurprisingly, the maximum likelihood estimate in this model (regardless of σ^2) is the sample average of the data.

MLE Regression with Gaussian Noise

We now revisit the linear regression problem with a maximum likelihood approach. As in the previous lecture, we assume our data are tuples of the form $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, where $\boldsymbol{x}_n \in \mathbb{R}^D$ and $y_n \in \mathbb{R}$. Rather than having a least-squares loss function, however, we now have to construct an explicit model for the noise that lets us reason about the conditional probability of the label. That is, we're now going to say that our data arise from a process like

$$y = \boldsymbol{x}^\top \boldsymbol{w} + \epsilon \tag{11}$$

where $\epsilon \in \mathbb{R}$ is a random variable capturing the noise. Just like we can construct different loss functions, we can think about different noise models for ϵ. Generalizing the previous section, a very natural idea is to say that this noise is from a zero-mean Gaussian distribution with variance σ^2,

i.e.,

$$\epsilon \mid \sigma^2 \sim \mathcal{N}(\epsilon \mid 0, \sigma^2)\,. \tag{12}$$

Adding a constant to a Gaussian just has the effect of shifting its mean, so the resulting conditional probability distribution for our generative probabilistic process is

$$\Pr(y_n \mid \boldsymbol{x}_n, \boldsymbol{w}, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(y_n - \boldsymbol{x}_n^\mathsf{T}\boldsymbol{w})^2\right\}\,. \tag{13}$$

This is very similar to Eq. 3, except now rather than conditioning on μ, we're conditioning on $\boldsymbol{x}_n$ and $\boldsymbol{w}$. Those two quantities combine to form the mean of the Gaussian distribution on y_n.

We denote the noise associated with the nth observation as ϵ_n and we will take these to be independent and identically distributed. This allows us to write the overall likelihood function as a product over these N terms:

$$\Pr(\{y_n\}_{n=1}^N \mid \{\boldsymbol{x}_n\}_{n=1}^N, \boldsymbol{w}, \sigma^2) = \prod_{n=1}^N \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(y_n - \boldsymbol{x}_n^\mathsf{T}\boldsymbol{w})^2\right\}\,. \tag{14}$$

Again, this is a function of $\boldsymbol{w}$, as that is the parameter we are seeking to fit to the data.

In the previous lecture, we employed some more compact notation and aggregated the labels into a vector $\boldsymbol{y}$ and the features into a design matrix $\boldsymbol{X}$. We do the same trick here, but with a new twist: we're going to turn this univariate Gaussian distribution into a multivariate Gaussian distribution with a diagonal covariance matrix. Recall that the PDF for a D-dimensional Gaussian distribution is

$$\Pr(\boldsymbol{z} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = |\boldsymbol{\Sigma}|^{-1/2}(2\pi)^{-D/2} \exp\left\{-\frac{1}{2}(\boldsymbol{z} - \boldsymbol{\mu})^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\boldsymbol{z} - \boldsymbol{\mu})\right\}\,. \tag{15}$$

The covariance matrix $\boldsymbol{\Sigma}$ must be square, symmetric, and positive definite. When $\boldsymbol{\Sigma}$ is diagonal, the D dimensions are independent of each other. Putting our regression likelihood into this form we write:

$$\Pr(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{w}, \sigma^2) = \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{X}\boldsymbol{w}, \sigma^2\mathbf{I}) = (2\sigma^2\pi)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^\mathsf{T}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})\right\}\,. \tag{16}$$

We can now think about how we'd maximize this with respect to $\boldsymbol{w}$ in order to find the maximum likelihood estimate. As in the simple Gaussian case, it is helpful to take the natural log first:

$$\log \Pr(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{w}, \sigma^2) = -\frac{N}{2}\log(2\sigma^2\pi) - \frac{1}{2\sigma^2}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^\mathsf{T}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})\,. \tag{17}$$

The additive term doesn't have a $\boldsymbol{w}$. We are then left with the following optimization problem:

$$\boldsymbol{w}^{\mathsf{MLE}} = \arg\max_{\boldsymbol{w}} \left\{-\frac{1}{2\sigma^2}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^\mathsf{T}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})\right\}\,. \tag{18}$$

The $\frac{1}{2\sigma^2}$ does not change the solution to this problem and of course could change the sign and make this maximization into a minimization:

$$\boldsymbol{w}^{\mathsf{MLE}} = \arg\min_{\boldsymbol{w}} (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^{\mathsf{T}} (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}) \,. \tag{19}$$

This is exactly the same optimization problem that we solved for the least-squares linear regression! While it seems like the loss function view and the maximum likelihood view are different, this reveals that they are often the same under the hood: least squares can be interpreted as assuming Gaussian noise, and particular choices of likelihood can be interpreted directly as (usually exponentiated) loss functions.

Fitting σ^2

One thing that is different about maximum likelihood, however, is that it gives us an additional parameter to play with that helps us reason about the *predictive distribution*. The predictive distribution is the distribution over the label, given parameters we have just fit. Rather than simply producing a single estimate, when we have a probabilistic model we can account for noise when we look at test data. That is, after finding $\boldsymbol{w}^{\mathsf{MLE}}$ if we have a query input $\boldsymbol{x}_{\mathsf{pred}}$ for which we don't know the $\boldsymbol{y}$, we could compute a guess via $y_{\mathsf{pred}} = \boldsymbol{x}_{\mathsf{pred}}^{\mathsf{T}} \boldsymbol{w}^{\mathsf{MLE}}$, or we could actually construct a whole distribution:

$$\Pr(y_{\mathsf{pred}} \mid \boldsymbol{x}_{\mathsf{pred}}, \boldsymbol{w}^{\mathsf{MLE}}, \sigma^2) = \mathcal{N}(y_{\mathsf{pred}} \mid \boldsymbol{x}_{\mathsf{pred}}^{\mathsf{T}} \boldsymbol{w}^{\mathsf{MLE}}, \sigma^2) \,. \tag{20}$$

This sounds great, but σ^2 went away when we constructed the optimization problem for $\boldsymbol{w}$. Why would it be any good? Well, it won't be any good — unless you fit it also. Fortunately, maximum likelihood estimation tells us how to do that one also, and we can start out by assuming that we've already computed $\boldsymbol{w}^{\mathsf{MLE}}$. We set up the problem the same way except we keep the additive term in Eq. 17:

$$\sigma^{\mathsf{MLE}} = \arg\max_{\sigma} \left\{ -\frac{N}{2} \log(2\sigma^2\pi) - \frac{1}{2\sigma^2} (\boldsymbol{X}\boldsymbol{w}^{\mathsf{MLE}} - \boldsymbol{y})^{\mathsf{T}} (\boldsymbol{X}\boldsymbol{w}^{\mathsf{MLE}} - \boldsymbol{y}) \right\} . \tag{21}$$

Solving this maximization problem is again just a question of differentiating and setting to zero:

$$\frac{\partial}{\partial \sigma^2} \left[-\frac{N}{2} \log \sigma^2 - \frac{1}{2\sigma^2} (\boldsymbol{X}\boldsymbol{w}^{\mathsf{MLE}} - \boldsymbol{y})^{\mathsf{T}} (\boldsymbol{X}\boldsymbol{w}^{\mathsf{MLE}} - \boldsymbol{y}) \right] = 0 \tag{22}$$

$$-\frac{N}{2\sigma^2} + \frac{1}{2\sigma^4} (\boldsymbol{X}\boldsymbol{w}^{\mathsf{MLE}} - \boldsymbol{y})^{\mathsf{T}} (\boldsymbol{X}\boldsymbol{w}^{\mathsf{MLE}} - \boldsymbol{y}) = 0 \tag{23}$$

$$-N + \frac{1}{\sigma^2} (\boldsymbol{X}\boldsymbol{w}^{\mathsf{MLE}} - \boldsymbol{y})^{\mathsf{T}} (\boldsymbol{X}\boldsymbol{w}^{\mathsf{MLE}} - \boldsymbol{y}) = 0 \tag{24}$$

$$\sigma^2 = \frac{1}{N} (\boldsymbol{X}\boldsymbol{w}^{\mathsf{MLE}} - \boldsymbol{y})^{\mathsf{T}} (\boldsymbol{X}\boldsymbol{w}^{\mathsf{MLE}} - \boldsymbol{y}) \,. \tag{25}$$

This is a satisfying result because it is just finding the sample average of the squared deviations between what $\boldsymbol{w}^{\mathsf{MLE}}$ predicts and what the training data actually are. It feels exactly like what happens when you compute the maximum likelihood estimate of the variance of a univariate Gaussian distribution.

Changelog

- 17 September 2018 – Initial version