

## Probabilistic Modeling

The goal of probabilistic modeling is to set up a statistical model  $p(\text{data} \mid \text{model})$  that “explains” our data (given a model) and then try to find the “best” model.

Suppose we believe there is a functional relationship  $y = \mathbf{w}^\top \mathbf{x}$  between some set of inputs  $\mathbf{x}_1, \dots, \mathbf{x}_N$  and some set of outputs  $y_1, \dots, y_n$ . However, in data that we observe, this relationship is corrupted by noise. That is, the  $y$  we observe are not precisely  $\mathbf{w}^\top \mathbf{x}$ , but is rather corrupted by some noise. We wish to model this noise stochastically; one choice is to say that the noise is Gaussian distributed. Namely,  $y = \mathbf{w}^\top \mathbf{x} + \epsilon$  and  $\epsilon \sim \mathcal{N}(0, \sigma^2)$ .

Here,  $\epsilon \sim \mathcal{N}(\mu, \sigma^2)$  denotes that  $\epsilon$  is drawn from a Gaussian (“Normal”) distribution with mean  $\mu$  and variance  $\sigma^2$ . This Gaussian has density:

$$\mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(x - \mu)^2\right] \quad (1)$$

To think about  $y = \mathbf{w}^\top \mathbf{x} + \epsilon$ , think about a true linear relationship that is then slightly corrupted. Since we assume that the error model is Gaussian with mean zero, this is the same as saying  $y \sim \mathcal{N}(\mathbf{w}^\top \mathbf{x}, \sigma^2)$ . (This is not hard to verify: if you don’t see it, try working it out.)

Now, we are ready to talk about  $p(\text{data} \mid \text{model})$ . Since the inputs  $\mathbf{x}$  are always given to us, we do not need to explicitly model them. Thus, although our data are the pairs  $(\mathbf{x}_n, y_n)$ , the  $\mathbf{x}$ s are always provided. Furthermore, our model (in this case) is totally specified by  $\mathbf{w}$ . We can therefore consider something of the form:

$$p(y_1, \dots, y_N \mid \mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{w}) \quad (2)$$

The first step is to apply the chain rule:

$$p(y_1, \dots, y_N \mid \mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{w}) = \prod_{n=1}^N p(y_n \mid \mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{w}, y_1, \dots, y_{n-1}) \quad (3)$$

Next, we make an assumption: conditioned on  $\mathbf{x}_n$  and  $\mathbf{w}$ , the  $y$ s are completely independent of each other, and  $y_n$  is independent of all  $\mathbf{x}_m$ s for  $m \neq n$ . Thus:

$$p(y_1, \dots, y_N \mid \mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{w}) = \prod_{n=1}^N p(y_n \mid \mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{w}, y_1, \dots, y_{n-1}) \quad (4)$$

$$= \prod_{n=1}^N p(y_n \mid \mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{w}) \quad (5)$$

$$= \prod_{n=1}^N p(y_n \mid \mathbf{x}_n, \mathbf{w}) \quad (6)$$

Now, we can substitute in our Gaussian model for  $p(y_n \mid \mathbf{x}_n, \mathbf{w})$ :

$$\prod_{n=1}^N p(y_n | \mathbf{x}_n, \mathbf{w}) = \prod_{n=1}^N \mathcal{N}or(y_n | \mathbf{w}^\top \mathbf{x}_n, \sigma^2) \quad (7)$$

Our first method of “solving” this is by using the *maximum likelihood estimator*. Namely, we want to choose the  $\mathbf{w}$  that maximizes the probability of the data given the model (probability of data given model is typically called the “likelihood”). Our standard methods for maximizing (or minimizing) require us to differentiate. Differentiating that product looks nasty. So instead of maximizing the *likelihood*, we will maximize the *log likelihood*

$$\ell\ell(\mathbf{w}) = \log p(\text{data} | \text{model}) \quad (8)$$

$$= \log \prod_{n=1}^N \mathcal{N}or(y_n | \mathbf{w}^\top \mathbf{x}_n, \sigma^2) \quad (9)$$

$$= \sum_{n=1}^N \log \mathcal{N}or(y_n | \mathbf{w}^\top \mathbf{x}_n, \sigma^2) \quad (10)$$

We can now plug in the definition of the Gaussian and do some simplification:

$$\ell\ell(\mathbf{w}) = \sum_{n=1}^N \log \mathcal{N}or(y_n | \mathbf{w}^\top \mathbf{x}_n, \sigma^2) \quad (11)$$

$$= \sum_{n=1}^N \log \left( \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{1}{2\sigma^2} (y_n - \mathbf{w}^\top \mathbf{x}_n)^2 \right] \right) \quad (12)$$

$$= \sum_{n=1}^N -\frac{1}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} (y_n - \mathbf{w}^\top \mathbf{x}_n)^2 \quad (13)$$

Now, this looks slightly messy, but remember that we’re just treating this as a function of  $\mathbf{w}$ . That means that the additive term in the front is irrelevant. The constant  $1/(2\sigma^2)$  is also irrelevant. Thus, to maximize the likelihood, we could instead maximize  $-\sum_n (y_n - \mathbf{w}^\top \mathbf{x}_n)^2$ . Or, equivalently, minimize  $\sum_n (y_n - \mathbf{w}^\top \mathbf{x}_n)^2$ . (Note the missing negative sign.)

This is *exactly* the linear regression model we came up with before by trying to minimize squared error! One way to interpret this is that by *choosing* squared error as our loss function, we are implicitly assuming a Gaussian noise model.

We can do the same for classification.

We might want to make some model like  $y = \text{sign}[\mathbf{w}^\top \mathbf{x}]$  for a binary classification problem with  $y \in \{-1, +1\}$ . However, this is incapable of dealing with noise. Instead, we transform  $\mathbf{w}^\top \mathbf{x}$  via the sigmoid and use the sigmoid’s output as the *probability* that  $y$  is  $+1$ . In other words:

$$p(y = +1 | \mathbf{w}, \mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x}) \quad (14)$$

$$p(y = -1 | \mathbf{w}, \mathbf{x}) = 1 - \sigma(\mathbf{w}^\top \mathbf{x}) \quad (15)$$

A convenient property of the sigmoid is that  $1 - \sigma(z) = \sigma(-z)$ . Thus, we get:

$$p(y \mid \mathbf{w}, \mathbf{x}) = \sigma(y\mathbf{w}^\top \mathbf{x}) = \frac{1}{1 + \exp[-y\mathbf{w}^\top \mathbf{x}]} \quad (16)$$

Following exactly the same steps as before, we obtain:

$$p(\text{data} \mid \text{model}) = \prod_{n=1}^N p(y_n \mid \mathbf{w}, \mathbf{x}_n) \quad (17)$$

$$= \prod_{n=1}^N \frac{1}{1 + \exp[-y_n \mathbf{w}^\top \mathbf{x}_n]} \quad (18)$$

$$\ell\ell(\mathbf{w}) = \sum_{n=1}^N \log \frac{1}{1 + \exp[-y_n \mathbf{w}^\top \mathbf{x}_n]} \quad (19)$$

$$= - \sum_{n=1}^N \log (1 + \exp[-y_n \mathbf{w}^\top \mathbf{x}_n]) \quad (20)$$

$$(21)$$

This is exactly the log loss that we were optimizing before. So optimizing log loss is the same as assuming a sigmoid probability in a statistical model!