Generative Stories

Informally, imagine your data is a list of

(humidity, temperature, is_raining)
x_0 x_1 y

triples. You want to build a probabilistic model. Here's a story that generates data like that:

For each training point,

1. choose whether it's raining, $y_i \sim \underline{\text{Disc}(\theta)}$

2. Choose the humidity given whether or not it was raining:

$$x_i \sim \text{Nor}(\mu_{y_i,0}, \sigma^2_{y_i,0})$$

3. same thing for temperature.

Now you have, for your training set, a likelihood:

$$p(D) = \prod_i p(x_{i0}, x_{i1}, y_i)$$

$$= \prod_i p(y_i) \cdot p(x_{i0}, x_{i1} \mid y_i) \qquad \text{(cond. prob.)}$$

$$= \prod_i p(y_i) \cdot p(x_{i0} \mid y_i) \cdot p(x_{i1} \mid y_i) \qquad \text{(from story)}$$

So we can attempt to find θ, μ, σ^2 that maximize likelihood of model. This is just like training an ML classifier.

CONDITIONAL MODELS

For predicting whether it is raining, we don't need to know θ. So maybe we can try to learn $p(y \mid x_0, x_1)$ directly. $p(y = \text{rain} \mid x_0 = \ldots\ x_1 = \ldots) > p(y = \text{dry} \mid x_0 \ldots)$

But let's start with a simpler setup. Instead of predicting "is-raining", let's predict "inches-of-rain-per-hour". Let's assume that a model for this is:

$$e_i \sim \mathrm{Nor}(0, \sigma^2)$$

$$y_i = x_{0i} \cdot w_0 + x_{1i} \cdot w_1 + b + e_i$$

temperature

humidity

rainfall

This is the same as

$$\vec{w} = (w_0, w_1, b)$$

$$\vec{x}_i = (x_{0i}, x_{1i}, 1)$$

$$y_i \sim \mathrm{Nor}(\langle \vec{w}, \vec{x} \rangle, \sigma^2)$$

$$p(x = x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

$$\text{So } p(D) = \prod_i^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \langle \vec{w}, \vec{x} \rangle)^2}{2\sigma^2}\right)$$

Take logs, and you get:

$$\log p(D) = -\frac{1}{2\sigma^2} \cdot \sum_i \left(y_i - \langle \vec{w}, \vec{x} \rangle\right)^2$$

To maximize the likelihood of linear model under Gaussian noise, we must minimize the squared loss!

LOGISTIC REGRESSION

What if we want to classify instead of regress?

Idea: regress on some value, then transform value so range is $[-1, 1]$.

The classic transformation is the logistic function:

$$\sigma(z) = \frac{1}{1+e^{-z}} = \frac{e^{z}}{1+e^{z}}$$

Returning to our "is_raining" prediction task,

$t_i = \sigma(\langle w, x \rangle)$ ← the smaller this is, ...

$z_i = \text{Bernoulli}(t_i)$ ← ... the more likely this will come out 0, ...

$y_i = 2z_i - 1$ ← ... and this, -1.

So now we write the likelihood:

$$\log p(D) = \sum_i [y_i = 1] \log \sigma(\langle w, \vec{x}_i \rangle) + [y_i = -1] \log \sigma(-\langle w, \vec{x}_i \rangle)$$

$$\sigma(x) = \frac{1}{1+e^{-x}} \qquad = \sum_i \log \sigma(y_i \langle w, \vec{x}_i \rangle)$$

$$= -\sum_i \log\left(1 + \exp\left(-y_i \langle w, \vec{x}_i \rangle\right)\right) -$$

So maximizing the likelihood of this model is minimizing the logistic loss!

REGULARIZATION IS JUST A PRIOR

Where does regularization fit in?

Remember M.A.P.: find the posterior, choose parameters that maximize it.

$$p(\theta \mid D) = \frac{p(D \mid \theta)\; p(\theta)}{p(D)}$$

For the purposes of M.A.P., $p(D)$ is a constant, so if we want to optimize $p(\theta \mid D)$, we can ignore $p(D)$. Now we proceed as follows:

$$\begin{aligned}
\arg\max_{\theta}\; p(\theta \mid D) &= \arg\max_{\theta}\; \log p(\theta \mid D) \\
&= \arg\max_{\theta}\; \log \frac{p(D \mid \theta)\; p(\theta)}{p(D)} \\
&= \arg\max_{\theta}\; \log p(D \mid \theta)\; p(\theta) \\
&= \arg\max_{\theta}\; \log p(D \mid \theta) + \log p(\theta)
\end{aligned}$$

Any prior looks like additive change to log-likelihood!

If probability distribution has simple log likelihood...

Consider a normal distribution centered at 0

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

$$\log p(x) = \cancel{\log\left(\frac{1}{\sqrt{2\pi}}\right) + \log\left(\frac{1}{\sigma}\right)} - \frac{x^2}{2\sigma^2} \leftarrow -\, x_0^2 + x_1^2 + x_2^2 = -\lambda \|\vec{x}\|^2$$

↑ constants, ↑ don't depend on x

$\lambda = \frac{1}{2\sigma^2}$

If we assume a prior where every parameter is drawn from a gaussian, we recover l_2 regularization!

This is nice because we now have different ways to look at the same problem:

Empirical loss minimization $\approx$ M.L.E.
Structural loss minimization $\approx$ M.A.P.

↑ loss-function view of the world.

↑ probabilistic view of the world.