Bayesian Inference

Muchang Bahng

Spring 2025

Contents

In frequentist inference, we would treat some true parameter θ as *fixed*, albeit unknown. We would do density estimation by maximizing the likelihood over some data $\mathcal{D}$ with respect so some parameters θ.

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, p(\mathcal{D} \mid \theta) \tag{1}$$

This maximum likelihood estimation, along with the method of moments, were two paradigms of estimation that we have seen. As long as we can solve this, we could get a point estimate, and there are a lot of ways to solve this. By assuming independence, we can decompose the probability term into a product of the likelihoods of each sample, which we can hopefully solve analytically or with numerical optimizers. We can even create confidence sets that give us some information about the uncertainty in our estimates. However, there is an Achilles heel of Frequentist inference, as depicted in the following example.

Example 0.1 (Conflicting Results in Significance Tests)

Let θ be the probability of a particular coin landing on heads, and suppose we want to test the hypotheses

$$H_0 : \theta = 1/2, \quad H_1 : \theta > 1/2 \tag{2}$$

at a significance level of $\alpha = 0.05$. Now suppose we observe the following sequence of flips:

heads, heads, heads, heads, heads, tails (5 heads, 1 tails)

To perform a frequentist hypothesis test, we must define a random variable to describe the data. The proper way to do this depends on exactly which of the following two ways the experiment was performed:

- Suppose that the experiment was "Flip six times and record the results." In this case, the random variable X counts the number of heads, and $X \sim \text{Binomial}(6, \theta)$. The observed data was $x = 5$, and the p-value of our hypothesis test is

$$\begin{aligned} \text{p-value} &= P_{\theta=1/2}(X \geq 5) && (3)\\ &= P_{\theta=1/2}(X = 5) + P_{\theta=1/2}(X = 6) && (4)\\ &= \frac{6}{64} + \frac{1}{64} = \frac{7}{64} = 0.109375 > 0.05. && (5) \end{aligned}$$

  So we fail to reject H_0 at $\alpha = 0.05$.
- Suppose instead that the experiment was "Flip until we get tails." In this case, the random variable X counts the number of the flip on which the first tails occurs, and $X \sim \text{Geometric}(1-\theta)$. The observed data was $x = 6$, and the p-value of our hypothesis test is

$$\begin{aligned} \text{p-value} &= P_{\theta=1/2}(X \geq 6) && (6)\\ &= 1 - P_{\theta=1/2}(X < 6) && (7)\\ &= 1 - \sum_{x=1}^{5} P_{\theta=1/2}(X = x) && (8)\\ &= 1 - \left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32}\right) = \frac{1}{32} = 0.03125 < 0.05. && (9) \end{aligned}$$

  So we reject H_0 at $\alpha = 0.05$.

The conclusions differ, which seems absurd. Moreover the p-values aren't even close—one is 3.5 times as large as the other. Essentially, the result of our hypothesis test depends on whether we would have stopped flipping if we had gotten a tails sooner. In other words, the frequentist approach requires us to specify what we would have done had the data been something that we already know it wasn't.

Note that despite the different results, the likelihood for the actual value of x that was observed is the same

for both experiments (up to a constant):

$$p(x|\theta) \propto \theta^5(1-\theta). \tag{10}$$

A Bayesian approach would take the data into account only through this likelihood and would therefore be guaranteed to provide the same answers regardless of which experiment was being performed. Therefore, Bayesian modeling on the other hand treats the true θ as intrinsically *random*. Therefore, we can "solve" for θ by completely characterizing its distribution given the data $\mathcal{D}$. This is a much harder problem than finding a point estimate, but we can relate this to the often computable likelihood using Bayes rule.

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta)\, p(\theta)}{p(\mathcal{D})} \tag{11}$$

Note that if we take the maximum of this, it is equivalent to finding

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, p(\mathcal{D} \mid \theta)\, p(\theta) \tag{12}$$

which is called a *maximum a-posteriori (MAP)* estimate and is similar to MLE, but with an extra term $p(\theta)$. This term, called the **prior distribution**, does not depend on the data and is usually thought of as an initial guess for θ before we see any data. Upon observing the data, our prior gets updated to the **posterior distribution** $p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta)p(\theta)$. Since the posterior must integrate to 1, we have the normalizing constant $p(\mathcal{D})$, known as our *marginal likelihood*.

The general framework is very simple but there are two big questions.

1. How do we know *which* prior to put? In the beginning, we will work with some friendly priors—called *conjugate priors*—that have nice analytic posteriors. In practice, this is an art.
2. How can we compute the normalizing constant $p(\mathcal{D})$? For very simple distributions, we can analytically solve it, but in most cases, computing it is impossible. Therefore, we are only given some scaled version of the posterior density. Fortunately, there are a range of *Markov Chain Monte Carlo (MCMC)* samplers that remarkably only require these values to sample from the distributions. Therefore, we use the following notation more often when calculating posteriors since the normalizing constant isn't as important as finding the shape of the posterior density.

$$\text{Posterior} \;\propto\; \text{Prior} \;\times\; \text{Likelihood} \tag{13}$$

   The MCMC techniques are covered in my sampling notes.

Once we have a distribution of θ, we can do prediction or density estimation by naively using only the posterior defined by our MAP estimate $p(x \mid \mathcal{D}) = p(x \mid \hat{\theta})$. But for a full Bayesian treatment, we should condition over the θ.

Definition 0.1 (Posterior Predictive Distribution)

The **posterior predictive distribution** is defined

$$p(x \mid \mathcal{D}) = \int p(x \mid \theta, \mathcal{D}) p(\theta \mid \mathcal{D}) d\,\theta \tag{14}$$

A lot of this is based off of [Hof09].

1 Single Parameter Families

1.1 Conjugate Priors

As we always do, let's start off with the simplest problems that we can solve analytically.

Example 1.1 (Uniform Prior, Binomial Likelihood, Beta Posterior)

The motivation behind the Beta distribution is that it satisfies **conjugacy** with a binomial likelihood. That is, assume that we have some data $\mathbf{x}$ of N observations containing m successes and $N-m$ failures (note that this observation $\mathbf{x}$ was in a way "reduced" to the information of only the number of successes m). We assume that there is some true success rate θ (between 0 and 1, of course) coming from these samples, and our job is to try and guess the true rate to the best of our abilities.
Before we even observe the data $\mathbf{x}$, our initial guess of θ might be modeled by the prior distribution $\theta \sim \text{Beta}(a, b)$. Furthermore, the likelihood is clearly a binomial (since it represents the probability of getting m successes out of N samples with fixed rate of success θ), so $m \mid \theta \sim \text{Binomial}(N, \theta)$. With these conditions, we claim that the posterior is also a beta, since

$$\begin{aligned} p(\theta \mid m) &\propto p_\theta(\theta)\, p(m \mid \theta) \\ &\propto \theta^{a-1}(1-\theta)^{b-1} \cdot \theta^m (1-\theta)^{N-m} \\ &= \theta^{a+m-1}(1-\theta)^{b+N-m-1} \end{aligned}$$

Theorem 1.1 ()

We know

$$\mathbb{E}_\theta[\theta] = \mathbb{E}_x[\mathbb{E}_\theta[\theta \mid x]] \tag{15}$$
$$\text{Var}_\theta[\theta] = \mathbb{E}_x[\text{Var}_\theta[\theta \mid x]] + \text{Var}_x[\mathbb{E}_\theta[\theta \mid x]] \tag{16}$$

The maximum likelihood framework gave point estimates for the parameters μ and Σ. Now we develop a Bayesian treatment by introducing prior distributions over these parameters. Given a set of N D-dimensional observations $\mathbf{X} = \{x_1, \dots, x_n\}$, the likelihood function is given by (the unnormalized function of μ):

$$p(\mathbf{X} \mid \mu, \Sigma) = \prod_{n=1}^{N} p(x_n \mid \mu, \Sigma) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp \sum_{n=1}^{N} \left(-\frac{1}{2}(x_n - \mu)^T \Sigma^{-1} (x_n - \mu) \right) \tag{17}$$

The likelihood function takes the form of the exponential of a quadratic form in μ. Thus, if we choose a prior $p(\mu)$ given by a Gaussian, it will be a conjugate distribution for this likelihood function. Taking our prior distribution to be

$$p(\mu, \Sigma) = \mathcal{N}(\mu, \Sigma \mid \mu_0, \Sigma_0) \tag{18}$$

The similarity of the symbols μ, Σ with μ_0, Σ_0 may be slightly confusing. We can think as such: μ, Σ are random variables that determine the parameters of some Gaussian distribution. But the values μ, Σ are uncertain, and their possible values with probabilities take the form of another distribution $\mathcal{N}(\mu_0, \Sigma_0)$. The posterior distribution is given by the familiar formula

$$p(\mu, \Sigma \mid \mathbf{X}) \propto p(\mathbf{X} \mid \mu, \Sigma)\, p(\mu, \Sigma) \tag{19}$$

which is another Gaussian $p(\mu \mid \mathbf{X}) = \mathcal{N}(\mu, \Sigma \mid \mu_N, \Sigma_N)$. Let us place a few conditions for simplification. Since every Gaussian density can be represented as a product of independent univariate Gaussians, we can work with univariate Gaussians. Furthermore, let us assume that the true value of σ is known, so all we have to do is find the posterior distribution of μ using the prior density $\mathcal{N}(\mu \mid \mu_0, \sigma_0^2)$. We have our prior and likelihood to be the following. Note that while the likelihood distribution is pretty much given, we have the

flexibility to choose what our prior distribution is. We have only set the prior as a Gaussian simply because it is a conjugate form and therefore will greatly simplify calculations.

$$p(\mu) = \mathcal{N}(\mu \,|\, \mu_0, \sigma_0^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(\mu - \mu_0)^2) \right)$$
$$p(\mathbf{X} \,|\, \mu) = \prod_{n=1}^{N} p(x_n \,|\, \mu) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{n=1}^{N} (x_n - \mu)^2 \right)$$

which gives a posterior $p(\mu \,|\, \mathbf{X}) = \mathcal{N}(\mu \,|\, \mu_N, \sigma_N^2)$ where

$$\mu_N = \frac{\sigma^2}{N\sigma_0^2 + \sigma^2}\mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2 + \sigma^2}\mu_{ML}$$
$$\frac{1}{\sigma_N^2} = \frac{1}{\sigma_0^2} + \frac{N}{\sigma^2}$$

and μ_{ML} is the maximum likelihood solution for μ given by the sample mean $\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N} x_n$. These values make sense. We can see that the mean of the posterior distribution μ_N is a compromise between the prior mean μ_0 and maximum likelihood solution μ_{ML}. If the number of observed data points $N = 0$, then it is simply the prior mean, but for $N \to \infty$, the posterior mean is given by the maximum likelihood solution since the data "overpowers" the prior mean assumption.

Now, suppose that the mean of the Gaussian over the data is known and we wish to infer the variance. For convenience, let us work with the precision $\lambda = \frac{1}{\sigma^2}$ over the variance. The likelihood function for λ is

$$p(\mathbf{X} \,|\, \lambda) = \prod_{n=1}^{N} \mathcal{N}(x_n \,|\, \mu, \lambda^{-1}) \propto \lambda^{N/2} \exp\left(-\frac{\lambda}{2} \sum_{n=1}^{N} (x_n - \mu)^2 \right) \tag{20}$$

Note that since this is a function of λ, it behaves differently than the likelihood function of μ, even though they are of the same form. Since the likelihood function is proportional to the product of a power of λ and the exponential of a linear function of λ, we must find a prior distribution $p(\lambda)$ with precisely these proportional properties identical to that of the likelihood. Fortunately, the Gamma distribution satisfies them, defined by

$$p(\lambda \,|\, a_0, b_0) = \text{Gamma}(\lambda \,|\, a_0, b_0) = \frac{1}{\Gamma(a_0)} b_0^{a_0} \lambda^{a_0 - 1} \exp(-b_0 \lambda) \tag{21}$$

Using Bayes rule and multiplying gives the posterior density

$$p(\lambda \,|\, \mathbf{X}) \propto \lambda^{a_0 - 1} \lambda^{N/2} \exp\left(-b_0 \lambda - \frac{\lambda}{2} \sum_{n=1}^{N} (x_n - \mu)^2 \right) \tag{22}$$

which is indeed the density of a $\text{Gamma}(\lambda \,|\, a_N, b_N)$ distribution, where

$$a_N = a_0 + \frac{N}{2}$$
$$b_N = b_0 + \frac{1}{2} \sum_{n=1}^{N} (x_n - \mu)^2 = b_0 + \frac{N}{2}\sigma_{ML}^2$$

where σ_{ML}^2 is the maximum likelihood estimator of the variance. Now, suppose that both the mean and precision are unknown. To find a conjugate prior, we consider the dependence of the likelihood function on

μ and λ.

$$\begin{aligned} p(\mathbf{X} \mid \mu, \lambda) &= \prod_{n=1}^{N} \left(\frac{\lambda}{2\pi}\right)^{1/2} \exp\left(-\frac{\lambda}{2}(x_n - \mu)^2\right) \\ &\propto \left(\lambda^{1/2} \exp\left(-\frac{\lambda\mu^2}{2}\right)\right)^N \exp\left(\lambda\mu \sum_{n=1}^{N} x_n - \frac{\lambda}{2}\sum_{n=1}^{N} x_n^2\right) \end{aligned}$$

We now wish to identify a prior distribution $p(\mu, \lambda)$ that has the same functional dependence on μ and λ as the likelihood function and that should therefore take the form

$$\begin{aligned} p(\mu, \lambda) &\propto \left(\lambda^{1/2} \exp\left(-\frac{\lambda\mu^2}{2}\right)\right)^{\beta} \exp\left(c\lambda\mu - d\lambda\right) \\ &= \exp\left(-\frac{\beta\lambda}{2}\Big(\mu - \frac{c}{\beta}\Big)^2\right) \lambda^{\beta/2} \exp\left(-\Big(d - \frac{c^2}{2\beta}\Big)\lambda\right) \end{aligned}$$

where c, d, β are constants. Since we can always write $p(\mu, \lambda) = p(\mu \mid \lambda)p(\lambda)$, we can find $p(\mu \mid \lambda)$ and $p(\lambda)$ by inspection. We have just shown that $p(\mu \mid \lambda)$ is a Gaussian whose precision is a linear function of λ and that $p(\lambda)$ is a gamma distribution, so the normalized prior takes the form

$$p(\mu, \lambda) = \mathcal{N}(\mu \mid \mu_0, (\beta\lambda)^{-1}) \text{ Gamma}(\lambda \mid a, b) \tag{23}$$

which is called the **Gaussian-Gamma distribution**. Note that this is not simply the product of an independent Gaussian prior over μ and a gamma prior over λ, because the precision of μ is a linear function of λ. The extension of this to multivariate random variables is straightforward.

1.2 Exponential Family of Distributions

The probability distributions so far are contained within the **exponential family** of distributions, which have important properties in common. The exponential family of distributions over $x \in \Omega \subset \mathbb{R}^D$, given parameters η, is defined to be the set of distributions of the form

$$p(x \mid \eta) = h(x)g(\eta) \exp\left(\eta^T u(x)\right) \tag{24}$$

where x may be a scalar or vector, discrete or continuous. Here, η are called the **natural parameters** of the distribution, and $u(x)$ is some function of x. The function $g(\eta)$ can be interpreted as the normalizing coefficient and therefore satisfies

$$g(\eta) \int_{x \in \Omega} h(x) \exp\left(\eta^T u(x)\right) dx = 1 \tag{25}$$

with the integration replaced by a summation if x is discrete.

Now, consider a set of iid data denoted by $\mathbf{X} = \{x_1, \dots, x_n\}$, for which the likelihood function is given by

$$p(X \mid \eta) = \left(\prod_{n=1}^{N} h(x_n)\right) g(\eta)^N \exp\left(\eta^T \sum_{n=1}^{N} u(x_n)\right) \tag{26}$$

Setting the gradient of $\ln p(\mathbf{X} \mid \eta)$ with respect to η to 0, we can the following condition to be satisfied by the maximum likelihood estimator η_{ML}:

$$-\nabla \ln g(\eta_{ML}) = \frac{1}{N} \sum_{n=1}^{N} u(x_n) \tag{27}$$

which can in principle be solved to obtain η_{ML}. The solution for the maximum likelihood estimator depends on the data only through $\sum_n u(x_n)$, which is therefore called the sufficient statistic of this distribution. Therefore, we do not need to store the entire data set itself but its sufficient statistic.

In general, for a given probability distribution $p(\mathbf{X} \mid \eta)$, we can seek a prior that is conjugate to the likelihood function, so that the posterior distribution has the same functional form as the prior. Given that the likelihood function is in the exponential family, there exists a conjugate prior that can be written in the form

$$p(\eta) = p(\eta \mid \chi, \nu) = f(\chi, \nu) g(\eta)^\nu \exp\left(\nu \eta^T \chi\right) \tag{28}$$

where $f(\chi, \nu)$ is a normalization coefficient, and $g(\eta)$ is the same function as the one appearing in the exponential family form of likelihood function. Indeed, multiplying this conjugate with the exponential family likelihood gives

$$p(\eta \mid \mathbf{X}, \chi, \nu) \propto g(\eta)^{\nu+N} \exp\left(\eta^T \left(\sum_{n=1}^{N} u(x_n) + \nu\chi\right)\right) \tag{29}$$

2 Bias Variance Decomposition

Determination of the predictive distribution $p(y \mid x)$ given data $\mathcal{D}$ is the goal of statistical inference, as we have seen. That is, posterior $p(y \mid x, \mathcal{D})$ tells us the distribution of y if we have a new data point x. But after this inference step, we must look now at the **decision step**: we must determine a function $h(x)$ that deterministically predicts a value y, without predictions. That is, we must have some algorithm to make a decision.

Let us zoom out for a better overview. Let $\mathcal{D}$ be our training data of N points. We can assume that each point $(x_i, y_i) \in \mathcal{D}$ was *generated* independently by a joint distribution $p(x, y)$. If we were to get another data point, we would just generate one from the density $p(x, y)$. Usually, we have fixed input data x and knew that the output y given x would be $p(y \mid x)$. But if we loosen our constraint on x, we would get

$$p(x, y) = p(y \mid x)p(x) \tag{30}$$

which states that each data point in $\mathcal{D}$ is gotten by generating a value of x with probability $p(x)$, and then generating a y given this x. Let us also denote $\mathcal{A}$ as our machine learning algorithm, which we can interpret as a function that takes in data $\mathcal{D}$ and outputs the hypothesis function $h_{\mathcal{D}}$.

$$\mathcal{A}(\mathcal{D}) = h_{\mathcal{D}} \tag{31}$$

Then, given that the next new data point (x, y) is generated, we can set our **test error**, or **loss/cost function**, of $h_{\mathcal{D}}$ to be

$$L\big(h_{\mathcal{D}}, (x, y)\big) = \big[h_{\mathcal{D}}(x) - y\big]^2 \tag{32}$$

This loss function basically calculates the inaccuracy of whatever hypothesis function $h_{\mathcal{D}}$ we have on the data (x, y), which in this case is the square of the residual. There can be other types of loss functions, but we will consider the squares loss function for now. Given $h_{\mathcal{D}}$, we can also calculate the expected test error by conditioning over all x, y drawn from P.

$$\text{Expected Test Error given } h_{\mathcal{D}} \implies \mathbb{E}_{x,y,\sim P}\big[(h_{\mathcal{D}}(x) - y)^2\big] = \int_x \int_y (h_{\mathcal{D}}(x) - y)^2 \, p(x, y) \, dy \, dx \tag{33}$$

However, note that we can treat the N data points $\mathcal{D}$ also as a random variable coming from the joint distribution of N P's. Therefore, we can take each possible dataset $\mathcal{D}$, calculate $h_{\mathcal{D}} = \mathcal{A}(\mathcal{D})$ with our algorithm, and average them out to get the expected hypothesis function $\overline{h}$. We can interpret $\overline{h}$ as the "ideal regressor" that we are trying to build, but with limited data $\mathcal{D}$, we can only build $h_{\mathcal{D}}$ that deviates from $\overline{h}$.

$$\overline{h} = \mathbb{E}_{\mathcal{D} \sim P^N}\big[\mathcal{A}(\mathcal{D})\big] = \int_{\mathcal{D}} h_{\mathcal{D}} P(\mathcal{D}) \, d\mathcal{D} \tag{34}$$

So, we can compute the expected error of the *entire algorithm* $\mathcal{A}$ by marginalizing over all x, y given $h_{\mathcal{D}}$ and marginalizing over all $\mathcal{D}$. Remember that $D \sim P^N$ is our training data of N points, and $(x, y) \sim P$ is our $(n+1)$th data point. Therefore, the expected test error of our *algorithm* for the $(n+1)$th data point is

$$\mathbb{E}_{(x,y)\sim P, \mathcal{D}\sim P^N}\big([h_{\mathcal{D}}(x) - y]^2\big) = \int_{\mathcal{D}} \int_x \int_y [h_{\mathcal{D}}(x) - y]^2 \, p(x, y) \, p(\mathcal{D}) \, dy \, dx \, d\mathcal{D} \tag{35}$$

The integral above looks quite intimidating, so let us decompose it. We just have do use a trick where we subtract and add the same term $\overline{h}(x)$.

$$\begin{aligned}
\mathbb{E}_{(x,y),\mathcal{D}}\big([h_{\mathcal{D}}(x) - y]^2\big) &= \mathbb{E}_{(x,y),\mathcal{D}}\big([(h_{\mathcal{D}}(x) - \overline{h}(x)) + (\overline{h}(x) - y)]^2\big) \\
&= \mathbb{E}_{(x,y),\mathcal{D}}\big([h_{\mathcal{D}}(x) - \overline{h}(x)]^2\big) + \mathbb{E}_{(x,y),\mathcal{D}}\big([\overline{h}(x) - y]^2\big) \\
&\quad + 2\mathbb{E}_{(x,y),\mathcal{D}}\big([h_{\mathcal{D}}(x) - \overline{h}(x)]\,[\overline{h}(x) - y]\big)
\end{aligned}$$

But I claim that the last term vanishes. It is easy to see why because

$$\begin{aligned}\mathbb{E}_{(x,y),\mathcal{D}}\left[\left(h_{\mathcal{D}}(x)-\bar{h}(x)\right)\left(\bar{h}(x)-y\right)\right] &= \mathbb{E}_{(x,y)}\left[E_{\mathcal{D}}\left[h_{\mathcal{D}}(x)-\bar{h}(x)\right]\left(\bar{h}(x)-y\right)\right]\\ &= \mathbb{E}_{(x,y)}\left[\left(E_{\mathcal{D}}\left[h_{\mathcal{D}}(x)\right]-\bar{h}(x)\right)\left(\bar{h}(x)-y\right)\right]\\ &= \mathbb{E}_{(x,y)}\left[\left(\bar{h}(x)-\bar{h}(x)\right)\left(\bar{h}(x)-y\right)\right]\\ &= \mathbb{E}_{(x,y)}\left[0\right]\\ &= 0\end{aligned}$$

Therefore, we can see that the expected value of the error of an algorithm consists of two terms: the variance and the second term.

$$\mathbb{E}_{(x,y),\mathcal{D}}\big([h_{\mathcal{D}}(x)-y]^2\big) = \mathbb{E}_{(x,y),\mathcal{D}}\big([h_{\mathcal{D}}(x)-\overline{h}(x)]^2\big) + \mathbb{E}_{(x,y),\mathcal{D}}\big([\overline{h}(x)-y]^2\big) \tag{36}$$

The second term is the expected value of the average prediction minus the y-value of the new point. Now, we do the same trick: Let the expected value of y given x be $\overline{y}(x) = \mathbb{E}_{y\,|\,x}(y) = \int y\,p(y\,|\,x)\,dx$. This function $\overline{y}(x)$ is the ideal regressor predicting y from x. Then, we have

$$\begin{aligned}\mathbb{E}_{x,y}\left[\left(\bar{h}(x)-y\right)^2\right] &= \mathbb{E}_{x,y}\left[\left(\bar{h}(x)-\bar{y}(x)\right)+\left(\bar{y}(x)-y\right)^2\right]\\ &= \underbrace{\mathbb{E}_{x,y}\left[\left(\bar{y}(x)-y\right)^2\right]}_{\text{Noise}} + \underbrace{\mathbb{E}_{x}\left[\left(\bar{h}(x)-\bar{y}(x)\right)^2\right]}_{\text{Bias}^2} + 2\,\mathbb{E}_{x,y}\left[\left(\bar{h}(x)-\bar{y}(x)\right)\left(\bar{y}(x)-y\right)\right]\end{aligned}$$

where the third term vanishes since

$$\begin{aligned}\mathbb{E}_{x,y}\left[\left(\bar{h}(x)-\bar{y}(x)\right)\left(\bar{y}(x)-y\right)\right] &= \mathbb{E}_{x}\left[\mathbb{E}_{y|x}\left[\bar{y}(x)-y\right]\left(\bar{h}(x)-\bar{y}(x)\right)\right]\\ &= \mathbb{E}_{x}\left[\left(\bar{y}(x)-\mathbb{E}_{y|x}\left[y\right]\right)\left(\bar{h}(x)-\bar{y}(x)\right)\right]\\ &= \mathbb{E}_{x}\left[\left(\bar{y}(x)-\bar{y}(x)\right)\left(\bar{h}(x)-\bar{y}(x)\right)\right]\\ &= \mathbb{E}_{x}\left[0\right]\\ &= 0\end{aligned}$$

Therefore, the expected test error is precisely the sum of three things.

$$\underbrace{\mathbb{E}_{x,y,\mathcal{D}}\left[\left(h_{\mathcal{D}}(x)-y\right)^2\right]}_{\text{Expected Test Error}} = \underbrace{\mathbb{E}_{x,\mathcal{D}}\left[\left(h_{\mathcal{D}}(x)-\bar{h}(x)\right)^2\right]}_{\text{Variance}} + \underbrace{\mathbb{E}_{x,y}\left[\left(\bar{y}(x)-y\right)^2\right]}_{\text{Noise}} + \underbrace{\mathbb{E}_{x}\left[\left(\bar{h}(x)-\bar{y}(x)\right)^2\right]}_{\text{Bias}^2}$$

To understand this term a bit deeper, recall the following: The function $\overline{y}(x)$, which outputs the expected value of y given x, is the best possible regressor we can have. There are many different algorithms that we can choose to approximate $\overline{y}(x)$, so let us choose one learning algorithm $\mathcal{A}$. We just feed an arbitrary dataset $\mathcal{D}$ to $\mathcal{A}$, which outputs a hypothesis function $h_{\mathcal{D}}$. But this hypothesis function $h_{\mathcal{D}}$ is really just an approximation of the *ideal* hypothesis function $\overline{h}$, which is the expectation of all hypotheses $h_{\mathcal{D}}$ (i.e. the hypothesis that $\mathcal{A}$ should generate when we feed it an infinite amount of data). So, by feeding $\mathcal{D}$ to $\mathcal{A}$, it generates a hypothesis function $h_{\mathcal{D}}(x)$, which approximates $\overline{h}(x)$, which hopefully is a good estimate of $\overline{y}(x)$.

1. The difference between a generated hypothesis function $h_{\mathcal{D}}(x)$ and the ideal hypothesis that it is trying to estimate according to learning algorithm $\mathcal{A}$ is represented by the variance. The variance term tells us how far each generated hypothesis $h_{\mathcal{D}}$ deviates from the ideal $\overline{h}$.
2. The difference between the ideal hypothesis $\overline{h}(x)$ (according to algorithm $\mathcal{A}$) and the ideal regressor *in general* $\overline{y}(x)$ is captured in the bias term. The bias term tells us how far our algorithm's ideal hypothesis deviates from the expectation of the conditional $p(y\,|\,x)$.

3. The noise term represents the difference between the true value of y and the best possible regressor $\overline{y}(x)$. But since the best we can do is find the expectation of the conditional $p(y \mid x)$, the deviation of the true values y from the mean $\overline{y}$ is simply the noise. For example, if we have $p(y \mid x) = \mathcal{N}\big(y \mid w^T \phi(x), \epsilon\big)$, then the noise would simply be ϵ. If the variance of ϵ is large, the noise would be large. Therefore, the same ideal regressor function $\overline{y}(x)$ would perform worse with a higher noise.

If we are comparing this to the throwing-darts analogy, we can imagine the ideal function $\overline{y}(x)$ to be the bull's eye that we must hit. The different algorithms $\mathcal{A}$ represent different players throwing the darts. When one algorithm (player) is chosen, their vision can be skewed (perhaps their glasses is off), leading them to think that the target $\overline{h}(x)$ is somewhere else. If their target is far away from the bull's eye (i.e. $[\overline{h}(x) - \overline{y}(x)]^2$ is high), then their bias is high. Their skills in darts may just be bad, so even if their vision is good and they have a good sense of where to hit (low bias), for each time they throw the dart (i.e. each time the regressor function $h_{\mathcal{D}}$ is generated from data), it may be very off from their ideal target $\overline{h}(x)$.

Therefore, if you are a data scientist and you find that your regression function is not accurate enough, it is your job to find out whether your bias is too high, your variance is too high, or whether there is too much noise, and fix the proper component. Generally, we would try to minimize this cost function, visualized below.

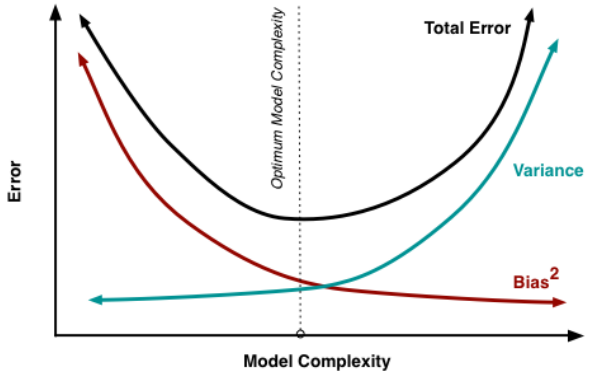

Figure 1: Visualization of the bias-variance tradeoff showing how model complexity affects error components

3 Hierarchical Modeling

Given a training data set $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$ comprised of N pairs of observations with corresponding target variables $\{(x_i, y_i)\}_{i=1}^N$ ($x_i \in \mathbb{R}^D, y_i \in \mathbb{R}$), the goal is to predict the value of y for a new value of x. We first construct a *statistical model* (more explained in next next subsection) by assuming that there exists some function $f(x)$ of some form such that the y_i's have been generated by inputting the x_i's into f, followed by a random residual term. We assume that the data $\mathcal{D}$ has been sampled independently, but this may not always be a justifiable assumption in practice. Under this model, which we denote $\mathcal{M}_i$, we further assume that f can be parameterized by a vector θ, so therefore, we assume that

$$y = f(x, \theta) + \epsilon, \qquad \epsilon \sim \text{Residual}(\beta) \tag{37}$$

where β is some collection of parameters that determine the error function.

- The frequentist perspective reduces this problem to finding the value of θ that maximizes the likelihood. That is, we must find

$$\theta^* = \arg\max_\theta p(\mathcal{D} \mid \theta) = \arg\max_\theta \prod_{i=1}^N p(y_i \mid x_i, \theta) \tag{38}$$

  and claiming that $y = f(x, \theta^*)$ is the function of best fit. This is a quite straightforward (hopefully convex) optimization problem, which can be done in many ways (e.g. batch/sequential gradient descent, solving normal equations, etc.).

- The Bayesian approach attempts to construct a *distribution* of the values of θ. Clearly, this vector θ would be an element in some multidimensional Euclidean space, and we want to define a posterior density $p(\theta \mid \mathcal{D})$ across this space that tells us the probability of θ. Using Bayes rule,

$$p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta)\, p(\theta) \tag{39}$$

  we see that we must define some prior distribution $p(\theta)$ on θ. We can assume that this prior is defined with some distribution

$$\theta \sim \text{Dist}_\theta(\gamma) \tag{40}$$

  where γ is a collection of parameters on θ. Knowing this prior of θ will allow us to get the posterior of $\theta \mid \mathcal{D}$. The not-so-complete Bayesian treatment would treat this γ as a known constant. But note that there is still uncertainty of whether θ comes from $\text{Dist}_\theta(\gamma)$ for one value of γ, compared to another value of γ. This uncertainty requires us to treat γ as now a **hyperparameter**, that is a parameter for the distribution of a parameter, and this distribution of γ, which we can denote

$$\gamma \sim \text{Dist}_\gamma(\xi) \tag{41}$$

  is called a **hyperprior**. We can construct higher and higher level hyperpriors on top of this as much as we want, which will lead to more flexibility in our model (but more computationally expensive). This is known as **hierarchical priors**. Generally, we will only go up to the level of one hyperparameter.

Let us summarize how we would conduct the Bayesian method step by step. We first have to determine how many levels of hierarchical priors we are accounting for. Say that we will treat ξ as a constant, and consider the parameter θ along with its hyperparameter γ. Our goal is to compute the posterior $p(\theta \mid \mathcal{D})$.

1. Since there is uncertainty over the value of θ depending on γ, we can marginalize over γ to get

$$p(\theta \mid \mathcal{D}) = \int p(\theta \mid \mathcal{D}, \gamma)\, p(\gamma \mid \mathcal{D})\, d\gamma \tag{42}$$

   If the situation calls for it, we could also compute the posterior by doing Bayes rule first to get $p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta)\ p(\theta)$, but then we would have to calculate both $p(\mathcal{D} \mid \theta)$ and $p(\theta)$ by marginalizing each over γ, which would lead to complications.

2. To calculate $p(\theta \mid \mathcal{D}, \gamma)$, note that the formula for the posterior density of θ given $\mathcal{D}$ is $p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta)p(\theta)$, where $p(\theta)$ is a density function of θ and parameter γ, which means that $p(\theta \mid \mathcal{D})$ would be a density function of θ and parameter γ. But since γ is fixed, the posterior

$$p(\theta \mid \mathcal{D}, \gamma) \propto p(\mathcal{D} \mid \theta, \gamma)p(\theta \mid \gamma) \tag{43}$$

   is a density function of θ with fixed constant γ. This can be easily calculated because the prior $p(\theta \mid \gamma)$ is of distribution $\text{Dist}_\theta(\gamma)$ and the likelihood $p(\mathcal{D} \mid \theta, \gamma)$ is the product of densities of y given fixed θ.

3. To calculate $p(\gamma \mid \mathcal{D})$, we first use Bayes rule to get

$$p(\gamma \mid \mathcal{D}) \propto p(\mathcal{D} \mid \gamma)\, p(\gamma) \tag{44}$$

   This can be easily calculated because the prior $p(\gamma)$ is of distribution $\text{Dist}_\gamma(\xi)$ of given ξ. The likelihood can be marginalized over θ to get

$$p(\mathcal{D} \mid \gamma) = \int p(\mathcal{D} \mid \theta, \gamma)\, p(\theta \mid \gamma)\, d\theta \tag{45}$$

   where $p(\theta \mid \gamma)$ is a function of θ with given parameter γ, and $p(\mathcal{D} \mid \theta)$ is the product of the individual likelihoods.

But remember that this was all assumed under model $\mathcal{M}_i$, so the posterior density $p(\theta^i \mid \mathcal{D})$ of the θ^i parameterizing our best-fit function is really

$$p(\theta^i \mid \mathcal{D}, \mathcal{M}_i) \tag{46}$$

where we index the parameter of model $\mathcal{M}_i$ to be θ^i, with a superscript (since we may mistake subscript indices to be the components of θ).

References

[Hof09] Peter D. Hoff. *A First Course in Bayesian Statistical Methods*. Springer Publishing Company, Incorporated, 1st edition, 2009.