# Bayesian Basics

Ryan P. Adams

These are notes to help clarify things and create context. Please note that they are **not** a replacement for the readings.

There are a lot of ideas that seem to carry the name *Bayesian* and so it can be unclear sometimes what this word actually means. At a high level, however, it is about being willing to use probability distributions to represent unknown quantities that are not necessarily random. That is, using probability to capture degrees of *belief* in which the uncertainty may be entirely in one's own head or the state of an algorithm. For example, we might have some noisy astronomical information about the question "how many rings does Saturn have?" We might have some evidence supporting some number of rings, but it is noisy and incomplete, so we are uncertain. A Bayesian is willing to place a probability distribution on this quantity and represent it as a random variable, because she is uncertain. A *frequenist* might assert that there is no possibility of repeating a random event that produces a new Saturn with a different number of rings, and so it is inappropriate to consider this a random variable with a probability distribution: there is an unknown truth and we must estimate it. This is a deep philosophical question that has been debated for a long time. In this class, we're going to take it as a given that some kinds of machine learning problems are noisy and uncertain and that it can be useful to reason about these using the calculus of probabilities.

The Bayesian model for machine learning is appealing for a few reasons. First, as I've said, it allows one to represent beliefs in the presence of noise. However, it also allows you to integrate out that uncertainty and account for it when making decisions and predictions from data. It provides a coherent way to balance old data against new data and accumulate more information as it arrives. It also enable one to separate out modeling assumptions from fitting (inference) procedures and separate algorithmic concerns from our inductive biases. Finally, it enables us to handle difficult tasks like model selection in a clear and rigorous way. Being Bayesian is not the only approach to machine learning and statistics, but it can be a nice one for many problems due to these and other properties.

Personally, I like to think of Bayesian inference as a kind of "hypothesis processing machine". Imagine that there is a space of possible (unobserved) states of the world and we'd like to reason about them. Let's imagine that our *a priori* beliefs about the world are captured by a prior distribution $p(\theta)$. Now, we see some data and those data are coupled to these hypotheses via a *likelihood function* $p(\text{data} \mid \theta)$. This likelihood function is a distribution over data, given a state of the world. The environment gives the data to us and we're stuck with it, so we think of likelihood functions as being functions over their parameters; this can be somewhat confusing because those parameters appear behind the vertical bars.[1] In any case, this hypothesis processing machine has

[1] Hard-core frequentists might say that you can't condition on something that isn't a random variable and so therefore likelihood function should be written as a parameterized family of densities like $f_\theta(\text{data})$.

two steps: first, multiply these two function together pointwise as a function of $\theta$; second, normalize so that you get a probability density back on $\theta$. This multiplication penalizes values of $\theta$ that assign low probability to the data, and upweights those that assign high probability to the data.

$$\text{prior } p(\theta) \to \text{multiply by } p(\text{data} \mid \theta) \to \text{divide by } \int p(\text{data} \mid \theta)\, p(\theta)\, d\theta \to \text{posterior } p(\theta \mid \text{data})$$

Bayes' theorem really is that simple:

$$p(\theta \mid \text{data}) = \frac{p(\theta)\, p(\text{data} \mid \theta)}{\int p(\theta')\, p(\text{data} \mid \theta')\, d\theta'}\,.$$

Here I'm using $\theta'$ instead of $\theta$ to make it clearer that this denominator doesn't depend on a specific value of $\theta$, but is an integral over *all* values. It is the normalization constant for this distribution over $\theta$, often called the *marginal likelihood*:

$$p(\text{data}) = \int p(\theta)\, p(\text{data} \mid \theta)\, d\theta\,.$$

## Conjugacy

It is often the case that your prior might have a simple form that you get to choose, but after you multiply it by one or more likelihood functions, it starts to become complicated. This typically means that you can evaluate it pointwise, but only up to a constant because the marginal likelihood integral becomes intractable. There are a variety of methods out there for dealing with this (common) situation, with the two most popular ones being Markov chain Monte Carlo and variational inference. These more advanced techniques are out of the scope of this course, so we will instead focus on the important situations in which the posterior distribution has the same form as the prior distribution. Likelihoods that have this form are generally *exponential family* distributions and the priors that are closed under the corresponding Bayesian updates are called *conjugate priors*. We won't go into too much detail about exponential family distributions in this course, but the basic idea is that these distributions have the form

$$p(\text{data} \mid \boldsymbol{\theta}) \propto \exp\left\{\boldsymbol{\theta}^\mathsf{T} \boldsymbol{T}(\text{data})\right\}\,,$$

that is, the log densities are linear in the parameters. Here I'm imagining that $\boldsymbol{\theta}$ is now a real-valued vector. The vector function $\boldsymbol{T}(\text{data})$ provides the *sufficient statistics*. The book goes into more detail about exponential families and why they are interesting, and CS281 discusses these topics further.

In any case, the reason these kinds of likelihoods are convenient is because if we have a prior that looks like

$$p(\theta) \propto \exp\left\{\boldsymbol{\psi}^\mathsf{T} \boldsymbol{\theta}\right\}\,,$$

then when you multiply it by one of those likelihoods, things inside the exponential function just add:

$$p(\boldsymbol{\theta} \mid \text{data}) \propto \exp\left\{\boldsymbol{\theta}^\mathsf{T} (\boldsymbol{\psi} + \boldsymbol{T}(\text{data}))\right\}\,.$$

I'm skipping a ton of details here about the conjugate prior setup. Bishop 2.4 has a more rigorous and thorough treatment in which these equations have all the other terms to make the math work out correctly with normalization constants and such. Nevertheless, this is the high level idea: make things multiply nicely.

### Example: Beta-Binomial

The Bernoulli distribution is just the "coin flip" distribution with some bias $\rho \in (0, 1)$. We'll say that a heads is 1 and a tails is 0. The probability mass function for the Bernoulli is

$$\Pr(X = x \mid \rho) = \rho^x (1 - \rho)^{1-x}.$$

You can write this as an exponential family using the "natural" parameterization $\theta = \ln\{\rho/(1-\rho)\}$:

$$\begin{aligned}
\rho^x (1-\rho)^{1-x} &= \exp\{x \ln \rho + (1-x)\ln(1-\rho)\} \\
&\propto \exp\{x \ln \rho - x \ln(1-\rho)\} \\
&\propto \exp\{x \ln \frac{\rho}{1-\rho}\}.
\end{aligned}$$

The conjugate prior for the Bernoulli distribution is the beta distribution, which is a density on the interval $(0, 1)$ given by

$$p(\rho \mid \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \rho^{\alpha-1}(1-\rho)^{\beta-1}.$$

Bishop 2.1.1 has nice pictures of the beta distribution and discusses some further properties. Now, in class I sort of threw around "let's imagine you see $J$ heads and $K$ tails", but if you were paying close attention, you know I was skipping over some important pieces. If you have $J$ heads out of $J + K$ tosses, then what you really have is a binomial distribution and there is a binomial coefficient out there in the likelihood:

$$\Pr(J \text{ heads}, K \text{ tails} \mid \rho) = \binom{J+K}{J} \rho^J (1-\rho)^K.$$

The binomial coefficient doesn't effect the math we do for the Bayesian update, however, since it just gets sucked into the normalization constant. To see how this works, we first denote our prior parameters for the beta distribution as $\alpha_0$ and $\beta_0$.

$$\begin{aligned}
p(\rho) &= \text{Beta}(\rho \mid \alpha_0, \beta_0) \\
p(\rho \mid J \text{ heads}, K \text{ tails}, \alpha_0, \beta_0) &\propto \overbrace{\left[\frac{\Gamma(\alpha_0+\beta_0)}{\Gamma(\alpha_0)\Gamma(\beta_0)} \rho^{\alpha_0-1}(1-\rho)^{\beta_0-1}\right]}^{\text{prior}} \times \overbrace{\left[\binom{J+K}{J} \rho^J (1-\rho)^K\right]}^{\text{likelihood}} \\
&\propto \rho^{\alpha_0+J-1}(1-\rho)^{\beta_0+K-1} \\
&= \text{Beta}(\rho \mid \alpha = \alpha_0 + J, \beta = \beta_0 + K).
\end{aligned}$$

So after seeing these flips we get a new beta distribution back out!

## Bayesian Updates for Gaussians with Known Covariance

Next, let's consider a very slightly more complex model: multivariate Gaussian data with known covariance. That is, we're imagining that we have $N$ data in $\mathbb{R}^D$ that have been drawn from a $D$-dimensional Gaussian distribution with *unknown* mean $\boldsymbol{\mu}$ but *known* covariance matrix $\boldsymbol{\Sigma}$. Gaussian distributions are very fundamental and I am going to assume that you've seen them before and are comfortable with them and what covariance matrices are, etc. For a review, see Bishop 2.3. The probability density function for a Gaussian with this parameterization is

$$\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = |2\pi|^{-D/2} |\boldsymbol{\Sigma}|^{-1/2} \exp\left\{ -\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \right\}.$$

This is the likelihood function, in terms of $\boldsymbol{\mu}$. The conjugate prior for $\boldsymbol{\mu}$ in this case is another Gaussian. We'll denote the prior parameters for that Gaussian as $\boldsymbol{m}_0$ and $\boldsymbol{S}_0$. We encounter $N$ data $\{\boldsymbol{x}_n\}_{n=1}^N$ and now we would like the posterior distribution on $\boldsymbol{\mu}$:

$$\begin{aligned} p(\boldsymbol{\mu} \mid \boldsymbol{m}_0, \boldsymbol{S}_0, \boldsymbol{\Sigma}, \{\boldsymbol{x}_n\}_{n=1}^N) &\propto \overbrace{\mathcal{N}(\boldsymbol{\mu} \mid \boldsymbol{m}_0, \boldsymbol{S}_0)}^{\text{prior}} \times \overbrace{\prod_{n=1}^N \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})}^{\text{likelihood}} \\ &\propto \exp\left\{ -\frac{1}{2} (\boldsymbol{\mu} - \boldsymbol{m}_0)^\top \boldsymbol{S}_0^{-1} (\boldsymbol{\mu} - \boldsymbol{m}_0) \right\} \prod_{n=1}^N \exp\left\{ -\frac{1}{2} (\boldsymbol{x}_n - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}) \right\}. \end{aligned}$$

I've thrown out all of the factors that did not involve $\boldsymbol{\mu}$. It's usually convenient to write this all in log space:

$$\begin{aligned} \ln p(\boldsymbol{\mu} \mid \boldsymbol{m}_0, \boldsymbol{S}_0, \boldsymbol{\Sigma}, \{\boldsymbol{x}_n\}_{n=1}^N) &= \text{const} - \frac{1}{2} \left( (\boldsymbol{\mu} - \boldsymbol{m}_0)^\top \boldsymbol{S}_0^{-1} (\boldsymbol{\mu} - \boldsymbol{m}_0) + \sum_{n=1}^N (\boldsymbol{x}_n - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}) \right) \\ &= \text{const} - \frac{1}{2} \left( \boldsymbol{\mu}^\top \boldsymbol{S}_0^{-1} \boldsymbol{\mu} - 2\boldsymbol{\mu}^\top \boldsymbol{S}_0^{-1} \boldsymbol{m}_0 - 2\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1} \sum_{n=1}^N \boldsymbol{x}_n + N \boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} \right). \end{aligned}$$

Here I just expanded the two quadratic forms and stuck terms that don't depend on $\boldsymbol{\mu}$ into the constant out front. I also observed that the $\boldsymbol{x}_n$ only participate in one of the terms. Now we collapse the like terms and write $\bar{\boldsymbol{x}}_N = \frac{1}{N} \sum_n \boldsymbol{x}_n$ for the sample mean of the data:

$$\ln p(\boldsymbol{\mu} \mid \boldsymbol{m}_0, \boldsymbol{S}_0, \boldsymbol{\Sigma}, \{\boldsymbol{x}_n\}_{n=1}^N) = \text{const} - \frac{1}{2} \left( \boldsymbol{\mu}^\top \left( \boldsymbol{S}_0^{-1} + N\boldsymbol{\Sigma}^{-1} \right) \boldsymbol{\mu} - 2\boldsymbol{\mu}^\top \left( \boldsymbol{S}_0^{-1} \boldsymbol{m}_0 + N\boldsymbol{\Sigma}^{-1} \bar{\boldsymbol{x}}_N \right) \right).$$

We now complete the square and write this as a quadratic form, which results in some other things getting baked into the constant. We don't care about any of these constants because we have to normalize later anyway. Remember that the (log) normalization constant is just a number we subtract in log space.

$$\begin{aligned} \ln p(\boldsymbol{\mu} \mid \boldsymbol{m}_0, \boldsymbol{S}_0, \boldsymbol{\Sigma}, \{\boldsymbol{x}_n\}_{n=1}^N) &= \text{const} - \frac{1}{2} (\boldsymbol{\mu} - \boldsymbol{m}_N)^\top \boldsymbol{S}_N^{-1} (\boldsymbol{\mu} - \boldsymbol{m}_N) \\ \boldsymbol{S}_N &= \left( \boldsymbol{S}_0^{-1} + N\boldsymbol{\Sigma}^{-1} \right)^{-1} \\ \boldsymbol{m}_N &= \boldsymbol{S}_N \left( \boldsymbol{S}_0^{-1} \boldsymbol{m}_0 + N\boldsymbol{\Sigma}^{-1} \bar{\boldsymbol{x}}_N \right). \end{aligned}$$

So now, in log space, we have a quadratic form and so we can see we'll wind up with a Gaussian in $\boldsymbol{\mu}$ if we exponentiate and normalize! We can compute the posterior mean and covariance, denoted $\boldsymbol{m}_N$ and $\boldsymbol{S}_N$, respectively, using a little bit of linear algebra.[2] A few things to think about to help get some intuition:

- The more data you get, the bigger $N$ will be and the more relative effect $\boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{\Sigma}^{-1}\bar{\boldsymbol{x}}_N$ will have on $\boldsymbol{S}_N$ and $\boldsymbol{m}_N$, respectively. This feels right because more data means that the prior should be overwhelmed.
- Consider what a strong prior would look like for $\boldsymbol{\mu}$, in the case where the dimensions were independent *a priori*. The matrix $\boldsymbol{S}_0$ would have small positive numbers on the diagonal and zeros off of it. When we took its inverse, it would still be a diagonal matrix, but now the values on the diagonal would be big. These would be "competing" with $N$ to center the posterior mean at $\boldsymbol{m}_0$ instead of $\bar{\boldsymbol{x}}_N$. This also feels right because a strong prior should compete more with the data.
- Note that the posterior covariance depends on the data only through $N$ and not on the actual values of $\boldsymbol{x}_n$. That means that our uncertainty in this case is entirely a function of the number of data. Note that this wouldn't be the case if $\boldsymbol{\Sigma}$ was unknown.

It is a useful exercise to work through this same math where both $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are unknown. In this case, there is still a conjugate prior, but now it is a more complicated distribution called a *Normal-Inverse-Wishart* distribution. The Wishart distribution is a distribution over positive definite matrices that is conjugate to Gaussian likelihoods with unknown covariances. Bishop goes through this in 2.3.6.

## Bayesian Linear Regression

We now have the tools to revisit linear regression in a Bayesian setting. Recall that are data are now *pairs*, $\{\boldsymbol{x}_n, t_n\}_{n=1}^N$. We'll assume that there are some basis function and our inputs become a design matrix $\boldsymbol{\Phi}$ which has $N$ rows and $J$ columns. The targets are real-valued and we stack them into a column vector $\boldsymbol{t} \in \mathbb{R}^N$. Our regression model assumes independent zero-mean Gaussian noise with precision $\beta$. Our weight parameter is a $J$-dimensional $\boldsymbol{w}$ and so we're saying the labels arise as

$$t_n = \boldsymbol{\phi}(\boldsymbol{x}_n)^\mathsf{T}\boldsymbol{w} + \epsilon_n \qquad\qquad \epsilon_n \sim \mathcal{N}(0, \beta^{-1})\,.$$

This becomes a likelihood function for the $n$th datum via

$$p(t_n \mid \boldsymbol{x}_n, \boldsymbol{w}, \beta) = \mathcal{N}(t_n \mid \boldsymbol{\phi}(\boldsymbol{x}_n)^\mathsf{T}\boldsymbol{w}, \beta^{-1})\,.$$

The noise is independent, so we can write the likelihood for all $N$ data as

$$\begin{aligned} p(\boldsymbol{t} \mid \{\boldsymbol{x}_n\}_{n=1}^N, \boldsymbol{w}, \beta) &= \prod_{n=1}^N \mathcal{N}(t_n \mid \boldsymbol{\phi}(\boldsymbol{x}_n)^\mathsf{T}\boldsymbol{w}, \beta^{-1}) \\ &= \mathcal{N}(\boldsymbol{t} \mid \boldsymbol{\Phi}\boldsymbol{w}, \beta^{-1}\mathbb{I}_N)\,. \end{aligned}$$

[2] It's pretty common to use the subscript $N$ to denote posterior parameters. This is a kind of reflection of using the subscript 0 for prior parameter; in the beginning you have zero data and afterward you have $N$ data.

Here we're writing this likelihood as a big multivariate Gaussian rather than a product of univariate ones, but is exactly the same. I'm using the notation $\mathbb{I}_N$ to indicate an $N \times N$ identity matrix. If you find this switch to matrix notation confusing, it might be worth working through it to convince yourself that it is correct. To do Bayesian linear regression, we'll need to put a prior on the weights $\boldsymbol{w}$. The convenient conjugate prior is a Gaussian and, as before, we'll use prior parameters $\boldsymbol{m}_0$ and $\boldsymbol{S}_0$; Bishop does the same in Equation (3.48). We proceed exactly as in the simple Gaussian case and write down the prior and likelihood to get the posterior:

$$p(\boldsymbol{w} \mid \boldsymbol{t}, \boldsymbol{\Phi}, \beta, \boldsymbol{m}_0, \boldsymbol{S}_0) \propto \overbrace{\mathcal{N}(\boldsymbol{w} \mid \boldsymbol{m}_0, \boldsymbol{S}_0)}^{\text{prior}} \times \overbrace{\mathcal{N}(\boldsymbol{t} \mid \boldsymbol{\Phi}\boldsymbol{w}, \beta^{-1}\mathbb{I}_N)}^{\text{likelihood}} .$$

We move to log space and collapse constants, as before:

$$\begin{aligned}
\ln p(\boldsymbol{w} \mid \boldsymbol{t}, \boldsymbol{\Phi}, \beta, \boldsymbol{m}_0, \boldsymbol{S}_0) &= \text{const} - \frac{1}{2}\left((\boldsymbol{w} - \boldsymbol{m}_0)^\mathsf{T}\boldsymbol{S}_0^{-1}(\boldsymbol{w} - \boldsymbol{m}_0) + \beta(\boldsymbol{t} - \boldsymbol{\Phi}\boldsymbol{w})^\mathsf{T}(\boldsymbol{t} - \boldsymbol{\Phi}\boldsymbol{w})\right) \\
&= \text{const} - \frac{1}{2}\left(\boldsymbol{w}^\mathsf{T}\boldsymbol{S}_0^{-1}\boldsymbol{w} - 2\boldsymbol{w}^\mathsf{T}\boldsymbol{S}_0^{-1}\boldsymbol{m}_0 - 2\beta\boldsymbol{w}^\mathsf{T}\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{t} + \beta\boldsymbol{w}^\mathsf{T}\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{\Phi}\boldsymbol{w}\right) .
\end{aligned}$$

Collect the quadratic and linear terms:

$$\ln p(\boldsymbol{w} \mid \boldsymbol{t}, \boldsymbol{\Phi}, \beta, \boldsymbol{m}_0, \boldsymbol{S}_0) = \text{const} - \frac{1}{2}\left(\boldsymbol{w}^\mathsf{T}\left(\boldsymbol{S}_0^{-1} + \beta\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{\Phi}\right)\boldsymbol{w} - 2\boldsymbol{w}^\mathsf{T}\left(\boldsymbol{S}_0^{-1}\boldsymbol{m}_0 + \beta\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{t}\right)\right) .$$

Complete the square:

$$\begin{aligned}
\ln p(\boldsymbol{w} \mid \boldsymbol{t}, \boldsymbol{\Phi}, \beta, \boldsymbol{m}_0, \boldsymbol{S}_0) &= \text{const} - \frac{1}{2}(\boldsymbol{w} - \boldsymbol{m}_N)^\mathsf{T}\boldsymbol{S}_N^{-1}(\boldsymbol{w} - \boldsymbol{m}_N) \\
\boldsymbol{S}_N &= \left(\boldsymbol{S}_0^{-1} + \beta\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{\Phi}\right)^{-1} \\
\boldsymbol{m}_N &= \boldsymbol{S}_N\left(\boldsymbol{S}_0^{-1}\boldsymbol{m}_0 + \beta\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{t}\right) .
\end{aligned}$$

So, it turns out that with Gaussian noise we have a Gaussian posterior on the weights. Now, think about what these parameters would look like if we made the prior very weak and zero mean. A very weak independent prior would mean that $\boldsymbol{S}_0$ was zero off of the diagonal and large positive values on the diagonal, i.e., large variances and large *a priori* uncertainty about $\boldsymbol{w}$. When this matrix is inverted, $\boldsymbol{S}_0^{-1}$ will have zeros off the diagonal and values that are nearly zero on the diagonal. That means that

$$\boldsymbol{S}_N \approx \beta^{-1}(\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{\Phi})^{-1}$$

and $\boldsymbol{S}_0^{-1}\boldsymbol{m}_0$ will be the zero vector. So now the posterior mean will be

$$\boldsymbol{m}_N \approx \overbrace{\beta^{-1}(\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{\Phi})^{-1}}^{\boldsymbol{S}_N}\beta\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{t} = (\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{t} ,$$

which we recognize as both the ordinary least squares and maximum likelihood estimates for $\boldsymbol{w}$. Bishop 3.3 has some very nice figures showing this posterior for simple data. Note that all through this we have assumed that $\beta$ is known. It is of course also possible to infer $\beta$ using an appropriate prior.

### Bayesian Linear Regression Posterior Predictive

We can also compute the posterior predictive in this case. Recall that the posterior predictive is what the model predicts about new data, integrating out the parameters. In this case, that means making a prediction of a new output at a new input location, taking into account all possible values of $\boldsymbol{w}$:

$$\begin{aligned} p(t \mid \boldsymbol{x}, \{\boldsymbol{x}_n, t_n\}_{n=1}^N, \boldsymbol{m}_0, \boldsymbol{S}_0, \beta) &= \int p(t \mid \boldsymbol{x}, \boldsymbol{w}, \beta)\, p(\boldsymbol{w} \mid \{\boldsymbol{x}_n, t_n\}_{n=1}^N, \boldsymbol{m}_0, \boldsymbol{S}_0, \beta)\, d\boldsymbol{w} \\ &= \int \overbrace{\mathcal{N}(t \mid \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{w}, \beta^{-1})}^{\text{predictive distribution}} \overbrace{\mathcal{N}(\boldsymbol{w} \mid \boldsymbol{m}_N, \boldsymbol{S}_N)}^{\text{posterior}}\, d\boldsymbol{w}\,. \end{aligned}$$

There are different ways to do this integral, including the kind of brute-force algebra we've been using. Personally, I like to think it through using some basic properties of the Gaussian distribution. In particular, if you have a Gaussian random variable $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and you apply the linear transformation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{b}$, the resulting distribution on $\boldsymbol{y}$ is also Gaussian with a simple form: $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}, \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^\top)$. This is just saying: *If I have a Gaussian random variable and I perform a linear transformation of it, what is the resulting distribution?* That is relevant here because this is exactly what the integral is computing: draw a random $\boldsymbol{w}$ from the posterior and then linearly transform it with $\phi(\boldsymbol{x})$. There's just one more piece of information we need: when we add two Gaussian random variables of the same dimension, the covariance of their sum is the sum of their covariance matrices. With these two pieces of knowledge, we see that:

$$p(t \mid \boldsymbol{x}, \{\boldsymbol{x}_n, t_n\}_{n=1}^N, \boldsymbol{m}_0, \boldsymbol{S}_0, \beta) = \mathcal{N}(t \mid \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{m}_N, \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}) + \beta^{-1})\,.$$

So the predictive distribution is nice and Gaussian. Bishop 3.3.2 has some nice figures of what this looks like with polynomial basis functions.