Lecture 18: Learning probabilistic models

Roger Grosse

1 Overview

In the first half of the course, we introduced backpropagation, a technique we used to train neural nets to minimize a variety of cost functions. One of the cost functions we discussed was cross-entropy, which encourages the network to learn to predict a probability distribution over the targets. This was our first glimpse into probabilistic modeling. But probabilistic modeling is so important that we're going to spend almost the last third of the course on it. This lecture introduces some of the key principles.

This lecture and the next one aren't about neural nets. Instead, they'll introduce the principles of probabilistic modeling in as simple a setting as possible. Then, starting next week, we're going to apply these principles in the context of neural nets, and this will result in some very powerful models.

1.1 Learning goals

- Know some terminology for probabilistic models: likelihood, prior distribution, posterior distribution, posterior predictive distribution, i.i.d. assumption, sufficient statistics, conjugate prior
- Be able to learn the parameters of a probabilistic model using maximum likelihood, the full Bayesian method, and the maximum a-posteriori approximation.
- Understand how these methods are related to each other. Understand why they tend to agree in the large data regime, but can often make very different predictions in the small data regime.

2 Maximum likelihood

The first method we'll cover for fitting probabilistic models is maximum likelihood. In addition to being a useful method in its own right, it will also be a stepping stone towards Bayesian modeling.

Let's begin with a simple example: we have flipped a particular coin 100 times, and it landed heads $N_H = 55$ times and tails $N_T = 45$ times. We want to know the probability that it will come up heads if we flip it again. We formulate the probabilistic model:

> The behavior of the coin is summarized with a parameter θ, the probability that a flip lands heads (H). The flips $\mathcal{D} = \left(x^{(1)}, \ldots, x^{(100)}\right)$ are independent Bernoulli random variables with parameter θ.

(In general, we will use $\mathcal{D}$ as a shorthand for all the observed data.) We say that the individual flips are **independent and identically distributed (i.i.d.)**; they are independent because one outcome does not influence any of the other outcomes, and they are identically distributed because they all follow the same distribution (i.e. a Bernoulli distribution with parameter θ).

We now define the **likelihood function** $L(\theta)$, which is the probability of the observed data, as a function of θ. In the coin example, the likelihood is the probability of the *particular* sequence of H's and T's being generated:

$$L(\theta) = p(\mathcal{D}) = \theta^{N_H}(1-\theta)^{N_T}. \tag{1}$$

Note that L is a function of the model parameters (in this case, θ), not the observed data.

This likelihood function will generally take on extremely small values; for instance, $L(0.5) = 0.5^{100} \approx 7.9 \times 10^{-31}$. Therefore, in practice we almost always work with the **log-likelihood function**,

$$\ell(\theta) = \log L(\theta) = N_H \log\theta + N_T \log(1-\theta). \tag{2}$$

For our coin example, $\ell(0.5) = \log 0.5^{100} = 100 \log 0.5 = -69.31$. This is a much easier value to work with.

In general, we would expect good choices of θ to assign high likelihood to the observed data. This suggests the **maximum likelihood criterion**: choose the parameter θ which maximizes $\ell(\theta)$. If we're lucky, we can do this analytically by computing the derivative and setting it to zero. (More precisely, we find *critical points* by setting the derivative to zero. We check which of the critical points, or boundary points, has the largest value.) Let's try this for the coin example:

$$\begin{aligned}\frac{\mathrm{d}\ell}{\mathrm{d}\theta} &= \frac{\mathrm{d}}{\mathrm{d}\theta}\left(N_H \log\theta + N_T \log(1-\theta)\right) \\ &= \frac{N_H}{\theta} - \frac{N_T}{1-\theta}\end{aligned} \tag{3}$$

Setting this to zero, we find the maximum likelihood estimate

$$\hat{\theta}_{\mathrm{ML}} = \frac{N_H}{N_H + N_T}, \tag{4}$$

i.e. the maximum likelihood estimate is simply the fraction of flips that came up heads. (We put a hat over the parameter to emphasize that it's an estimate.) Hopefully this seems like a sensible guess for θ. Now let's look at some more examples.

Example 1. Suppose we are interested in modeling the distribution of temperatures in Toronto in March. Here are the high temperatures, in Celsius, from the first week of March 2014:

$$\text{-2.5} \quad \text{-9.9} \quad \text{-12.1} \quad \text{-8.9} \quad \text{-6.0} \quad \text{-4.8} \quad \text{2.4}$$

Call these observations $x^{(1)}, \ldots, x^{(N)}$, where $N = 7$. In order to formulate a probabilistic model, we first choose a parametric form for the distribution over temperatures. Often we choose a Gaussian distribution, not because we believe it's an especially good model, but because it makes the computations easy. So let's assume the temperatures are drawn from a Gaussian distribution with unknown mean μ and known standard deviation $\sigma = 5$. Our likelihood function is given by:

$$\begin{aligned}
\ell(\mu) &= \log \prod_{i=1}^{N} \left[\frac{1}{\sqrt{2\pi} \cdot \sigma} \exp\left(-\frac{(x^{(i)} - \mu)^2}{2\sigma^2} \right) \right] \\
&= \sum_{i=1}^{N} \log \left[\frac{1}{\sqrt{2\pi} \cdot \sigma} \exp\left(-\frac{(x^{(i)} - \mu)^2}{2\sigma^2} \right) \right] \\
&= \sum_{i=1}^{N} -\frac{1}{2} \log 2\pi - \log \sigma - \frac{(x^{(i)} - \mu)^2}{2\sigma^2} \qquad (5)
\end{aligned}$$

Since μ can take any possible real value, the maximum must occur at a critical point, so let's look for critical points. Setting the derivative to 0,

$$\begin{aligned}
0 = \frac{\mathrm{d}\ell}{\mathrm{d}\mu} &= \frac{1}{2\sigma^2} \sum_{i=1}^{N} \frac{\mathrm{d}}{\mathrm{d}\mu} (x^{(i)} - \mu)^2 \\
&= -\frac{1}{\sigma^2} \sum_{i=1}^{N} x^{(i)} - \mu \qquad (6)
\end{aligned}$$

Therefore, $\sum_{i=1}^{N} x^{(i)} - \mu = 0$, and solving for μ, we get $\mu = \frac{1}{N} \sum_{i=1}^{N} x^{(i)}$. The maximum likelihood estimate of the mean of a normal distribution is simply the mean of the observed values, or the **empirical mean**. Plugging in our temperature data, we get $\hat{\mu}_{\text{ML}} = -5.97$.

Example 2. In the last example, we pulled the standard deviation $\sigma = 5$ out of a hat. Really we'd like to learn it from data as well. Let's add it as a parameter to the model. The likelihood function is the same as before, except now it's a function of both μ and σ, rather than just μ. To maximize a function

of two variables, we find critical points by setting the *partial* derivatives to 0. In this case,

$$0 = \frac{\partial \ell}{\partial \mu} = -\frac{1}{\sigma^2} \sum_{i=1}^{N} x^{(i)} - \mu \tag{7}$$

$$\begin{aligned} 0 = \frac{\partial \ell}{\partial \sigma} &= \frac{\partial}{\partial \sigma} \left[\sum_{i=1}^{N} -\frac{1}{2} \log 2\pi - \log \sigma - \frac{1}{2\sigma^2} (x^{(i)} - \mu)^2 \right] \\ &= \sum_{i=1}^{N} -\frac{1}{2} \frac{\partial}{\partial \sigma} \log 2\pi - \frac{\partial}{\partial \sigma} \log \sigma - \frac{\partial}{\partial \sigma} \frac{1}{2\sigma} (x^{(i)} - \mu)^2 \\ &= \sum_{i=1}^{N} 0 - \frac{1}{\sigma} + \frac{1}{\sigma^3} (x^{(i)} - \mu)^2 \\ &= -\frac{N}{\sigma} + \frac{1}{\sigma^3} \sum_{i=1}^{N} (x^{(i)} - \mu)^2 \end{aligned} \tag{8}$$

From the first equality, we find that $\hat{\mu}_{\text{ML}} = \frac{1}{N} \sum_{i=1}^{N} x^{(i)}$ is the empirical mean, just as before. From the second inequality, we find $\hat{\sigma}_{\text{ML}} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x^{(i)} - \mu)^2}$. In other words, $\hat{\sigma}_{\text{ML}}$ is simply the empirical standard deviation. In the case of the Toronto temperatures, we get $\hat{\mu}_{\text{ML}} = -5.97$ (as before) and $\hat{\sigma}_{\text{ML}} = 4.55$.

Example 3. We've just seen two examples where we could obtain the exact maximum likelihood solution analytically. Unfortunately, this situation is the exception rather than the rule. Let's consider how to compute the maximum likelihood estimate of the parameters of the gamma distribution, whose PDF is defined as

$$p(x) = \frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx}, \tag{9}$$

where $\Gamma(a)$ is the gamma function, which is a generalization of the factorial function to continuous values.[1] The model parameters are a and b, both of which must take positive values. The log-likelihood, therefore, is

$$\begin{aligned} \ell(a, b) &= \sum_{i=1}^{N} a \log b - \log \Gamma(a) + (a - 1) \log x^{(i)} - b x^{(i)} \\ &= Na \log b - N \log \Gamma(a) + (a - 1) \sum_{i=1}^{N} \log x^{(i)} - b \sum_{i=1}^{N} x^{(i)}. \end{aligned} \tag{10}$$

[1] The definition is $\Gamma(t) = \int_0^\infty x^{t-1} e^{-x} \, \mathrm{d}x$, but we're never going to use the definition in this class.

Most scientific computing environments provide a function which computes $\log \Gamma(a)$. In SciPy, for instance, it is `scipy.special.gammaln`.

To maximize the log-likelihood, we're going to use gradient ascent, which is just like gradient descent, except we move uphill rather than downhill. To derive the update rules, we need the partial derivatives:

$$\frac{\partial \ell}{\partial a} = N \log b - N \frac{\mathrm{d}}{\mathrm{d}a} \log \Gamma(a) + \sum_{i=1}^{N} \log x^{(i)} \tag{11}$$

$$\frac{\partial \ell}{\partial b} = N \frac{a}{b} - \sum_{i=1}^{N} x^{(i)}. \tag{12}$$

Our implementation of gradient ascent, therefore, consists of computing these derivatives, and then updating $a \leftarrow a + \alpha \frac{\partial \ell}{\partial a}$ and $b \leftarrow b + \alpha \frac{\partial \ell}{\partial b}$, where α is the learning rate. Most scientific computing environments provide a function to compute $\frac{\mathrm{d}}{\mathrm{d}a} \log \Gamma(a)$; for instance, it is `scipy.special.digamma` in SciPy.

Here are some observations about these examples:

- In each of these examples, the log-likelihood function ℓ decomposed as a sum of terms, one for each training example. This results from our independence assumption. Because different observations are independent, the likelihood decomposes as a product over training examples, so the log-likelihood decomposes as a sum.

- The derivatives worked out nicely because we were dealing with log-likelihoods. Try taking derivatives of the likelihood function $L(\theta)$, and you'll see that they're much messier.

- All of the log-likelihood functions we looked at wound up being expressible in terms of certain **sufficient statistics** of the dataset, such as $\sum_{i=1}^{N} x^{(i)}$, $\sum_{i=1}^{N} [x^{(i)}]^2$, or $\sum_{i=1}^{N} \log x^{(i)}$. When we're fitting the maximum likelihood solution, we can forget the data itself and just remember the sufficient statistics. This doesn't happen for all of our models; for instance, it didn't happen when we fit the neural language model in Assignment 1. However, it does happen for many of the distributions commonly used in practice.[2]

- We made a lot of questionable assumptions in formulating these models. E.g., we assumed that temperatures on different days were independent; in practice, the temperature tomorrow is likely to be similar to the temperature today. This is also true of models we fit previously; e.g., the Markov assumption we used to justify our neural

[2] If you're interested in learning more, the families of distributions whose likelihoods can be written in terms of sufficient statistics are known as *exponential families*.

language model is clearly bogus. If this were a statistics class, we'd talk about ways to test your modeling assumptions. But because this is a machine learning class, we'll throw caution to the wind and fit models that we know are wrong. Hopefully they'll still be good enough that they can make sensible predictions (in the supervised setting) or reveal interesting structure (in the unsupervised setting).

2.1 Beware of data sparsity

Maximum likelihood is a very powerful technique, but it has a pitfall: if you have too little training data, it can seriously overfit. The most severe pathology is when it assigns probability 0 to things that were never seen in the training set, but which still might actually happen. For instance, suppose we flip a coin twice, and it lands H both times. The maximum likelihood estimate of θ, the probability of H, would be 1. But this is pretty extreme — effectively we're considering it impossible that the coin will ever come up T! This problem is known as **data sparsity**.

This problem isn't so different in principle from examples of overfitting which we discussed for other loss functions. We would like our model to **generalize** well to data it hasn't seen before, i.e. assign the new data high likelihood. We can measure the generalization performance by holding out a separate test set, and measuring the log-likelihood on this test set at the very end. (As before, if we want to choose model hyperparamters, we'd hold out a separate validation set.) In our coin example, if we choose $\theta = 1$ and the coin comes up T even *a single time* in the test set, this will give us a test log-likelihood of $-\infty$. Clearly it's a bad idea to assign any outcome probability 0 if it might ever occur.

Last week, we talked about regularization as a way to attenuate overfitting. Would that work here? One naive approach would be to add an L_2 penalty, $-\frac{1}{2}\theta^2$, to the objective function. (We subtract the penalty since we're maximizing.) But this isn't quite what we want: it would allow (in fact, encourage) the degenerate solution $\theta = 0$. Instead, let's look at Bayesian techniques for parameter estimation. These techniques will turn out to be closely related to regularization.

3 Bayesian parameter estimation

In the maximum likelihood approach, the observations (i.e. the x_i's) were treated as random variables, but the model parameters were not. In the Bayesian approach, we treat the parameters as random variables as well. We define a model for the joint distribution $p(\boldsymbol{\theta}, \mathcal{D})$ over parameters $\boldsymbol{\theta}$ and data $\mathcal{D}$. (In our coin example, $\boldsymbol{\theta}$ would be the probability of H, and $\mathcal{D}$ would be the sequence of 100 flips that we observed.) Then we can perform the usual operations on this joint distribution, such as marginalization and conditioning.

In order to define this joint distribution, we need two things:

- A distribution $p(\boldsymbol{\theta})$, known as the **prior distribution**. It's called the prior because

it's supposed to encode your "prior beliefs," i.e. everything you believed about the parameters *before* looking at the data. In practice, we normally choose priors to be computationally convenient, rather than based on any sort of statistical principle. More on this later.

- The **likelihood** $p(\mathcal{D} \mid \boldsymbol{\theta})$, the probability of the observations given the parameters, just like in maximum likelihood.

Bayesians are primarily interested in computing two things:

- The **posterior distribution** $p(\boldsymbol{\theta} \mid \mathcal{D})$. This corresponds to our beliefs about the parameters *after* observing the data. In general, the posterior distribution can be computed using Bayes' Rule:

$$p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{p(\boldsymbol{\theta})p(\mathcal{D} \mid \boldsymbol{\theta})}{\int p(\boldsymbol{\theta}')p(\mathcal{D} \mid \boldsymbol{\theta}') \, \mathrm{d}\boldsymbol{\theta}'}. \tag{13}$$

  However, we don't normally compute the denominator directly. Instead we work with unnormalized distributions as long as possible, and normalize only when we need to. Bayes' Rule can therefore be written in a more succinct form, using the symbol $\propto$ to denote "proportional to":

$$p(\boldsymbol{\theta} \mid \mathcal{D}) \propto p(\boldsymbol{\theta})p(\mathcal{D} \mid \boldsymbol{\theta}). \tag{14}$$

- The **posterior predictive distribution** $p(\mathcal{D}' \mid \mathcal{D})$, which is the distribution over future observables given past observations. For instance, given that we've observed 55 H's and 45 T's, what's the probability that the next flip will land H? We can compute the posterior predictive distribution by computing the posterior over $\boldsymbol{\theta}$ and then marginalizing out $\boldsymbol{\theta}$:

$$p(\mathcal{D}' \mid \mathcal{D}) = \int p(\boldsymbol{\theta} \mid \mathcal{D})p(\mathcal{D}' \mid \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}. \tag{15}$$

3.1 The full Bayesian approach

Let's figure out the posterior distribution and posterior predictive distribution for our coin example. We've already specified the likelihood, so it remains to specify the prior. One option is to use an **uninformative prior**, which assumes as little as possible about the problem. In the case of the coin, this might correspond to the uniform distribution $p(\theta) = 1$. (There is no single recipe for choosing an uninformative prior; statisticians have a few different recipes which often, but not always, agree with each other.)

Alternatively, we can draw upon our lifetime of experience flipping coins. Most coins tend to be fair, i.e. the come up heads around 50% of the time. So perhaps our prior should make $\theta = 0.5$ more likely. There are a lot of distributions which can do this, but a

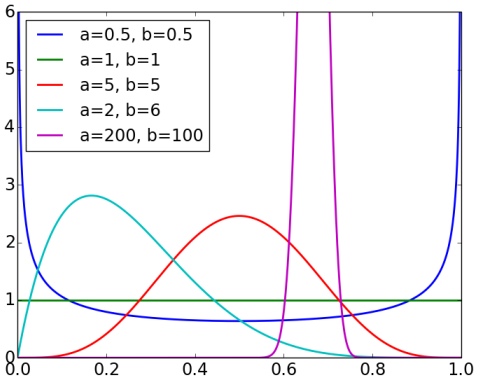

Figure 1: The PDF of the beta distribution for various values of the parameters a and b. Observe that the distribution becomes more peaked as a and b become large, and the peak is near $a/(a+b)$.

particularly useful one is the **beta distribution**, parameterized by $a, b >$ and defined as:

$$p(\theta; a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \theta^{a-1}(1-\theta)^{b-1}. \tag{16}$$

This distribution is visualized in Figure 1. Why did we choose the beta distribution, of all things? Once we work through the example, we'll see that it's actually pretty convenient. Observe that the first term (with all the Γ's) is just a normalizing constant, so it doesn't depend on θ. In most of our computations, we'll only need to work with unnormalized distributions (i.e. ones which don't necessarily integrate to 1), so we can drop the cumbersome normalizing constant and write

$$p(\theta; a, b) \propto \theta^{a-1}(1-\theta)^{b-1}. \tag{17}$$

A few values are plotted in Figure 1. From these plots, we observe a few things:

- The distribution appears to be centered around $a/(a+b)$. In fact, it's possible to show that if $\theta \sim \text{Beta}(a, b)$, then $\mathbb{E}[\theta] = a/(a+b)$.
- The distribution becomes more peaked for larger values of a and b.
- The values $a = b = 1$ correspond to the uniform distribution. Therefore, we can simply treat the uniform prior as a special case of the beta prior.

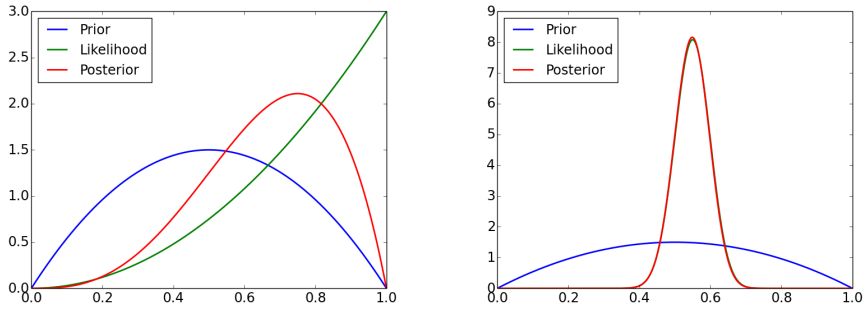

Figure 2: Plots of the prior, likelihood, and posterior for the coin flip example, with the prior Beta$(2, 2)$. **(Left)** Small data setting, $N_H = 2$, $N_T = 0$. **(Right)** Large data setting, $N_H = 55$, $N_T = 45$. In this case, the data overwhelm the prior, so the posterior is determined by the likelihood. Note: for visualization purposes, the likelihood function is normalized to integrate to 1, since otherwise it would be too small to see.

Now let's compute the posterior and posterior predictive distributions. When we plug in our prior and likelihood terms for the coin example, we get:

$$p(\boldsymbol{\theta} \mid \mathcal{D}) \propto p(\boldsymbol{\theta}) p(\mathcal{D} \mid \boldsymbol{\theta}) \tag{18}$$

$$\propto \left[\theta^{a-1}(1-\theta)^{b-1}\right]\left[\theta^{N_H}(1-\theta)^{N_T}\right] \tag{19}$$

$$= \theta^{a-1+N_H}(1-\theta)^{b-1+N_T}. \tag{20}$$

But this is just a beta distribution with parameters $N_H + a$ and $N_T + b$. Let's stop and check if our answer makes sense. As we observe more flips, N_H and N_T both get larger, and the distribution becomes more peaked around a particular value. Furthermore, the peak of the distribution will be near $N_H/(N_H + N_T)$, our maximum likelihood solution. This reflects the fact that the more data we observe, the less uncertainty there is about the parameter, and the more the likelihood comes to dominate. We say that the **data overwhelm the prior**.

We now compute the posterior predictive distribution over the next flip x':

$$\begin{aligned}
\theta_{\text{pred}} &= \Pr(x' = H \mid \mathcal{D}) \\
&= \int p(\theta \mid \mathcal{D}) \Pr(x' = H \mid \theta) \, \mathrm{d}\theta \\
&= \int \text{Beta}(\theta; N_H + a, N_T + b) \cdot \theta \, \mathrm{d}\theta \\
&= \mathbb{E}_{\text{Beta}(\theta; N_H + a, N_T + b)}[\theta] \\
&= \frac{N_H + a}{N_H + N_T + a + b},
\end{aligned} \tag{21}$$

where the last line follows from the formula for the expectation of a beta random variable. Again, let's check if this is reasonable. If N_H and N_T are large, this is approximately $N_H/(N_H+N_T)$, so our predictions are similar to the ones we get using maximum likelihood. However, if N_H and N_T are relatively small, then the predictive distribution is **smoothed**, i.e. less extreme than the maximum likelihood one. The value θ_{pred} is somewhere in between the prior and the maximum likelihood estimate.

OK, back to an earlier question. Where did our choice of prior come from? The key thing to notice is Eqn 20, where the posterior wound up belonging to the same family of distributions as the prior. Why did this happen? Let's compare the formulas for the beta distribution and the likelihood:

$$p(\theta) = \text{Beta}(\theta; a, b) \propto \theta^{a-1}(1-\theta)^{b-1} \tag{22}$$

$$p(\mathcal{D} \mid \theta) \propto \theta^{N_H}(1-\theta)^{N_T} \tag{23}$$

In other words, the prior was chosen to have the same functional form as the likelihood.[3] Since we multiply these expressions together to get the (unnormalized) posterior, the posterior will also have this functional form. A prior chosen in this way is called a **conjugate prior**. In this case, the parameters of the prior distribution simply got added to the observed counts, so they are sometimes referred to as **pseudo-counts**.

Let's look at some more examples.

> **Example 4.** Let's return to our problem of estimating the mean temperature in Toronto, where our model assumes a Gaussian with unknown mean μ and known standard deviation $\sigma = 5$. The first task is to choose a conjugate prior. In order to do this, let's look at the PMF of a single data point:
>
> $$p(x \mid \mu) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{24}$$

[3] The $\propto$ notation obscures the fact that the normalizing constants in these two expressions may be completely different, since $p(\theta)$ is a distribution over parameters, while $p(\mathcal{D} \mid \theta)$ is a distribution over observed data. In this example, the latter normalizing constant happens to be 1, but that won't always be the case.

If we look at this as a function of μ (rather than x), we see that it's still a Gaussian! This should lead us to conjecture that the conjugate prior for a Gaussian is a Gaussian. Let's try it and see if it works.

Our prior distribution will be a Gaussian distribution with mean μ_{pri} and standard deviation σ_{pri}. The posterior is then given by:

$$
\begin{aligned}
p(\mu \mid \mathcal{D}) &\propto p(\mu) p(\mathcal{D} \mid \mu) \\
&= \left[\frac{1}{\sqrt{2\pi}\sigma_{\text{pri}}} \exp\left(-\frac{(\mu-\mu_{\text{pri}})^2}{2\sigma_{\text{pri}}^2}\right)\right]\left[\prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x^{(i)}-\mu)^2\right)\right] \\
&\propto \exp\left(-\frac{(\mu-\mu_{\text{pri}})^2}{2\sigma_{\text{pri}}^2} - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x^{(i)}-\mu)^2\right) \\
&\propto \exp\left(-\frac{\mu^2}{2\sigma_{\text{pri}}^2} + \frac{\mu_{\text{pri}}\mu}{\sigma_{\text{pri}}^2} - \frac{\mu_{\text{pri}}^2}{2\sigma_{\text{pri}}^2} - \frac{1}{2\sigma^2}\sum_{i=1}^{N}[x^{(i)}]^2 + \frac{1}{\sigma^2}\sum_{i=1}^{N} x^{(i)}\mu - \frac{N}{2\sigma^2}\mu^2\right) \\
&= \exp\left(-\frac{\mu_{\text{pri}}^2}{2\sigma_{\text{pri}}^2} - \frac{1}{2\sigma^2}\sum_{i=1}^{N}[x^{(i)}]^2 + \left[\frac{\mu_{\text{pri}}}{\sigma_{\text{pri}}^2} - \frac{\sum_{i=1}^{N} x^{(i)}}{\sigma^2}\right]\mu - \frac{1}{2}\left[\frac{1}{\sigma_{\text{pri}}^2} + \frac{N}{\sigma^2}\right]\mu^2\right) \\
&\propto \exp\left(-\frac{(\mu-\mu_{\text{post}})^2}{\sigma_{\text{post}}^2}\right),
\end{aligned}
$$

where

$$
\sigma_{\text{post}} = \frac{1}{\sqrt{\frac{1}{\sigma_{\text{pri}}^2} + \frac{N}{\sigma^2}}} \tag{25}
$$

$$
\mu_{\text{post}} = \frac{\frac{1}{\sigma_{\text{pri}}^2}\mu_{\text{pri}} + \frac{N}{\sigma^2}\frac{1}{N}\sum_{i=1}^{N} x^{(i)}}{\frac{1}{\sigma_{\text{pri}}^2} + \frac{N}{\sigma^2}}. \tag{26}
$$

The last step uses a technique called *completing the square.* You've probably done this before in a probability theory class. So the posterior distribution is a Gaussian with mean μ_{post} and standard deviation σ_{post}.

The formulas are rather complicated, so let's break them apart. First look how σ_{post} changes if we vary the prior or the data.

- As we increase the number of observations N, the denominator gets larger, so σ_{post} gets smaller. This should be intuitive: as we observe more data, the posterior gets more peaked, which corresponds to the posterior standard deviation decreasing.

- What if we increase the prior standard deviation σ_{pri} or the observation standard deviation σ? Then the denominator gets smaller, which means σ_{post} gets larger. This should be intuitive, because increasing the uncertainty in either the prior or the likelihood should increase the uncertainty in the posterior.

Now let's look at the formula for μ_{post}. It takes the form of a **weighted average** of the prior mean μ_{pri} and the maximum likelihood mean $\frac{1}{N}\sum_{i=1}^{N} x^{(i)}$. By weighted average, I mean something of the form

$$\frac{ar + bs}{a + b}.$$

where the weights a and b are both positive. This is a weighted average of r and s; if a is larger, it is closer to r, while if b is larger, it is closer to s. For μ_{post}, the weights for the prior mean and maximum likelihood mean are $1/\sigma^2_{\text{pri}}$ and N/σ^2, respectively. Let's see what happens if we change the problem.

- As the number of observations N gets larger, the weighted average becomes closer and closer to the maximum likelihood estimate. This should make sense: as we get more information, our prior beliefs become less relevant.
- If we increase the prior standard deviation σ_{pri}, then $1/\sigma^2_{\text{pri}}$ gets smaller, so there is less weight on the prior. On the other hand, if we increase the standard deviation σ of the observations, then N/σ^2 gets smaller, and there is less weight on the maximum likelihood solution. In other words, whichever of the two terms we are more confident about should count for more.

Observe that $\sum_{i=1}^{N} x^{(i)}$ is a sufficient statistic, since it is the only thing we need to remember about the data in order to compute the posterior.

Finally, let's take a look at the posterior predictive distribution. We compute this as

$$\begin{aligned} p(x' \mid \mathcal{D}) &= \int p(\mu \mid \mathcal{D}) p(x' \mid \mu) \, \mathrm{d}\mu \\ &= \int \text{Gaussian}(\mu; \mu_{\text{post}}, \sigma_{\text{post}}) \text{Gaussian}(x'; \mu, \sigma) \, \mathrm{d}\mu \\ &= \text{Gaussian}(x'; \mu_{\text{post}}, \sqrt{\sigma^2_{\text{post}} + \sigma^2}) \end{aligned} \tag{27}$$

The last step uses the formula for the convolution of two Gaussian distributions. Now let's see how it behaves.

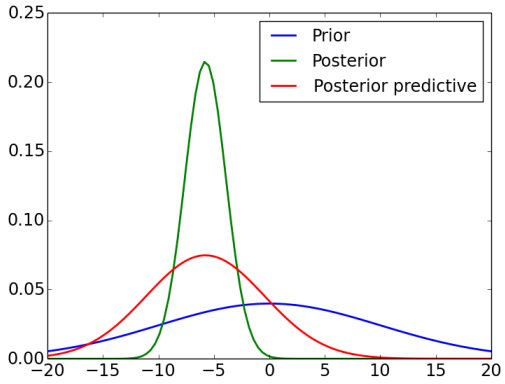

Figure 3: The prior, posterior, and posterior predictive distributions for the Toronto temperatures example.

- When there are no observations (i.e. $N = 0$), then μ_{post} and σ_{post} are the prior mean and standard deviation. The predictive distribution is centered at μ_{post}, but more spread out than the prior.
- When N is very large, the mean of the predictive distribution is close to the maximum likelihood mean, and the standard deviation is very close to σ. In other words, it makes almost the same predictions as the maximum likelihood estimate.

The prior, posterior, and posterior predictive distributions are all shown in Figure 3.

For both the coin and Gaussian examples, the posterior predictive distribution had the same parametric form as the model. (I.e., it was a Bernoulli distribution for the coin model, and a Gaussian distribution for the Gaussian model.) This does *not* happen in general; often the posterior predictive distribution doesn't have a convenient form, which is part of what makes the full Bayesian approach difficult to apply.

3.2 The difficulty of the full Bayesian approach

We've seen two different ways to learn the parameters of a probabilistic model. Maximum likelihood is based on optimization, while the full Bayesian approach is based on computing integrals. In either case, for some of the commonly used distributions, we can derive a closed-form solution. However, for many important models (such as multilayer neural nets), there's no closed-form solution. As we saw in Example 3, if we can't find a closed form, we can still maximize the log-likelihood using gradient ascent.

But for the Bayesian approach, we need to compute an *integral* in order to marginalize out the model parameters. If we only have a few parameters, we can do this using numerical quadrature methods. Unfortunately, these methods are exponential in the number of variables being integrated out. If we're trying to fit a neural net with thousands (or even millions) of parameters, this is completely impractical. There are other methods for integration which perform well in high dimensional spaces; we'll discuss one such set of techniques, called Markov chain Monte Carlo, later in the course. However, integration still tends to be a much more difficult problem than optimization, so if possible we would like to formulate our learning algorithms in terms of optimization. Let's now look at the maximum a-posteriori (MAP) approximation, a way of converting the integration problem into an optimization problem.

3.3 Maximum a-posteriori (MAP) approximation

We worked through two examples of the full Bayesian approach: Bernoulli and Gaussian models. In both cases, we saw that as more data is observed, the posterior distribution becomes more and more peaked around a single value. This suggests that maybe we can get away with summarizing the posterior with a single point estimate. The **maximum a-posteriori (MAP) approximation** chooses the parameters which are most likely under the posterior, i.e.

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} \; p(\boldsymbol{\theta} \mid \mathcal{D}) \tag{28}$$

$$= \arg\max_{\boldsymbol{\theta}} \; p(\boldsymbol{\theta}, \mathcal{D}) \tag{29}$$

$$= \arg\max_{\boldsymbol{\theta}} \; p(\boldsymbol{\theta})\, p(\mathcal{D} \mid \boldsymbol{\theta}) \tag{30}$$

$$= \arg\max_{\boldsymbol{\theta}} \; \log p(\boldsymbol{\theta}) + \log p(\mathcal{D} \mid \boldsymbol{\theta}) \tag{31}$$

Observe that maximizing $\log p(\mathcal{D} \mid \boldsymbol{\theta})$ is equivalent to maximum likelihood estimation, so the only difference between MAP and ML is the addition of the prior term $\log p(\boldsymbol{\theta})$. The prior is therefore somewhat analogous to a regularizer. In fact, if $p(\boldsymbol{\theta})$ is a Gaussian distribution centered at 0, you get L2 regularization!

Example 5. Let's return to our coin flip example. The joint probability is given by:

$$\begin{aligned}
\log p(\theta, \mathcal{D}) &= \log p(\theta) + \log p(\mathcal{D} \mid \theta) \\
&= \text{const} + (a-1)\log\theta + (b-1)\log(1-\theta) + N_H \log\theta + N_T \log(1-\theta) \\
&= \text{const} + (N_H + a - 1)\log\theta + (N_T + b - 1)\log(1-\theta) \qquad (32)
\end{aligned}$$

(Here, const is a shorthand for terms which don't depend on θ.) Let's maximize this by finding a critical point:

$$\frac{\mathrm{d}}{\mathrm{d}\theta}\log p(\theta,\mathcal{D}) = \frac{N_H+a-1}{\theta} - \frac{N_T+b-1}{1-\theta} \tag{33}$$

Setting this to zero, we get

$$\hat{\theta}_{\mathrm{MAP}} = \frac{N_H+a-1}{N_H+N_T+a+b-2} \tag{34}$$

We can summarize the results of the three different methods in the following table, for $a = b = 2$.

	Formula	$N_H = 2, N_T = 0$	$N_H = 55, N_T = 45$
$\hat{\theta}_{\mathrm{ML}}$	$\frac{N_H}{N_H+N_T}$	1	$\frac{55}{100} = 0.55$
θ_{pred}	$\frac{N_H+a}{N_H+N_T+a+b}$	$\frac{4}{6} \approx 0.67$	$\frac{57}{104} \approx 0.548$
$\hat{\theta}_{\mathrm{MAP}}$	$\frac{N_H+a-1}{N_H+N_T+a+b-2}$	$\frac{3}{4} = 0.75$	$\frac{56}{102} \approx 0.549$

When we have 100 observations, all three methods agree quite closely with each other. However, with only 2 observations, they are quite different. $\hat{\theta}_{\mathrm{ML}} = 1$, which as we noted above, is dangerous because it assigns no probability to T, and it will have a test log-likelihood of $-\infty$ if there is a single T in the test set. The other methods smooth the estimates considerably. MAP behaves somewhere in between ML and FB; this happens pretty often, as MAP is a sort of compromise between the two methods.

Example 6. Let's return to our Gaussian example. Let's maximize the joint probability:

$$\log p(\mu,\mathcal{D}) = \mathrm{const} - \frac{1}{2\sigma_{\mathrm{pri}}^2}(\mu-\mu_{\mathrm{pri}})^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x^{(i)}-\mu)^2 \tag{35}$$

$$\frac{\mathrm{d}}{\mathrm{d}\mu}\log p(\mu,\mathcal{D}) = -\frac{1}{\sigma_{\mathrm{pri}}^2}(\mu-\mu_{\mathrm{pri}}) + \frac{1}{\sigma^2}\sum_{i=1}^{N}(x^{(i)}-\mu) \tag{36}$$

When we set this to 0, we get exactly the same formula for $\hat{\mu}_{\mathrm{MAP}}$ as we derived earlier for μ_{post}. This doesn't mean the two methods make the same predictions, though. The two predictive distributions have the same mean, but the MAP one has standard deviation $\hat{\sigma}_{\mathrm{MAP}} = \sigma$, compared with $\sigma_{\mathrm{pred}} = \sqrt{\sigma_{\mathrm{post}}^2 + \sigma^2}$ for

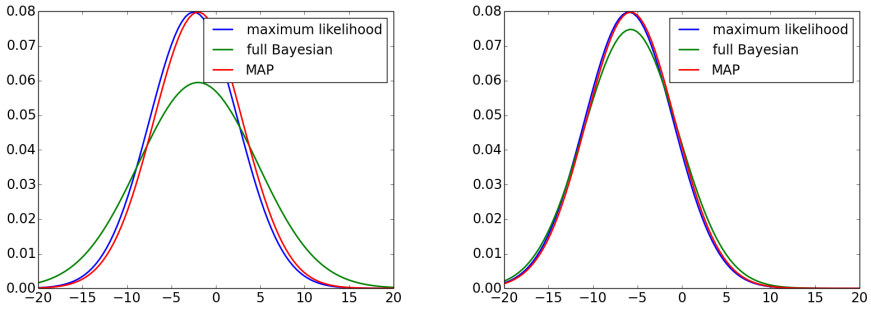

Figure 4: Comparison of the predictions made by the ML, FB, and MAP methods about future temperatures. **(Left)** After observing one training case. **(Right)** After observing 7 training cases, i.e. one week.

the full Bayesian approach. In other words, the full Bayesian approach smooths the predictions, while MAP does not. Therefore, the full Bayesian approach tends to make more sensible predictions in the small data setting. A comparison of the three methods is shown in Figure 4.

3.4 Is MAP a good approximation?

In both of the examples we looked at, ML, FB, and MAP all made very similar predictions in the large data regime, but very different ones in the small data regime. Which setting is more typical in practice?

On one hand, we typically use a lot more data than we did in these toy examples. In typical neural net applications, we'd have thousands or millions of training cases. On the other hand, we'd also have a lot more parameters: typically thousands or millions. Depending on the precise dataset and model architecture, there might or might not be a big difference between the methods.

3.5 Can the full Bayesian method overfit?

We motivated the Bayesian approach as a way to prevent overfitting. It's sometimes claimed that you *can't* overfit if you use the full Bayesian approach. Is this true? In a sense, it is. If your prior and likelihood model are both accurate, then Bayesian inference will average the predictions over all parameter values that are consistent with the data. Either there's enough data to accurately pinpoint the correct values, or the predictions will be averaged over a broad posterior which probably includes values close to the true one.

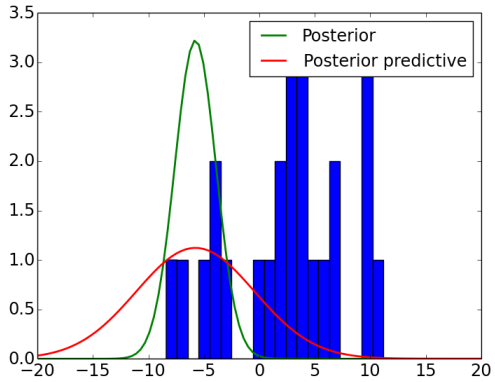

Figure 5: The full Bayesian posterior predictive distribution given the temperatures for the first week, and a histogram of temperatures for the remainder of the month. Observe that the predictions are poor because of model misspecification.

However, in the presence of **model misspecification**, the full Bayesian approach can still overfit. This term is unfortunate because it makes it sound like misspecification only happens when we do something wrong. But pretty much all the models we use in machine learning are vast oversimplifications of reality, so we can't rely on the theoretical guarantees of the Bayesian approach (which rely on the model being correctly specified). We can see this in our Toronto temperatures example. Figure 5 shows the posterior predictive distribution given the first week of March, as well as a histogram of temperature values for the rest of the month. A lot of the temperature values are outside the range predicted by the model! There are at least two problems here, both of which result from the erroneous i.i.d. assumption:

- The data are not identically distributed: the observed data are for the start of the month, and temperatures may be higher later in the month.
- The data are not independent: temperatures in subsequent days are correlated, so treating each observation as a new independent sample results in a more confident posterior distribution than is actually justified.

Unfortunately, the data are rarely independent in practice, and there are often systematic differences between the datasets we train on and the settings where we'll need to apply the learned models in practice. Therefore, overfitting remains a real possibility even with the full Bayesian approach.

4 Summary

We've introduced three different methods for learning probabilistic models:

- Maximum likelihood (ML), where we choose the parameters which maximize the likelihood:

$$\hat{\boldsymbol{\theta}}_{\text{ML}} = \arg\max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \log p(\mathcal{D} \mid \boldsymbol{\theta}) \tag{37}$$

  Sometimes we can compute the optimum analytically by setting partial derivatives to 0. Otherwise, we need to optimize it using an iterative method such as SGD.

- The full Bayesian (FB) approach, where we make predictions using the posterior predictive distribution. To do this, we condition on the data and integrate out the parameters:

$$p(\mathcal{D}' \mid \mathcal{D}) = \int p(\boldsymbol{\theta} \mid \mathcal{D}) p(\mathcal{D}' \mid \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}. \tag{38}$$

  Sometimes there's a closed-form solution to the integral, but otherwise we need to solve a difficult high-dimensional integration problem. This is what makes FB so difficult to apply in practice.

- Maximum a-posteriori (MAP), a compromise between ML and FB. We approximate the posterior distribution with a single value $\boldsymbol{\theta}_{\text{MAP}}$, which maximizes the posterior probability

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} \; \log p(\boldsymbol{\theta} \mid \mathcal{D}) = \arg\max_{\boldsymbol{\theta}} \; \log p(\boldsymbol{\theta}) + \log p(\mathcal{D} \mid \boldsymbol{\theta}). \tag{39}$$

  This is similar to ML in that it's an optimization problem, and the prior term $\log p(\boldsymbol{\theta})$ is analogous to a regularization term.

All three approaches behave similarly in the setting where there are many more data points than parameters. However, in settings where there isn't enough data to accurately fit the parameters, the Bayesian methods have a smoothing effect, which can result in much more sensible predictions. Later in this course, we'll see models where each of these methods is useful.