6.867 Section 3: Classification

Reading:

- Probabilistic models: Bishop 4.1, 4.2
- Bayesian methods: Bishop 4.3.1–4.3.4, 4.4–4.5
- SVMs: Bishop (briefly) first part of 7.1; Murphy (grudgingly) 14.5; Hastie Tibshirani Friedman 12.1–3

Contents

1 Intro

Now we will look at the supervised learning problem of *classification*, in which the data given is a set of pairs

$$\mathcal{D} = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\} ,$$

with $x^{(i)} \in \mathbb{R}^D$ and $y^{(i)} \in \{c_1, \ldots, c_k\}$; that is, the output is a discrete value indicating the *class* of the example. By default, we'll think about the two-class classification problem, with the *classes* or *labels* often drawn from the set $\{0, 1\}$ or $\{+1, -1\}$. This problem isn't conceptually different from regression, but it offers a different kind of structure to be exploited during learning.

We'll look at learning prediction rules, probabilistic models, and distributions over models, and focus on the case in which the model represent a thresholded linear relationship between inputs and outputs. In later parts of the course, we will return to classification and look at non-linear and non-parametric approaches.

Because we have recently been looking at probabilistic approaches to regression, we will begin by looking at probabilistic and Bayesian approaches to classification. When we are done with that, we will consider some interesting strategies for finding linear separators directly, without probabilistic modeling.

2 Representation

What does it mean to have a linear model for classification? Generally, it will be that we can express the output value $y^{(i)}$ by specifying a $D-1$-dimensional *hyperplane* in the D-dimensional feature space. Then, points that are on one side of the hyperplane are considered to be in one class, points on the other side, in the other class.

That is, there are some weight values w_0 and $w = (w_1, \ldots, w_D)$ such that

$$y = \begin{cases} +1 & \text{if } w_0 + w_1 x_1 + \ldots + w_D x_D > 0 \\ -1 & \text{otherwise} \end{cases} .$$

Such a model is known as a *linear separator*. So, in a 2D feature space, our separator would be a 1D hyperplane (otherwise known as a line). We would use three parameters (which is really one more than necessary) to describe the line.

As in regression, we can transform the input space, via a set of non-linear basis functions, and find a linear separator in the transformed space. Such a separator will be non-linear when projected back down into the original space.

In the following, to simplify notation, we will omit the possibility of using basis function to transform the input values, but the extension is completely straightforward.

3 Probabilistic models

	No model	Prediction rule	Prob model	Dist over models
Classification			❋	

3.1 Estimating $\Pr(X, Y)$

Estimating the joint distribution is often referred to as learning a *generative* model. We will make a point estimate of the parameters governing the distribution of the data, and then we can use our loss function to decide how to make decisions or predictions.

3.1.1 Linear discriminant analysis

If we're going to estimate the joint distribution, we need to make some distributional assumptions. A common model is:

$$\begin{aligned} Y &\sim \text{Bernoulli}(\pi) \\ X \mid Y = c &\sim \text{Gaussian}(\mu_c, \Sigma_c) \end{aligned}$$

where π_c specifies the unconditional probability of getting an object of class c, and π is a vector of π_c for the possible class values c. We will use θ to name all of the parameters: π, μ, Σ.

We can find the maximum-likelihood parameter estimates for this model straightforwardly. Letting $n_c = |\{y^{(i)} = c \mid i = 1 \ldots n\}|$ be the number of examples in $\mathcal{D}$ of class c, we have:

$$\begin{aligned} \pi_c &= \frac{n_c}{n} \\ \mu_c &= \frac{1}{n_c} \sum_{\{i|y^{(i)}=c\}} x^{(i)} \\ \Sigma_c &= \frac{1}{n_c} \sum_{\{i|y^{(i)}=c\}} (x^{(i)} - \mu_c)(x^{(i)} - \mu_c)^T \end{aligned}$$

Now, how should we make predictions? If we have 0-1 loss,

$$h(x) = \begin{cases} 1 & \text{if } \Pr(Y = 1 \mid X = x; \theta) > \Pr(Y = 0 \mid X = x; \theta) \\ 0 & \text{otherwise} \end{cases}$$

Let's concentrate on the conditions under which we predict 1:

$$\begin{aligned} \Pr(Y = 1 \mid X = x; \theta) &> \Pr(Y = 0 \mid X = x; \theta) \\ \Pr(X = x \mid Y = 1; \theta) \Pr(Y = 1; \theta) &> \Pr(X = x \mid Y = 0; \theta) \Pr(Y = 0; \theta) \\ \log \Pr(X = x \mid Y = 1; \theta) + \log \Pr(Y = 1; \theta) &> \log \Pr(X = x \mid Y = 0; \theta) + \log \Pr(Y = 0; \theta) \\ \log \Pr(X = x \mid Y = 1; \theta) - \log \Pr(X = x \mid Y = 0; \theta) &> \log \Pr(Y = 0; \theta) - \log \Pr(Y = 1; \theta) \\ -\frac{1}{2} \log \det(\Sigma_1) - \frac{1}{2}(x - \mu_1)\Sigma_1^{-1}(x - \mu_1)^T & \\ +\frac{1}{2} \log \det(\Sigma_0) + \frac{1}{2}(x - \mu_0)\Sigma_0^{-1}(x - \mu_0)^T &> \log \pi_0 - \log \pi_1 \\ -(x - \mu_1)\Sigma_1^{-1}(x - \mu_1)^T + (x - \mu_0)\Sigma_0^{-1}(x - \mu_0)^T &> 2(\log \pi_0 - \log \pi_1) + \log \det(\Sigma_1) - \log \det(\Sigma_0) \end{aligned}$$

This is pretty clearly quadratic in x, so it's what we would call a *quadratic separator* or *quadratic discriminant*.

If we assume that the covariances of the two classes are equal, $\Sigma_1 = \Sigma_0$, then things simplify and we are doing *linear discriminant analysis.* Now we have that (determinant of Σ cancels):

$$\Pr(Y = c \mid X = x) \propto \exp\left(\mu_c^T \Sigma^{-1} x - \frac{1}{2}\mu_c^T \Sigma^{-1} \mu_c + \log \pi_c\right) \exp\left(-x^T \Sigma^{-1} x\right) \quad .$$

Define

$$\begin{aligned} \beta_c &= \Sigma^{-1} \mu_c \\ \gamma_c &= -\frac{1}{2}\mu_c^T \Sigma^{-1} \mu_c + \log \pi_c \end{aligned}$$

Then

$$\begin{aligned}\Pr(Y = c \mid X = x) &= \frac{\exp\left(\beta_c^T x + \gamma_c\right)\exp\left(x^T\Sigma^{-1}x\right)}{\sum_{c'}\exp\left(\beta_{c'}^T x + \gamma_{c'}\right)\exp\left(x^T\Sigma^{-1}x\right)} \\ &= \frac{\exp\left(\beta_c^T x + \gamma_c\right)}{\sum_{c'}\exp\left(\beta_{c'}^T x + \gamma_{c'}\right)}\end{aligned}$$

Note that, in the two class case, this reduces to

$$\begin{aligned}\Pr(Y = 1 \mid X = x) &= \frac{\exp\left(\beta_1^T x + \gamma_1\right)}{\exp\left(\beta_0^T x + \gamma_0\right) + \exp\left(\beta_1^T x + \gamma_1\right)} \\ &= \frac{1}{\exp\left(\beta_0^T x + \gamma_0 - \beta_1^T x - \gamma_1\right) + 1} \\ &= \operatorname{sigmoid}(\beta_1^T x + \gamma_1 - \beta_0^T x - \gamma_0)\end{aligned}$$

where the *sigmoid* function and its inverse, the *logit* function are defined as follows:

$$\begin{aligned}\operatorname{sigmoid}(a) &= \frac{1}{1 + \exp(-a)} \\ &= \frac{\exp a}{1 + \exp a} \\ \operatorname{logit}(\sigma) &= \log\left(\frac{\sigma}{1-\sigma}\right) \ .\end{aligned}$$

The sigmoid is a "soft" step function, which takes a real number and maps it to the interval $(0, 1)$. If we have an expression of the form $\operatorname{sigmoid}(W^T X)$, then the larger the magnitude of W, the steeper the slope on the sigmoid.

So, with two classes, we predict class 1 when

$$\begin{aligned}\Pr(Y = 1 \mid X = x; \theta) &> \Pr(Y = 0 \mid X = x; \theta) \\ \exp\left(\beta_1^T x + \gamma_1\right) &> \exp\left(\beta_0^T x + \gamma_0\right) \\ \beta_1^T x + \gamma_1 &> \beta_0^T x + \gamma_0 \\ x\left(\beta_1 - \beta_0\right) + \gamma_1 - \gamma_0 &> 0\end{aligned}$$

This is definitely a linear separator.

3.1.2 Factoring the class conditional probability

In the previous section, we assumed a special structure, namely, that all the elements in a particular class were drawn from a Gaussian distribution (over R^D). Now, we'll pursue another method that is also based on a special distributional structure.

This is a method called **'Naive Bayes'**. Let's assume now that $x^{(i)} \in \{0, 1\}^D$. You could interpret the X values as numbers, encoded in binary, and put a multinomial distribution over all 2^D values for each class. But that is a lot of parameters to estimate!

Assume: Features are independent, given the class. That is, that

$$\Pr(X \mid Y = c) = \prod_{j=1}^{D} \Pr(X_j \mid Y = c) \ .$$

So

$$\begin{aligned}Y &\sim \operatorname{Bernoulli}(\pi) \\ X_j \mid Y = c &\sim \operatorname{Bernoulli}(\theta_{c,j})\end{aligned}$$

So, now, if we have two classes, we have 2D parameters, each of which is easy to estimate:

$$\begin{aligned}\Pr(X_j = 1 \mid Y = 1) &= \theta_{1,j} \\ \Pr(X_j = 1 \mid Y = 0) &= \theta_{0,j}\end{aligned}$$

Use Bernoulli ML estimate, maybe with Laplace "correction":

$$\widehat{\theta}_{1j} = \frac{\#(X_j = 1, Y = 1) + 1}{\#(Y = 1) + 2} \quad .$$

Now, for prediction. Given x, predict $C = 1$ if

$$\begin{aligned}
\Pr(x \mid C = 1)\Pr(C = 1) &> \Pr(x \mid C = 0)\Pr(C = 0) \\
\prod_{j=1}^{D} \Pr(x_j \mid C = 1)\Pr(C = 1) &> \prod_{j=1}^{D} \Pr(x_j \mid C = 0)\Pr(C = 0) \\
\sum_{j=1}^{D} \log \Pr(x_j \mid C = 1) + \log \Pr(C = 1) &> \sum_{j=1}^{D} \log \Pr(x_j \mid C = 0) + \log \Pr(C = 0) \\
\sum_{j=1}^{D} \log(\theta_{1j}^{x_j}(1-\theta_{1j})^{(1-x_j)}) + \log \pi_1 &> \sum_{j=1}^{D} \log(\theta_{0j}^{x_j}(1-\theta_{0j})^{(1-x_j)}) + \log \pi_0 \\
\sum_{j=1}^{D} (x_j \log \theta_{1j} + (1-x_j)\log(1-\theta_{1j})) + \log \pi_1 &> \sum_{j=1}^{D} (x_j \log \theta_{0j} + (1-x_j)\log(1-\theta_{0j})) + \log \pi_0 \\
\sum_{j=1}^{D} x_j \left(\log \frac{\theta_{1j}}{(1-\theta_{1j})} - \log \frac{\theta_{0j}}{1-\theta_{0j}} \right) &> \log \pi_0 - \log \pi_1 - \sum_{j=1}^{D} \log \frac{1-\theta_{1j}}{1-\theta_{0j}}
\end{aligned}$$

So, this is a linear separator of the form $x^T w + w_0 > 0$ with

$$W_j = \log \frac{\theta_{1j}}{1-\theta_{1j}} - \log \frac{\theta_{0j}}{1-\theta_{0j}} \quad ,$$

and

$$W_0 = \log \frac{\pi_1}{\pi_0} + \sum_{j=1}^{n} \frac{\log(1-\theta_{1j})}{\log(1-\theta_{0j})} \quad .$$

Interestingly, the probability model is also sigmoidal. We have

$$\begin{aligned}
\Pr(C = 1 \mid X = x) &= \frac{\Pr(x \mid C = 1)\Pr(C = 1)}{\Pr(x \mid C = 1)\Pr(C = 1) + \Pr(x \mid C = 0)\Pr(C = 0)} \\
&= \frac{\exp(f_1(x))}{\exp(f_1(x)) + \exp(f_2(x))} \\
&= \frac{1}{1 + \exp(f_2(x) - f_1(x))} \\
&= \text{sigmoid}(f_1(x) - f_2(x))
\end{aligned}$$

where

$$f_c(x) = \sum_{j=1}^{D} (x_j \log \theta_{cj} + (1-x_j)\log(1-\theta_{cj})) + \log \pi_c \quad .$$

3.1.3 Exponential family

A really cool family of distributions.

- Only family for which conjugate priors exist
- Finite-size sufficient statistics
- Includes Normal, Bernoulli, Multinomial, Poisson, Gamma, Exponential, Beta, Dirichlet, various combinations

A (somewhat simplified) subset of exponential-family distributions can be written in the form:

$$\Pr(x; \eta) = h(x) g(\eta) \exp(\eta^T u(x))$$

where x may be a scalar or vector, discrete or continuous; η is called the *natural parameters*, and $g(\eta)$ is a normalization constant.

3.1.3.1 Bernoulli is in exponential family

$$\begin{aligned}
\Pr(x \mid p) &= p^x (1-p)^{(1-x)} \\
&= \exp\left(x \log(p) + (1-x)\log(1-p)\right) \\
&= \exp\left(x \log(p) - x\log(1-p) + \log(1-p)\right) \\
&= (1-p)\exp\left(x \log \frac{p}{1-p}\right)
\end{aligned}$$

This fits in the family with: $u(x) = x$, $\eta = \log \frac{p}{1-p}$, $h(x) = 1$, and $g(\eta) = 1 - p$. So

$$\Pr(x \mid \eta) = \text{sigmoid}(-\eta)\exp(\eta x) \quad .$$

3.1.3.2 Normal is in exponential family We'll just look at the one-dimensional case, but it's true for multi-variate Gaussian as well.

$$\begin{aligned}
\Pr(x \mid \mu, \sigma^2) &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) \\
&= \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2\sigma^2}x^2 + \frac{\mu}{\sigma^2}x - \frac{1}{2\sigma^2}\mu^2\right)
\end{aligned}$$

To make this match up, let

$$\begin{aligned}
\eta &= \begin{pmatrix} \mu/\sigma^2 \\ -1/2\sigma^2 \end{pmatrix} \\
u(x) &= \begin{pmatrix} x \\ x^2 \end{pmatrix} \\
h(x) &= (2\pi)^{1/2} \\
g(\eta) &= \sqrt{-2\eta_2}\exp\left(\frac{\eta_1^2}{4\eta_2}\right) .
\end{aligned}$$

3.1.3.3 LDA for exponential family Cool result! If we have two classes, and $\Pr(X \mid Y = c)$ is a distribution in the exponential family, with the restrictions that $u(x) = x$, and that the scale parameters are shared among the classes (e.g., the covariance in the Gaussian case) so that the form is

$$\Pr(x \mid \lambda_c, s) = \frac{1}{s} h\left(\frac{1}{s}x\right) g(\lambda_c) \exp\left(\frac{1}{s}\lambda_c^T x\right) \ ,$$

then

- The separator is linear and
- The predictive probability $\Pr(Y = 1 \mid x)$ is a sigmoid on an activation function which is the difference between the logs of the class membership probabilities of the two classes. That is,

  $$\Pr(Y = 1 \mid x) = \text{sigmoid}(a(x)) \ ,$$

  where

  $$a(x) = (\lambda_1 - \lambda_0)^T x + \log g(\lambda_1) - \log g(\lambda_0) + \log \pi_1 - \log \pi_0$$

3.2 Estimating $\Pr(Y \mid X)$

This is called estimating a *discriminative model*.

3.2.1 Least squares

A favorite trick is to reduce our current problem to a previous problem we already know how to solve. In this case, we could try to treat classification as a regression problem, by taking the Y values to be in $\{+1, -1\}$ and applying one of our standard regression methods. We could then predict class $+1$, given x, if

$$W^T x + W_0 > 0 \ .$$

That will generate a separator, but it turns out to be a bad idea. There is generally no hypothesis that does a good job of representing the data, and it is easy to show that there are situations in which a linear separator for the data exists, but the hypothesis that regression comes up with does not separate the data.

One reason that this doesn't work out is that it isn't founded on a sensible probabilistic model: the squared error criterion fundamentally assumes that the Y values are normally distributed, but they're not.

3.2.2 Logistic regression

So, what assumption can we reasonably make about the distribution of outputs, in the classification case? Rather than directly trying to predict the value, we could try to find a regression model that predicts $\Pr(Y = 1 \mid X = x)$ as a function of x. One thought would be to use the form

$$\Pr(Y = 1 \mid X = x) = W^T x \ ,$$

but the problem is that we need the probabilities to be in the interval $[0, 1]$, and that linear form is unconstrained.

We can take inspiration from what we saw in the LDA case: that in at least two cases we considered, the predictive distribution could be described as a sigmoid applied to a dot product. So, let's consider models of the form

$$\Pr(Y = 1 \mid X = x) = \text{sigmoid}(W^T x) \ .$$

Such models are called *logistic regression* models.

Which is confusing, since we're using them for classification...but the idea is that we're doing a regression to find a probability value.

So, how does this differ from LDA? We are not making any distributional assumptions about X. So, we're going to train a predictive model from the same class, but *using a different criterion*.

Given data $\mathcal{D}$, what are the maximum-likelihood estimates of the parameters W? Assuming $y^{(i)} \in \{0, 1\}$, the likelihood function is:

$$\Pr(\mathcal{D} \mid W) = \prod_{i=1}^{n} s\left(W^T x^{(i)}\right)^{y^{(i)}} \left(1 - s\left(W^T x^{(i)}\right)\right)^{(1-y^{(i)})} \quad ,$$

where s is short for sigmoid. The negative log likelihood is:

Which we want to minimize.

$$\mathrm{NLL}(W) = -\sum_{i=1}^{n} \left(y^{(i)} \log s\left(W^T x^{(i)}\right) + (1 - y^{(i)}) \log\left(1 - s\left(W^T x^{(i)}\right)\right)\right) \quad .$$

The gradient with respect to the weights is

Here's the cool thing about the derivative of the sigmoid: $\frac{d}{dx}s(x) = s(x)(1 - s(x))$

$$\nabla_W \mathrm{NLL}(W) = \sum_{i=1}^{n} \left(s\left(W^T x^{(i)}\right) - y^{(i)}\right) x^{(i)} \quad .$$

This is the same as the gradient of the error function for the linear regression model!

We'll see why in the next section.

If the data is linearly separable, the optimization will want to make the weights as large as possible (to make the sigmoid as steep as possible, which will push the output values closer and closer to +1 and −1. Furthermore, again, if the data are separable, there is an infinity of separators that are equally good; so the optimization problem is not well posed. This can be addressed by adding a regularization penalty or using a prior.

An old trick from neural network days was to use targets of +0.6 and −0.6, which are actually attainable with finite weights.

Unfortunately, there is no closed form solution for the maximum likelihood values of W, so we need to use an iterative optimization method. Two common choices are:

- *Gradient descent*, especially *stochastic gradient descent*, which goes through the data points one by one and does an update to the weights with a very small step size based on each individual point. This can be guaranteed to converge, though it might be slow, but it can be less likely than batch gradient descent to get stuck in local optima on non-convex functions.

- *Iterative reweighted least squares*, which is essentially Newton's method. It can be computationally challenging, because it uses the Hessian (matrix of second derivatives), but is more reliable than regular gradient descent.

Fortunately, the error function is convex, so these methods, appropriately tuned, will find the global optimum.

3.2.3 Generalized linear models

Both linear regression and logistic regression are instances of a more general framework of *generalized linear models*:

- The distribution of $Y \mid X$ is an exponential family distribution

- The goal is to predict $E[Y \mid X; \theta]$

- The *natural parameter* η and the inputs are linearly related: $\eta = \theta^T x$.

What is cool about this is that any regression-type problem that satisfies the requirements above can be handled in common way. So, there are general purpose methods for

- Maximum likelihood parameter estimation (IRLS)
- Bayesian estimation
- Various statistical tests

This also means that these models are not too sensitive to the distributional assumptions made (beyond the general GLM assumptions), since the parameter-fitting is independent of which exponential-family distribution you have.

3.3 Generative versus discriminative

Given a choice, should we choose a generative or discriminative model?

This discussion cribbed from *Machine Learning: A Probabilistic Perspective* by Kevin Murphy.

- Generative classifiers are usually computationally easier to fit.
- Generative classifiers, in the multi-class case, generalize more easily to adding a new class (since the per-class models are independent).
- Generative models can be estimated relatively easily in the presence of missing or unlabeled data.
- Generative models can be run "backwards" (used to predict X from Y).
- Discriminative models apply very well when we expand the input space using a basis set of feature functions; generative models can get into trouble due to correlation among the inputs.
- *If the distributional assumptions are true*, generative models can be well estimated with fewer training examples than discriminative models.
- *If the distributional assumptions are not true*, generative models can result in really bad predictions.

4 Distributions over models

	No model	Prediction rule	Prob model	Dist over models
Classification				❋

Rather than finding a single best weight vector W^* and using that to make predictions $\Pr(y \mid x; W^*)$, we might again wish to be Bayesian, by putting a prior on W and then using the posterior on W, that is $\Pr(W \mid \mathcal{D})$, to make predictions that take uncertainty in the weights into account.

Unfortunately, there is no conjugate prior for logistic regression. Here, we will show one strategy (that can be used in all sorts of cases with non conjugate priors, sometimes to good effect, sometimes less so) for approximating the posterior. The idea is to start with a Gaussian prior, compute the posterior, which is

$$\log \Pr(W \mid \mathcal{D}) = -\frac{1}{2}(W - \mathbf{m}_0)^T S_0^{-1}(W - \mathbf{m}_0) + \sum_{i=1}^{n}(y^{(i)} \log o^{(i)} + (1 - y^{(i)}) \log(1 - o^{(i)})) \enspace ,$$

where $o^{(i)} = \text{sigmoid}(W^T x^{(i)})$.

This clearly does not have the form of a Gaussian. So we want to find a Gaussian approximation to the posterior, which means finding parameters $\mathbf{m}_n$ and $\mathbf{S}_n$ that make a good approximation. Using a second-order Taylor-series expansion about the mode, we find that the MAP estimate of the weights, which under a simple diagonal Gaussian prior is

$$W_{\text{map}} = \arg\min_W \text{NLL}(W) + \lambda W^T W \ ,$$

the appropriate choice of $\mathbf{m}_n$.

This is known as the Laplace approximation; see Bishop 4.4 for more details.

The gradient of the negative log likelihood at the mode is:

$$\left.\frac{\partial \text{NLL}(W)}{\partial W}\right|_{W_{\text{MAP}}} = \mathbf{s}_0^{-1}(W - \mathbf{m}_0) - \sum_{i=1}^{n}(y^{(i)} - o^{(i)})x^{(i)} \ ,$$

where output $o^{(i)} = \text{sigmoid}({x^{(i)}}^T W_{\text{MAP}})$.

The covariance matrix is then the inverse of the Hessian, which is a matrix of second derivatives of NLL. So,

$$\mathbf{s}_n^{-1} = H = \left.\frac{\partial^2 \text{NLL}(W)}{\partial W \partial W^T}\right|_{W_{\text{MAP}}} = \mathbf{s}_0^{-1} + \sum_{i=1}^{n} o^{(i)}(1 - o^{(i)})x^{(i)}{x^{(i)}}^T \ .$$

Going from here to a predictive distribution requires further approximation. We won't go into it in detail. However, there is one more useful concept to get out of this, which is the *Bayesian information criterion* or BIC. We can approximate

$$\log \Pr(\mathcal{D}) \approx \log \Pr(\mathcal{D} \mid W_{\text{MAP}}) + \log \Pr(W_{\text{MAP}}) + \frac{D}{2}\log(2\pi) - \frac{1}{2}\log|H| \ .$$

The second two terms, called the *Occam factor*, measure the complexity of the model. If the prior on weights is uniform, then we can simplify to

$$\log \Pr(\mathcal{D}) \approx \log \Pr(\mathcal{D} \mid W_{\text{ML}}) - \frac{1}{2}\log|H| \ .$$

The determinant of a covariance matrix can be seen informally as characterizing the "size" of the covariance ellipse: if it's large then we are more uncertain about the weight values, which means the marginal likelihood will be lower.

If we think of H as being a sum of H_i over the individual data points, and assume that they are approximable by single $\widehat{H}$, then

$$\begin{aligned} \log|H| &= \log|n\widehat{H}| \\ &= \log\left(n^D|\widehat{H}|\right) \\ &= D\log n + \log|\widehat{H}| \end{aligned}$$

We drop $\log|\widehat{H}|$ because it is independent of n and D, and wind up with a fairly gross approximation:

$$\log \Pr(\mathcal{D}) \approx \log \Pr(\mathcal{D} \mid W_{\text{ML}}) - \frac{D}{2}\log n \ .$$

This is the BIC score, which can be used as a quick-and-dirty way of selecting model complexity.

BIC is asymptotically consistent as a selection criterion: that is, given a set of models, as $n \to \infty$, BIC will select the correct model. It has a tendency to select models that are too simple, with small n, though.