Introductory Tutorial on Gaussian Processes and Hyperparameter Selection

Philippe Wenk

December 6, 2019

Abstract

Summary of the PAI tutorial on Bayesian hyperparameter selection, briefly introducing Gaussian processes, cross-validation, empirical Bayes and hyperpriors.

1 Problem Setting

Let us assume that we are given the task of modeling a one-dimensional dynamical system. At N time points $\boldsymbol{t} = [t_0, \dots, t_{N-1}]$, we are provided with noisy observations $\boldsymbol{y} = [y(t_0), \dots, y(t_{N-1})]$ of the system's state. Our goal is to retrieve the true states $\boldsymbol{x} = [x(t_0), \dots, x(t_{N-1})]$, assuming an i.i.d. Gaussian observation noise model.

A simple Approach The problem description gives rise to the graphical model shown in Figure 1. Here, σ denotes the (unknown) standard deviation of the additive Gaussian noise. Formally, we would describe the noise model using the conditional density $p(\boldsymbol{y}|\boldsymbol{x}, \sigma) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{x}, \sigma^2 \boldsymbol{I}_N)$. Here, we are treating $\boldsymbol{x}$ and σ as parameters of the model, since we do not observe either. To obtain their values, a natural approach might be to just try and maximize this likelihood. However, this leads to problems: If we choose $\boldsymbol{x} = \boldsymbol{y}$ and let σ decrease, we will get arbitrarily large $p(\boldsymbol{y}|\boldsymbol{x}, \sigma)$. Clearly, this would not be a good solution though. $\boldsymbol{x} = \boldsymbol{y}$ would mean that we are fitting our data perfectly and $\sigma \to 0$ means that we are estimating that there is no noise at all. These are both clear signs of overfitting, we are just blindly believing our data without question. Thus, we need a different way to reliably extract information from our data.

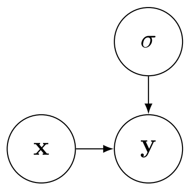

Figure 1: Simple Data Model.

2 Gaussian Process Regression

In Gaussian Process Regression[1], we try to avoid this overfitting by treating $\boldsymbol{x}$ as random variables as well. In classic Bayesian fashion, we expand our model by additional hyperparameters $\boldsymbol{\phi}$. These hyperparameters then govern a prior distribution over $\boldsymbol{x}$, that should encode some smootheness information to avoid fitting the data (and thus the noise) perfectly. Formally, we have to start by defining a positive-definite kernel function $k_{\boldsymbol{\phi}}(t_i, t_j)$ parametrized by the hyperparameters $\boldsymbol{\phi}$. This kernel function is then used to define the prior on $\boldsymbol{x}$, namely

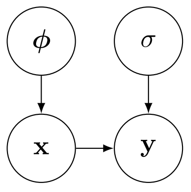

Figure 2: GP Regression Model.

$$p(\boldsymbol{x}|\boldsymbol{\phi}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{0}, \boldsymbol{C}_{\boldsymbol{\phi}}), \tag{1}$$

where $\boldsymbol{C}_{\boldsymbol{\phi}}^{i,j} = k(t_i, t_j)$. Combining this prior with the Gaussian noise observation model $p(\boldsymbol{y}|\boldsymbol{x}, \sigma) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{x}, \sigma^2 \boldsymbol{I}_N)$, we obtain the classical GP regression as shown in Figure 2. Due to the Gaussianity of all components involved, we can calculate many interesting quantities analytically.

First, using Bayes rule[2], we can now do full Bayesian inference to infer the posterior of the state values $\boldsymbol{x}$, arriving at

$$p(\mathbf{x}|\boldsymbol{y}, \boldsymbol{\phi}, \sigma) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{C}_{\boldsymbol{\phi}}(\boldsymbol{C}_{\boldsymbol{\phi}} + \sigma^2 \boldsymbol{I}_N)^{-1}\boldsymbol{y}, \sigma^2(\boldsymbol{C}_{\boldsymbol{\phi}} + \sigma^2 \boldsymbol{I}_N)^{-1}\boldsymbol{C}_{\boldsymbol{\phi}}). \tag{2}$$

Furthermore, it is straight forward to marginalize out $\boldsymbol{x}$ to obtain

$$p(\boldsymbol{y}|\sigma, \boldsymbol{\phi}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{0}, \boldsymbol{C}_{\boldsymbol{\phi}} + \sigma^2 \boldsymbol{I}_N). \tag{3}$$

Finally, we can use the underlying GP assumption that all $\boldsymbol{x}$ are jointly Gaussian distributed to obtain the joint Gaussian distribution including an as of yet unobserved point τ. Define $\boldsymbol{k}(\boldsymbol{t}, \tau) := [k(t_0, \tau), \dots, k(t_{N-1}, \tau)]$. By the definition of a GP, we then have

$$p(\boldsymbol{y}, y(\tau)|\sigma, \phi) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{y} \\ y(\tau)\end{bmatrix} \mid \boldsymbol{0}, \begin{bmatrix}\boldsymbol{C}_{\boldsymbol{\phi}} & \boldsymbol{k}(\boldsymbol{t}, \tau) \\ \boldsymbol{k}^T(\boldsymbol{t}, \tau) & k(\tau, \tau)\end{bmatrix}\right), \tag{4}$$

which implies the following conditional distribution

$$p(y(\tau)|\boldsymbol{y}, \sigma, \phi) = \mathcal{N}(y(\tau)|\mu_\tau, \Sigma_\tau), \tag{5}$$

where

$$\mu_\tau = \boldsymbol{k}^T(\boldsymbol{t}, \tau)(\boldsymbol{C}_{\boldsymbol{\phi}} + \sigma^2 \boldsymbol{I}_N)^{-1}\boldsymbol{y} \tag{6}$$

and

$$\Sigma_\tau = k(\tau, \tau) - \boldsymbol{k}^T(\boldsymbol{t}, \tau)(\boldsymbol{C}_{\boldsymbol{\phi}} + \sigma^2 \boldsymbol{I}_N)^{-1}\boldsymbol{k}(\boldsymbol{t}, \tau) + \sigma^2. \tag{7}$$

If we were to compare this model to the naive approach presented in the first section, it seems that we just shifted the problem of inferring $\boldsymbol{x}$ and σ to inferring $\boldsymbol{\phi}$ and σ and it is still unclear how to find good values for those. However, as we shall see in the next section, the new parameters can be found using a few techniques that have shown to work well in numerous practical applications.

[1] for more details, please refer to Williams and Rasmussen [2006]

[2] For dealing with the algebraic difficulties of these calculations, please refer to Petersen and Pedersen

3 Hyperparameter Fitting

In this section, we will look at different ways of inferring "good" values for the hyperparameters $\boldsymbol{\phi}$ and σ given some observed data $\boldsymbol{y}$. For more information, see the model selection section of Williams and Rasmussen [2006].

3.1 Cross-Validation

Ignoring the Bayesian framework, the classical way of tuning hyperparameters would be cross-validation. In essence, we would split our $\boldsymbol{t}$ into a training set $\boldsymbol{t}_{\text{train}}$ and a validation set $\boldsymbol{t}_{\text{validation}}$. We then use $\boldsymbol{t}_{\text{train}}$ instead of the full vector $\boldsymbol{t}$ in defining $\boldsymbol{C}_{\boldsymbol{\phi}}$ and all other relevant quantities and then use the prediction Equations (5)-(7) to check what probabilities the model would assign to the unseen examples $\boldsymbol{y}(\boldsymbol{t}_{\text{validation}})$. This provides us with a score for the chosen hyperparameters, which we can then use e.g. by searching over a grid or randomly searching over all possible hyperparameter settings. Naturally, this comes with a lot of issues, including poor scaling in the amount of hyperparameters and the fact that we will always train on a smaller dataset than you actually have available, as you only train on a subset $\boldsymbol{y}(\boldsymbol{t}_{\text{train}})$ of your data.

3.2 Empirical Bayes

One method that seems to avoid these problems is called Empirical Bayes. Here, we maximize the observation marginals $p(\boldsymbol{y}|\sigma, \boldsymbol{\phi})$ w.r.t. the unknown quantities $\boldsymbol{\phi}$ and σ. In practice, this is done by minimizing the loss

$$\begin{aligned}\mathcal{L} &:= -\log(p(\boldsymbol{y}|\sigma, \boldsymbol{\phi})) && (8)\\ &= \frac{1}{2}\log(\det(\boldsymbol{C}_{\boldsymbol{\phi}} + \sigma^2 \boldsymbol{I}_N)) + \frac{1}{2}\boldsymbol{y}^T(\boldsymbol{C}_{\boldsymbol{\phi}} + \sigma^2 \boldsymbol{I}_N)^{-1}\boldsymbol{y} + \frac{N}{2}\log(2\pi) && (9)\end{aligned}$$

At first glance, this looks very similar to the naive approach in the first section, where we just maximized w.r.t. $\boldsymbol{x}, \sigma$. However, there is a key difference: When deriving Equation (3), we marginalized over $\boldsymbol{x}$. For a fixed $\boldsymbol{\phi}$, this can be interpreted as calculating a weighted average over many $p(\boldsymbol{y}|\hat{\boldsymbol{x}}, \sigma)$, where the weight of each $\hat{\boldsymbol{x}}$ is given by the prior distribution $p(\hat{\boldsymbol{x}}|\boldsymbol{\phi})$. Thus, we are not looking for one specific $\boldsymbol{x}$ that maximizes the likelihood of our observations, we are looking for a set of plausible $\hat{\boldsymbol{x}}$ that all provide a decent observation likelihood. If the prior is adequately chosen, then this will keep us from overfitting to $\boldsymbol{x} = \boldsymbol{y}$, since putting a lot of mass on $\boldsymbol{x} = \boldsymbol{y}$ would mean that we also have to put mass on many $\hat{\boldsymbol{x}}$ that would explain the data very badly (e.g. functions that would be not smooth enough).

Another way of intuitively understanding why Empirical Bayes might be a good idea is to look directly at Equation (9). In principle, $\boldsymbol{C}_{\boldsymbol{\phi}}$ encodes how much points should correlate if they are close to each other [3]. Thus, a small variance tends to correspond to less complicated (i.e. quickly varying) $\boldsymbol{x}$, while a large

[3] Here, distance is implicitly encoded in the structure of the kernel function. For some kernels, like the RBF, this can be directly associated with the Euclidean distance.

variance will allow for much "wigglier" functions. While the first term of Equation (9) punishes large variances, the second term would like the variance to be large, at least in the direction of the observed vectors $\boldsymbol{y}$. Thus, there is an automatic trade-off between observation fit and model complexity.

From a practical perspective, Empirical Bayes is quite appealing, since we can readily deploy gradient based optimizers to find good hyperparameters. However, one should always be aware that, since the objective is in general not convex, we might get stuck in local optima.

3.3 Hyperpriors

The key difference between Empirical Bayes and the naive approach of the first section was the fact that we introduced an additional layer of priors and then marginalized over the quantities we previously inferred, obtaining a cleaner optimization problem with automatic complexity trade-offs. Since every good thing is worth overdoing, we could go one step further and introduce priors for our priors, so called hyperpriors. The key idea here is the hope that we might inherit some benefits of the previous step, i.e. that the resulting model might be less sensitive to errors of the parameters of our hyperpriors or that our optimization procedure might become even more robust. The corresponding Bayes net is shown in Figure 3.

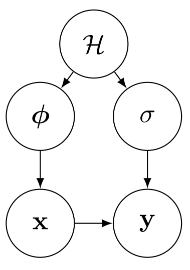

Figure 3: GP With Hyperpriors.

Unfortunately, the marginalization over $\boldsymbol{\phi}$ and σ is not as simple as marginalizing over $\boldsymbol{x}$ in the previous section. The special structure of Gaussian processes and the Gaussianity of all likelihoods allowed for an analytical derivation of the closed-form marginals of Equation (3). However, for $\boldsymbol{\phi}$ and σ, similar calculations are not possible in general. So, marginalization would need to be done using some approximation schemes (like e.g. MC sampling).

However, we can use our hyperpriors in a straight forward application of Bayes rule. Treating $\mathcal{H}$ as a fixed constant for readability, we get the posterior

$$p(\boldsymbol{\phi}, \sigma|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\phi}, \sigma)p(\boldsymbol{\phi}, \sigma)}{p(\boldsymbol{y})} \tag{10}$$

Optimizing this posterior is closely related to empirical Bayes. Since $p(\boldsymbol{y})$ is independent of $\boldsymbol{\phi}$ and σ, it can be ignored when optimizing for our hyperparameters. Thus, the only part left is the optimization objective of empirical Bayes, $p(\boldsymbol{y}|\boldsymbol{\phi}, \sigma)$, which is multiplied by the prior for $\boldsymbol{\phi}$ and σ. Thus, the only difference to Empirical Bayes is the additional regularization introduced by the priors. Such regularization might be a good idea if our hyperpriors are well chosen (i.e. we know that our hyperparameters should be in a certain region). However, if our prior knowledge is unreliable, introducing these priors might actually hurt our model's performance. In practice, people thus choose weak priors (i.e. close to the uniform distribution) if the prior knowledge is uncertain. Of course, if we choose our priors to be more and more uniformly distributed, maximizing Equation (10) will ultimately become equivalent to empirical Bayes.

References

Kaare Brandt Petersen and Michael Syskind Pedersen. The matrix cookbook.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.