How to fit a simple model

Carl Edward Rasmussen

Ellis 2022 Cambridge Machine Learning Summer School

July 11-15th, 2022

Motivation

Simplifying models are ubiquitous throughout science and engineering.

Such models typically need to be fit to empirical observations.

In this talk I compare two different approaches:

- the simpler, classical approach, which is used almost universally in practise
- the Bayesian view

On a simple, generic example, we show that the differences in accuracy can be massive.

An illustrative example

Data: We have N pairs of observations $\mathcal{D} = \{x_n, y_n\}_{n=1,\ldots,N}$ from

$$y \;=\; f(x) + \varepsilon, \quad \text{and} \quad \varepsilon \;\sim\; \mathcal{N}(0, \sigma_\varepsilon^2),$$

where f is an unknown function, and ε is independent, zero mean Gaussian observation noise of unknown variance σ_ε^2.

Goal: Find a value a, such that the approximate model

$$y \;=\; ax + \epsilon$$

where $x \sim \mathcal{N}(0, \sigma^2_{x,\text{objective}} = 1)$, is as good as possible, judged by squared error loss.

Sometimes this is known as Best Linear Approximation (BLA).

This problem is simple to state, yet contains qualitative features characteristic of more challenging problems.

Classical solution

Make a further assumption, that the training inputs x_n are drawn from the same distribution as our goal criterion

$$x_n \sim \mathcal{N}(0, \sigma^2_{x,\text{objective}}).$$

The classical, *least squares estimate* is

$$\hat{a}_{\text{LS}} = \Big[\sum_{n=1}^{N} x_n^2\Big]^{-1} \sum_{n=1}^{N} x_n y_n.$$

This estimate has a number of well known properties (eg. minimum variance unbiased estimator).

Numerous variants of this method exist too.

Bayesian procedure, overview

The Bayesian procedure consists of two steps

Inference Write down a generative model (containing unknown quantities) which describes how the data could have been generated. Compute the posterior distribution over all unknown quantities (conditioned on the observations)

Projection Find the solution to your problem, by projecting the posterior on to your approximate model, by minimising the expected loss. Formally, this is called Bayesian decision theory. Also, report the confidence in the answer.

Notice, that the eventual goal of modeling is irrelevant to the inference step (so you can reuse inference to solve every problem).

Minimising expected loss ≡ Bayesian decision theory

The *loss function*, in our example specified to be quadratic

$$L(a, a_{\text{best}}) \;=\; (a - a_{\text{best}})^2,$$

gives the loss associated with predicting a if the best answer was actually a_{best}.

The problem is that, of course, we don't know a_{best}. But, we will have a posterior distribution over a. Therefore, choose $\hat{a}$ which minimizes expected loss

$$\hat{a} \;=\; \mathsf{argmin}_a \int L(a, a_{\text{best}}) p(a_{\text{best}}|\mathcal{D}) da_{\text{best}}.$$

For squared loss

$$\frac{\partial}{\partial a} \int L(a, a_{\text{best}}) p(a_{\text{best}}|\mathcal{D}) da_{\text{best}} \;= 0 \implies$$

$$2 \int (a - a_{\text{best}}) p(a_{\text{best}}|\mathcal{D}) da_{\text{best}} \;= 0 \implies \boxed{\hat{a} \;=\; \mu_a.}$$

where μ_a is the posterior mean. Notice: we weren't free to arbitrarily choose the posterior mean, it was optimal given the loss and our posterior knowledge.

Model projection

Therefore, we seek μ_a from the distribution $p(a) = \mathcal{N}(\mu_a, \sigma_a^2)$ of the linear projection $a = \langle x^2 \rangle^{-1} \langle x f(x) \rangle$ from our generative model (not from our samples)

$$\mu_a = \iint x f(x) p(f(x)|x, \mathcal{D}, \hat{\mathbf{h}}) df(x) p(x) dx = \int x \bar{f}_{\mathcal{D}}(x) p(x) dx$$

$$= \boxed{\mathbf{z}^\top [k(\mathbf{x}, \mathbf{x}) + \sigma_\varepsilon^2 I]^{-1} \mathbf{y},} \qquad z_i = \frac{\sigma_w^2}{1+\ell^2} \sqrt{\frac{\ell^2}{1+\ell^2}} x_i \exp\Big[- \frac{x_i^2}{2(1+\ell^2)} \Big].$$

averaged over the posterior distribution over functions, and averaged over the distribution over inputs, $p(x)$. Here $\bar{f}_{\mathcal{D}}(x)$ is the posterior mean. Analogously, for the variance (where $k_{\mathcal{D}}(x, x')$ is the posterior covariance)

$$\sigma_a^2 = \int \Big[\int x f_{\mathcal{D}}(x) p(x) dx - \int x' \bar{f}_{\mathcal{D}}(x') p(x') dx' \Big]^2 p(f|\mathcal{D}, \hat{\mathbf{h}}) df$$

$$= \iiint x x' [f_{\mathcal{D}}(x) - \bar{f}_{\mathcal{D}}(x)][f_{\mathcal{D}}(x') - \bar{f}_{\mathcal{D}}(x')] p(f|\mathcal{D}, \hat{\mathbf{h}}) p(x) p(x') df dx dx'$$

$$= \iint x x' k_{\mathcal{D}}(x, x') p(x) p(x') dx dx' = \boxed{\frac{\sigma_w^2 \ell}{(2+\ell^2)^{3/2}} - \mathbf{z}^\top [k(\mathbf{x}, \mathbf{x}') + \sigma_\varepsilon^2]^{-1} \mathbf{z}.}$$

A practical illustration

Let's investigate a simple example:

Use a cubic $f(x) = x^3$. Note, that this is smooth (in agreement with our prior), but not stationary (contrary to our prior).

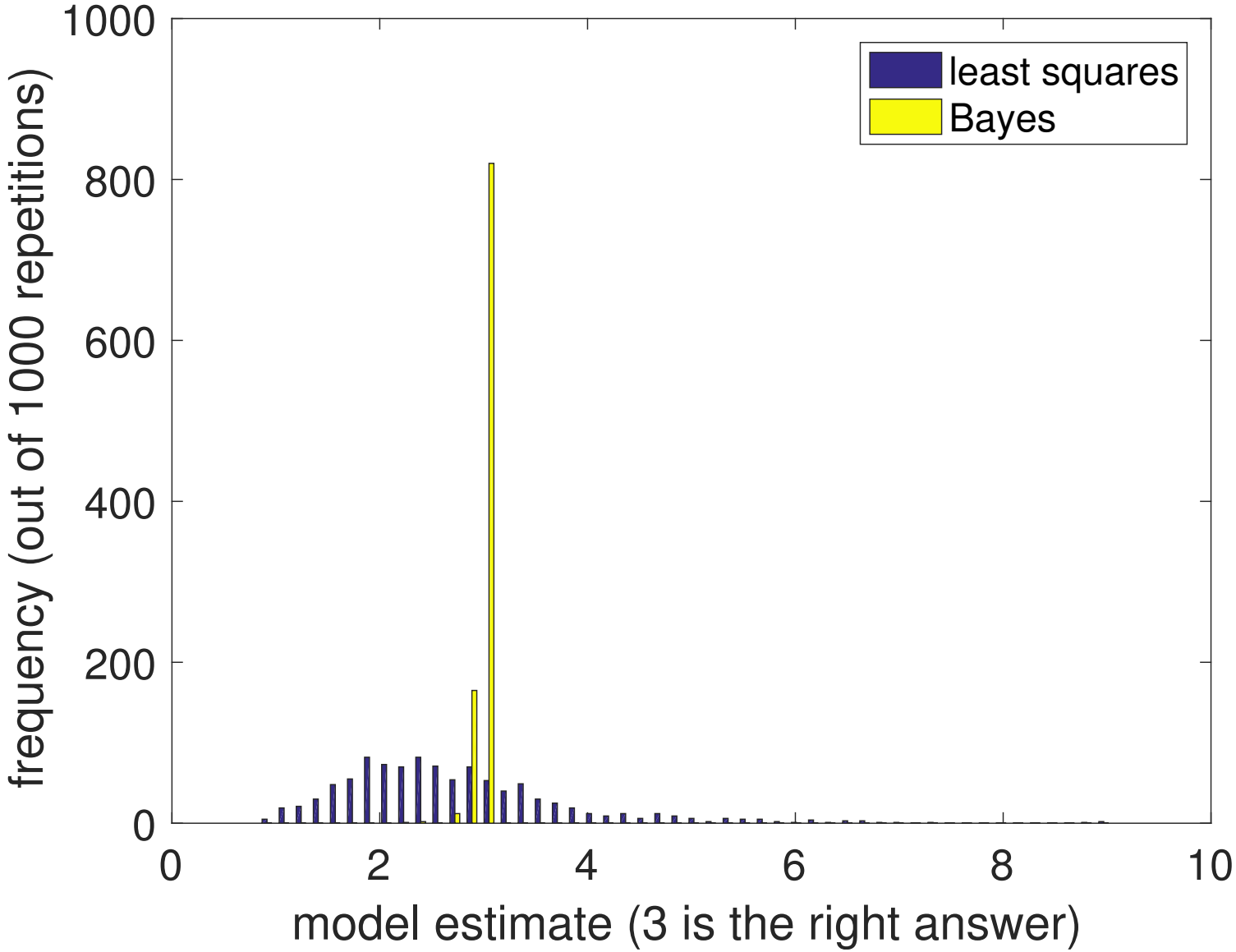
The inputs are drawn from $x \sim \mathcal{N}(0, 1)$ (in agreement with $\sigma^2_{x,\text{objective}}$).

The y values are observed with a small amount of noise $\sigma_\varepsilon = 0.001$.

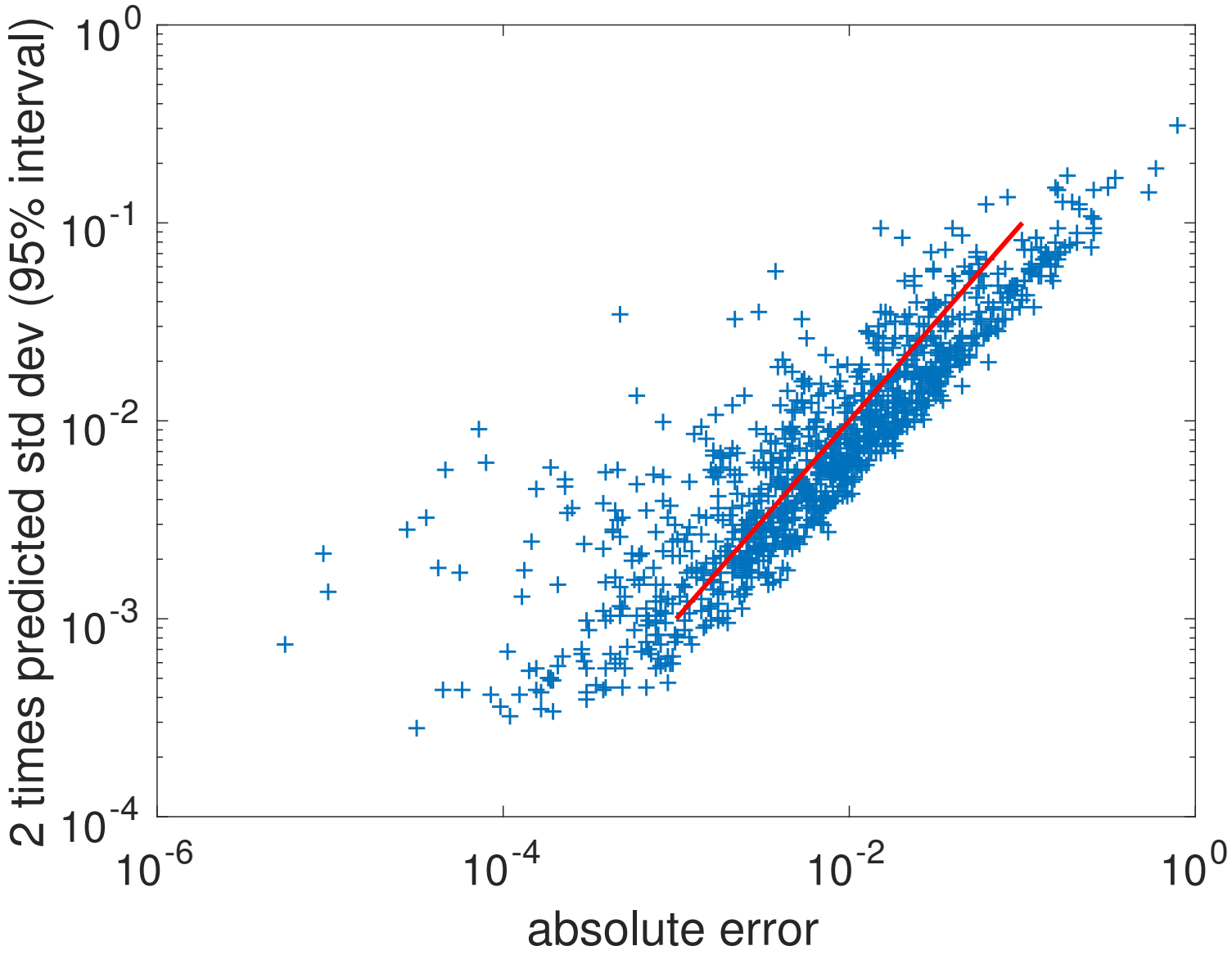
Number of available samples, $n = 25$.

Perform 1000 repetitions with data drawn at random.

Comparing predictions



The Bayesian model is well calibrated



Results

The average squared error for the Bayesian model is 400 times smaller than for Least Squares.

The Bayesian model is more complicated (8 lines of matlab).

Computational load for classical method is trivial, for the Bayesian method the CPU time is about 0.1 seconds (principally for the 3 dimensional non-linear optimization for the hyperparameters).

The Bayesian method gives you both a good guess for the value of a, as well as a well calibrated measure of confidence.

Bottom line: in this example, the Bayesian method is completely automatic, has no free parameters, is utterly practical, and highly accurate.

Take home messages . . .

- The role of the model, is different in the two approaches
  - Bayesian: the distinction between generative model and approximate model elegantly separates inference and approximations.
  - classical: using a single model entangles approximation and estimation, leading to poor performance
- the Bayesian generative model should be flexible
  - in our example, computing one number required inference over an infinite number of parameters
  - don't fear overfitting: in inference, there is no fitting
- inference and projection are conceptually simple, automatic (no user choices) and practical
- the Bayesian framework will of course also work in the particular case with the identity projection, such that the generative model and the approximate model are identical. However
  - this case usually doesn't correspond to a realistic modeling problem
  - it totally obscures the neat division between inference and approximation

. . . more take home messages

- in the Bayesian setting, it is not necessary that the training and test data come from identical distributions
- notice the role of the prior
  - notions like

    the only difference between Bayesian and classical methods is the prior, which is arbitrary anyway

    rely on fundamental misconceptions
  - our prior contained no quantitative information
  - the hierarchical specification: *smooth on some (unknown) scale* is powerful and practical
  - the prior contained absolutely no information about the desired *a*
  - the notion that I can't use Bayesian methods because I have no prior information is usually wrong