Robust Regression with online Gaussian Processes

Lehel Csató
Max-Planck Institute for Biological Cybernetics,
72076, Tübingen, Germany

July 17, 2003

Abstract

Robust regression is defined in this framework as function estimation from noisy data with additive, but *non-gaussian* noise. We present GP regression estimates assuming heavy-tailed Laplace noise. A second example is when the noise is further "worsened" (in the classical estimation theory): it is heavy-tailed and positive.

Two noise models, exponential but heavy-tailed, are presented. The first noise model has a symmetric Laplace distribution:

$$P_s(y|f_{\boldsymbol{x}}, \lambda) = \frac{\lambda}{2} \exp\left(-\lambda|y - f_{\boldsymbol{x}}|\right) \tag{1}$$

where λ is the inverse of the noise variance, y is the observed noisy output, and $f_{\boldsymbol{x}}$ is the true (de-noised) output. We also consider a model where the noise can only be positive, thus the likelihood function is:

$$P_p(y|f_{\boldsymbol{x}}, \lambda) = \begin{cases} \lambda \exp\left(-\lambda|y - f_{\boldsymbol{x}}|\right) & \text{if } y > f_{\boldsymbol{x}} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Given a likelihood function, to apply the online learning, we need to compute the average of the likelihood with respect to a *one-dimensional Gaussian* random variable $f_{\boldsymbol{x}} \sim N(\mu_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}}^2)$. To obtain the update coefficients we then need to compute the derivatives of the log-average with respect to the prior mean $\mu_{\boldsymbol{x}}$.

1 Positive noise

First we compute the average-likelihood for the non-symmetric noise model, eq. (2). We define the function g as this average:

$$g(\mu_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}}, \lambda) = \langle P_p(y|f_{\boldsymbol{x}}, \lambda)\rangle_{N(\mu_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}}^2)} = \lambda \exp\left(\theta\left(\frac{\theta}{2} - a\right)\right) \Phi\left(-\theta + a\right) \tag{3}$$

where

$$a = \frac{\mu_{\boldsymbol{x}} - y}{\sigma_{\boldsymbol{x}}}, \qquad \theta = \lambda\sigma_{\boldsymbol{x}}, \qquad \text{and} \qquad \Phi(z) = \int_{-\infty}^{z} \exp\left(-\frac{t^2}{2}\right)\frac{dt}{\sqrt{2\pi}}. \tag{4}$$

Note that the pair of parameters (θ, a) is enough for parametrising g since the noise parameter λ is fixed in this derivation, Thus, in the following we will use $g(\theta, a)$ but keep in mind the dependence $\partial_a f = \partial_\mu f / \sigma_{\boldsymbol{x}}$, coming from the definition of a.

We have the update coefficient $q^{\boldsymbol{x}}$ for the posterior mean function based on the new data $(\boldsymbol{x}, y)$ as:

$$q^{\boldsymbol{x}} = \partial_{\mu_{\boldsymbol{x}}} \log g(\theta, a) = -\lambda + \frac{\lambda}{\theta\sqrt{2\pi}} \frac{\exp\left(-\frac{(\theta-a)^2}{2}\right)}{\Phi(-\theta + a)} \tag{5}$$

and similarly one has the updates for the posterior kernel $r^{\boldsymbol{x}}$ as

$$r^{\boldsymbol{x}} = \partial^2_{\mu_{\boldsymbol{x}}} \log g(\theta, a) = \frac{\partial^2_{\mu_{\boldsymbol{x}}} g(\theta, a)}{g(\theta, a)} - (q^{\boldsymbol{x}})^2 \tag{6}$$

where

$$\frac{\partial^2_{\mu_{\boldsymbol{x}}} g(\theta, a)}{g(\theta, a)} = \lambda^2 - \frac{(\theta - a)\lambda^2}{\theta^2\sqrt{2\pi}} \frac{\exp\left(-\frac{(\theta-a)^2}{2}\right)}{\Phi(-\theta + a)}$$

When applying online learning, we iterate over the inputs $[(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)]$, at time $t < N$ having an estimate of the GP marginal as a normal distribution $N(\mu_{t+1}, \sigma^2_{t+1})$ and computing $(q^{(t+1)}, r^{(t+1)})$ based on eqs. (5) and (6).

2 Symmetric noise

The update coefficients for the symmetric noise are obtained similarly to the positive case, based on the following averaged likelihood:

$$\langle P_s(y|f_{\boldsymbol{x}}, \lambda)\rangle_{N(\mu_{\boldsymbol{x}}, \sigma^2_{\boldsymbol{x}})} = \frac{1}{2}\bigg(g(\theta, a) + g(\theta, -a)\bigg). \tag{7}$$

Repeating the deduction from the positive noise case, we have the first and second derivatives as

$$\begin{aligned} q^{\boldsymbol{x}} &= \lambda \frac{g(\theta, -a) - g(\theta, a)}{g(\theta, -a) + g(\theta, a)} \\ r^{\boldsymbol{x}} &= \lambda^2 \left[1 - \frac{1}{\sigma_{\boldsymbol{x}}\sqrt{2\pi}} \exp\left(-\frac{a^2}{2}\right) \frac{1}{g(\theta, -a) + g(\theta, a)}\right] - \left(q^{\boldsymbol{x}}\right)^2 \end{aligned}$$

3 Numerical problems

The above equations need to estimate the logarithm of the error function (Erf), which can be very unstable. In coding the Matlab implementation of the robust regression, an asymptotic expansion was used whenever the direct estimation of the log Erf function became numerically unstable. See the Matlab code for details.

4 Changing the likelihood parameters

Since all models are exponential, we can adapt the likelihood parameters. This is a multi-step procedure and it takes place *after* the online iterations: it assumes that there is an approximation to the posterior GP (obtained with fixed likelihood parameters). This is the *E-step* from the EM algorithm.

In the *M-step* then we maximise the following lower-bound to the marginal likelihood (model evidence):

$$\ln p(\mathcal{D}) \geq \int d\boldsymbol{f}\; p_{post}(\boldsymbol{f}) \ln P(\mathcal{D}|\boldsymbol{f}) \tag{8}$$

which again involves only one-dimensional integrals (i.i.d. data were assumed), which leads to the following values for the likelihood parameters:

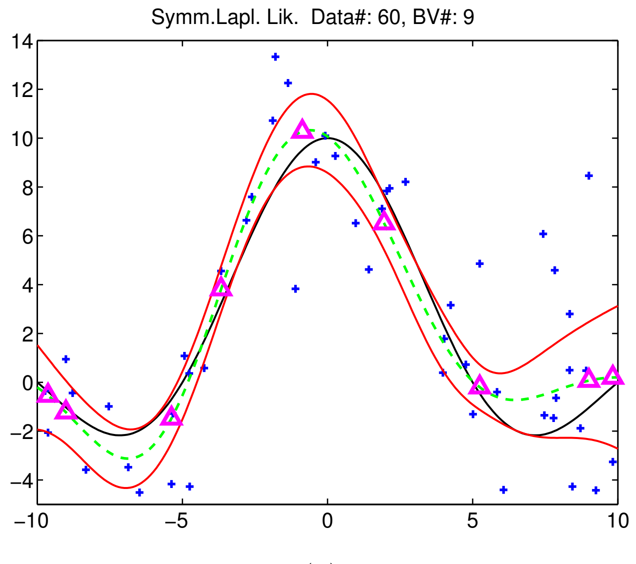
$$\begin{aligned} \frac{N}{\lambda_p} &= \sum_{n=1}^{N} \langle H(y_n - f_n) \rangle_{N(\mu_n, \sigma_n^2)} \\ \frac{N}{\lambda_s} &= \sum_{n=1}^{N} \langle |y_n - f_n| \rangle_{N(\mu_n, \sigma_n^2)} \end{aligned}$$

where $f_n \sim N(\mu_n, \sigma_n^2)$ is the marginal of the posterior GP at $\boldsymbol{x}_n$ and $H(t)$ is the step function.

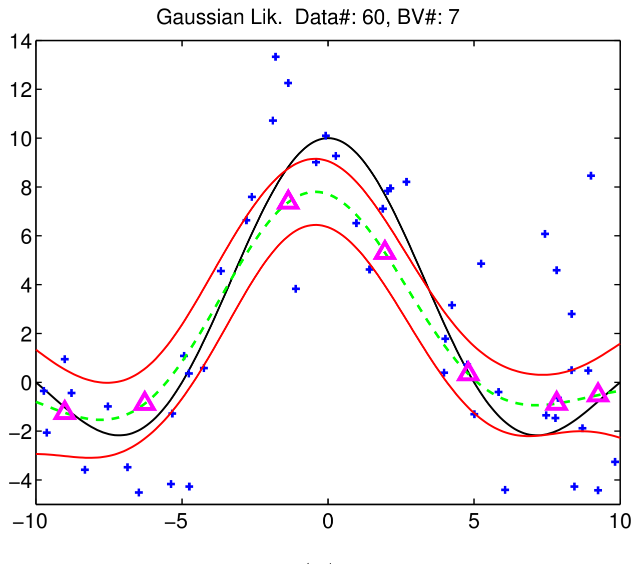
5 Examples

In the following the regression with robust models is demonstrated on a toy example: the estimation of the noisy sinc function.

The estimation was compared with Gaussian noise assumption, thus involving three models for the likelihood function for which we can estimate the noise (see the EM algorithm from the previous section). See figure captions for more explanation.

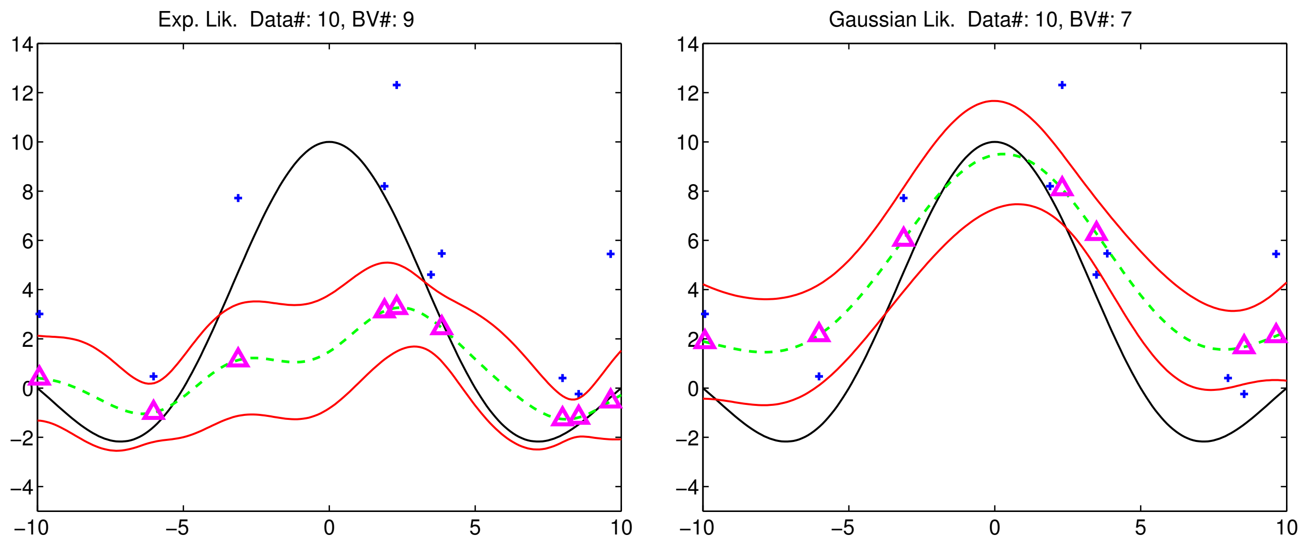


(a)

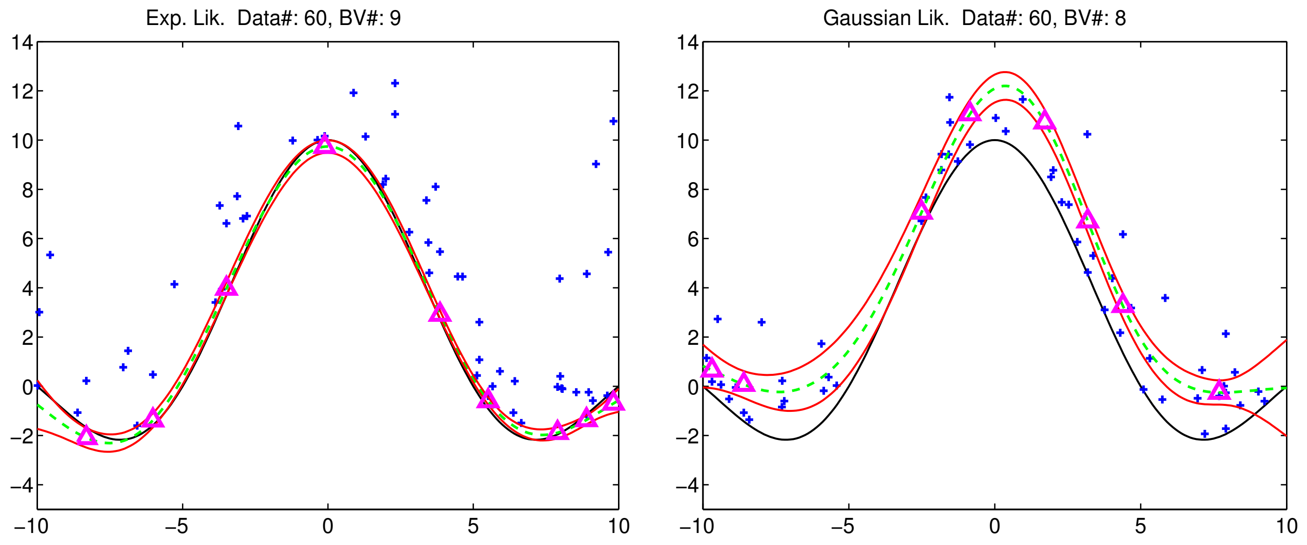


(b)

Figure 1: Comparing regression estimates with symmetric Laplace (a) and Gaussian (b) likelihoods. The true noise is symmetric Laplace ($\lambda = 0.3$). The algorithm estimates the length-scale and amplitude of the GP kernel (model parameters) and the noise of the likelihood. **Subfigure (a):** regression estimation with the true noise model. The error bars are the Bayesian error bars, specifying the variation of the latent variables which cannot be directly translated into error bars on the *outputs* y: the outputs have heavy tails. **Subfigure (b):** regression estimation with Gaussian noise assumption. With approximately the same and amplitude of the GP kernel function (determined using training), the noise estimation is $\sigma_0^2 \approx 35$ which is a crude over-estimation of the actual variance of the outputs.



(a) (b)



(c) (d)

Figure 2: Comparing regression estimates with positive exponential (a) and Gaussian (b) likelihoods. The true noise is a positive exponential ($\lambda = 0.4$). Similarly to Fig. 1, the algorithm estimates the length-scale and amplitude of the GP kernel (model parameters) and the noise of the likelihood. **Subfigure (a):** regression estimation with the true noise model. The error bars are the Bayesian error bars, specifying the variation of the latent variables which cannot be directly translated into error bars on the *outputs* y: the outputs have heavy tails. **Subfigure (b):** regression estimation with Gaussian noise assumption. With approximately the same and amplitude of the GP kernel function (determined by the training procedure), the noise estimate is $\sigma_0^2 \approx 3$. *Subfigures (c) and (d):* the regression functions for more data, which allows more exact estimation of the true function. Notice that the *Bayesian uncertainty* for the correct model (left) shrinks more, reflecting more evidence.