Inclass 15: Bayesian Inference II

[SCS4049] Machine Learning and Data Science

Seongsik Park (s.park@dgu.edu)

AI Department, Dongguk University

Bayesian linear regression

Let $\mathcal{S} = \{(x^{(i)}, y^{(i)}\}_{i=1}^{n}$ be a training set of i.i.d. examples from some unknown distribution. The standard probabilistic interpretation of linear regression states that

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)}, \quad i = 1, ..., n \tag{1}$$

where $\epsilon^{(i)}$ are i.i.d. white noise variables with independent $\mathcal{N}(0, \sigma^2)$ distributions. It follows that $y^{(i)} - \theta^T x^{(i)} \sim \mathcal{N}(0, \sigma^2)$, or equivalently,

$$P(y^{(i)} \mid x^{(i)}, \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right). \tag{2}$$

For notational convenience, we define

$$X = \begin{bmatrix} — \ (x^{(1)})^T \ — \\ — \ (x^{(2)})^T \ — \\ \vdots \\ — \ (x^{(n)})^T \ — \end{bmatrix} \in \mathcal{R}^{n \times d} \quad \vec{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix} \in \mathcal{R}^n \quad \vec{\epsilon} = \begin{bmatrix} \epsilon^{(1)} \\ \epsilon^{(2)} \\ \vdots \\ \epsilon^{(n)} \end{bmatrix} \in \mathcal{R}^n \tag{3}$$

Bayesian linear regression

In Bayesian linear regression, we assume that a **prior distribution** over parameters is also given; a typical choice, for instance, is $\theta \sim \mathcal{N}(0, \tau^2 I)$. Using Bayes' rule, we obtain the **parameter posterior**,

$$p(\theta \mid \mathcal{S}) = \frac{p(\theta)p(\mathcal{S} \mid \theta)}{\int_{\theta'} p(\theta')p(\mathcal{S} \mid \theta')d\theta'} = \frac{p(\theta)\prod_{i=1}^{n} p(y^{(i)} \mid x^{(i)}, \theta)}{\int_{\theta'} p(\theta')\prod_{i=1}^{n} p(y^{(i)} \mid x^{(i)}, \theta')d\theta'}. \tag{4}$$

Assuming the same noise model on testing points as on our training points, the "output" of Bayesian linear regression on a new test point x_* is not just a single guess "y_*", but rather an entire probability distribution over possible outputs, known as the **posterior predictive distribution**:

$$p(y_* \mid x_*, \mathcal{S}) = \int_{\theta} p(y_* \mid x_*, \theta)p(\theta \mid \mathcal{S})d\theta. \tag{5}$$

For many types of models, the integrals are difficult to compute, and hence, we often resort to approximations, such as MAP estimation.

Bayesian linear regression

In the case of Bayesian linear regression, however, the integrals actually are *tractable*! In particular, for Bayesian linear regression, one can show that

$$\theta \mid \mathcal{S} \sim \mathcal{N}\left(\frac{1}{\sigma^2} A^{-1} X^T \vec{y}, A^{-1}\right) \tag{6}$$

$$y_* \mid x_*, \mathcal{S} \sim \mathcal{N}\left(\frac{1}{\sigma^2} x_*^T A^{-1} X^T \vec{y}, x_*^T A^{-1} x_* + \sigma^2\right) \tag{7}$$

where $A = \frac{1}{\sigma^2} X^T X + \frac{1}{\tau^2} I$.

Bayesian linear regression

The derivation of these formulas is somewhat involved. Nonetheless, from these equations, we get at least a flavor of what Bayesian methods are all about: the posterior distribution over the test output y_* for a test input x_* is a Gaussian distribution–this distribution reflects the uncertainty in our predictions $y_* = \theta^T x_* + \epsilon_*$ arising from both the randomness in ϵ_* and the uncertainty in our choice of parameters θ. In contrast, classical probabilistic linear regression models estimate parameters θ directly from the training data but provide no estimate of how reliable these learned parameters may be (see Figure 1).

Bayesian linear regression

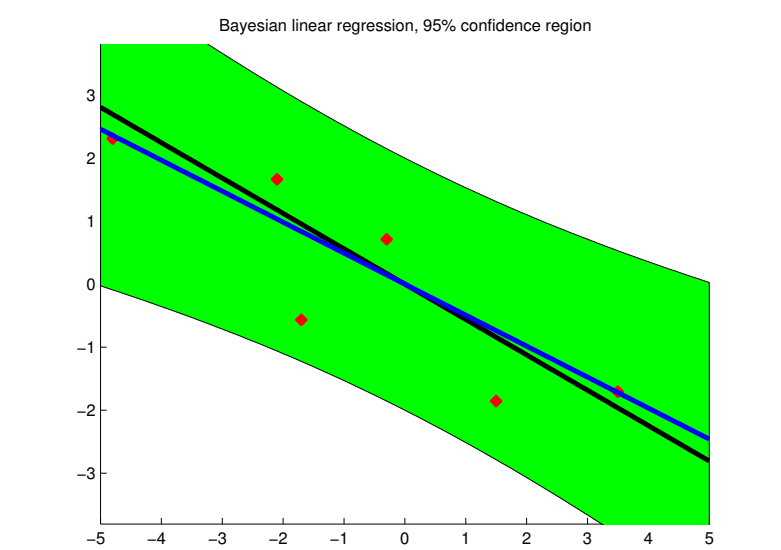

Figure 1: Bayesian linear regression for a one-dimensional linear regression problem, $y^{(i)} = \theta x^{(i)} + \epsilon^{(i)}$, with $\epsilon^{(i)} \sim \mathcal{N}(0, 1)$ i.i.d. noise. The green region denotes the 95% confidence region for predictions of the model. Note that the (vertical) width of the green region is largest at the ends but narrowest in the middle. This region reflects the uncertain in the estimates for the parameter θ. In contrast, a classical linear regression model would display a confidence region of constant width, reflecting only the $\mathcal{N}(0, \sigma^2)$ noise in the outputs.

Learning a distribution over functions: two approaches

Parametric models

- Parameterized model $f_\theta(x)$ with assuming a prior for θ
- Learn a posterior distribution over parameters $p(\theta \mid \mathcal{D})$ to represent a distribution over functions $f_\theta(x)$
- Randomness in θ
- Bayesian linear regression

Nonparametric models

- Stochastic process or random functions $f(x)$
- Learn a distribution over functions directly from data
- Randomness in $f(x)$
- Gaussian process regression

Bayesian regression: parameteric vs. nonparametric

Given a set of training examples, $\mathcal{D} = \{(x_n, y_n) \mid n = 1, ..., N\}$, the goal of Bayesian regression is to make a prediction given new input x_*, computing $p(y_* \mid x_*, \mathcal{D})$.

Parametric approach

- Model $x_n, y_n \mid \theta \sim p(x, y \mid \theta)$, assuming a parametric representation for $f(\cdot) = f_\theta(\cdot)$
- Prior over parameters: $p(\theta)$
- Posterior over parameters:

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta)p(\theta)}{p(\mathcal{D})} \tag{8}$$

- Prediction is computed by

$$p(y_* \mid x_*, \mathcal{D}) = \int p(y_* \mid x_*, \theta)p(\theta \mid \mathcal{D})d\theta \tag{9}$$

Bayesian regression: parameteric vs. nonparametric

Given a set of training examples, $\mathcal{D} = \{(x_n, y_n) \mid n = 1, ..., N\}$, the goal of Bayesian regression is to make a prediction given new input x_*, computing $p(y_* \mid x_*, \mathcal{D})$.

Nonparametric approach

- Model $x_n, y_n \sim f$, without parametric representations for $f(\cdot)$
- Prior over function: $f \sim p(f)$
- Posterior over function:

$$p(f \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid f)p(f)}{p(\mathcal{D})} \tag{10}$$

- Prediction is computed by

$$p(y_* \mid x_*, \mathcal{D}) = \int p(y_* \mid x_*, f)p(f \mid \mathcal{D})df \tag{11}$$