Machine Learning

Bayesian Regression & Classification

learning as inference, Bayesian Kernel Ridge regression & Gaussian Processes, Bayesian Kernel Logistic Regression & GP classification, Bayesian Neural Networks

Marc Toussaint
U Stuttgart
Learning as Inference

- The parameteric view

\[
P(\beta|\text{Data}) = \frac{P(\text{Data}|\beta) P(\beta)}{P(\text{Data})}
\]

- The function space view

\[
P(f|\text{Data}) = \frac{P(\text{Data}|f) P(f)}{P(\text{Data})}
\]

- Today:
  - Bayesian (Kernel) Ridge Regression $\leftrightarrow$ Gaussian Process (GP)
  - Bayesian (Kernel) Logistic Regression $\leftrightarrow$ GP classification
  - Bayesian Neural Networks (briefly)
Beyond learning about specific Bayesian learning methods:

Understand relations between

\[
\text{loss/error} \leftrightarrow \text{neg-log likelihood}
\]

\[
\text{regularization} \leftrightarrow \text{neg-log prior}
\]

\[
\text{cost (reg.+loss)} \leftrightarrow \text{neg-log posterior}
\]
Ridge regression as Bayesian inference

- We have random variables $X_{1:n}, Y_{1:n}, \beta$

- We observe data $D = \{(x_i, y_i)\}_{i=1}^n$ and want to compute $P(\beta \mid D)$

- Let’s assume:
  $P(X)$ is arbitrary
  $P(\beta)$ is Gaussian: $\beta \sim \mathcal{N}(0, \frac{\sigma^2}{\lambda}) \propto e^{-\frac{\lambda}{2\sigma^2} \|\beta\|^2}$
  $P(Y \mid X, \beta)$ is Gaussian: $y = x^T \beta + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2)$
Ridge regression as Bayesian inference

- Bayes’ Theorem:

\[
P(\beta \mid D) = \frac{P(D \mid \beta) P(\beta)}{P(D)}
\]

\[
P(\beta \mid x_{1:n}, y_{1:n}) = \frac{\prod_{i=1}^{n} P(y_i \mid \beta, x_i) P(\beta)}{Z}
\]

\(P(D \mid \beta)\) is a \textit{product} of independent likelihoods for each observation \((x_i, y_i)\)
Ridge regression as Bayesian inference

- Bayes’ Theorem:

\[
P(\beta \mid D) = \frac{P(D \mid \beta) P(\beta)}{P(D)}
\]

\[
P(\beta \mid x_1:n, y_1:n) = \frac{\prod_{i=1}^{n} P(y_i \mid \beta, x_i) P(\beta)}{Z}
\]

\(P(D \mid \beta)\) is a product of independent likelihoods for each observation \((x_i, y_i)\)

Using the Gaussian expressions:

\[
P(\beta \mid D) = \frac{1}{Z'} \prod_{i=1}^{n} e^{-\frac{1}{2\sigma^2} (y_i - x_i^T \beta)^2} e^{-\frac{\lambda}{2\sigma^2} \|\beta\|^2}
\]
Ridge regression as Bayesian inference

- **Bayes’ Theorem:**
  
  \[
  P(\beta | D) = \frac{P(D | \beta) P(\beta)}{P(D)}
  \]
  
  \[
  P(\beta | x_{1:n}, y_{1:n}) = \frac{\prod_{i=1}^{n} P(y_i | \beta, x_i) P(\beta)}{Z}
  \]

  \(P(D | \beta)\) is a product of independent likelihoods for each observation \((x_i, y_i)\)

Using the Gaussian expressions:

\[
P(\beta | D) = \frac{1}{Z'} \prod_{i=1}^{n} e^{-\frac{1}{2\sigma^2} (y_i - x_i^\top \beta)^2} e^{-\frac{\lambda}{2\sigma^2} \|\beta\|^2}
\]

\[
- \log P(\beta | D) = \frac{1}{2\sigma^2} \left[ \sum_{i=1}^{n} (y_i - x_i^\top \beta)^2 + \lambda \|\beta\|^2 \right] - \log Z'
\]

\[
- \log P(\beta | D) \propto L^{\text{ridge}}(\beta)
\]

**1st insight:** The **neg-log posterior** \(P(\beta | D)\) is equal to the cost function \(L^{\text{ridge}}(\beta)\)!
Ridge regression as Bayesian inference

- Let us compute \( P(\beta | D) \) explicitly:

\[
P(\beta | D) = \frac{1}{Z'} \prod_{i=1}^{n} e^{-\frac{1}{2\sigma^2} (y_i - x_i^T \beta)^2} e^{-\frac{\lambda}{2\sigma^2} \|\beta\|^2}
\]

\[
= \frac{1}{Z'} e^{-\frac{1}{2\sigma^2} \sum_i (y_i - x_i^T \beta)^2} e^{-\frac{\lambda}{2\sigma^2} \|\beta\|^2}
\]

\[
= \frac{1}{Z'} e^{-\frac{1}{2\sigma^2} [(y - X\beta)^T (y - X\beta) + \lambda \beta^T \beta]}
\]

\[
= \frac{1}{Z'} e^{-\frac{1}{2} \left[ \frac{1}{\sigma^2} y^T y + \frac{1}{\sigma^2} \beta^T (X^T X + \lambda I) \beta - \frac{2}{\sigma^2} \beta^T X^T y \right]}
\]

\[
= \mathcal{N}(\beta | \hat{\beta}, \Sigma)
\]

This is a Gaussian with covariance and mean

\[
\Sigma = \sigma^2 (X^T X + \lambda I)^{-1} , \quad \hat{\beta} = \frac{1}{\sigma^2} \Sigma X^T y = (X^T X + \lambda I)^{-1} X^T y
\]

- **2nd insight:** The mean \( \hat{\beta} \) is exactly the classical \( \text{argmin}_\beta L_{\text{ridge}}(\beta) \).

- **3rd insight:** The Bayesian inference approach not only gives a mean/optimal \( \hat{\beta} \), but also a variance \( \Sigma \) of that estimate!
Predicting with an uncertain $\beta$

- Suppose we want to make a prediction at $x$. We can compute the **predictive distribution** over a new observation $y^*$ at $x^*$:

$$P(y^* \mid x^*, D) = \int_{\beta} P(y^* \mid x^*, \beta) \, P(\beta \mid D) \, d\beta$$

$$= \int_{\beta} N(y^* \mid \phi(x^*)^\top \beta, \sigma^2) \, N(\beta \mid \hat{\beta}, \Sigma) \, d\beta$$

$$= N(y^* \mid \phi(x^*)^\top \hat{\beta}, \sigma^2 + \phi(x^*)^\top \Sigma \phi(x^*))$$

Note $P(f(x) \mid D) = N(f(x) \mid \phi(x)^\top \hat{\beta}, \phi(x)^\top \Sigma \phi(x))$ without the $\sigma^2$

- So, $y^*$ is Gaussian distributed around the mean prediction $\phi(x^*)^\top \hat{\beta}$:
Wrapup of Bayesian Ridge regression

- **1st insight:** The *neg-log posterior* $P(\beta \mid D)$ is equal to the cost function $L^{\text{ridge}}(\beta)$!

  This is a very very common relation: optimization costs correspond to neg-log probabilities; probabilities correspond to exp-neg costs.

- **2nd insight:** The mean $\hat{\beta}$ is exactly the classical $\arg\min_\beta L^{\text{ridge}}(\beta)$.

  More generally, the most likely parameter $\arg\max_\beta P(\beta \mid D)$ is also the least-cost parameter $\arg\min_\beta L(\beta)$. In the Gaussian case, mean and most-likely coincide.

- **3rd insight:** The Bayesian inference approach not only gives a mean/optimal $\hat{\beta}$, but also a variance $\Sigma$ of that estimate!

  This is a core benefit of the Bayesian view: It naturally provides a probability distribution over predictions (“error bars”), not only a single prediction.
Kernelized Bayesian Ridge Regression

- As in the classical case, we can consider arbitrary features $\phi(x)$
- .. or directly use a kernel $k(x, x')$:

$$
P(f(x) \mid D) = \mathcal{N}(f(x) \mid \phi(x)\top \hat{\beta}, \, \phi(x)\top \Sigma \phi(x))$$

$$
\phi(x)\top \hat{\beta} = \phi(x)\top X\top (XX\top + \lambda I)^{-1} y
$$

$$
= \kappa(x)(K + \lambda I)^{-1} y
$$

$$
\phi(x)\top \Sigma \phi(x) = \phi(x)\top \sigma^2 \, (X\top X + \lambda I)^{-1} \phi(x)
$$

$$
= \frac{\sigma^2}{\lambda} \phi(x)\top \phi(x) - \frac{\sigma^2}{\lambda} \phi(x)\top X (XX\top + \lambda I_k)^{-1} X\top \phi(x)
$$

$$
= \frac{\sigma^2}{\lambda} \kappa(x, x) - \frac{\sigma^2}{\lambda} \kappa(x)(K + \lambda I_n)^{-1} \kappa(x)
$$

3rd line: As on slide 02:24
last lines: Woodbury identity $(A + UBV)^{-1} = A^{-1} - A^{-1}U(B^{-1} + VA^{-1}U)^{-1}VA^{-1}$
with $A = \lambda I$

- In standard conventions $\lambda = \sigma^2$, $P(\beta) = \mathcal{N}(\beta \mid 0, 1)$
  - Regularization: scale the covariance function (or features)
Kernelized Bayesian Ridge Regression

is equivalent to Gaussian Processes
(see also Welling: “Kernel Ridge Regression” Lecture Notes; Rasmussen & Williams sections 2.1 & 6.2; Bishop sections 3.3.3 & 6)

- As we have the equations already, I skip further math details. (See Rasmussen & Williams)
Gaussian Processes

• The function space view

\[ P(f|\text{Data}) = \frac{P(\text{Data}|f) \, P(f)}{P(\text{Data})} \]

• Gaussian Processes define a probability distribution over functions:
  - A function is an infinite dimensional thing – how could we define a Gaussian distribution over functions?
  - For every finite set \( \{x_1, \ldots, x_M\} \), the function values \( f(x_1), \ldots, f(x_M) \) are Gaussian distributed with mean and cov.

\[
\begin{align*}
\langle f(x_i) \rangle & = \mu(x_i) \quad \text{(often zero)} \\
\langle [f(x_i) - \mu(x_i)][f(x_j) - \mu(x_j)] \rangle & = k(x_i, x_j)
\end{align*}
\]

Here, \( k(\cdot, \cdot) \) is called **covariance function**

• Second, Gaussian Processes define an observation probability

\[ P(y|x, f) = \mathcal{N}(y|f(x), \sigma^2) \]
Gaussian Processes

(a), prior

(b), posterior

(from Rasmussen & Williams)
GP: different covariance functions

- These are examples from the $\gamma$-exponential covariance function

$$k(x, x') = \exp\{-|x - x'|/l|^\gamma\}$$
GP: derivative observations

(from Rasmussen & Williams)
• Bayesian Kernel Ridge Regression = Gaussian Process

• GPs have become a standard regression method

• If exact GP is not efficient enough, many approximations exist, e.g. sparse and pseudo-input GPs
Bayesian (Ridge) Logistic Regression
Bayesian Logistic Regression

- $f$ now defines a logistic probability over $y \in \{0, 1\}$:

$$P(X) = \text{arbitrary}$$

$$P(\beta) = \mathcal{N}(\beta|0, \frac{2}{\lambda}) \propto \exp\{-\lambda\|\beta\|^2\}$$

$$P(Y = 1 \mid X, \beta) = \sigma(\beta^T \phi(x))$$

- Recall

$$L_{\text{logistic}}(\beta) = -\sum_{i=1}^{n} \log p(y_i \mid x_i) + \lambda\|\beta\|^2$$

- Again, the parameter posterior is

$$P(\beta|D) \propto P(D \mid \beta) \ P(\beta) \propto \exp\{-L_{\text{logistic}}(\beta)\}$$
Bayesian Logistic Regression

- Use **Laplace approximation** (2nd order Taylor for $L$) at $\beta^* = \arg\min_\beta L(\beta)$:

$$L(\beta) \approx L(\beta^*) + \bar{\beta}^\top \nabla + \frac{1}{2} \bar{\beta}^\top H \bar{\beta}, \quad \bar{\beta} = \beta - \beta^*$$

$$P(\beta|D) \propto \exp\{-\bar{\beta}^\top \nabla - \frac{1}{2} \bar{\beta}^\top H \bar{\beta}\}$$

$$= \mathcal{N}[\bar{\beta}|-\nabla, H] = \mathcal{N}(\bar{\beta}|-H^{-1}\nabla, H^{-1})$$

$$= \mathcal{N}(\beta|\beta^*, H^{-1}) \quad \text{(because } \nabla = 0 \text{ at } \beta^*)$$

- Then the predictive distribution of the **discriminative function** is also Gaussian!

$$P(f(x)|D) = \int_\beta P(f(x)|\beta) \ P(\beta|D) \ d\beta$$

$$= \int_\beta \mathcal{N}(f(x)|\phi(x)^\top \beta, 0) \ \mathcal{N}(\beta|\beta^*, H^{-1}) \ d\beta$$

$$= \mathcal{N}(f(x)|\phi(x)^\top \beta^*, \phi(x)^\top H^{-1} \phi(x)) =: \mathcal{N}(f(x)|f^*, s^2)$$

- The predictive distribution over the label $y \in \{0, 1\}$:

$$P(y(x)=1|D) = \int_{f(x)} \sigma(f(x)) \ P(f(x)|D) \ df$$

$$\approx \varphi(\sqrt{1 + s^2 \pi/8 f^*})$$

the approximation replaced $\sigma$ by the probit function $\varphi(x) = \int_{-\infty}^{x} \mathcal{N}(0, 1) dx$. 

18/24
Kernelized Bayesian Logistic Regression

- As with Kernel Logistic Regression, the MAP discriminative function $f^*$ can be found iterating the Newton method $\leftrightarrow$ iterating GP estimation on a re-weighted data set.
- The rest is as above.
Kernel Bayesian Logistic Regression

is equivalent to Gaussian Process Classification

- GP classification became a standard classification method, if the prediction needs to be a meaningful probability that takes the model uncertainty into account.
Bayesian Neural Networks
General non-linear models

- Above we always assumed $f(x) = \phi(x)^\top \beta$ (or kernelized)
- Bayesian Learning also works for non-linear function models $f(x, \beta)$
- Regression case:
  $P(X)$ is arbitrary.
  $P(\beta)$ is Gaussian: $\beta \sim \mathcal{N}(0, \frac{\sigma^2}{\lambda}) \propto e^{-\frac{\lambda}{2\sigma^2} \|\beta\|^2}$
  $P(Y \mid X, \beta)$ is Gaussian: $y = f(x, \beta) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2)$
General non-linear models

- To compute $P(\beta|D)$ we first compute the most likely
  \[
  \beta^* = \operatorname{argmin}_\beta L(\beta) = \operatorname{argmax}_\beta P(\beta|D)
  \]

- Use Laplace approximation around $\beta^*$: 2nd-order Taylor of $f(x, \beta)$ and then of $L(\beta)$ to estimate a Gaussian $P(\beta|D)$

- Neural Networks:
  - The Gaussian prior $P(\beta) = \mathcal{N}(\beta|0, \frac{\sigma^2}{\lambda})$ is called weight decay
  - This pushes “sigmoids to be in the linear region”.
Conclusions

• Probabilistic inference is a very powerful concept!
  – Inferring about the world given data
  – Learning, decision making, reasoning can view viewed as forms of (probabilistic) inference

• We introduced Bayes’ Theorem as the fundamental form of probabilistic inference

• Marrying Bayes with (Kernel) Ridge (Logistic) regression yields
  – Gaussian Processes
  – Gaussian Process classification