732A96/TDDE15 Advanced Machine Learning

Gaussian Process Regression and Classification

Jose M. Peña
IDA, Linköping University, Sweden

Lectures 12: Gaussian Process Classification

Contents

- Linear Logistic Regression
- Bayesian Linear Logistic Regression
- Gaussian Process Classification
- Laplace Approximation
- Gaussian Process Classification: Iris Data

Literature

- Main source
  - Rasmussen, C. E. and Williams, K. I. *Gaussian Processes for Machine Learning*. MIT Press, 2006. Chapters 3.1-3.4.1 and 3.7.
- Additional source
  - Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006. Chapters 6.4.5-6.4.6.

Linear Logistic Regression

- Consider a binary classification problem $y \in \{-1, +1\}$. Then,

$$p(y=+1|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|y=+1)p(y=+1)}{p(\boldsymbol{x}|y=+1)p(y=+1) + p(\boldsymbol{x}|y=-1)p(y=-1)}$$
$$= \frac{1}{1+\exp(-s(\boldsymbol{x}))} = \sigma(s(\boldsymbol{x})).$$

  where $s(\boldsymbol{x}) = \log \frac{p(\boldsymbol{x}|y=+1)p(y=+1)}{p(\boldsymbol{x}|y=-1)p(y=-1)} = \log \frac{p(y=+1|\boldsymbol{x})}{p(y=-1|\boldsymbol{x})}$, and σ is called logistic sigmoid function.
- We assume that $p(\boldsymbol{x}|y)$ is a member of the exponential family (e.g., Gaussian, multinomial), which implies that $s(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{w}$. The model $p(y=+1|\boldsymbol{x},\boldsymbol{w}) = \sigma(\boldsymbol{x}^T\boldsymbol{w})$ is called logistic regression.
- Given some training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)|i=1,\ldots,n\} = (X, \boldsymbol{y})$, we determine the parameters $\boldsymbol{w}$ by maximizing the log lik function:

$$\log p(\boldsymbol{y}|X,\boldsymbol{w}) = -\sum_{i=1}^{n}[y_i \log\sigma(\boldsymbol{x}_i^T\boldsymbol{w}) + (1-y_i)(1-\log\sigma(\boldsymbol{x}_i^T\boldsymbol{w}))].$$

- No closed form solution exists, but the log lik function is concave and thus easy to maximize via gradient ascent. The gradient of the log lik function wrt $\boldsymbol{w}$ is $\sum_{i=1}^{n}(\sigma(\boldsymbol{x}_i^T\boldsymbol{w}) - y_i)\boldsymbol{x}_i$.
- Beware of overfitting for linearly separable datasets: Log lik maximization causes $|\boldsymbol{w}|$ to tend to infinity, i.e. the sigmoid function becomes a Heaviside step function.

Bayesian Linear Logistic Regression

- Prior distribution: $\boldsymbol{w} \sim \mathcal{N}(0, \Sigma_p)$, e.g. ridge regression $\Sigma_p = \alpha^{-1} I$.
- Posterior distribution:

$$\log p(\boldsymbol{w}|X, \boldsymbol{y}) \propto -\frac{1}{2}\boldsymbol{w}^T \Sigma_p^{-1} \boldsymbol{w} + \sum_{i=1}^{n} \log \sigma(y_i f_i)$$

  where $f_i = \boldsymbol{x}_i^T \boldsymbol{w}$.

- No closed form solution exists, but the penalty term is quadratic on $\boldsymbol{w}$ and thus the log posterior is concave and thus easy to maximize via gradient ascent or related methods.
- A full Bayesian approach uses the predictive distribution:

$$p(y_* = +1|\boldsymbol{x}_*, X, \boldsymbol{y}) = \int p(y_* = +1|\boldsymbol{x}_*, \boldsymbol{w}) p(\boldsymbol{w}|X, \boldsymbol{y}) d\boldsymbol{w}.$$

- No closed form expression for the predictive distribution exists.
- The above carries over to multi-class classification problems by using the multiple logistic function, a.k.a. softmax.

Gaussian Process Classification

- Given a test case $\boldsymbol{x}_*$, use a GP for regression to predict a real number f_* that is then "squashed" through the sigmoid logistic function to produce a class label $y_* = \sigma(f_*)$.
- However, the training data only include class labels $\boldsymbol{y}$ and, thus, $\boldsymbol{f}$ are latent variables.
- In other words, prediction occurs in two steps:
  - Compute the distribution of the latent variable f_*:

$$p(f_*|\boldsymbol{x}_*, X, \boldsymbol{y}) = \int p(f_*, \boldsymbol{f}|\boldsymbol{x}_*, X, \boldsymbol{y}) d\boldsymbol{f}.$$

  - Compute the prediction y_*, since the latent variable f_* is uninteresting:

$$p(y_* = +1|\boldsymbol{x}_*, X, \boldsymbol{y}) = \int \sigma(f_*) p(f_*|\boldsymbol{x}_*, X, \boldsymbol{y}) df_*.$$

  - No closed form solutions exist for these integrals. Solutions: Laplace approximation or MC sampling.

Laplace Approximation

- Computing the distribution of the latent variable can be rewritten as

$$p(f_*|\boldsymbol{x}_*, X, \boldsymbol{y}) = \int p(f_*, \boldsymbol{f}|\boldsymbol{x}_*, X, \boldsymbol{y}) d\boldsymbol{f} = \int p(f_*|\boldsymbol{x}_*, X, \boldsymbol{f}) p(\boldsymbol{f}|X, \boldsymbol{y}) d\boldsymbol{f}$$

  where

  - the first term is $\mathcal{N}(K(\boldsymbol{x}_*, X)K(X,X)^{-1}\boldsymbol{f}, K(\boldsymbol{x}_*, \boldsymbol{x}_*) - K(\boldsymbol{x}_*, X)K(X,X)^{-1}K(X, \boldsymbol{x}_*))$ since it is a GP for regression, and
  - the second term is Gaussian around its modes as the size of the training data grows, due to the central limit theorem. Therefore, it can be approximated by a second order Taylor expansion around a mode via Laplace's method:

$$p(\boldsymbol{f}|X, \boldsymbol{y}) \approx \mathcal{N}(\hat{\boldsymbol{f}}, A^{-1})$$

    where $\hat{\boldsymbol{f}} = \arg\max_{\boldsymbol{f}} p(\boldsymbol{f}|X, \boldsymbol{y})$ and $A = -\nabla\nabla \log p(\boldsymbol{f}|X, \boldsymbol{y})|_{\boldsymbol{f}=\hat{\boldsymbol{f}}}$.

- Laplace's method to approximate $p(u) = \frac{1}{Z}q(u)$:
  - Find a mode u_0 of $\log q$, i.e. $\frac{\partial}{\partial u} \log q(u)|_{u=u_0} = 0$ (numerical methods are typically used).
  - Consider a second order Taylor expansion of $\log q$ centered at u_0 (second order because a Gaussian distribution is quadratic in the variables):

$$\log q(u) \approx \log q(u_0) - \frac{1}{2}A(u - u_0)^2$$

    where $A = \frac{\partial^2}{\partial u^2} \log q(u)|_{u=u_0}$ (the first order term is gone because u_0 is a mode of q).
  - Then, $q(u) \approx q(u_0) exp[-\frac{1}{2}A(u - u_0)^2]$ and thus $p(u) \approx \mathcal{N}(u_0, A^{-1})$.

Laplace Approximation

- Computing the distribution of the latent variable can be rewritten as

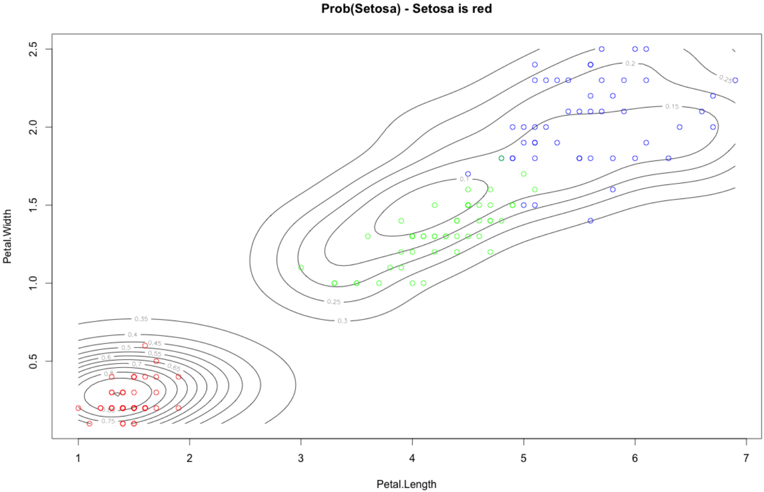
  $$p(f_*|\boldsymbol{x}_*, X, \boldsymbol{y}) = \int p(f_*, \boldsymbol{f}|\boldsymbol{x}_*, X, \boldsymbol{y})d\boldsymbol{f} = \int p(f_*|\boldsymbol{x}_*, X, \boldsymbol{f})p(\boldsymbol{f}|X, \boldsymbol{y})d\boldsymbol{f}$$

  where

  - the first term is $\mathcal{N}(K(\boldsymbol{x}_*, X)K(X, X)^{-1}\boldsymbol{f}, K(\boldsymbol{x}_*, \boldsymbol{x}_*) - K(\boldsymbol{x}_*, X)K(X, X)^{-1}K(X, \boldsymbol{x}_*))$ since it is a GP for regression, and
  - the second term is approximated by $\mathcal{N}(\hat{\boldsymbol{f}}, A^{-1})$ where $\hat{\boldsymbol{f}} = \arg\max_{\boldsymbol{f}} p(\boldsymbol{f}|X, \boldsymbol{y})$ and $A = -\nabla\nabla \log p(\boldsymbol{f}|X, \boldsymbol{y})|_{\boldsymbol{f}=\hat{\boldsymbol{f}}}$.
- Moreover, $A = -\nabla\nabla \log p(\boldsymbol{f}|X, \boldsymbol{y})|_{\boldsymbol{f}=\hat{\boldsymbol{f}}} = -W - K(X, X)^{-1}$ where W is a diagonal matrix with elements $\sigma(\hat{f}_i)(1 - \sigma(\hat{f}_i))$.
- Then, $p(f_*|\boldsymbol{x}_*, X, \boldsymbol{y}) = \mathcal{N}(K(X, \boldsymbol{x}_*)^T K(X, X)^{-1}\hat{\boldsymbol{f}}, K(\boldsymbol{x}_*, \boldsymbol{x}_*) - K(X, \boldsymbol{x}_*)^T(K(X, X) + W^{-1})^{-1}K(X, \boldsymbol{x}_*))$.
- Finally, note that the (approximate) prediction requires one-dimensional numerical integration.
- In general, the prediction (expected sigmoid) differs from the sigmoid of the expectation $(\sigma(K(X, \boldsymbol{x}_*)^T K(X, X)^{-1}\hat{\boldsymbol{f}}))$. However, either both or none are greater than 0.5. So, we can use the latter if we are only interested in the most probable class label.
- Demo of `KernLabDemo.R`.

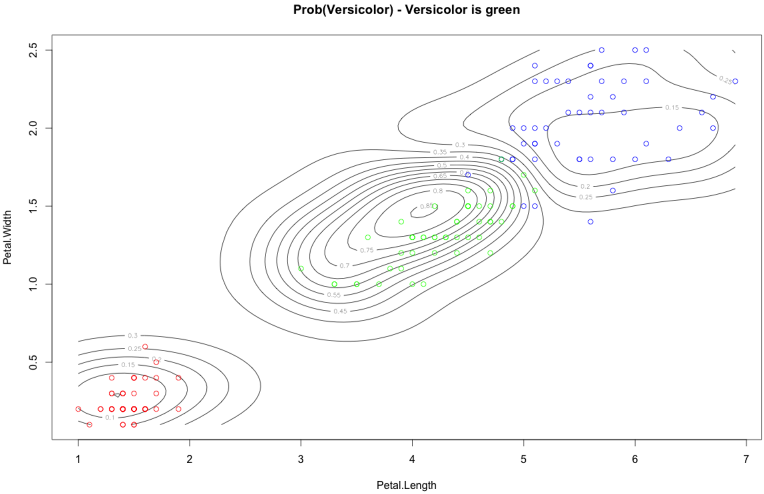
Gaussian Process Classification: Iris Data

- SE kernel with automatic ℓ estimation and $\sigma_f = 1$.
- $Species \sim Petal.Length + Petal.Width$.
- $p(Setosa|Petal.Length, Petal.Width)$:



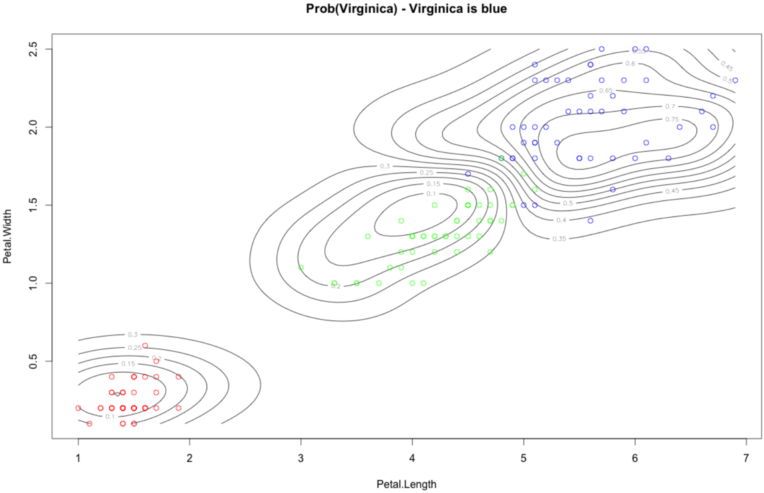
Gaussian Process Classification: Iris Data

- SE kernel with automatic ℓ estimation and $\sigma_f = 1$.
- $Species \sim Petal.Length + Petal.Width$.
- $p(Versicolor|Petal.Length, Petal.Width)$:



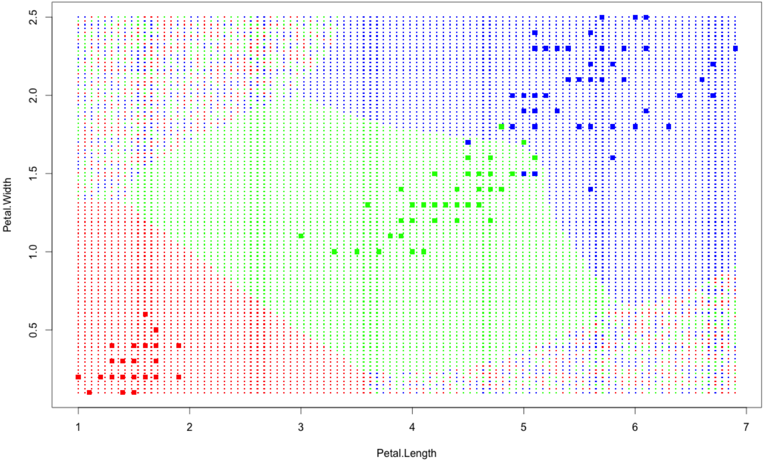
Gaussian Process Classification: Iris Data

- SE kernel with automatic ℓ estimation and $\sigma_f = 1$.
- $Species \sim Petal.Length + Petal.Width$.
- $p(Virginica | Petal.Length, Petal.Width)$:



Gaussian Process Classification: Iris Data

- SE kernel with automatic ℓ estimation and $\sigma_f = 1$.
- $Species \sim Petal.Length + Petal.Width$.
- Decision boundary:



Contents

- Linear Logistic Regression
- Bayesian Linear Logistic Regression
- Gaussian Process Classification
- Laplace Approximation
- Gaussian Process Classification: Iris Data

Thank you