

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,

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

where $s(\mathbf{x}) = \log \frac{p(\mathbf{x}|y=+1)p(y=+1)}{p(\mathbf{x}|y=-1)p(y=-1)} = \log \frac{p(y=+1|\mathbf{x})}{p(y=-1|\mathbf{x})}$ is the log odds ratio, and $\sigma(a) = \frac{1}{1+\exp(-a)}$ is the logistic sigmoid function.

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

$$\log p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \sum_{i=1}^n \log \sigma(y_i(\mathbf{x}_i^T \mathbf{w}))$$

since $\sigma(-a) = 1 - \sigma(a)$.

- No closed form solution exists, but the log lik function is concave and thus easy to maximize via gradient ascent.
- Beware of overfitting for linearly separable datasets: Log lik maximization causes $\|\mathbf{w}\|$ to tend to infinity, i.e. the sigmoid function becomes a Heaviside step function.

Bayesian Linear Logistic Regression

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

$$\log p(\mathbf{w}|X, \mathbf{y}) \propto \sum_{i=1}^n \log \sigma(y_i(\mathbf{x}_i^T \mathbf{w})) - \frac{1}{2} \mathbf{w}^T \Sigma_p^{-1} \mathbf{w}.$$

- ▶ No closed form solution exists, but the penalty term is quadratic on \mathbf{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:

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

- ▶ 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

- ▶ Consider a **binary** classification problem $y \in \{-1, +1\}$.
- ▶ Given a test case \mathbf{x}_* , use a GP for regression to predict a real number f_* that is then “squashed” through the logistic function to produce a class label $y_* = \sigma(f_*)$.
- ▶ However, the training data only include class labels \mathbf{y} and, thus, \mathbf{f} are **latent** variables.
- ▶ In other words, prediction occurs in two steps:
 - ▶ Compute the distribution of the latent variable f_* :

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

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

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

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

Laplace Approximation

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

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

where

- ▶ the first term is $\mathcal{N}(K(\mathbf{x}_*, X)K(X, X)^{-1}\mathbf{f}, K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, X)K(X, X)^{-1}K(X, \mathbf{x}_*))$ 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:

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

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

- ▶ Laplace's method to approximate $p(u) = \frac{1}{2} q(u)$:
 - ▶ 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 $A = -\frac{\partial^2}{\partial u^2} \log q(u)|_{u=u_0}$ (since u_0 is a mode of q , the first order term is zero and the second order term is negative).

- ▶ 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

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

where

- ▶ the first term is $\mathcal{N}(K(\mathbf{x}_*, X)K(X, X)^{-1}\mathbf{f}, K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, X)K(X, X)^{-1}K(X, \mathbf{x}_*))$ since it is a GP for regression, and
 - ▶ the second term is approximated by $\mathcal{N}(\hat{\mathbf{f}}, A^{-1})$ where $\hat{\mathbf{f}} = \arg \max_{\mathbf{f}} p(\mathbf{f}|X, \mathbf{y})$ and $A = -\nabla \nabla \log p(\mathbf{f}|X, \mathbf{y})|_{\mathbf{f}=\hat{\mathbf{f}}}$.
- ▶ Moreover,

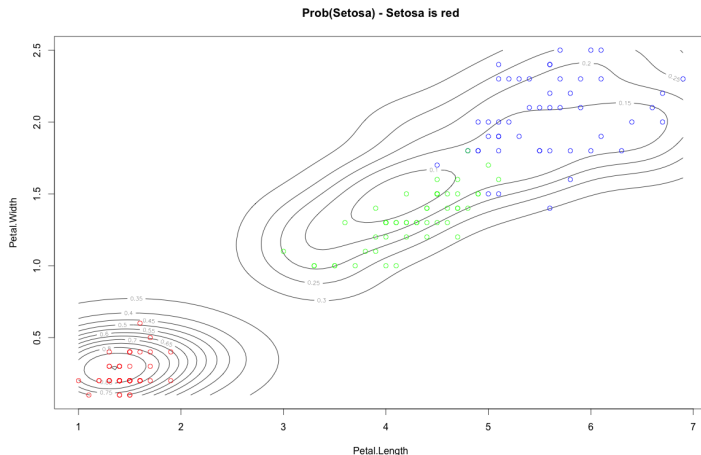
$$p(\mathbf{f}|X, \mathbf{y}) = p(\mathbf{f}, \mathbf{y}|X)/p(\mathbf{y}|X) \propto p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|X)$$

i.e. logistic function times GP prior. Typically, numerical methods are used to maximize it.

- ▶ Moreover, $A = -\nabla \nabla \log p(\mathbf{f}|X, \mathbf{y})|_{\mathbf{f}=\hat{\mathbf{f}}} = -W - K(X, X)^{-1}$ where W is a diagonal matrix with elements $\sigma(\hat{f}_i)(1 - \sigma(\hat{f}_i))$.
- ▶ Then, $p(f_*|\mathbf{x}_*, X, \mathbf{y}) = \mathcal{N}(K(X, \mathbf{x}_*)^T K(X, X)^{-1} \hat{\mathbf{f}}, K(\mathbf{x}_*, \mathbf{x}_*) - K(X, \mathbf{x}_*)^T (K(X, X) + W^{-1})^{-1} K(X, \mathbf{x}_*))$.
- ▶ Finally, note that the (approximate) prediction requires one-dimensional numerical integration, or MC sampling.
- ▶ The prediction (expected sigmoid) differs from the sigmoid of the expectation ($\sigma(K(X, \mathbf{x}_*)^T K(X, X)^{-1} \hat{\mathbf{f}})$). Luckily, either both or none are greater than 0.5. So, the latter suffices to find the most probable class.

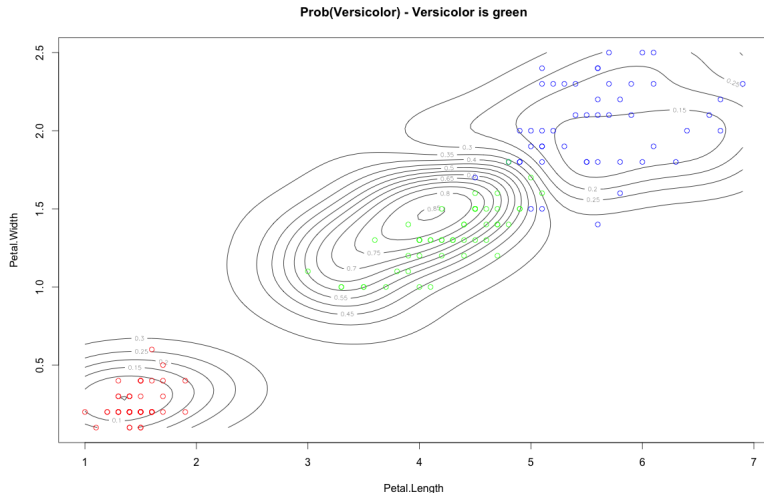
Gaussian Process Classification: Iris Data

- ▶ Multi-class classification is solved similarly, i.e. latent variable + softmax + Laplace's approximation + numerical integration or MC sampling.
- ▶ Demo of KernLabDemo.R.
- ▶ SE kernel with automatic ℓ estimation and $\sigma_f = 1$.
- ▶ $\text{Species} \sim \text{Petal.Length} + \text{Petal.Width}$.
- ▶ $p(\text{Setosa} | \text{Petal.Length}, \text{Petal.Width})$:



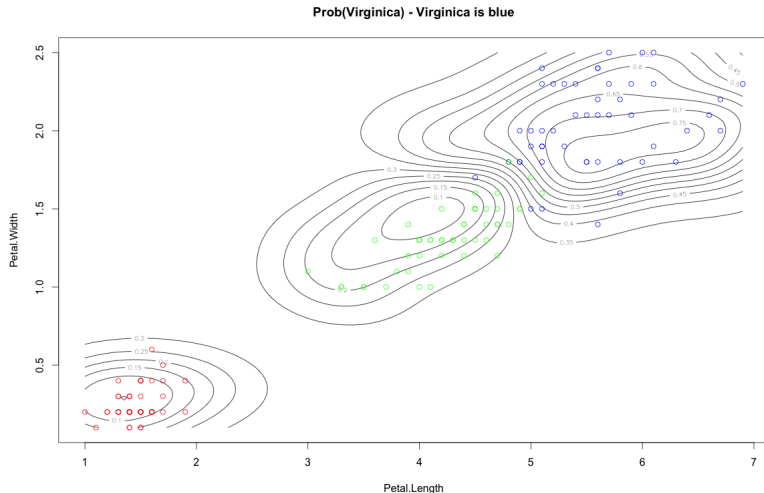
Gaussian Process Classification: Iris Data

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



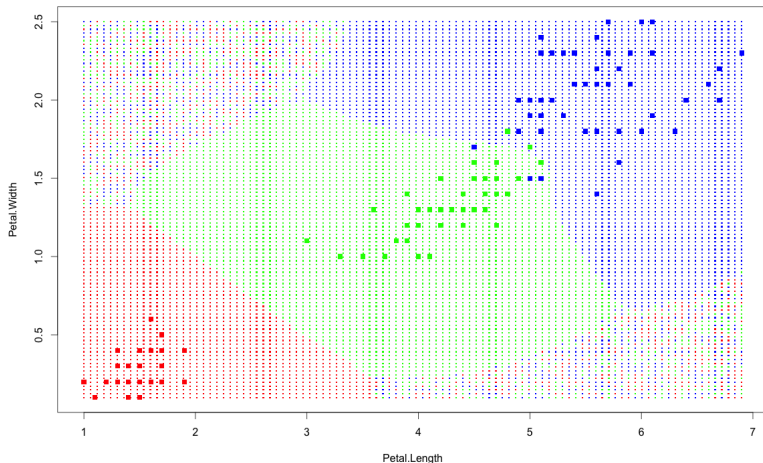
Gaussian Process Classification: Iris Data

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



Gaussian Process Classification: Iris Data

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



Contents

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

Thank you