Gaussian Processes for Machine Learning

P. Izmailov

April 15, 2016

Section 1

Gaussian Process

Gaussian Process

Definition

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

$$f \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot)) \Leftrightarrow \forall t_1, \ldots, t_n \quad (f(t_1), \ldots, f(t_n)) \sim \mathcal{N}(\mu, K),$$

where $\mu = (m(t_1), \ldots, m(t_n))^T$, $K \in \mathbb{R}^{n \times n}$, $K_{ij} = k(t_i, t_j)$.
$m : \mathbb{R} \to \mathbb{R}$ is called the mean function of the gaussian process f.
$k : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ is the covariance function of f.

Mean and covariance functions completely determine a gaussian process.

Example

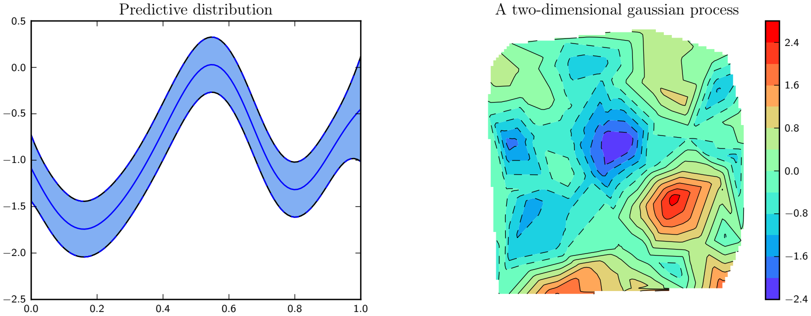
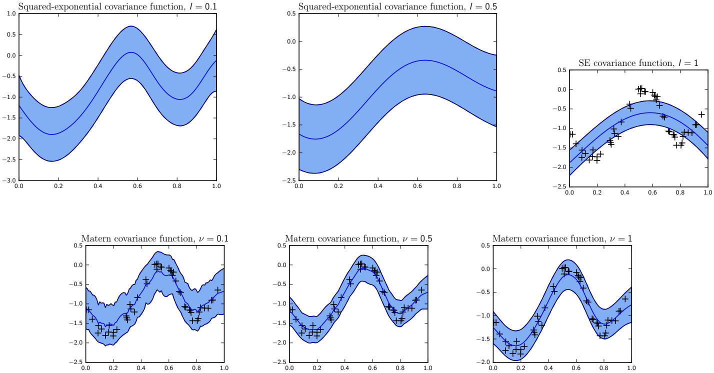


Figure: Examples of gaussian processes

Covariance functions



Squared-exponential covariance function, $l = 0.1$

Squared-exponential covariance function, $l = 0.5$

SE covariance function, $l = 1$

Matern covariance function, $\nu = 0.1$

Matern covariance function, $\nu = 0.5$

Matern covariance function, $\nu = 1$

Figure: Squared-exponential and Matern covariance functions

Section 2

Gaussian Process Regression

Notation

- $\{(x_i, f_i) | i = 1, \dots, n\}$ — dataset, considered to be generated from a Gaussian process $f \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$, let $x \in \mathbb{R}^d$.
- $X \in \mathbb{R}^{n \times d}$ — the matrix, comprised of data points $x_1, \dots, x_n$.
- $f \in \mathbb{R}^n$ — the vector of target values $f_1, \dots, f_n$.
- $y \in \mathbb{R}^n$ — the noisy version of f: $y \sim \mathcal{N}(y|f, \sigma_n^2 I)$
- $X_* \in \mathbb{R}^{k \times d}$ — new (test) data points.
- $f_* \in \mathbb{R}^k$ — the desired vector of process values at new data points X_*.
- $K(X, X) \in \mathbb{R}^{n \times n}$ — the matrix, comprised of pairvise values of the covariance function $k(\cdot, \cdot)$ of the underlying process:

$$K(X, X)_{ij} = k(x_i, x_j).$$

- $K(X, X_*) \in \mathbb{R}^{n \times k}$ — the matrix, defined similarly to the $K(X, X)$.
- $K(X_*, X) = K(X, X_*)^T$.

Problem statement

The model is as follows. f is a vector of unobserved samples from a gaussian process $\mathcal{GP}(m(\cdot), k(\cdot, \cdot))$ at data points x_i and y is it's noisy version we observe.

$$f \sim \mathcal{N}(m(X), K(X, X)),$$

$$y = f + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma_n^2 I).$$

What we want to obtain is the predictive distribution at a set of new data points X_*

$$p(f_* | X_*, X, y).$$

We put the following prior on our model: the data is generated from a zero-mean gaussian process with covariance function $k(\cdot, \cdot)$:

$$f \sim \mathcal{GP}(0, k(\cdot, \cdot)).$$

This prior is not limiting, because zero-mean prior does not imply a zero-mean predictive distribution.

GP-Regression

The joint distribution for the process values f and f_* is given by

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right).$$

We slightly change the covariance matrix to obtain the joint distribution for y and f_*.

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right).$$

Predictive distribution

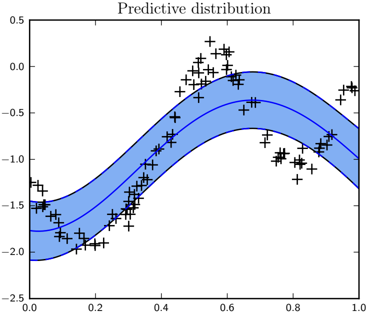
Conditioning the joint distribution, we obtain the predictive

$$f_*|y \sim \mathcal{N}(\hat{m}, \hat{K}),$$

$$\mathbb{E}[f_*|y] = \hat{m} = K(X_*, X)(K(X, X) + \sigma_n^2 I)^{-1} y,$$

$$\mathrm{cov}(f_*|y) = \hat{K} = K(X_*, X_*) - K(X_*, X)(K(X, X) + \sigma_n^2 I)^{-1} K(X, X_*).$$

Example



As we can see, the model does not explain the data very well. In order to deal with this problem, we can tweak the covariance function. The covariance functions usualy have a set of parameters, which we will refer to as covariance (or kernel) hyper-parameters. Varying these parameters, we can find a better model for the data.

Marginal likelihood

In order to find the best set of kernel hyper-parameters, we maximize the marginal likelihood with respect to them. In the case of gaussian process regression, this likelihood is given by

$$\log p(y) = \log \int p(y|f)p(f)df = \log \mathcal{N}(y|0, K(X,X) + \sigma_n^2 I) =$$

$$= -\frac{1}{2} y^T (K(X,X) + \sigma_n^2 I)^{-1} y - \frac{1}{2} \log |K(X,X) + \sigma_n^2 I| - \frac{n}{2} \log 2\pi.$$

If $k(\cdot,\cdot)$ is a differentiable function of it's hyper-parameters (which is usually true), $p(y)$ also is, and can be maximized with gradient-based optimization methods.

Example

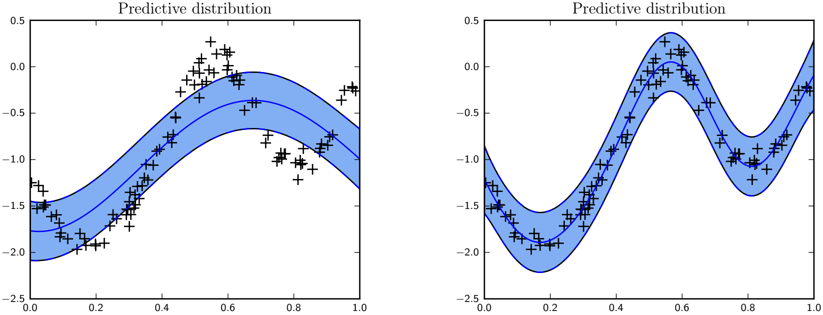


Figure: Predictive distribution before and after hyper-parameter adaptation

As we can see, after adaptation of kernel hyper-parameters, the method does a better job, explaining the data.

Computational Complexity

The predictive mean and covariance are given by

$$\mathbb{E}[f_*|y] = \hat{m} = K(X_*, X)(K(X, X) + \sigma_n^2 I)^{-1} y,$$

$$\mathrm{cov}(f_*|y) = \hat{K} = K(X_*, X_*) - K(X_*, X)(K(X, X) + \sigma_n^2 I)^{-1} K(X, X_*).$$

Marginal likelihood is given by

$$\log p(y) = -\frac{1}{2} y^T (K(X, X) + \sigma_n^2 I)^{-1} y - \frac{1}{2} \log |K(X, X) + \sigma_n^2 I| - \frac{n}{2} \log 2\pi.$$

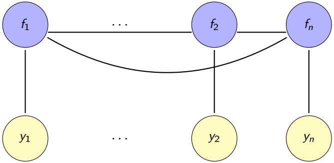
The computational complexity of the gaussian process regression is determined by the complexity of inversing the $K(X, X)$ matrix and computing the detrminant of $K(X, X) + \sigma_n^2 I$, and thus scales as $O(n^3)$. This complexity makes the method inapplicable to big problems and thus approximate approaches are needed.

Section 3

Inducing Inputs

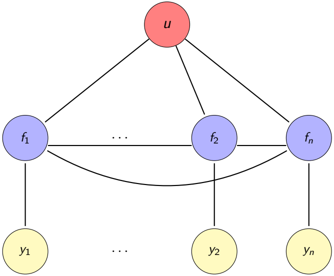
Inducing Inputs

The graphical model for the gaussian process regression looks like this.



Inducing Inputs

Now we slightly change the model, adding a set of latent variables u.



The joint probability of latent and observable variables now is given by

$$p(y, f, u) = p(y|f)p(f|u)p(u).$$

Inducing Inputs

The latent variables u are referred to as inducing inputs. The intuition behind them is that they are considered as the values of the process at new data points $z_1, \dots, z_m$. We will have to introduce some more notation now.

- $Z_m \in \mathbb{R}^{m \times d}$ — the matrix, comprised of the coordinates of the inducing inputs inputs $z_1, \dots, z_m$.
- $K_{nn} = K(X, X)$
- $K_{mm} = K(Z_m, Z_m)$
- $K_{mn} = K(Z_m, X)$
- $K_{nm} = K(X, Z_m) = K_{mn}^T$

As u_i are considered to be generated from the same gaussian process, as f_i, we have the following formulas.

$$p(u) = \mathcal{N}(u|0, K_{mm}),$$

$$p(f|u) = \mathcal{N}(f|K_{nm}K_{mm}^{-1}u, \tilde{K}),$$

where $\tilde{K} = K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}$.

Evidence Lower Bound

The standard variational lower bound for the marginal likelihood $p(y)$ for our augmented model is

$$\log p(y) \geq \mathbb{E}_{q(u,f)} \log \frac{p(y,u,f)}{q(u,f)} = \mathbb{E}_{q(u,f)} \log p(y|f) - \text{KL}\left(q(u,f) \mid\mid p(u,f)\right).$$

Our model implies $\mathbb{E}_{q(u,f)} \log p(y|f) = \mathbb{E}_{q(f)} \log p(y|f)$, where $q(f)$ is the marginal of $q(u,f)$.
We will consider the variational distributions of the following form:

$$q(u,f) = p(f|u)q(u),$$

where $q(u) \sim \mathcal{N}(u|\mu, \Sigma)$. This implies $q(f)$

$$q(f) = \int p(u|f)q(u)du =$$

$$\mathcal{N}(f|K_{nm}K_{mm}^{-1}\mu, K_{nn} + K_{nm}K_{mm}^{-1}(\Sigma - K_{mm})K_{mm}^{-1}K_{mn}).$$

Evidence Lower Bound

Now, consider the KL-divergence in the lower bound we've devised.

$$\mathrm{KL}\left(q(u,f) \mid\mid p(u,f)\right) = \mathrm{KL}\left(q(u)p(f|u) \mid\mid p(u)p(f|u)\right) = \mathrm{KL}\left(q(u) \mid\mid p(u)\right).$$

Finally, the lower bound is

$$\log p(y) \geq \mathbb{E}_{q(f)} \log p(y|f) - \mathrm{KL}\left(q(u) \mid\mid p(u)\right) =$$

$$= \sum_{i=1}^{n} \mathbb{E}_{q(f_i)} \log p(y_i|f_i) - \mathrm{KL}\left(q(u) \mid\mid p(u)\right).$$

Note, that although, we've devised this bound for the regression problem, we never used the fact, that we are actually performing regression. This bound holds for binary classification problem as well.
However, in the case of GP-regression, the right-hand side of the bound can be computed analytically in a closed form.

SVI method

Substituting the normal distributions $q(u)$, $p(u)$, $q(f)$ and $p(y|f)$ back into the lower bound, we obtain the following inequality.

$$\log p(y) \geq \sum_{i=1}^{n} \left(\log \mathcal{N}(y_i | k_i^T K_{mm}^{-1} \mu, \sigma_n^2) - \frac{1}{2\sigma_n^2} \tilde{K}_{ii} - \frac{1}{2} \mathrm{tr}(\Sigma \Lambda_i) \right) -$$

$$-\frac{1}{2} \left(\log \frac{|K_{mm}|}{|\Sigma|} - m + \mathrm{tr}(K_{mm}^{-1}\Sigma) + \mu^T K_{mm}^{-1} \mu \right),$$

where $\Lambda_i = \frac{1}{\sigma_n^2} K_{mm}^{-1} k_i k_i^T K_{mm}^{-1}$, and k_i is the i-th column of the matrix K_{mn}. This lower can be maximized with respect to kernel hyper-parameters and variational parameters μ, Σ using the stochastic optimization techniques. The method was proposed at [Hensman et al., 2013]. The authors suggest using the stochastic gradient descent with natural gradients for variational parameters. The complexity of computing a stochastic update for one object is $O(m^3)$.

Titsias's method

The lower bound we devised can also be maximized with respect to variational parameters analytically, which was suggested in [Titsias, 2009]. The optimal distribution is $q^*(u) \sim \mathcal{N}(u|\hat{u}, \Lambda^{-1})$, where

$$\Lambda = \frac{1}{\sigma_n^2} K_{mm}^{-1} K_{mn} K_{nm} K_{mm}^{-1} + K_{mm}^{-1},$$

$$\hat{u} = \frac{1}{\sigma_n^2} \Lambda^{-1} K_{mm}^{-1} K_{mn} y.$$

Substituting this distribution back to the ELBO, we obtain

$$\log p(y) \geq -\frac{1}{2}\left(n \log 2\pi + \log|B| + y^T B^{-1} y + \frac{1}{\sigma_n^2} \mathrm{tr}(\tilde{K}) \right),$$

where $B = \sigma_n^2 I + K_{nm} K_{mm}^{-1} K_{mn}$. The complexity of computing the optimal distribution parameters, the lower bound and it's gradients is $O(nm^2)$. However, we can not apply stochastic optimization in this case.

Example

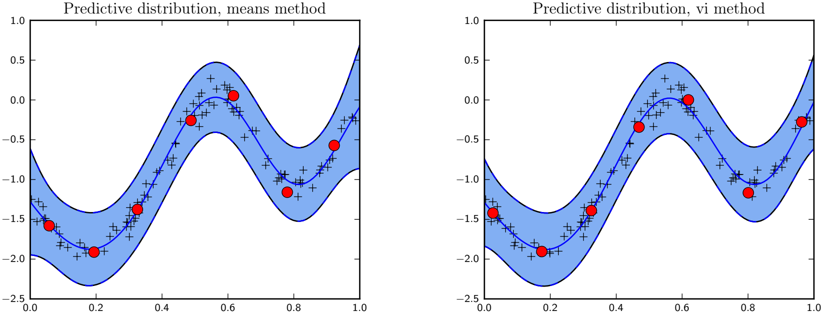


Figure: Example of two implementations of the Titsias's method. The vi method maximizes the lower bound with respect to the positions of inducing inputs, while the means method just uses the K-means cluster centers as inducing point positions.

Section 4

Experiments

Experiments

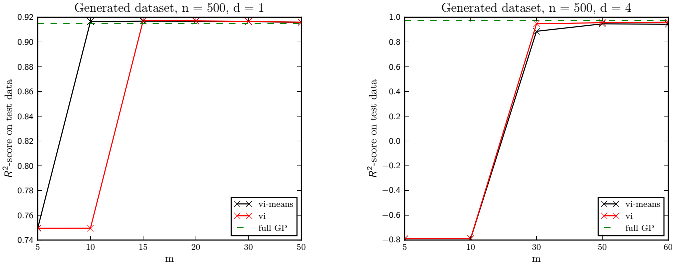


Figure: Comparison of the Titsias's method with and without optimization with respect to inducing point positions

As we can see, for these small problems optimization with respect to positions of inducing points does not affect the quality too much. However, this optimization dramaticly increases the number of optimized parameters, and makes the optimization much harder. Thus, we didn't perform this optimization in further experiments.

Experiments

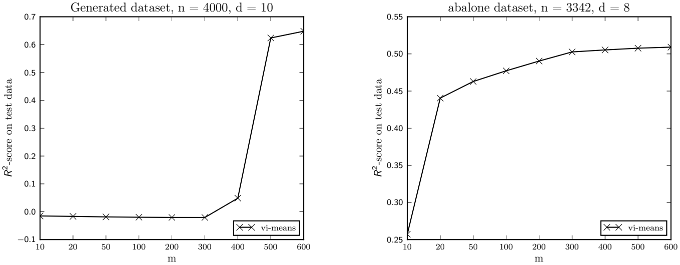


Figure: The dependence between quality and number of inducing points for slightly bigger datasets

Experiments

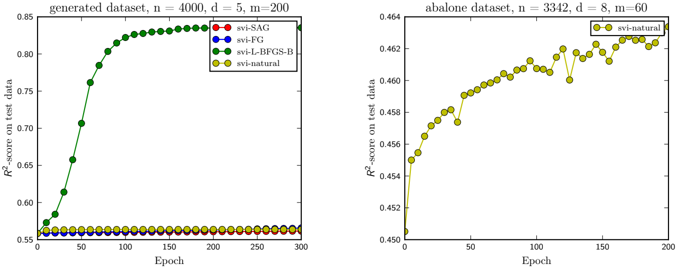


Figure: Comparison of variaous optimization methods for svi lower bound

Experiments

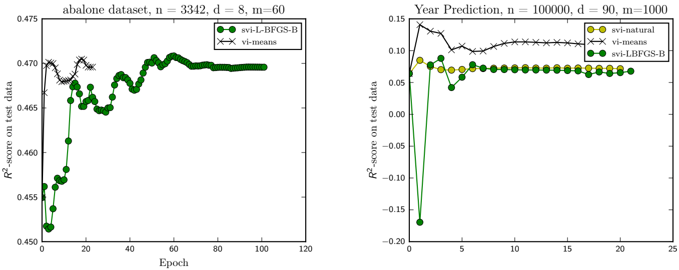


Figure: Comparison of vi and svi methods

References

- Titsias M. K. (2009). Variational Learning of Inducing Variables in Sparse Gaussian Processes. In: *International Conference on Artificial Intelligence and Statistics*, pp. 567–574.

- Hensman J., Fusi N., Lawrence D. (2013). Gaussian Processes for Big Data. In: *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*.