Introduction to Gaussian Processes: Regression, Classification, Active Learning, Experimental Design and Bayesian Optimization

Ruben Martinez-Cantin
Defense University Center
Zaragoza, Spain
rmcantin@unizar.es

Abstract

A Gaussian process is a simple, yet powerful, non-parametric Bayesian model. Originally developed as an stochastic process for time series analysis, it can be also used as a prior distribution over functions, which can be used for inference. In this laboratory we will learn how it can be used for simple examples of nonlinear regression, classification, experimental design and Bayesian optimization.

1 Introduction

During this laboratory we are going to use different toolboxes. All of them are written to work both in MATLAB and GNU OCTAVE:

GPML The Gaussian process toolbox by Carl Edward Rasmussen and Hannes Nickisch. The toolbox is based on the methods presented in the book also by Carl Rasmussen. Both the toolbox and the book [7] are freely available[1]. Some of the examples that weare going to address here are extracted directly from the book, so you are encouraged to check it during the lab. There is also a short manual in [6].

SFO The Submodular function obtimization by Andreas Krause[2]. It includes different methods and functions to perform minimization or maximization of submodular functions. We will focus on the problem of experimental design. There is also a short manual in [4].

DIRECT Originally developed by Jones, Perttunen and Stuckman [3], the DIRECT algorithm is a efficient global optimization method that relies on Lipschitz optimization. This toolbox has been written by Dan Finkel[3]. We will use it also for experimental design and Bayesian optimization.

LHS The latin hypercube sampling is a well known strategy for random sampling with orthogonality and coverage guarantees. It has been extensively used also in the experimental design literature. This code has been implemented by Budiman Minasny[4].

> **Important:** Before doing anything, run the *startup* scritp. It will make sure all the toolboxes, datasets and other files are in MATLAB path.

[1] GPML: http://www.gaussianprocess.com
[2] SFO: http://las.ethz.ch/sfo/index.html
[3] DIRECT: http://www4.ncsu.edu/~ctk/Finkel_Direct/
[4] LHS: http://www.mathworks.com/matlabcentral/fileexchange/4352-latin-hypercube-sampling

2 Part 3: Active Learning

In this part of the practical sessions we are going to cover the problems of active learning, experimental design, Bayesian optimization, stochastic bandits... *all in a simple and common framework!*

> **Important:** If you have not used GPML toolbox and have not attended the first part of the practical session about Gaussian process, please read Section 1 of that session handouts and run the code that it is explained there before doing anything else.

2.1 Optimality and decisions

In the previous part, we have seen how to do inference and prediction. However, as Peter Green explained during his talk, when we need to take a decision, we need the posterior distribution over our data, plus a cost or regret function $\delta(f, d)$ that modulates our decisions d. Therefore, we need to see what is the regret function of different problem.

Being in a Bayesian setup, we are going to focus on the average cost or regret, which is sometimes defined as best response case. In this case, our decisions are:

$$d_{ac} = \arg\min_d \int_F \delta(f, d) dP(f) \tag{1}$$

where $P(f)$ is a distribution over functions, that is, our Gaussian process.

2.1.1 Global optimization

The objective of a global optimization algorithm is to find the sequence of points

$$x_n \in \mathcal{A} \subset \mathbb{R}^m, \quad n = 1, 2, \dots \tag{2}$$

which converges to the point x^*, which corresponds to the extremum of the target function, when n is large for all problems from a given family. this search procedure is a sequential decision making problem where point at step $n + 1$ is based on decision d_n based on all previous data:

$$x_{n+1} = d_n(x_{1:n}, y_{1:n}) \tag{3}$$

where $y_i = f(x_i)$. The search method is the sequence of decisions $d = d_0, \dots, d_N$, which leads to the final decision $x_{N+1} = x_{N+1}(d)$. The *regret* of the search can be expressed as:

$$\delta(f, d) = f\left(x_{N+1}(d)\right) - f(x^*) \tag{4}$$

In global optimization there is no convergence guarantees apart from dense sampling, that is, sampling every point in the search space. In practice, the user defines a stopping criteria, such as a *budget*, which corresponds to he maximum number of iterations N.

2.1.2 Stochastic bandits

The problem of stochastic bandits is very similar to global optimization. The main difference is that, while in the global optimization case, we pay the regret only at the end, in the bandits setting, we pay regret *at every iteration.* That is,

$$\delta_n(f, d) = f\left(x_n(d)\right) - f(x^*) \tag{5}$$

In this setting, it is a standard approach to consider the cumulative and average regret:

$$\delta_{cum}(f, d) = \sum_{n=1}^{N} \delta_n(f, d)\ ; \qquad \delta_{ave}(f, d) = \frac{\sum_{n=1}^{N} \delta_n(f, d)}{N} \tag{6}$$

In practice, global optimization algorithm tends to be *anytime* algorithms, that is, it always tends to provide the best possible solution for the current number of iterations. If the number of iterations increases, the performance of the algorithm will increase. The purpose of this methodology is based on the assumption that the algorithm does not have a priori knowledge of the available budget.

That means that global optimization algorithms intrinsically are considering the average regret instead of the final regret. For that reason, bandits and global optimization strategies are interchangeable.

2.1.3 Bayesian experimental design

Sequential experimental design and active learning focus on the problem of finding the decisions that provides more information about the model that we are compute, for example, the Gaussian process.

This case is more involved, since there is no single criterion for that. A nice review can be found in [1]. The most extended methods are the A-optimality, the D-optimality and the E-optimality.

A-optimality seeks to minimize average variance of the latent variables of the model. For the Gaussian case, it is equivalent to minimize the trace of the covariance matrix.

D-optimality seeks to maximize the information gain, or in other words, maximize the KL-divergence between the prior and posterior distribution. For the Gaussian case, it is equivalent to maximize the determinant of the inverse of the posterior covariance matrix.

E-optimality works in a minimax fashion, where it seeks to maximize the minimum eigenvalue of the information matrix. However, this criteria does not seems to corresponds to any utility function.

2.2 Continuous inputs: A simple greedy strategy

In the first case we are going to assume that the set of input points that are available are infinite but bounded. Typically, we are going to have box bounds over continuous data such that our parameter space is defined as $\mathcal{A} = [0, 1]^m$.

This kind of search is a double exponential. First, it is exponential in the number of dimensions of our parameter space. Second, it is exponential in the number of points that we want to select. However, due to the continuity assumption, if a point is found as the optimum for a certain criteria, all the points in the ϵ-neighborhood of that point are also optima.

For that reason, in the continuous case, we rather take a greedy strategy, where we select a single point at each step, updating the model after every new data point arrives. As said before, we can apply this framework for active learning, global optimization and stochastic bandits.

Note that in this greedy setup, the Bayesian experimental design criteria are equivalent to find the point which has maximum predicted uncertainty.

Thus, depending on the problem we may have different criteria. Jones made a complete review of different criteria for global optimization [2]. Let be $p(y_t|\mathbf{x}_{1:t-1}, y_{1:t-1}, \mathbf{x}_t) = \mathcal{N}(\mu_t, \sigma_t^2)$, the predicted distribution given the data and a new query point $\mathbf{x}_t$. Let assume also that $y_{min} = \inf(y_{1:t-1})$ and $y_{max} = \sup(y_{1:t-1})$. Then, we can base our decision on:

Predicted mean which corresponds to μ_t. In the bandits and optimization setups, it is sometimes called *pure exploitation strategy*. In classification we may want to find the points

$$x_* = \arg\min_x |\mu_t|.$$

Can you figure out why?

Predicted variance which corresponds to σ_t. In the bandits and optimization setups, it is sometimes called *pure exploration strategy*.

Predicted mean and variance ratio which for classification is defined as $|\mu_t|/\sigma_t$.

Upper or lower bound which are $\mu_t \pm \beta_t \sigma_t$. Srinivas et al. [8] found the optimal values of β_t to guarantee no regret in the bandits setting. Changing β_t with time can be used to vary the exploratory behavior of the method. This is a standard strategy for optimization and bandits.

Probability of improvement which in the case of seeking the minimum is defined as

$$p(y_t \leq y_{min}) = \Phi\left(\frac{y_{min} - \mu_t}{\sigma_t}\right)$$

where Φ is the standard Gaussian cumulative density function. In the case of maximimization, the signs switch and y_{max} replaces y_{min}.

Expected improvement which was defined by [5] for global optimization. If we define the improvement for minimization as $I = \max(y_{min} - y, 0)$, and compute the expected value with respect to our function distributions:

$$EI(x) = \mathbb{E}[I(x)] = \int \max(y_{min} - f(x), 0)\ dp(f)$$

Thus

$$EI(x_t) = (y_{min} - \mu_t)\Phi(d) + \sigma_t\phi(d)$$

where, again, Φ is the standard Gaussian cumulative density function, ϕ is the standard Gaussian probability density function and $d = (y_{min} - \mu_t)/\sigma_t$.

Can you guess which criteria are good for optimization? For bandits? For regression? For classification?

3 DIRECT usage

Once we have selected the criterion to be used, we still need to found the optimal value (maximum or minimum, depending on the application) to select the next point. This requires solving another non-convex optimization problem on a box bounded domain. For that reason, we are going to use the DIRECT algorithm, developed by Jones et al [3] and implemented by D. Finkel.

This algorithm is a deterministic sampling algorithm based on the Lipschitz assumption to compute lower bounds on the function value. All the criteria that we have defined are Lipschitz. The code and some examples can be found in the *gpml/direct* folder.

Important: DIRECT always seeks for the minimum of a function. If you want to maximize your criterion, just send the negative value to DIRECT.

4 Exercises: Continuous case

During the exercises, we are going to see how we can apply our knowledge of Gaussian processes to compute the expectation of the different criteria based on the distribution over functions that the Gaussian process provides.

First, we are going to start with a simple regression case *act_reg_continous.m*. Note that this function is highly multimodal, which will be interesting when we want to find the global minimum. As we did before, just fill the gaps on the GP configuration.

The function *simple_criterion.m* is an example of the interface that the criterion function needs to satisfy in order to work with DIRECT and the rest of the code. Just edit that file o create a new one with the same interface to implement the rest of the criteria. Try to find the best criterion for active learning of the regression function, finding the minimum or minimize the bandits regret.

Second, the file *act_class_continous.m* provides a similar example but for binary classification. Again, try to find the best criterion for active classification.

References

[1] K Chaloner and I Verdinelli. Bayesian experimental design: A review. *J. of Statistical Science*, 10:273–304, 1995.

[2] D.R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21:345–383, 2001.

[3] D.R. Jones, C.D. Perttunen, and B.E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, October 1993.

[4] Andreas Krause. Sfo: A toolbox for submodular function optimization. *Journal of Machine Learning Research*, 11:1141–1144, 2010.

[5] J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. In L.C.W. Dixon and G.P. Szego, editors, *Towards Global Optimisation 2*, pages 117–129. Elsevier, 1978.

[6] Carl Edward Rasmussen and Hannes Nickisch. Gaussian processes for machine learning (gpml) toolbox. *Journal of Machine Learning Research*, 11:3011–3015, 2010.

[7] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, Massachusetts, 2006.

[8] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proc. International Conference on Machine Learning (ICML)*, 2010.