HYPER-PARAMETER SELECTION WITH BAYESIAN OPTIMIZATION

Amal Rannen-Triki & Matthew B. Blaschko

June 2019

1 Setup

This tutorial will use: GPflow, GPFlowOpt, sklearn and jupyter notebooks.

Environment for Gaussian prcesses Create a new environment and run:

1. pip install tensorflow
2. pip install gpflow

Environment for Bayesian optimization To install GPFlowOpt:

- Download the codes from
- Go into the folder GPflowOpt
- run "pip install . –process-dependency-links"

This command breaks with the most recent versions of pip. If you have an error, please run "pip install pip==18.1"

Note that GPflowOpt has its own implementation of GPflow. While the overall structure is the same, there are few differences in the folder/naming structure.

2 Gaussian processes

In the following GPML refers to: C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006. `http://www.gaussianprocess.org/gpml/chapters/RW.pdf`

Prelim exercises

1. Let $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12} & \Sigma_{22} \end{bmatrix}\right)$

   Show that $p(x_2|x_1) = \mathcal{N}\left(\frac{\Sigma_{12}}{\Sigma_{11}} x_1, \Sigma_{22} - \frac{\Sigma_{12}^2}{\Sigma_{11}}\right)$

2. (*) Kernel ridge regression (KRR) can be thought of as L_2 regularized least squares after projecting your data through a non-linear (possibly infinite dimensional) map $x \mapsto \phi(x) \in \mathcal{H}$, and predicting by an inner product $\langle \phi(x), w \rangle_{\mathcal{H}}$.

   What is the solution to KRR? (You will have to make use of the representer theorem for kernels.)

   How does it compare to a GP?

Exercise 1: Consider a GP regression problem with n, with past observations X, past target y and a new set of observations X_*. We consider that the measurements are corrupted with a noise of variance σ_n^2. Derive the predictive equations for the new target f_*:

$$f_*|X, y, X_* \sim \mathcal{N}(\overline{f}_*, \mathrm{cov}(f_*)) \text{ where} \tag{1}$$

$$\overline{f}_* = K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} y \tag{2}$$

$$\mathrm{cov}(f_*) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*) \tag{3}$$

Exercise 2: (GPML - p.31) Let $var_n(f(x_*))$ be the predictive variance of a Gaussian process regression model at x_* given a dataset of size n. The corresponding predictive variance using a dataset of only the first $n-1$ training points is denoted $var_{n-1}(f(x_*))$. Show that $var_n(f(x_*)) \leq var_{n-1}(f(x_*))$, i.e. that the predictive variance at x_* cannot increase as more training data is obtained. One way to approach this problem is to use the partitioned matrix equations given in section A.3 (p. 201) (GPML) to decompose $var_n(f(x_*)) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*)$. Note that while this conclusion is true for Gaussian process priors and Gaussian noise models it does not hold generally [1].

Exercise 3: GPML - p.187 We consider the notations introduced in p.173 (eq. 8.8, 8.9 and 8.10). It can be shown that:

$$E(C_{opt}) = \mathrm{tr}(K - \tilde{K}) \text{ where } \tilde{K} = K_{nm} K_{mm}^{-} 1 K_{mn}. \tag{4}$$

Now consider adding one datapoint into set I, so that K_{mm} grows to $K_{(m+1)(m+1)}$. Using again the matrix identities of section A.3 (p. 201), show that the change in E due to adding the extra datapoint can be computed in time $\mathcal{O}(mn)$.

** Bonus - show (4).

Exercise 4: GP Regression with GPflow Open the notebook Gaussian_processes.ipynb. For this notebook, use the kernel with the (indendant) gpfow package. Run the notebook to get familiar with the different settings of a GP regression.

3 Bayesian optimization

Exercise 1: Consider the function $f(x) = \text{round}(x)$. Run few steps of Bayesian optimization to maximize f by hand with observations in $[0, 1]$.

Exercise 2: Run the Bayesian optimization in the section Bias effect in the Bayesian_opt notebook. Does it succeed to find the minimum? If it fails, how to correct it? Change the kernel, acquisition function and/or the optimizer and retry.

Exercise 3: Run the Bayesian optimization in the section Curse of dimensionality in the Bayesian_opt notebook for increasing dimensions. What do you observe?

Exercise 4: Run the Bayesian optimization in the section Uncertainty of measurement in the Bayesian_opt notebook. What do you observe? Can you improve the result?

Exercise 5: Run the Bayesian optimization in the section Peaked vs wiggly in the Bayesian_opt notebook. What do you observe? Can you improve the result?

Exercise 6: Select the hyperparameters C and gamma for the SVM defined in svm.py using:

- Grid search
- Random search
- Bayesian optimization

Grid search and Random search can be done using the methods in sklearn.model_selection.

Exercise 7: Select the learning rate, batch size, and hidden sizes for the MLP defined in mlp.py using:

- Grid search
- Random search
- Bayesian optimization

Exercise 8: Competition Find the maximum of the functions mlp_function and mysterious_function given in competition.py

Exercise 9: Model selection Using classification models from e.g. scikit-learn `https://scikit-learn.org/` and your favorite dataset (e.g. `http://yann.lecun.com/exdb/mnist/`) set up a model selection experiment in which you simultaneously determine which algorithm to use and which parameters of the algorithm to select.

References

[1] David Barber and David Saad. Does extra knowledge necessarily improve generalization? *Neural Computation*, 8(1):202–214, 1996.