Gaussian processes applied to the study of chemotactic movements of droplets

Carlotta Porcelli, qbp693

August 20, 2017

Abstract

The research hereby presented investigates the use of a Gaussian Process regression model on a dataset collected from experiments lead in the context of Artificial Chemical Life for the EVOBLISS project [1] at the REAL laboratory in the ITUniversity of Copenhagen [2] . The experiments performed examine the chemotactic behavior of 1-Decanol droplets placed in Decanoate solutions with different pHs and molarities when a chemical gradient is introduced in the external aqueous environment. The Gaussian process model has been fit to the data collected in order to look for a combination of ph and molarity values of the decanoate solution that would allow the droplet to increase the efficiency of its chemotactic movement.

1 Introduction

The movement of non-living objects in aqueous environments has been used to reproduce the chemotaxis process of a living cell, movement in a particular direction. [3]
The development of chemical systems that mimic the behavior of living systems can be useful in studying related processes in natural living systems.
The data used in this research were collected from experiments based on two phases. The droplet composition and volume along with the salt gradient created are the same in all the experiments. On the other hand, the experimental space for the water phase is generated by two continuous variables: the molarity and the pH of the decanoate solutions. The experiments carried out by EvoBot [4] span in continuous ranges of molarity, from 5mM to 20mM and pH from 7.0 to 12.3 to create the aqueous phase.
The aqueous combinations explored were created by mixing decanoate solutions at different pHs and adding water to get the molarity desired. In each experiment the droplet movement is tracked for a total time of 60 seconds. The implementation of the tracking assigns a *fitness value* based on the speed and the spatial precision of the droplet. High values represent good chemotactic movements.
Figure 1 shows an example of the tracking of one individual. The tracking starts at the initial position where the droplet is placed. The blue path is composed of the centroids of the droplet after its movement.

Figure 1: Example of the tracking of an individual. This image is saved for all the experiments and contains the information about the individual, including the pH and molarity of the solution tested, in this case pH=9.66 and molarity=13.50.

The 'END' point is where the program stopped tracking the droplet because it reached the proximities where the salt was added, *salt source*. The droplet reached efficiently the salt source and the fitness value registered is 0.120086.

The aim of this project is make an estimation on the values of pH and molarity forming a decanoate solution in which the droplet performs a smooth and precise chemotactic movement. In this mean, a Gaussian regression model is used to characterize a non-linear relationship between the two variables (pH and molarity). In a classic linear regression algorithm, it is assumed that a dependent variable y can be modeled as a function of one or more independent variables X in the form of $y = f(x) + \epsilon$. The choice of the GP approach relies on the facts that it finds a distribution over the possible functions *f(x)* consistent with the observed data. Moreover, the observations occur in a continuous domain.

2 Materials and methods

2.1 The Gaussian Processes

The task of building regression models to characterize non-linear relationships between variables often involves extensive model selection procedures to ensure that the most appropriate model is chosen. A alternative is to use a Bayesian strategy to model the unknown underlying function. Modeling data using Gaussian distributions in some cases would not seem to be any gain because of the reduced flexibility that those distributions have. A Gaussian process is defined as probability distribution over functions y(x) so that the set of values of y(x) evaluated on a set of points $x_1, ..., x_N$ jointly have a Gaussian distribution.
A Gaussian process assumes that a $p(f(x_1), ..., f(x_N))$ is jointly Gaussian with some mean $\mu(x)$ and covariance $\sum(x)$ given by $\sum_{ij} = k(x_i, x_j)$, where k is a positive definite kernel function [6]. The covariance represents a form of distance or similarity. Considering two input points x_i and x_j with corresponding observed values y_i andy_j. If the inputs are close to each other, it is expected that thaty_i andy_j will be close as well. This measure of similarity is embedded in the covariance function. Since the key assumption in GP modelling is that the data can be represented as a sample from a multivariate Gaussian distribution, we have that

$$\begin{bmatrix} y \\ y_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K & K^T \\ K_* & K_{**} \end{bmatrix}\right) \tag{1}$$

Where y is the observed value, y_* is the prediction and K is the covariance matrix. The interest is on the conditional probability $p(y_*|y)$ that follows a Gaussian distribution:

$$y_*|y \sim N(K_* K^{-1} y, K_{**} - K_* K^{-1} K_*^T) \tag{2}$$

Table 1: First 4 data points

pH	molarity	fitness values
10.07222914	11.35659835	-0.027302565
10.94524279	10.03621951	-0.017098116
9.72623402	8.979547637	-0.032969753
9.841845331	12.44303142	0.105694953

2.2 The dataset

The data set used is composed of 73 experiments. The first two columns contain the values of pH and molarity of the solution tested, the last column specifies the fitness values computed after the tracking of the droplet. The values of the first 4 data points are shown in Table 1.

2.3 Software used

The library from which the Gaussian process regression model has been implemented is the GPy framework from the Sheffield machine learning group. [5]

2.3.1 Choice of Kernel

The kernel used to describe the covariance function expected in the dataset is the Radial-basis function (RBF) kernel defined on two samples x and x' as:

$$K(x, x') = exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \tag{3}$$

Where $\|x-x'\|^2$ is the squared Euclidean distance between the two samples and σ is the length-scale free parameter describing how smooth the function has to be. The kernel used in th GP model has been implemented as follows:
kg = GPy.kern.RBF(input_dim=2, variance=0.5, lengthscale=1., ARD=True).
Where the **input_dim** parameter represents the dimension of the input space, equals to two variables; the **variance**, determining variation of function values from their mean, has been set to 0.5 and the **lengthscale** to 1.

2.3.2 Model parameters and optimization

The model has been implemented using the kernel in the above section and the noise variance parameter set to 1. This parameter specifies how much noise is expected to be present in the data.
model = GPy.models.GPRegression(XTrain, y_train_set_, kernel=kg, normalizer=False, noise_var=1.)
The model has been later fit to the data using the **model.optimize()** function with maximum number of function evaluations equals to 1000. A value of 10 has been specified for the parameter **n_restarts_optimizer**, which rusn the optimization algorithm as many times as specified, using randomly-chosen starting coordinates to avoid finding a local, rather than a global, maximum in the marginal likelihood.

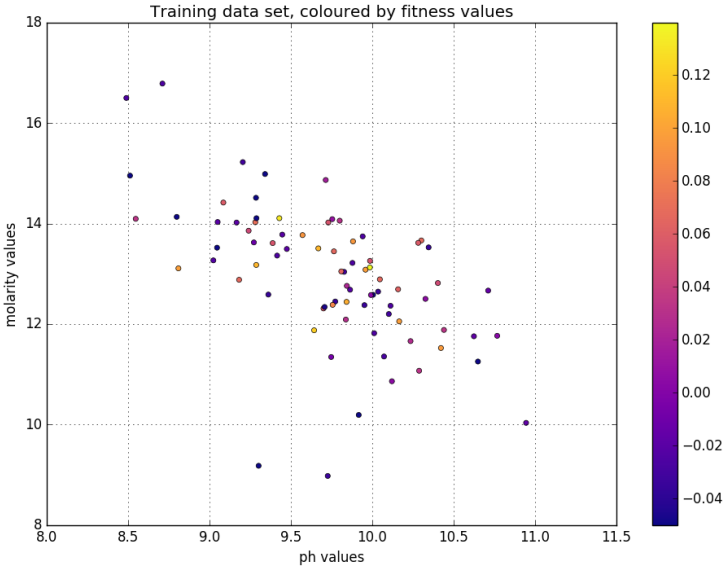

Figure 2: Distribution of the data set colored by fitness values

3 Results

The plot in Figure 2 shows the distribution over the experimental space of the data set. The points are colored accordingly to their fitness value. The plot in Figure 3 shows the Gaussian density distribution of the data before fitting the model.
The plot in Figure 4 shows the Gaussian density distribution after fitting the model on the data set.

4 Discussion

From the plot in Figure 4 it is possible to observe the lighter area bounded by values 9.5¡pH¡10.2 and 12¡molarity¡14. In this area the samples should get higher fitness values than the darker areas depicted around it. However, it can be observed that many samples in that area have darker color, meaning lower fitness values. This could be explained by the fact that a high number of tested individuals, even having very similar pH and molarity values, produced very different results. This is also shown in the plot in Figure 2 where many points close to each other have different fitness values. The chemical system studied is unfortunately highly unstable. In many cases it has been noticed that Marangoni flows, superficial tensions between droplets and the edge of the Petri dish or air flows, affected the movements of the droplet resulting in low fitness values even though the same aqueous phase might have produced higher results in the previous experiment. To smooth the noise introduced in the system the number of experiment for each individual has been increased to four and the fitness value of the experiment was the median of the four replicas instead of

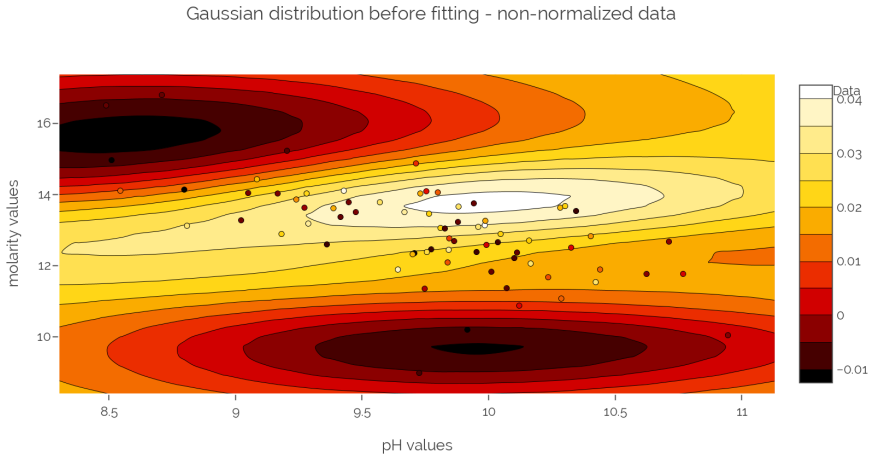

Figure 3: Gaussian distribution before fitting the non normalized data

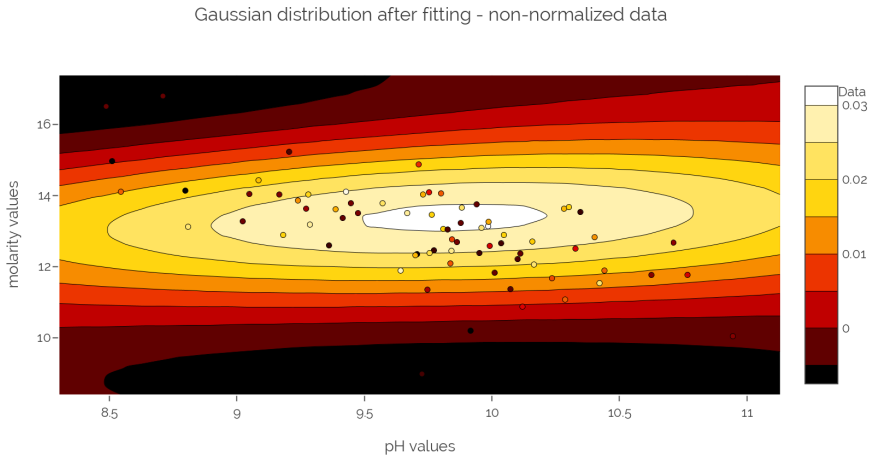

Figure 4: Gaussian distribution after fitting the non normalized data

the average, in order to minimize the influence of the outliers. Regrettably it seems that there still is a high level of noise in the evaluation.

4.1 Future work

To solve the problems related to the chemical system, future work has been programmed to improve the results of these experiments. A change in the protocol for making the solutions is being studied and a covering to avoid air flows from the outside has been designed. On the computational side, different type of kernels or combinations of them could be explored to check for different results. Even though the system is noisy, experiments within the above found range of pH and molarity could be lead in order to check if the fitness values match the ones indicated from the model.

References

[1] https://blogit.itu.dk/evoblissproject/overview/progress/artificial-chemical-life/

[2] https://real.itu.dk/projects/evobliss/

[3] J. Cejkova, M. Novák, F. Stěpánek and M.M. Hanczyc (2014) Dynamics of tactic droplets in salt concentration gradients. *Langmuir* 30(40): 11937-44.

[4] https://blogit.itu.dk/evoblissproject/overview/progress/evolutionary-robotic-platform/

[5] GPy: A Gaussian process framework in python http://github.com/SheffieldML/GPy

[6] Robert, Christian. "Machine Learning, a Probabilistic Perspective." (2014): 62-63.