Extending Bayesian Optimization to non-Classical Problems selected from Industry

Betül Cetinkaya

Hochschule Darmstadt, Fachbereiche MN & I

Abstract

In order to apply Bayesian Optimization to real-world problems, the method got extended for selected problems and tested on simulated data. The selected problems were motivated by industry and are: Discrete Input, Categorical Input and Continuous Context. For each of these problems a solution approach was worked out and applied. Finally the solutions for all three problems were combined to test the effectiveness under combined conditions.

Motivation

The advantages of Bayesian Optimization make it attractive for real-world problems. The method seeks via iterating for a global optimum of an objective without approximating the corresponding objective completely. The intern strategy function, called Acquisition function, selects intelligently the next sample to evaluate. However in real-world, there exist divers properties the method can not handle. Classically, the Gaussian Process is used to model the observations made so far. However, the Gaussian Process can handle only continuous data. The Acquisition function on the other side only handles feasible values without considering environmental influences on the objective. Those environmental influences are meant to be observable but non-controllable. The following problems were selected motivated by industry: discrete input, categorical input and context. The last is an environmental influence. In order to solve the selected problems the Bayesian Optimization was extended for possible solutions.

Theoretical Foudations

Gaussian Process The basic assumption of the Gaussian process is as follows:

1. $\boldsymbol{d(x, x') = 0 \Rightarrow d(y, y') \approx 0}$
2. $\boldsymbol{d(x, x') \approx 0 \Rightarrow d(y, y') \approx 0}$
3. $\boldsymbol{d(x, x') >> 0 \Rightarrow d(y, y')?}$

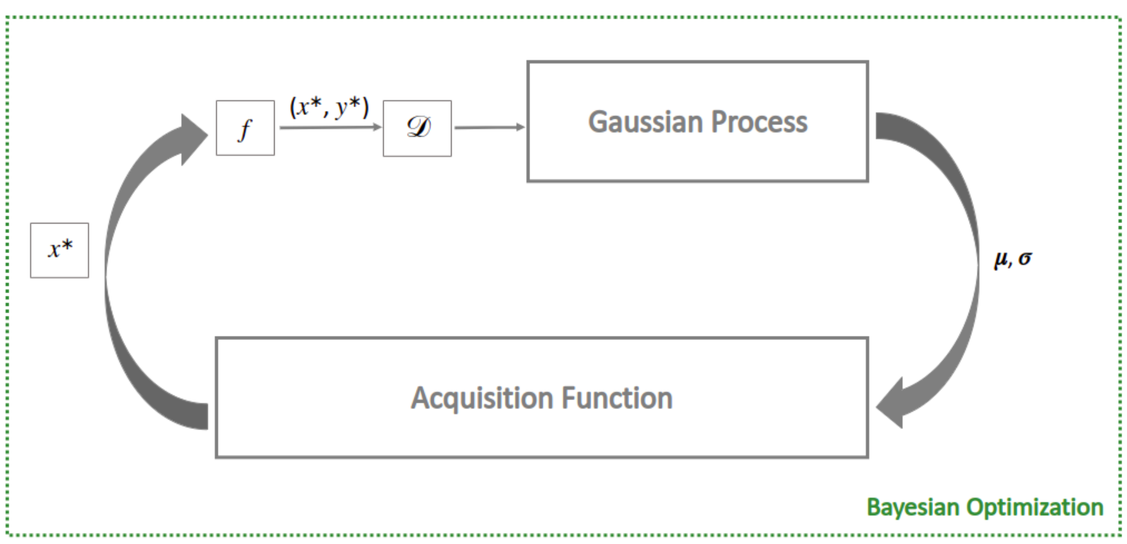
Therefore similar or identical inputs tend to have similar outputs. About unsimilar inputs no information is given. The similarity of inputs is therefore essential when modelling with the Gaussian Process. Information transfer can be done by using the similarity of inputs and resources can be saved.

Problem Description

Discrete input Discrete inputs are numerical discrete and do not have to be integer valued. They also can be equidistant with $\mathbf{0.5}$ step size. However, only equidistant discrete values were considered here. Examples for a discrete input are number of workers, machines or any machine parameter which has only discrete values. Discrete inputs may have the distance 0 if they belong to the same interval. This information is not consider by every distance measure, e.g. euclidean distance. By not recognizing this property, the Acquisition function may suggest a point which is not observed due to lack of feasibility. Consider the Acquisition function suggest the next point to sample $\boldsymbol{x = 4.6}$. However, only integer values are realized, therefore the realized value is $\mathbf{5}$. If this information is not considered by the Gaussian Process the observations of the objective are modelled incorrect and the Acquisition function selects repeatedly the same or similar point.

Categorical input No distinction between ordinal and nominal categories were made. Therefore, the categories were considered as nominal. Examples for categorical inputs are: (ordinal) speed of conveyor belt (low, medium, fast) and (nominal) type of resource (type of grain for producing muesli). Same problem as in discrete input exists here: the similarity can not be recognized correctly and therefore no information transfer can be done. Theretically, the categories could be encoded as dummy one hot encoding(i.e. categ1: $\mathbf{[1, 0, 0]}$, categ2: $\mathbf{[0, 1, 0]}$, categ3: $\mathbf{[0, 1, 0]}$) but in this case the representation of the categories would be orthogonal and therefore independent from each other. In case of independence, no information transfer can be done. Most of the distance measures only distinguish between identical and non-identical and no gradiation of similarity can be done. Therefore an intelligent numerical representation is needed, which maps similar categories closer to each other than unsimilar ones.

Context Context is meant to be an influence on the objective which is not controllable. In real world those are called environmental variable or disruption. A context is observable but non-controllable because it is no subject to human influence. Examples are outdoor temperature (continuous) and number of sick workers (numerical discrete). The problems with context are, they are either modelled by the Gaussian Process and therefore non-feasible next to evaluate points are suggested by the Acquisition function. Or they are not modelled by the Gaussian Process and therefore their influence is not considered resulting in high noise of data.



Methodology

Discrete input In order to recognize values of one interval as the same, a transformation was done for the discrete input. Every incoming value was first mapped to an interval by the transformer and afterwards modelled by the Gaussian Process. In doing so, elements of one interval were recognized as identic and therefore discrete behavior was modelled by the Gaussian Process.

Categorical input The solution of this problem was selected from Word Embedding, a field of Natural Language Process. There, a numeric representation is learnt for each word or category. This numeric representation allows mathematical operations like summation and subtraction. This numerical representation is trained by labelled data and generated by a hidden state after finishing the training process.

Context The solution for the context problem was to model the continuous context by the Gaussian Process first. Afterwards, all feasible values were passed to the Acquisition function so that the best one of these was suggested. The restriction to feasible values was done by considering the current context value as given and non-changeable. Therefore, all others are changeable.

Results

By doing as described, two of three problems were solved. Discrete input and context were solved: the similarities were recognized correctly in case of discrete input and the context was modelled as well as only feasible values were returned by the Acquisition function. However, the solution for categorical input sometimes fails when mapping the categories to numerical space. It may happen that unsimilar categories are recognized as similar or even almost identical. This depends on the training method of the implemented Bayesian Optimization which based on gradient descent and back propagation. The gradient based training method is an obstacle when it comes to order changing: the weights of the linear embedding are initialized randomly. However, the resulting order or the weights may be undesired. Suppose there are 4 categories, categories 1 and 2 are similar to each other and categories 3 and 4 are also similar. If the initialized weights result in categories 2 and 3 have a more similar representation and categories 2 and 1 have an unsimilar representation, meaning 2 and 3 are closer to each other than 2 and 1. Now consider category 3 is located between the both similar categories 1 and 2. Then during training, category 2 learns its unsimilarity to category 3 and therefore deviades from it and therefore can not be close to category 1. However another problem occured in case of categorical input: sometimes unsimilar categories are anyway recognized as similar. In these cases, the model complexity got very high in order to be able being dynamic (after a high point at $\boldsymbol{f(2) = 10}$ may follow a low point $\boldsymbol{f(2.2) = -1.2}$ because the first point belongs to one category and the second one to another. But if the solution for categorical input performed good, information transfer was performed between categories and resources were saved just like in the other problems.

Last but not least, combining all solutions and problems behaved the same: discrete inputs were modelled in a good manner as well as context and categorical inputs sometimes performed good and sometimes bad.

Outlook

- test ordinal categories → is the ordering useful?
- linear embedding
  → multiple ramdom initializations
  → selection model with the lowest complexity
- find & test solutions for further problems (e.g. delaying input)
- analyze Performance of Bayesian Optimization at different dimensions/ different problems

Quellen

[1] S. Almasian, A. Spitz, and M. Gertz, *Word embeddings for entity-annotated texts*, in European Conference on Information Retrieval, Springer, 2019, pp. 307–322.

[2] E. C. Garrido-Merchán and D. Hernández-Lobato, *Dealing with categorical and integer-valued variables in bayesian optimization with gaussian processes*, Neurocomputing, 380 (2020), pp. 20–35.

[3] A. Krause and C. Ong, *Contextual gaussian process bandit optimization*, Advances in neural information processing systems, 24 (2011).

[4] C. E. Rasmussen, *Gaussian processes in machine learning*, in Summer school on machine learning, Springer, 2003, pp. 63–71.