Probabilistic Optimization with Latent Search for Automatic Model Selection

Xiaoyu Lu
University of Oxford
xiaoyu.lu@new.ox.ac.uk

Javier González
Amazon Research Cambridge
gojav@amazon.com

Zhenwen Dai
Amazon Research Cambridge
zhenwend@amazon.co.uk

Neil D. Lawrence
Amazon Research Cambridge & University of Sheffield
lawrennd@amazon.co.uk

Abstract

We tackle the problem of automatic model selection: given a large class of models and a data set, we propose a new procedure to select the best model according to some pre-defined, possibly very expensive, model selection criterion. Although our approach is general, we focus on the class of models defined by the families of kernel combinations in Gaussian Processes (GP). We tackle this problem by making use of a new latent variable probabilistic model, a *Kernel Grammar Variational Autoencoder* (KG-VAE) that we use to embed the model space into some low dimensional continuous manifold. This provides a coherent Euclidean representation of the kernel combinations and a direct way of navigating the kernel space more efficiently in a Bayesian optimization (BO) fashion. The key aspect of this approach is that kernel combinations can be generated *off-line* by using a context-free grammar that can be used to produce the input data used to train the KG-VAE. Some experiments illustrate the utility of this approach.

1 Introduction

The problem of automatic model selection is ubiquitous in science and engineering. In deep learning, choosing the right architecture of the network is crucial to guarantee susses in practical applications [1]. In kernel-based methods, the problem reduces to the selection of an appropriate kernel function together with its hyper-parameters [2, 3]. Despite there has been considerable effort in these fields to propose automatic model selection methods, the *truth* is that in most scenarios previous knowledge is used as the main driver to select the type of model to use.

The *Automatic Statistician* (AS) is one of the most prominent attempts to automate the model selection process. With a Gaussian process (GP) as the class of models of choice, the goal is to automatically select the best kernel structure to explain a data set, which is chosen by enumerating a countably infinite space of arbitrarily complex kernels composed via additions and multiplication of simple ones [4]. Interestingly, the final goal of the AS is not just to fit a model to data, but to write a report that uses the type of combination as the main element to interpret the data. This makes the problem specially hard. Several kernel combinations may have similar prediction power, but when looking for the most interpretable one a balance between the kernel complexity as its prediction power must be taken into account. This implies that $\mathcal{O}(n^3)$ goodness of fit (GOF) measures like Bayesian Information Criterion (BIC) or the model *evidence* need to be optimized to select the best model, which makes brute force computationally intractable.

Several approaches have been proposed to select 'interpretable' kernels in the context of the AS. In the original AS paper a 'greedy' search [3, 4] us used to find the best combination. [5] scales this

7th NIPS Bayesian Optimization workshop in Science and Engineering (NIPS 2017), Long Beach, CA, USA.

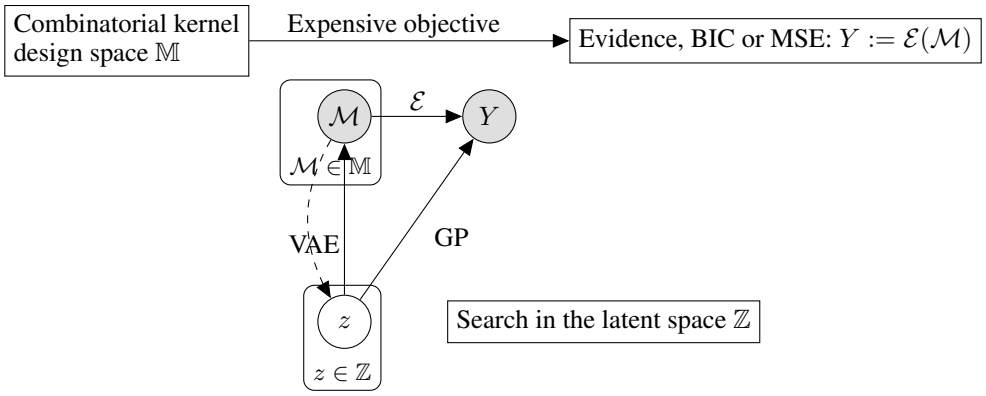

Figure 1: Main elements of PROLAS: a VAE is used to learn the latent space $\mathcal{Z}$ based on data produced by a context-free grammar. BO is applied over the latent space via a GP model to find the optimal kernel combination in GP models.

approach to big data scenarios. In [2] a parametric measure, the *Hellinger distance* is used to define a search over the kernels space using Bayesian optimization (BO). In this work we use a similar idea, but instead of using a parametric distance for a few candidates, we *learn* a low dimensional space where all possible combination are represented. Our algorithm, PROLAS, which stands for Probabilistic Optimization with Latent Search, leverages the idea that kernel combinations can be expressed as operations of a context-free grammar. We use a Variational Auto-encoder (VAE) [6, 7] to learn a well-behaved low-dimensional manifold where the models are naturally represented without making further hypothesis about their distance. Interestingly, this latent space can be learned *off-line* using data produced by the grammar. BO is later used to connect the latent space with the GOF of interest by means of a GP. See Figure 1 for a graphical illustration of the method proposed in this work. Section 2 describes the proposed approach. In Section 3 we illustrate its performance with a series of experimental results. In Section 4 we include some conclusions and further lines of research derived from this work.

2 Variational Auto-encoders for optimal search in Kernel Spaces

2.1 Problem description

Our goal is to solve a supervised learning problem between the spaces $\mathcal{X}$ and $\mathcal{Y}$ given a dataset of observations $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ where $\mathbf{x}_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. Suppose that we have a class of probabilistic models $\mathbb{M}$ that we consider adequate to represent the data. We call $\mathcal{M}$ each probabilistic model of this set and we denote by $\Theta_{\mathcal{M}}$ its associated parameter space. The models in $\mathbb{M}$ will represent different structural assumptions about the data as trends of periodicity and the values of $\theta \in \Theta$, the model hyper-parameters, will differentiate models within the same family. In particular we will consider $\mathbb{M}$ to be the family of GPS with different kernel combinations.

The problem to solve is to select one single model from $\mathbb{M}$ that explains our data $\mathcal{D}$ the best. In this paper we will use the BIC for consistency with [4] although the evidence would be a more appropriate measure in this scenario. The problem is therefore reduced to find

$$\mathcal{M}_\theta^* := \arg\max_{\mathcal{M} \in \mathbb{M}} \mathcal{E}(\mathcal{M}). \tag{1}$$

where $\mathcal{E} : \mathbb{M} \to \mathbb{R}$ is the chosen 'goodness of fit measure'. Both the evidence and and the BIC scale cubically with the number of observations N and gradient information is not available given the discrete nature of $\mathbb{M}$. Therefore standard optimization methods do not apply to solve (1). The direction we take in this work is to find low dimensional Euclidean representation of $\mathbb{M}$ where Bayesian optimization principles can be applied.

2.2 Grammar and Data-driven based Kernel Representation

Following [4, 3, 2], we know that it is possible to generate a countably infinite kernel space through closure of kernels by means of a context free grammar. In particular, given a set of base kernels $\mathcal{B}$ we can generate an expression $\mathcal{S}$ representing a kernel combination by subsequent operations $\mathcal{O}$,

Algorithm 1 Context-free grammar for kernel expressions generation.

Input: N_{max}, $p_{\mathcal{B}}$, $p_{\mathcal{O}}$, $\mathcal{S} = \emptyset$.
repeat
  Update $\mathcal{S}$ by choosing one of the base kernel from $\mathcal{B}$ with prior probability $p_{\mathcal{B}}$.
  Update $\mathcal{S}$ by choosing one operation from $\mathcal{O}$ from with prior probability $p_{\mathcal{O}}$.
until Operation is *Stop* or the number of applied operations is N_{max}.

additions, multiplications, replacements or stop. Both the kernels and the operations are chosen according to pre-specified probabilities $p_{\mathcal{B}}$ and $p_{\mathcal{O}}$. See Algorithm 1 for details.

To represent the expression $\mathcal{S}$ in a way that it is useful for our purposes, we use 1-vectors for both the kernels and the operations. Suppose that $\mathcal{B} = \{K_1, K_2, K_3, K_4\}$ is the set of four base kernels. We also have the set of operations $\mathcal{O} = \{+, \times, Stop\}$. We transform the expression $\mathcal{S}$ into a binary vector by recurrently attaching the 1-hot vector of each kernel and operation. When the operation is $Stop$, we complete the vector with zeros. For instance, in the following example, four kernels are combined before termination:

$$\underbrace{K_2}_{1000}\ \underbrace{+}_{100}\ \underbrace{K_1}_{0100}\ \underbrace{*}_{010}\ \underbrace{K_3}_{0010}\ \underbrace{*}_{010}\ \underbrace{K_1}_{1000}\ \underbrace{Stop}_{001}\ \underbrace{...}_{000}$$

This representation, that we denote by $\mathbf{r}_g$, captures the complexity of the combination but it does not take into account of the differences in the kernels due to the dataset $\mathcal{D}$. To this end we combine the previous *grammar-driven* representation of the kernels with a *data-driven* representation. In particular, we use a measure of distance between the kernel matrices of the base kernels evaluated in the data and the kernel matrices of the combinations for some values of the hyper-parameters θ of the kernels. For each combination, this provides a vector of size $|\mathcal{B}|$ that takes into account the properties of the data of the problem. A proper distance to use is the *Hellinger distance* but it is computationally very demanding [2]. Although not optimal, we observed that the Frobenius distance, which is very quick to compute, was doing a reasonable job. We denote this representation of the kernel combinations by $\mathbf{r}_d$. The final representation for any each combination is therefore $\mathbf{r} = [\mathbf{r}_g, \mathbf{r}_d]$, which has dimension $(N_{max}+1)|\mathcal{B}| + (N_{max}-1)|\mathcal{O}|$ for N_{max} the maximum number of allowed operations (added kernels).

2.3 Kernel Grammar Variational Auto-encoder (KG-VAE)

This section describes a bespoke VAE for our problem, the Kernel Grammar Variational Auto-encoder (KG-VAE). It is used to learn a latent where all possible candidate models in $\mathbb{M}$ are represented. The key elements of KG-VAE are the encoder and the decoder. The *encoder* is a Gaussian distribution where the mean is the output of one feed-forward layer with *Relu* activation function:

$$\log q(\mathbf{z}|\mathbf{r}) = \log \mathcal{N}(\mathbf{z} : \mu, \sigma^2 \mathbf{I}) \ \text{ for } \mu = \mathbf{W}_2 \mathbf{h} + b_2 \ , \ h = \text{Relu}(\mathbf{W}_1 \mathbf{r} + b_1),$$

where $\mathbf{W}_1$, $\mathbf{W}_2$ and b_1 and b_2 are the weights of the model, which can be extended to with more intermediate layers. For the *decoder* let $\pi = \exp(\mathbf{W}_4 \text{Relu}(\mathbf{W}_3 \mathbf{z} + b_4) + b_5)$ be the mapping parameters of a point $\mathbf{z}$ in the latent space where $\mathbf{W}_3$, $\mathbf{W}_4$ and b_4 and b_5 are extra parameters of the model. Consider the partitions of $\boldsymbol{\pi} = [\boldsymbol{\pi}_d, \boldsymbol{\pi}_g]$ and $\mathbf{r} = [\mathbf{r}_d, \mathbf{r}_g]$ where the subindice g refers to the components of the grammar representation and d for the distance based representation. The likelihood of the decoder can by factored as $p(\mathbf{r}|\mathbf{z}) = p(\mathbf{r}_d|\boldsymbol{\pi}_d) p(\mathbf{r}_g|\boldsymbol{\pi}_g)$ where $p(\mathbf{r}_g|\boldsymbol{\pi}_g) = p(\mathbf{r}_{g_1}) \prod_{j=1}^{N_{max}} p(\mathbf{r}_{g_{j+1}}|\mathbf{r}_{g_j})$ is the likelihood of the grammar representation. Each factor $p(\mathbf{r}_{g_{j+1}}|\mathbf{r}_{g_j})$ follows a Multinomial distribution with normalised parameter $\boldsymbol{\pi}_j / \textit{sum}(\boldsymbol{\pi}_j)$. The conditional dependency on the previous factor refers to the appearance of Stop operation. An advantage of the KG-VAE is that the kernel we learn is always feasible. Note also that the generation of the grammars and the training of KG-VAE can all be done off-line, as we do not need to evaluate the objective in (1) for each kernel data point used in the training phase.

2.4 Probabilistic Search on Latent Model Spaces

Let $\mathbb{Z}$ the latent space learned by the KG-VAE. To approach (1) we apply BO on $\mathbb{Z}$. Denote by $\mathcal{M}_{\mathbf{r}}$ the model associated to the representation $\mathbf{r}$. We reformulate (1) as

$$\mathbf{z}^* := \arg\max_{\mathbf{z} \in \mathbb{Z}} \ \mathcal{E}(\mathcal{M}_{\tilde{\mathbf{r}}|\mathbf{z}}). \tag{2}$$

Algorithm 2 The PROLAS algorithm.

Input: $\mathcal{D}$, $\mathcal{B}$, $p_{\mathcal{B}}$, $\mathcal{O}$, $p_{\mathcal{O}}$, N_{max} and N_{iter}.

Use Algorithm 1 to generate a series of models $\mathcal{M}$ and their representations $\mathbf{r}$.
Train a KG-VAE on the generated set.
Compute $\tilde{\mathcal{D}}_1 = \{(\mathbf{z}_1, \mathcal{E}(\mathcal{M}_{\tilde{\mathbf{r}}|\mathbf{z}_1}))\}$

Off-line

for $j = 0$ **to** N_{iter} **do**

1. Fit a GP with kernel k to the generated representation.
2. Optimize the acquisition $\alpha(\mathbf{z})$ and obtain $\mathbf{z}_{j+1}$.
5. Augment $\tilde{\mathcal{D}}_{j+1} = \{\tilde{\mathcal{D}}_j \cup [\mathbf{z}_{j+1}, \mathcal{E}(\mathcal{M}_{\tilde{\mathbf{r}}|\mathbf{z}_{j+1}}))\}$.

end for

Returns: Report $\mathbf{z}^{\star}_{N_{iter}}$.

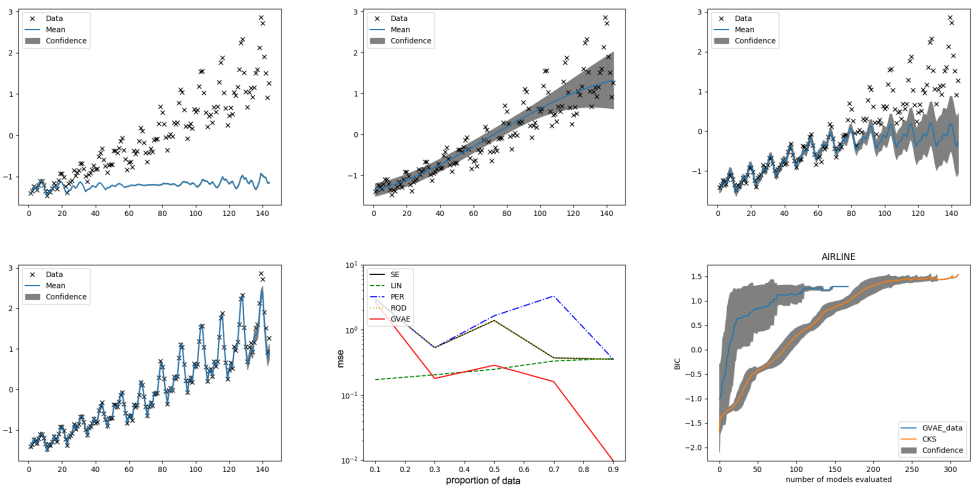

Figure 2: Results for the Airline dataset using PROLAS and comparison with CSK (bottom right).

where $\tilde{\mathbf{r}}|\mathbf{z}$ is the mode of $p(\mathbf{r}|\mathbf{z})$. We use BO to search the optimal $\mathbf{z}^*$: we select a series of locations $\mathbf{z}_1, \ldots, \mathbf{z}_{N_{iter}}$ such that the minimum of $\mathcal{E}(\mathcal{M}_{\tilde{\mathbf{r}}|\mathbf{z}})$ is evaluated as quickly as possible. We use a GP $p(f) = \mathcal{GP}(\mu; k)$ with mean function μ and positive-definite kernel k that we fit to the currently observed locations. Under Gaussian likelihoods, the posterior distribution of f (for a sample of size n) is also a GP, with posterior mean and variance available in closed form [8]. The posterior of the GP is used to form the acquisition function α. The next evaluation is placed at the global maximum of this acquisition function [9]. See Algorithm 2.

3 Experiments

We applied the PROLAS algorithm to fit the the Airline Passenger Data, a time series with 144 time steps. We used 4 base kernels, Squared exponential, Linear, Periodic and Rational Quadratic. We used uniform prior probabilities in both the grammar operations and the base kernels. We allow a maximum of 5 operations in the grammar generating process. The used KG-VAE we has 2 hidden layers with 400 hidden units. In Figure 2 (top row and bottom left) we show the fit of the model proposed by PROLAS to the data set when an increasing number of training data are used (10%, 25%, 50% and 75%). Interestingly, 10% of the data already some of the seasonal structure is captured. With 75% of the data the fit in the test set is excellent. In Figure Figure 2 (bottom row, center) we show the mean square error in the prediction set for base kernels and in the combination proposed by PROLAS, which consistently improve the base kernels, specially when the data size increases. Finally we compare PROLAS with the greedy *compositional kernel search (*CSK*)* [4]. In Figure 2 (bottom row, right) we show the BIC found in terms of the number of evaluated models for 20 runs of with different initialization. PROLAS converges to a better solution faster than CSK.

4 Conclusions and future work

In this work we have presented a new algorithm, PROLAS, to perform Bayesian optimization in the context of the Automatic Statistician. The methods follow some previous efforts to optimize in combinatorial spaces [7]. It is based on a new KG-VAE that allows to map an arbitrarily large number of GP models into a low dimensional Euclidean space in which BO is feasible. A first set of experiments show the utility of this approach. Further validation in other datasets and the deep analysis of the properties of the method are left as future work. Further efficiency by using parallel approaches [10] will also be investigated.

References

[1] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. *Journal of Machine Learning Research*, 2015.

[2] Gustavo Malkomes, Charles Schaff, and Roman Garnett. Bayesian optimization for automated model selection. In *Advances in Neural Information Processing Systems*, pages 2900–2908, 2016.

[3] Roger Grosse, Ruslan R Salakhutdinov, William T Freeman, and Joshua B Tenenbaum. Exploiting compositionality to explore a large space of model structures. *arXiv preprint arXiv:1210.4856*, 2012.

[4] David Duvenaud, James Robert Lloyd, Roger Grosse, Joshua B Tenenbaum, and Zoubin Ghahramani. Structure discovery in nonparametric regression through compositional kernel search. *arXiv preprint arXiv:1302.4922*, 2013.

[5] Hyunjik Kim and Yee Whye Teh. Scalable structure discovery for regression using gaussian processes. *AutoML 2016 Proceedings, Journal of Machine Learning Research Workshop and Conference Proceedings*, 2016.

[6] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[7] Matt J. Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 1945–1954. PMLR, 2017.

[8] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[9] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[10] J. González, Z. Dai, P. Hennig, and N. Lawrence. Batch bayesian optimization via local penalization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS 2016)*, volume 51 of *JMLR Workshop and Conference Proceedings*, pages 648–657, 2016.