Biological assay modelling with adaptive deep kernel learning

Prudencio Tossou[*]
InVivo AI, Université Laval
prudencio@invivoai.com

Basile Dura
InVivo AI, Mila
basile@invivoai.ca

Mario Marchand
Université Laval
mario.marchand@ift.ulaval.ca

François Laviolette
Université Laval
francois.laviolette@ift.ulaval.ca

Alexandre Lacoste
Element AI
allac@elementai.com

Abstract

During an active drug discovery program, biological assay modelling is best viewed as a few-shot regression problem. In this work, we developed a method better suited for this problem, as existing ones do not perform well on noisy and complex molecular task distributions. The method learns a kernel family suitable for a task distribution and selects the appropriate kernel for each task during inference. To demonstrate its effectiveness, we introduce bioassay modelling datasets and demonstrate better performances than the state of the art in the noisy and uncertain environments of biological experiments.

1 Motivation

Deep learning methods are now pushing the frontiers of pharmaceutical R&D following breakthroughs in domains like computer vision, autonomous driving, and natural language processing [1–4]. So far, the common characteristic among successful applications is the availability of high quality/quantity of training data. However, relatively few tasks in pharmaceutical R&D can fulfill these data requirements. For example, under the constraints of an active drug discovery program, the data from biological assays (or bioassays) is often relatively small and noisy. Modelling such assays is thus best viewed as a few-shot regression problem, with many variables (e.g. organisms, cell lines, readouts, experiment conditions, etc) accounting for the data distribution generated within each assay. Due to the molecular space (10^60) and the dataset (< 100) sizes, obtaining models that are highly accurate, noise-resistant with well-calibrated uncertainty estimation is a challenging task. However, accuracy and noise-robustness are critical as bioassay models are primarily intended to become surrogates for the actual assays. Uncertainty estimation is also required because models will be used for prioritizing molecules in subsequent experiments and better exploring of the chemical space. To meet those needs powerful few-shot regression algorithms are required.

Recent advances in episodic meta-learning (or few-shot learning) have led to new algorithms that learn efficiently and generalize well from small amounts of training data [5, 6]. Most operate by learning some prior knowledge across a large collection of tasks, to then transfer and adapt it to new tasks that have limited amounts of data [7, 8]. The nature of the meta-knowledge captured or the amount of adaptation performed for new tasks are the main differences between few-shot algorithms. To successfully use those algorithms for bioassay modelling, the meta-knowledge must be sufficiently rich to allow quick extrapolation, prediction and uncertainty estimation in unseen regions of the huge

[*]

3rd NeurIPS workshop on Meta-Learning (MetaLearn 2019), Vancouver, Canada.

chemical space. Moreover, given the rapidly changing data distribution from one bioassay to another, greater adaptation capacity is required and must be accounted for during modelling.

In previous work, metric learning methods [9–13] accumulate meta-knowledge in high capacity and rich metric functions and use simple base-learners such as k-nearest neighbor [11, 10] or low capacity neural networks [12] to produce adequate models for new tasks. However, they do not adapt the covariance functions nor the base-learners at test-time. Initialization- and optimization-based methods [14–16] that learn the initialization points and update rules for gradient descent-based algorithms, respectively, allow for improved adaptation on new tasks but remain compute-intensive and memory inefficient. As both improved adaptation and rich meta-knowledge are required for bioassay modelling, we combine the strengths of existing methods and propose ADKL (Adaptive Deep Kernel Learning) , which frames few-shot regression as a deep kernel learning (DKL) problem. It relieves the burden put on the feature extractor in vanilla DKL methods by adapting the kernel to the distribution at test time.

2 Method

ADKL is inspired by the DKL framework introduced by Wilson et al. [17], which combines the flexibility of kernel methods with deep neural networks for single task learning. This new algorithm allows us to learn a rich data-driven prior during meta-learning, increase test-time adaptation capacity, incorporate domain-specific knowledge, and estimate the predictive uncertainty.

Preliminaries: For a task t, the DKL framework takes a training dataset $D^t_{\text{train}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$, and produces a predictor of the form:

$$h^t_*(\mathbf{x}) = \sum_{(\mathbf{x}_i, y_i) \in D^t_{\text{train}}} \alpha^t_i k_{\boldsymbol{\rho}}(\boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x}), \boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x}_i)), \tag{1}$$

where $\boldsymbol{\phi}_{\boldsymbol{\theta}}$ is a deep neural network that embeds inputs in a space $\mathcal{H}$, $k_{\boldsymbol{\rho}}$ is a kernel function over $\mathcal{H}$ with hyperparameters $\boldsymbol{\rho}$ and $\boldsymbol{\alpha}^t = (\alpha^t_1, \cdots, \alpha^t_m)$ are weights given by a differentiable kernel method such as Gaussian Process (GP) or Kernel Ridge Regression (KRR). The framework is highly expressive and efficient, but is limited to single task learning settings because it requires as many samples as deep learning methods and is as slow as kernel methods. Generalizing it to few-shot learning, we are able to alleviate these concerns to obtain a more powerful framework. It also enables the incorporation of domain-specific knowledge in the meta-learning through the choice of the kernel family.

ADKL: For the following, let us consider that one has access to a meta-training collection $\mathscr{D}_{\text{meta}-\text{train}} := \left\{(D^{t_j}_{\text{train}}, D^{t_j}_{\text{valid}})\right\}_{j=1}^T$, of T tasks to *learn how to learn* from few datapoints. Each task t_j has its own training (or support) set $D^{t_j}_{\text{train}}$ and validation (or query) set $D^{t_j}_{\text{valid}}$. A meta-testing collection $\mathscr{D}_{\text{meta}-\text{test}}$ is also available to assess the generalization across unseen tasks. DKL can be generalized to few-shot learning by sharing the deep neural network parameters $\boldsymbol{\theta}$ and the kernel hyperparameters $\boldsymbol{\rho}$ among tasks in the meta-training loop and optimizing the following expected loss on all tasks:

$$\underset{\boldsymbol{\theta}, \boldsymbol{\rho}}{\text{argmin}} \underset{t \sim \mathscr{D}_{\text{meta}-\text{train}}}{\mathbf{E}} \underset{\mathbf{x}, y \sim D^{t_j}_{\text{valid}}}{\mathbf{E}} l(h^{t_j}(\mathbf{x}), y), \tag{2}$$

where h^{t_j} is given by Equation 1 using the training set $D^{t_j}_{\text{train}}$, and l is the loss function optimized by the kernel method (i.e. the quadratic loss for KRR and the negative marginal likelihood for GP). Briefly, DKL for few-shot learning (FS-DKL) consists of finding a representation common to all tasks such that a kernel method (in our case, GP or KRR) will generalize well from a small number of samples. It gives principled meta-learning algorithms with better adaptation capacity than metric learning at test-time (as enabled by the kernel methods).

FS-DKL assumes that one kernel will adequately model all tasks, which may be impossible for our bioassay modelling tasks. Consequently, we increase modelling capacity by proposing ADKL, which learns an adaptive and task-dependent kernel rather than a single shared kernel. We define an adaptive kernel as follows:

$$k_{\text{ADKL}}(\mathbf{x}, \mathbf{x}'; \mathbf{z}_{t_j}) := k_{\boldsymbol{\rho}}(\boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x}; \mathbf{z}_{t_j}), \boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x}'; \mathbf{z}_{t_j})) \tag{3}$$

where $\mathbf{z}_{t_j} = \boldsymbol{\psi}_{\boldsymbol{\eta}}(D_{\text{train}}^{t_j})$ is a task embedding obtained by transforming a training set $D_{\text{train}}^{t_j}$ with a task encoding network $\boldsymbol{\psi}_{\boldsymbol{\eta}}$ with parameters $\boldsymbol{\eta}$. The latter is a modified version of DeepSets, an order invariant network proposed by Zaheer et al. [18], and captures the first and second moments of the data distribution that have generated each $D_{\text{train}}^{t_j}$. Once $\mathbf{z}_{t_j}$ is obtained, we compute the conditional input embedding using the function $\boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x}; \mathbf{z}_{t_j})$ by transforming $\mathbf{x}$ with a neural network $\mathbf{u}$, concatenating $\mathbf{u}(\mathbf{x})$ with $\mathbf{z}_{t_j}$, and applying a non-linear mapping to this concatenation. This nonlinear mapping allows the capture of complex interactions between the task and the input representations. The adaptive kernel is then computed using Equation 3 and h^{t_j} is obtained using a differentiable kernel method. ADKL-GP and ADKL-KRR will be used to refer to the algorithms we derived from the framework when the respective task-level regressors are obtained using GP and KRR.

3 Experiments

3.1 Datasets

We introduce and use two benchmarks for bioassay modelling in our experiments: Binding and Antibacterial (available here).

1. **Binding**: The goal of each task in this collection is to predict the binding affinity of small, drug-like molecules to a protein. The collection was extracted from the public database BindingDB and encompasses 7,620 tasks, each containing between 10 and 9,000 samples. Each task is characterized by the protein for which molecule bindings are measured.
2. **Antibacterial**: The goal here is to predict the antimicrobial activity of small molecules for various bacteria. The task collection was extracted from the public database PubChem and contains 3,842 tasks, each consisting of 10 to 225 samples. A task is characterized by a bacterial strain.

For both collections, the molecules are represented by their SMILES (documentation here), which describe of the molecular structures using short ASCII strings. All models evaluated on these collections share the same input feature extractor configuration: a 1-D CNN of 2 layers of 128 hidden units each and a kernel size of 5. We use CNN instead of LSTM or advanced graph convolution methods for scalability reasons. Moreover, the targets were scaled linearly between 0 and 1.

We also include the same Sinusoids task collection proposed by Kim et al. [15] in our experiments to see our method performance outside of bioassay modelling tasks.

3.2 Benchmarking analysis

We evaluate model performance against R2-D2 [13], CNP[19], and MAML[14]. R2-D2 is a natural comparison to ADKL-KRR (when the latter uses the linear kernel) to show whether the adapted deep kernel provides more test-time adaptation. CNP is also a natural comparison to ADKL-GP and will help measure performance differences between the task-level Bayesian models generated within the GP and CNP frameworks. MAML is considered herein for its fast-adaptation at test-time and as the representative of initialization and optimization based models. In the following experiments, all DKL methods use the linear kernel.

Our first set of experiments evaluates performance on both the real-world and toy tasks. We train each method using support and query sets of size $m = 10$. During meta-testing, the support set size is also $m = 10$, but the query set consists of the remaining samples of each task. For datasets lacking sufficient samples (from the Binding and Antibacterial collections), we use half of the samples in the support set and the remaining in the query set. For each task, during meta-testing, we average the Mean Squared Error (MSE) over 20 random partitions of the query and support sets. We refer to this value as the task MSE. Table 1 shows the mean MSE over tasks in the meta-test set for all the benchmarks and algorithms. In general, we observe that the real-world datasets are challenging for all methods but ADKL-KRR methods consistently outperform R2-D2 and CNP. The gap between ADKL-KRR and R2-D2 shows the importance of adapting the kernel to each task rather than sharing a single kernel. The direct comparison of ADKL-KRR and R2-D2 shows that adapting the kernel to each task helps significantly compared to a shared single kernel.

model	**Antibacterial**	**Binding**	**Sinusoids**
ADKL-GP	0.087	0.093	0.327 ± 0.042
ADKL-KRR	**0.078**	**0.088**	**0.138** ± 0.004
BMAML	0.106	0.094	0.744± 0.020
CNP	0.083	0.094	0.607± 0.056
Learned Basis	0.091	0.101	0.475 ± 0.063
Proto-MAML	0.092	0.090	1.457 ± 0.029
R2-D2	0.093	0.098	0.172± 0.011
MAML	–	–	1.651± 0.040

Table 1: Meta-test MSE. The uncertainty was computed over 10 runs for the Sinusoids

3.3 Active Learning

In our second set of experiments, we intent to measure the effectiveness of the uncertainty captured by the predictive distribution of ADKL-GP for active learning. CNP, in comparison, serves to measure which of CNP and GP better captures the data uncertainty for improving FSR under active sample selection. For this purpose, we meta-train both algorithms using support and query sets of size $m = 5$ (and $T \in \{100, 1000\}$ for the Sinusoids collection). During meta-test time, five samples are randomly selected to constitute the support set D_{train} and build the initial hypothesis for each task. Then, from a pool U of unlabeled data, we choose the input $\mathbf{x}^*$ of maximum predictive entropy, i.e., $\mathbf{x}^* = \text{argmax}_{\mathbf{x} \in U} \mathbb{E}\left[\log p(y|\mathbf{x}, D_{\text{train}})\right]$. The latter is removed from U and added to D_{train} with its predicted label. The within-task adaptation is performed on the new support set to obtain a new hypothesis which is evaluated on the query set D_{valid} of the task. This process is repeated until we reach the allowed budget of 20 queries.

Figure 1 highlights that, in the active learning setting, ADKL-GP consistently outperforms CNP. Very few samples are queried by ADKL-GP to capture the data distribution, while CNP performance is far from optimal, even when allowed the maximum number of queries. Also, since using the maximum predictive entropy strategy is better than querying samples at random for ADKL-GP (solid vs. dashed line), these results suggest that the predictive uncertainty obtained with GP is informative and more accurate than that of CNP. Moreover, when the number of queries is greater than 10, we observe a performance degradation for CNP, while ADKL-GP remains consistent. This observation highlights the generalization capacity of DKL methods, even outside the few-shot regime where they have been trained — this same property does not hold true for CNP. We attribute this property of DKL methods to their use of kernel methods. In fact, their role in adaptation and generalization increases as we move away from the few-shot training regime.

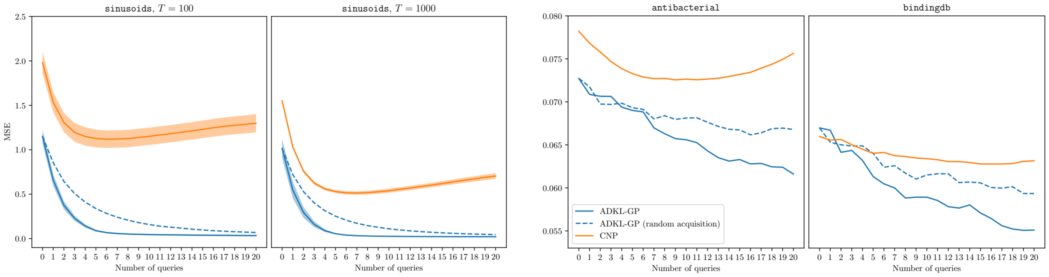

Figure 1: Average MSE performance on the meta-test during active learning. The width of the shaded regions denotes the uncertainty over five runs for the `sinusoidal` collection. No uncertainty is shown for the real-world tasks as they were too time consuming.

Conclusion

In this paper, we investigated bioassay modelling using few-shot regression (FSR) algorithms. We propose a new kernel based framework for FSR and demonstrate its superiority in test-time adaptation to the current state of the art methods. Also, by making our bioassay modelling collections publicly available, we hope that the community will leverage them to propose FSR algorithms that are ready to be deployed in real-life settings, in turn having a positive impact on drug discovery.

References

[1] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. Deep learning in bioinformatics. *Briefings in bioinformatics*, 18(5):851–869, 2017.

[2] Youjun Xu, Kangjie Lin, Shiwei Wang, Lei Wang, Chenjing Cai, Chen Song, Luhua Lai, and Jianfeng Pei. Deep learning for molecular generation. *Future medicinal chemistry*, 11(6): 567–597, 2019.

[3] Marwin HS Segler and Mark P Waller. Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Chemistry–A European Journal*, 23(25):5966–5971, 2017.

[4] Marwin HS Segler, Mike Preuss, and Mark P Waller. Learning to plan chemical syntheses. *arXiv preprint arXiv:1708.04202*, 2017.

[5] Yaqing Wang and Quanming Yao. Few-shot learning: A survey. *CoRR*, abs/1904.05046, 2019. URL `http://arxiv.org/abs/1904.05046`.

[6] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

[7] Sebastian Thrun and Lorien Pratt. Learning to learn: Introduction and overview. In *Learning to learn*, pages 3–17. Springer, 1998.

[8] Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18(2):77–95, 2002.

[9] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, 2015.

[10] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[11] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

[12] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017.

[13] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

[14] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[15] Taesup Kim, Jaesik Yoon, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. *arXiv preprint arXiv:1806.03836*, 2018.

[16] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

[17] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378, 2016.

[18] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, pages 3391–3401, 2017.

[19] Marta Garnelo, Dan Rosenbaum, Chris J Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo J Rezende, and SM Eslami. Conditional neural processes. *arXiv preprint arXiv:1807.01613*, 2018.