Black-Box Hyperparameter Optimization for Nuclei Segmentation in Prostate Tissue Images

Thomas Wollmann[1], Patrick Bernhard[1], Manuel Gunkel[2], Delia M. Braun[3], Jan Meiners[4], Ronald Simon[4], Guido Sauter[4], Holger Erfle[2], Karsten Rippe[3], Karl Rohr[1]

[1] University of Heidelberg, BioQuant, IPMB, and DKFZ Heidelberg,
Biomedical Computer Vision Group
[2] High-Content Analysis of the Cell (HiCell) and Advanced Biological Screening Facility, BioQuant, University of Heidelberg, Germany
[3] Division of Chromatin Networks, DKFZ and BioQuant, Heidelberg, Germany
[4] Department of Pathology, University Medical Center Hamburg-Eppendorf, Germany
thomas.wollmann@bioquant.uni-heidelberg.de

Abstract. Segmentation of cell nuclei is essential for analyzing high-content histological screens. Often, parameters of automatic approaches need to be optimized, which is tedious and difficult to perform manually. We propose a novel hyperparameter optimization framework, which formulates optimization as a combination of candidate sampling and an optimization strategy. We present a clustering based and a deep neural network based pipeline for nuclei segmentation, for which the parameters are optimized using state of the art optimizers as well as a novel optimizer. The pipelines were applied to challenging prostate cancer tissue images. We performed a quantitative evaluation using 28,388 parameter settings. It turned out that the deep neural network outperforms the clustering based pipeline, while the results for different optimizers vary slightly.

1 Introduction

The segmentation of cell nuclei in histological prostate tissue images is a crucial task to stratify prostate cancer. In particular, the properties of the microscopy data with regard to contrast, noise, cell clustering, edge information, shape variation, and intensity variation determine the complexity of the required segmentation pipeline. Generally, a complex pipeline is necessary for robustly segmenting heterogeneous data (Fig. 1), while the segmentation result highly depends on the used parameters. Since manual parameter optimization of complex algorithms is very time-consuming and difficult, automated parameter optimization is required. However, for complex pipelines the objective function is usually not fully differentiable, which prevents using first or higher order optimization methods. Instead, zero order optimization (black-box optimization) [1] can be performed without using further information of the objective function. Black-box optimization uses a limited number of evaluations of the objective function, and the non-convex optimization tries to determine the (local) optimum by finding the best parameters. For machine learning systems, black-box optimization can be used for automatically tuning hyperparameters as done for denoising algorithms [2], for simulated objective functions [3] or for cell segmentation in tissue images [4]. However, to our knowledge, a systematic evaluation on the applicability of automated black-box optimization has not been conducted for hyperparameter optimization of cell nuclei segmentation pipelines. Existing optimization frameworks like Spearmint [5], Hyperopt [6], Scikit-Optimize [7], or Google Vizier [8] do not satisfy the demands of cell nuclei segmentation as they have a low ease of use (e.g., mix of programming languages, workspace management), employ only few optimizers or only offer limited expandability. Furthermore, Google Vizier is not publicly available.

In this work, we introduce a novel black-box optimization framework, where hyperparameter optimization is formulated as a combination of candidate sampling and an optimization strategy. Our framework allows a modular design of new optimizers as well as quickly implementing state of the art optimizers. We applied our framework to a clustering based pipeline as well as a deep neural network based pipeline to segment cell nuclei in challenging prostate cell tissue images. We evaluated the pipelines using different optimizers and 28388 parameter settings. We provide insights into cell nuclei segmentation and suggest common practices for hyperparameter optimization in this application.

2 Methods

We investigated two nuclei segmentation pipelines, one based on K-means clustering and the other based on a U-Net convolutional neural network (CNN). The hyperparameters of the pipelines were optimized using our novel distributed black-box optimization framework.

2.1 Segmentation pipelines

Clustering based segmentation The pipeline involves several parameters (in the following highlighted in italic). An image is smoothed by a Gaussian filter (sigma) before performing K-means clustering using intensity values (cluster initialization method). Cluster initialization with a random seed value leads to a

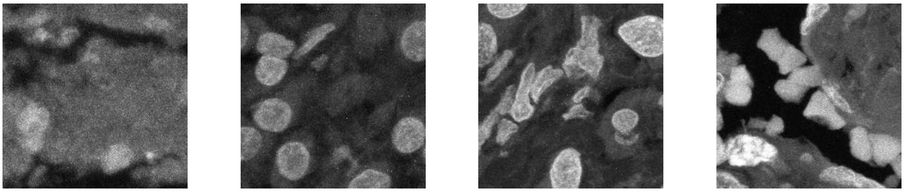


Fig. 1. Examples of prostate tissue images with various challenges for image analysis.(a) Strong background noise, (b) Low contrast, (c) Strong shape variation, (d) Strong intensity variation.

non-deterministic pipeline. To avoid this, we set the seed value to a fix value. Median filter and morphological closing of small holes are applied subsequently. By comparing a selected geometric feature of each cluster to the mean of all clusters, one cluster is assigned as foreground, whose labels are subsequently thresholded with regard to the geometric features area (upper and lower threshold) and solidity before using the foreground cluster as segmentation result.

CNN based segmentation We train a U-Net [9] on the respective training and validation datasets using the Adam optimizer and early stopping. For training we perform offline data augmentation using rotation, flipping, and elastic deformation. The local minimum found by Adam highly depends on the initialization of the network. Therefore, for a fair comparison we use the same seed value for sampling the initial network weights in all experiments. Small segmented objects are discarded using a threshold for the area. The parameters of this pipeline are the learning rate, batch size, and area threshold.

2.2 HyperHyper optimization framework

Our proposed distributed black-box optimization framework *HyperHyper* subdivides hyperparameter optimization in a hyperparameter candidate sampler and an optimization strategy. Candidate sampler and optimization strategy can be selected from a model zoo to form an optimizer for a specific application. Sequential model-based optimization (SMBO) is performed by sampling candidates and evaluating or dismissing them (Fig. 2). The candidate sampler employs a specified hyperparameter space definition as prior, which allows using numerical and categorical parameters with various distributions (e.g., discrete/continuous uniform, Gaussian, log Gaussian, exponential). The sampled hyperparameters are applied to the segmentation pipeline by a worker (compute node) and a performance score with respect to manually annotated ground truth is calculated. In our experiments, we use the Dice coefficient as performance measure. For each hyperparameter evaluation, a dedicated workspace is created and managed by the framework. The optimization can be performed by highly distributed computation. A database is used for distributing compute jobs including the hyperparameters as well as the compute pipeline, and collecting respective results. For each available compute cluster, a coordinator node manages the instantiation

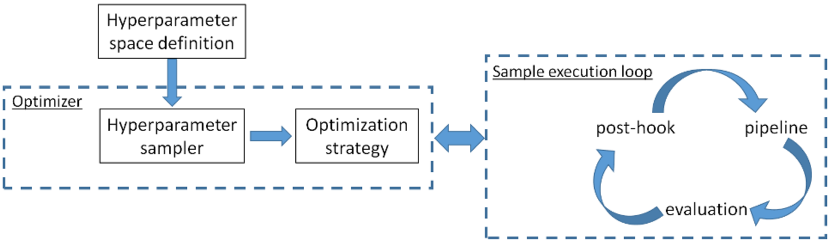
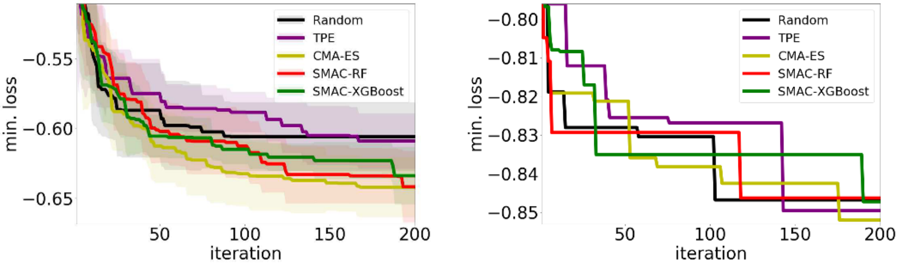


Fig. 2. Schematic representation of the black-box optimization framework.

of workers within the respective cluster. Since the used pipelines contain non-ordinal parameters, we decided to choose optimizers which can handle variables without a natural order. Besides random search (Random) we used sequential model-based algorithm configuration (SMAC) [10], which combines a random forest regression model and sampling from the prior, and represents a more sophisticated version of the general sequential model-based optimization (SMBO) framework [10]. We also modified SMAC by using the XGBoost [11] regression model (SMAC-XGBoost) instead of random forest (SMAC-RF), since XGBoost is currently one of the most popular decision tree based models. Alternatively, we use an evolutionary optimizer with a covariance matrix adaptation evolution strategy (CMA-ES), which is a generic population-based meta-heuristic based optimizer, where feature sets are assumed as "genomes", which undergo evolutionary processes like selection, recombination or mutation [12]. We further investigated the tree of parzen estimator (TPE) surrogate, which performs a nonparametric density approximation of a random variable [13].

3 Experimental results

We applied our hyperparameter optimization framework using multiple optimizers to two pipelines for cell segmentation in challenging prostate cancer tissue images (Fig. 1). The tissue microarray (TMA) images of varying sizes were divided into 256 × 256 pixel image patches before randomly splitting the dataset into 75 % for training and 25 % for testing. We used 60 ground truth images which were manually annotated by an expert. The clustering based pipeline includes six parameters, whereas the CNN based pipeline involves three parameters. As global optimum we used the result from extensive Grid Search. For each optimizer, 200 evaluations were performed on 20 compute nodes (clustering: 27280, CNN: 1108 parameter settings). The pipelines are deterministic, since we used a fix seed value. However, the hyperparameter optimization itself is stochastic. Therefore, we performed 10 runs per optimizer and report mean and standard deviation of the results for the clustering based pipeline (Tab. 1).



(a) Clustering pipeline (b) CNN pipeline

Fig. 3. Comparison of the loss for different optimizers as a function of the number of training iterations. The clustering pipeline is averaged over ten runs (standard deviation highlighted).

Table 1. Results for different optimizers. Shown is the improvement Δ Dice after the warm-up phase and the absolute Dice value. Best results are highlighted in bold.

Pipeline	Optimizer	Δ Dice (Improvement)	Dice
Clustering	Random	0.030	0.606 ± 0.025
	TPE	0.045	0.609 ± **0.020**
	CMA-ES	0.077	**0.642** ± 0.021
	SMAC-RF	**0.094**	**0.642** ± 0.026
	SMAC-XGBoost	0.064	0.634 ± 0.021
	Grid Search	–	*0.654*
CNN	Random	0.019	0.847
	TPE	0.038	0.850
	CMA-ES	0.033	**0.852**
	SMAC-RF	0.017	0.846
	SMAC-XGBoost	**0.039**	0.847
	Grid Search	–	*0.864*

Due to computational resources needed for training deep neural networks, we ran the CNN based pipeline once per optimizer. In addition to Dice, we report the difference (Δ Dice) to the Dice value after the warm-up phase. The framework performs a warm-up phase for exploring the parameter space by evaluating 20 random samples before performing optimization. Thus, Δ Dice reflects the improvement achieved by the optimizer. For the clustering based pipeline, it turns out that SMAC-RF performs best regarding Δ Dice (Fig. 3), whereas for the CNN pipeline our proposed SMAC-XGBoost achieves the best value for Δ Dice (Fig. 4). Considering the absolute Dice value, CMA-ES and SMAC-RF perform best, deviating only 0.012 from the global minimum, whereas TPE yields the lowest standard deviation. SMAC-XGBoost achieves a slightly lower Dice value than the best performing SMAC-RF. However, SMAC-XGBoost outperforms SMAC-RF at the beginning of the training. For the CNN based pipeline, the CMA-ES achieves the best absolute Dice value. Overall, the CNN based pipeline significantly outperforms the clustering based pipeline.

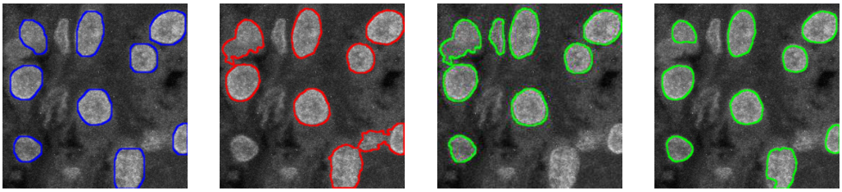

(a) Ground truth (b) Clustering (c) Clustering (d) CNN

Fig. 4. Example image with ground truth (blue) and segmentation using SMAC-RF (red) and SMAC-XGBoost (green).

4 Conclusion

We presented a novel framework for hyperparameter optimization of nuclei segmentation pipelines. The framework allows implementing common optimizers as well as designing novel optimizers. From our study using two pipelines for segmenting cell nuclei in prostate tissue images, it turned out that CMA-ES and SMAC derivatives perform best.

Acknowledgement. Support of the BMBF within the projects CancerTelSys (e:Med, #01ZX1602) and de.NBI (HD-HuB, #031A537C) is gratefully acknowledged.

References

1. Wang Y, Du S, Balakrishnan S, et al. Stochastic Zeroth-order Optimization in High Dimensions. arXiv:1710.10551; 2017.
2. Ramani S, Blu T, Unser M. Monte-carlo SURE: a black-box optimization of regularization parameters for general denoising algorithms. IEEE Trans Image Process. 2008;17(9):1540–1554.
3. Hansen N, Auger A, Ros R, et al.; ACM. Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. Proc GECCO. 2010; p. 1689–1696.
4. Teodoro G, Kurç TM, Taveira LF, et al. Algorithm sensitivity analysis and parameter tuning for tissue image segmentation pipelines. Bioinformatics. 2016;33(7):1064–1072.
5. Snoek J, Larochelle H, Adams RP. Practical bayesian optimization of machine learning algorithms. Proc Adv Neural Inf Process Syst. 2012; p. 2951–2959.
6. Bergstra J, Yamins D, Cox DD; Citeseer. Hyperopt: a python library for optimizing the hyperparameters of machine learning algorithms. Proc SciPy. 2013; p. 13–20.
7. Komer B, Bergstra J, Eliasmith C. Hyperopt-sklearn: automatic hyperparameter configuration for scikit-learn. Proc ICML Workshop AutoML. 2014; p. 2825–2830.
8. Golovin D, Solnik B, Moitra S, et al.; ACM. Google vizier: a service for black-box optimization. Proc SIGKDD. 2017; p. 1487–1495.
9. Ronneberger O, Fischer P, Brox T; Springer. U-net: convolutional networks for biomedical image segmentation. Proc MICCAI. 2015; p. 234–241.
10. Hutter F, Hoos HH, Leyton-Brown K; Springer. Sequential model-based optimization for general algorithm configuration. Proc LION. 2011; p. 507–523.
11. Chen T, Guestrin C; ACM. Xgboost: a scalable tree boosting system. Proc SIGKDD. 2016; p. 785–794.
12. Goldberg DE. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley; 1989.
13. Parzen E. On estimation of a probability density function and mode. Ann Math Stat. 1962;33(3):1065–1076.