Supplementary material for: Initializing Bayesian Hyperparameter Optimization via Meta-Learning

Matthias Feurer and **Jost Tobias Springenberg** and **Frank Hutter**
{feurerm,springj,fh}@cs.uni-freiburg.de
Computer Science Department, University of Freiburg
Georges-Köhler-Allee 52
79110 Freiburg, Germany

Implemented Metafeatures

To evaluate our approach in a realistic setting we implemented 46 metafeatures from the literature listed in Table 1.[1] These metafeatures are computed only for the training set. While most of them can be computed for a whole dataset, some of them (e.g., skewness) are defined for each attribute of a dataset. In this case, we compute the metafeature for each attribute of the dataset and use the mean, standard deviation, minimum and maximum of the resulting vector as proposed in Reif, Shafait, and Dengel (2012b).

Since previous empirical results suggested that landmarking metafeatures are superior to other metafeatures (Pfahringer, Bensusan, and Giraud-Carrier 2000; Reif, Shafait, and Dengel 2011; 2012a), we experimented with using only the landmarking features used in the first experiment of Pfahringer, Bensusan, and Giraud-Carrier (2000). We also experimented with the subsets of metafeatures used in previous works on collaborative SMBO (Bardenet et al. 2013; Yogatama and Mann 2014). The exact subset are:

- *Pfahringer*: number of features, number of numeric features, number of categorical features, number of classes, class probability max, landmark lda, landmark naive bayes, landmark decision tree
- *Bardenet(Experiment 1)*: number of classes, log number of features, log inverse dataset ratio, pca 95percent
- *Bardenet(Experiment 2)*: number of classes, log number of features, log inverse dataset ratio, pca kurtosis first pc, pca skewness first pc
- *Yogatama*: log number of features, log number of instances, number of classes

Datasets and Preprocessing

The 57 datasets from the OpenML project website (Vanschoren et al. 2013) that we used are listed in Figure 1.

[1] These are the same metafeatures as listed in Table 1 in the main paper. In contrast to the main paper, Table 1 contains additional information on the metafeatures values.

Additional Experimental Results

In this section we provide additional results for the SVM benchmark.

Warmstarting Spearmint for Optimizing SVMs

For space constraints, only part of this experiment was shown in the main paper. Here, we give the full results.

Figure 2 (top) shows how Spearmint (Snoek, Larochelle, and Adams 2012) compares to the other state-of-the-art SMBO frameworks for optimizing the hyperparameters of a SVM. The plot on left-hand side shows that Spearmint captured the relationship between hyperparameters and response values in its model, converging to the optimum very fast. In contrast, the plots in the middle and on the right-hand side show cases where Spearmint performs worse than every other model (which actually happened only on 3 out 57 datasets).

Figure 2 (bottom) shows how meta-learning can improve the vanilla Spearmint method. For the dataset on the left, both version of meta-learning work well and slightly improve Spearmint. The middle dataset shows how meta-learning can help Spearmint on datasets where it performs badly, advancing it from being worse to being best. In contrast, for the dataset on the right, meta-learning only yielded small improvements (a comparison to the right top plot in Figure 2 shows that neither variant of Spearmint performed better than random search in this case).

To complete the above analysis, Figure 3 (top) quantifies on how many datasets MI-Spearmint based on the learned distance performed statistically significant better than its competitors according to the two-sided t-test. The lower plot of Figure 3 shows the statistically significant losses. Both of these quantities are plotted over time, as the function evaluation budget increases. We observe that MI-Spearmint started off much better than all other methods. Given larger function evaluation budgets, using its Spearmint part, it even increased the performance advantage over random search, TPE, and SMAC. Compared to Spearmint, MI-Spearmint started off significantly better in 70% of the datasets, but these differences leveled off over time. There was very little difference between the two MI-Spearmint variants (based on the L_1 and the learned distance). We can conclude that given a large enough budget vanilla Spearmint already yields good hyperparameter configurations, making it hard to improve

Table 1: List of implemented metafeatures

Metafeature	Value			Calculation time (s)		
	Minimum	**Mean**	**Maximum**	**Minimum**	**Mean**	**Maximum**
class-entropy	0.64	1.92	4.70	0.00	0.00	0.00
class-probability-max	0.04	0.43	0.90	0.00	0.00	0.00
class-probability-mean	0.04	0.28	0.50	0.00	0.00	0.00
class-probability-min	0.00	0.19	0.48	0.00	0.00	0.00
class-probability-std	0.00	0.10	0.35	0.00	0.00	0.00
dataset-ratio	0.00	0.06	0.62	0.00	0.00	0.00
inverse-dataset-ratio	1.62	141.90	1620.00	0.00	0.00	0.00
kurtosis-max	-1.30	193.43	4812.49	0.00	0.01	0.05
kurtosis-mean	-1.30	24.32	652.23	0.00	0.01	0.05
kurtosis-min	-3.00	-0.59	5.25	0.00	0.01	0.05
kurtosis-std	0.00	48.83	1402.86	0.00	0.01	0.05
landmark-1NN	0.20	0.79	1.00	0.01	0.61	8.97
landmark-decision-node-learner	0.07	0.55	0.96	0.00	0.13	1.34
landmark-decision-tree	0.20	0.78	1.00	0.00	0.49	5.23
landmark-lda	0.26	0.79	1.00	0.00	1.39	70.08
landmark-naive-bayes	0.10	0.68	0.97	0.00	0.06	1.05
landmark-random-node-learner	0.07	0.47	0.91	0.00	0.02	0.26
log-dataset-ratio	-7.39	-3.80	-0.48	0.00	0.00	0.00
log-inverse-dataset-ratio	0.48	3.80	7.39	0.00	0.00	0.00
log-number-of-features	1.10	2.92	5.63	0.00	0.00	0.00
log-number-of-instances	4.04	6.72	9.90	0.00	0.00	0.00
number-of-Instances-with-missing-values	0.00	96.00	2480.00	0.00	0.00	0.01
number-of-categorical-features	0.00	13.25	240.00	0.00	0.00	0.00
number-of-classes	2.00	6.58	28.00	0.00	0.00	0.00
number-of-features	3.00	33.91	279.00	0.00	0.00	0.00
number-of-features-with-missing-values	0.00	3.54	34.00	0.00	0.00	0.00
number-of-instances	57.00	2126.33	20000.00	0.00	0.00	0.00
number-of-missing-values	0.00	549.49	22175.00	0.00	0.00	0.00
number-of-numeric-features	0.00	20.67	216.00	0.00	0.00	0.00
pca-95percent	0.02	0.52	1.00	0.00	0.00	0.00
pca-kurtosis-first-pc	-2.00	13.38	730.92	0.00	0.00	0.01
pca-skewness-first-pc	-27.07	-0.16	6.46	0.00	0.00	0.04
percentage-of-Instances-with-missing-values	0.00	0.14	1.00	0.00	0.00	0.00
percentage-of-features-with-missing-values	0.00	0.16	1.00	0.00	0.00	0.00
percentage-of-missing-values	0.00	0.03	0.65	0.00	0.00	0.00
ratio-categorical-to-numerical	0.00	1.35	33.00	0.00	0.00	0.00
ratio-numerical-to-categorical	0.00	0.49	7.00	0.00	0.00	0.00
skewness-max	0.00	5.34	67.41	0.00	0.00	0.04
skewness-mean	-0.56	1.27	14.71	0.00	0.00	0.04
skewness-min	-21.19	-0.62	1.59	0.00	0.00	0.04
skewness-std	0.00	1.60	18.89	0.00	0.01	0.05
symbols-max	0.00	13.09	429.00	0.00	0.00	0.00
symbols-mean	0.00	3.01	41.38	0.00	0.00	0.00
symbols-min	0.00	1.44	12.00	0.00	0.00	0.00
symbols-std	0.00	3.06	107.21	0.00	0.00	0.00
symbols-sum	0.00	71.04	1648.00	0.00	0.00	0.00

Figure 1: List of the 57 datasets used for the experiments from the OpenML project website (Vanschoren et al. 2013).

Dataset name	# Features	# Patterns	# Classes
abalone	8	4177	28
anneal.ORIG	38	898	5
arrhythmia	279	452	13
audiology	69	226	24
autos	25	205	6
balance-scale	4	625	3
braziltourism	8	412	7
breast-cancer	9	286	2
breast-w	9	699	2
car	6	1728	4
cmc	9	1473	3
credit-a	15	690	2
credit-g	20	1000	2
cylinder-bands	39	540	2
dermatology	34	366	6
diabetes	8	768	2
ecoli	7	336	8
eucalyptus	19	736	5
glass	9	214	6
haberman	3	306	2
heart-c	13	303	2
heart-h	13	294	2
heart-statlog	13	270	2
hepatitis	19	155	2
ionosphere	34	351	2
iris	4	150	3
kr-vs-kp	36	3196	2
labor	16	57	2
letter	16	20000	26
liver-disorders	6	345	2
lymph	18	148	4
mfeat-factors	216	2000	10
mfeat-fourier	76	2000	10
mfeat-karhunen	64	2000	10
mfeat-morphological	6	2000	10
mfeat-pixel	240	2000	10
mfeat-zernike	47	2000	10
mushroom	22	8124	2
nursery	8	12960	5
optdigits	64	5620	10
page-blocks	10	5473	5
pendigits	16	10992	10
postoperative-patient-data	8	90	3
primary-tumor	17	339	21
satimage	36	6430	6
segment	19	2310	7
sonar	60	208	2
soybean	35	683	19
spambase	57	4601	2
tae	5	151	3
tic-tac-toe	9	958	2
vehicle	18	846	4
vote	16	435	2
vowel	13	990	11
waveform-5000	40	5000	3
yeast	8	1484	10
zoo	17	101	7
Minimum	3.0	57.0	2.0
Maximum	279.0	20000.0	28.0
Mean	33.9	2126.3	6.6
10% quantile	6.0	153.4	2.0
90% quantile	64.0	5531.8	11.8
median	16.0	699.0	4.0

further. However, as already mentioned in the main paper, meta-learning initialization does substantially improve performance with few function evaluations.

References

Bardenet, R.; Brendel, M.; Kégl, B.; and Sebag, M. 2013. Collaborative hyperparameter tuning. In *Proc. of ICML*, 199–207.

Pfahringer, B.; Bensusan, H.; and Giraud-Carrier, C. 2000. Meta-learning by landmarking various learning algorithms. In *Proc. of ICML*, 743–750.

Reif, M.; Shafait, F.; and Dengel, A. 2011. Prediction of classifier training time including parameter optimization. In *KI 2011: Advances in Artificial Intelligence*.

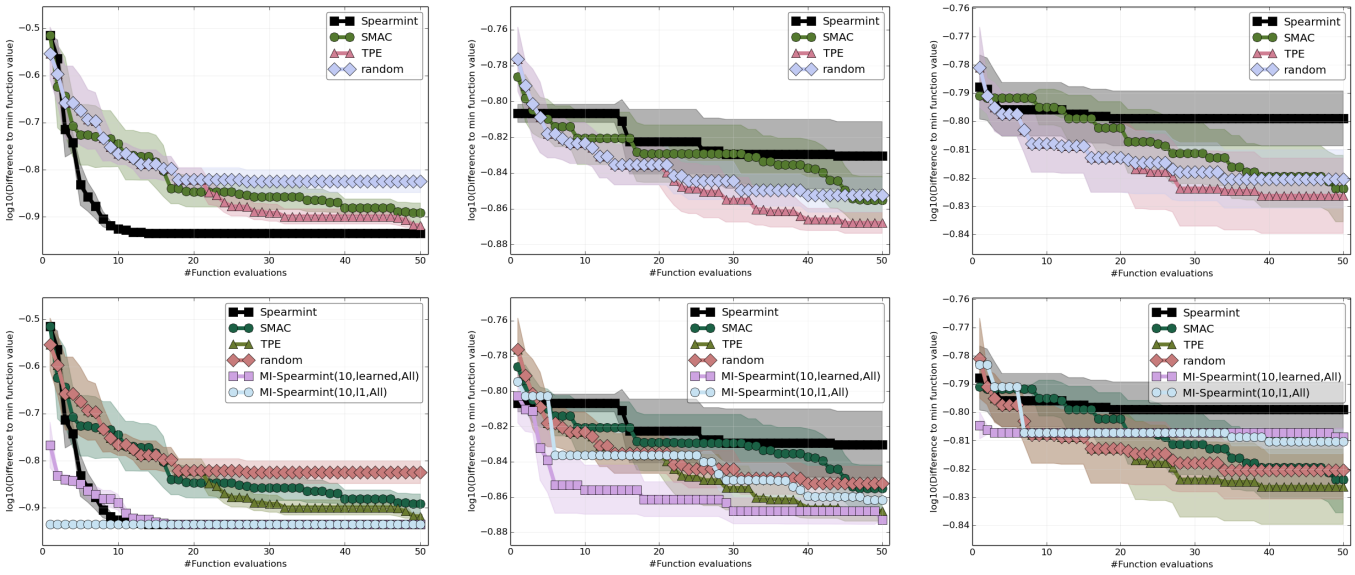
Reif, M.; Shafait, F.; and Dengel, A. 2012a. Meta-learning for evolutionary parameter optimization of classifiers. *Machine Learning* 87:357–380.

Reif, M.; Shafait, F.; and Dengel, A. 2012b. Meta2-features: Providing meta-learners more information. Poster and Demo Track of the 35th German Conference on AI.

Snoek, J.; Larochelle, H.; and Adams, R. 2012. Practical bayesian optimization of machine learning algorithms. In *Proc. of NIPS*, 2951–2959.

Vanschoren, J.; van Rijn, J. N.; Bischl, B.; and Torgo, L. 2013. OpenML: Networked science in machine learning. *SIGKDD Explorations* 15(2):49–60.

Yogatama, D., and Mann, G. 2014. Efficient transfer learning method for automatic hyperparameter tuning. In *Proc. of AISTATS*, 1077–1085.



dataset: heart-c dataset: haberman dataset: breast-cancer

Figure 2: Difference in SVM validation error between the best found hyperparameters at evaluation t and the best value obtained via a full grid search on three datasets. MI-Spearmint$(10, d, X)$ stands for MI-Spearmint with an initial design of $t = 10$ configurations suggested by meta-learning using metafeatures X with distance function d.

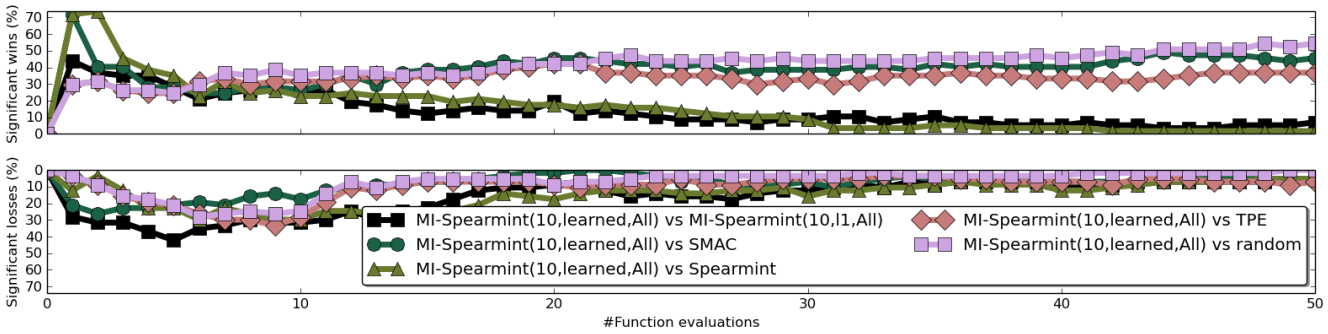


Figure 3: Percentage of wins of MI-Spearmint with an initial design of $t = 10$ configurations suggested by meta-learning using the learned distance with all metafeatures. The upper plot shows significant wins of MI-Spearmint against each other approach according to the two-sided t-test while the lower plot shows the statistically significant losses.