Effects of Random Sampling on SVM Hyper-parameter Tuning

Tomáš Horváth[1⋆], Rafael G. Mantovani[2], and André C. P. L. F. de Carvalho[2]

[1] Faculty of Informatics, Eötvös Loránd University
Budapest, Hungary
tomas.horvath@inf.elte.hu
[2] Institute of Mathematical and Computer Sciences, University of São Paulo
São Carlos, Brazil
{rgmantov,andre}@icmc.usp.br

Abstract. Hyper-parameter tuning in use is one of the crucial steps in the application of machine learning algorithms. In general, the tuning process is modeled as an optimization problem for which several methods have been proposed. For complex algorithms, the evaluation of a hyper-parameter configuration is expensive and their runtime is sped up through data sampling. In this paper, the effect of sample sizes to the results of hyper-parameter tuning process is investigated. Hyper-parameters of Support Vector Machines are tuned on samples of different sizes generated from a dataset. Hausdorff distance is proposed for computing the differences between the results of hyper-parameter tuning on two samples of different size. 100 real-world datasets and two tuning methods (Random Search and Particle Swarm Optimization) are used in the experiments revealing some interesting relations between sample sizes and results of hyper-parameter tuning which open some promising directions for future investigation in this direction.

Keywords: Sampling, Hyper-parameter tuning, Suport Vector Machines

1 Introduction

Hyper-parameter (HP) tuning of Machine Learning (ML) algorithms is a challenging task because of some practical difficulties: First, good hyper-parameter (HP) settings depend on the dataset used, so the HP tuning should take into account the algorithm-dataset combination. Second, the individual HP values in good HP settings are often related to each other, thus, independent tuning of individual HP values should be avoided. Finally, evaluating the fitness of a specific HP setting can be computationally expensive, an issue on which this paper is focused on.

The choice of the most adequate technique for HP tuning approach depends on the complexity of the ML algorithm used w.r.t. the given data, i.e. if the

⋆ Tomáš Horváth is also a member of the Institute of Computer Science, Faculty of Science, Pavol Jozef Šafárik University in Košice, Slovakia.

evaluation of the HP setting for a ML algorithm has a low or high computational cost. In other words, if many different HP settings can be easily evaluated or HP have to be tuned from a small number of evaluations.

An alternative to speed up the runtime of ML algorithms is sampling [5], when a reduced but representative sample of the data is used to induce a model, instead of using the whole dataset. In addition to random sampling [13], there are more sophisticated (stratified) sampling methods [9, 19] which can be used.

The motivation behind the research presented in this paper lies in the following question: In what extent does sampling of a dataset affect the results of HP tuning? A preliminary research, an experiment, was conducted in order to investigate the previous question. Experiments were conducted on 100 real-world datasets and two tuning techniques for HP tuning pf Support Vector Machines (SVM). For comparing the results of HP tuning on different sample sizes, the use of the Hausdorff distance was proposed. Experiments reveal interesting relations which serve as the basis for the further research in this direction.

2 Hyper-parameter Tuning

HP tuning is, in general, treated as an optimization problem whose objective function $f : \mathcal{A} \times \mathcal{D} \times \mathcal{H} \rightarrow \mathbb{R}$ captures the predictive performance of an algorithm $a \in \mathcal{A}$ with the HP setting $\mathbf{h} = (h_1, h_2, \ldots, h_k) \in \mathcal{H}$ on the dataset $\mathbf{D} \in \mathcal{D}$ where $\mathcal{A}$ is the set of ML algorithms, $\mathcal{D}$ is the set of datasets and $\mathcal{H} = \mathcal{H}_1 \times \mathcal{H}_2 \times \cdots \times \mathcal{H}_k$ is the space of admissible values of HP for the algorithm a. The task of HP tuning is, given a, $\mathcal{H}$ and $\mathbf{D}$, to find $\mathbf{h}^\star \in \mathcal{H}$ such that

$$\mathbf{h}^\star = \underset{\mathbf{h} \in \mathcal{H}}{arg\,max}\, f(a, \mathbf{D}, \mathbf{h}) \tag{1}$$

Several techniques for algorithm HP tuning have been proposed in the literature: The simplest techniques are the widely used Grid Search (GS) [3] in case of low-dimensional $\mathcal{H}$ and Random Search (RS) [10] for higher dimensional $\mathcal{H}$. Local and pattern search techniques [14] extend the palette, however, these techniques tend to get stuck in local minima. Another large family of HP tuning techniques includes Sequential Model-Based Optimization (SMBO) techniques [8]. Nature inspired techniques, like Genetic algorithms (GAs) [6] and Particle Swarm Optimization (PSO) [12], have also been largely utilized for HP tuning. Recent works focus on Bayesian techniques [16] and meta-learning [17].

SVM have been shown to be very efficient when used for classification tasks. Consequently, several authors have proposed solutions for SVM HP tuning. All previously mentioned techniques have been applied to SVM HP tuning , initialization of optimization methods with promising HP configuration values [12] and recommendation of when optimization techniques should be used [11], have been investigated.

3 Experiment Settings

Classification experiments were carried out using SVM with the Radial Basis Function (RBF) Kernel [7]. SVM with RBF kernel has two HP to tune, cost C of SVM and width γ of the kernel. Given a dataset (or its sample), these HPs are tuned using the average per-class accuracy measured over the folds of a 10-fold cross-validation re-sampling strategy.

3.1 Experimental Methodology

The methodology adopted in the experiments, illustrated in Alg. 1, can be briefly described as follows: a random sample of instances is extracted from each dataset. For this sample, the best HP setting is found using a given HP tuning technique. Iteratively, the sample is extended with instances randomly chosen from the so far not sampled instances of the dataset. Iteration continues while the extended sample does not contain all the instances from the dataset. Since the HP tuning techniques used are stochastic, the whole process is repeated 30 times for each dataset. The resulting set of the best found HP settings is further analyzed.

Algorithm 1 Experiment methodology

Require: $K, \mathcal{D} = \{\mathbf{D}_1, \ldots, \mathbf{D}_N\}, \{p_1, \ldots, p_n\}, f, a, \mathcal{H}, hpt$
1: **for** $k = 1 \rightarrow K$ **do**
2: **for** $i = 1 \rightarrow N$ **do**
3: $\mathbf{D}_i^k \leftarrow \emptyset$
4: $\mathcal{H}_i^{k\star} \leftarrow \emptyset$
5: **for** $j = 1 \rightarrow n$ **do**
6: $\mathbf{S} \leftarrow$ sample p_j% from $\mathbf{D}_i$ $(\mathbf{D}_i^k \cap \mathbf{S} = \emptyset)$
7: $\mathbf{D}_i^k \leftarrow \mathbf{D}_i^k \cup \mathbf{S}$
8: $\mathbf{h}_{i_j}^{k\star} \leftarrow \underset{\mathbf{h} \in \mathcal{H}}{arg\,max}\, f(a, \mathbf{D}_i^k, \mathbf{h})$ ▷ using hpt
9: $\mathcal{H}_i^{k\star} \leftarrow \mathcal{H}_i^{k\star} \cup \{\mathbf{h}_{i_j}^{k\star}\}$
10: **return** $\mathcal{H}_i^{k\star}$

3.2 Parameters used in the experiments

The parameters of the experiment methodology, described in Alg. 1, are the following:

1. *Datasets*: 100 (multi-class) classification datasets from the UCI Machine Learning Repository[3]. Their main characteristics, such as number of instances (Inst.), number of attributes (Attr.) and number of classes (Class.), are summarized in Tab. 1. Thus, $\mathcal{D} = \{\mathbf{D}_1, \ldots, \mathbf{D}_{100}\}$ and N=100.

[3] http://archive.ics.uci.edu/ml/

2. *Sample sizes*: The parameter n of Alg. 1 is set to 11, such that $\{p_1, \ldots, p_{11}\}$ is equal to $\{50, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5\}$. Thus, the initial sample size is 50% of the size of the dataset and at each iteration 5% of all the remaining instances in the dataset are added to the sample. Sample sizes are, thus, 50%, 55%, ..., 95%, 100%, respectively, of the original dataset. A simple random sampling mechanism is utilized in experiments.
3. *SVM HP-space*: As previously mentioned, the chosen algorithm a is SVM with RBF kernel. The fitness function f is a simple classification predictive accuracy defined as the ratio of correctly classified instances to all instances averaged over 10-folds of cross-validation. The HP-space is set to $\mathcal{H} = \mathcal{H}_{cost} \times \mathcal{H}_{\gamma} = [2^{-2}, 2^{15}] \times [2^{-15}, 2^{3}] \subset \mathbb{R}^2$.
4. *HP-Tuning methods*: Two HP tuning techniques (the parameter *hpt* of Alg. 1) are used in the experiments:
   - Exhaustive RS of the HP space $\mathcal{H}$, suggested in [2] as a good alternative for HP tuning with number of trials set to 2500.
   - PSO[4] [20]. The number of maximum evaluations was set to 2500. It corresponds to a maximum of 100 iterations with a population composed by 25 particles. The default HP values recommended by LibSVM [4] were added to the initial population. The other PSO parameters are those recommended by the package used for the experiments (see next).

   Since both of techniques are stochastic, each HP tuning process was run 30 times, i.e. the parameter K=30.

All techniques were implemented using the `R` framework[5]. The `e1071` package, which has an interface to the LibSVM[6] library [4], was used for the SVM implementation. For PSO, the `pso` package [1] was used. RS was implemented by the authors.

4 Results and Discussion

Figure 1 shows the results for the *'analcatdata_germangss'* and *'artificial-characters'* datasets, in the first and second rows, respectively. In the columns from left to right, the 30 best[7] HP settings found for samples with 50%, 60%, 70%, 80%, 90% and 100%, respectively, of the original data, are presented.

The first finding is that, not only for the two illustrated dataset, but in general, the results from HP tuning using RS are very similar to those using PSO. This can be due to the relatively large number of evaluations in the experiment, which can allow RS to cover a large portion of the HP space $\mathcal{H}$, enough to find good HP values.

[4] Particle swarm optimization has been successfully used in partially irregular or noisy optimization problems, and, often performs well, finding good solutions because it does not make any assumption about the search landscape.

[5] `https://www.r-project.org/`

[6] https://www.csie.ntu.edu.tw/ cjlin/libsvm/

[7] According to the parameter K of the Alg. 1 set to 30 in the experiment.

Table 1. Classification datasets used in the experiments. Datasets with the maximum (analcatdata_germangss) and minimum (artificial-characters) average distances (d^H_{mean}) are highlighted.

Name	d^H_{RS}	d^H_{PSO}	d^H_{mean}	Name	d^H_{RS}	d^H_{PSO}	d^H_{mean}
abalone-11class	**3.53**	4.09	3.81	ozone-eighthr	6.79	**6.40**	6.60
abalone-28class	**3.30**	3.34	3.32	ozone-onehr	**3.48**	4.84	4.16
abalone-3class	3.20	**2.70**	2.95	page-blocks	**2.40**	3.31	2.86
abalone-7class	**3.43**	4.03	3.73	parkinsons	4.87	**4.31**	4.59
acute-infl.-nephr.	3.90	**2.87**	3.38	pima-ind.-diab.	3.62	**3.09**	3.36
analcat._author.	**3.72**	6.96	5.34	planning-relax	**6.20**	6.86	6.53
analcat._boxing2	4.00	**3.60**	3.80	plant-sp.-leav.-marg.	7.68	**6.95**	7.31
analcat._credit.	**3.38**	3.52	3.45	plant-sp.-leav.-shape	4.40	**2.89**	3.65
analcat._dmft	**4.05**	4.08	4.06	plant-sp.-leav.-text.	**2.44**	3.47	2.96
analcat._germ,	**12.49**	13.24	**12.87***	prnn_crabs	**2.43**	3.36	2.90
analcat._lawsuit	**4.02**	7.11	5.56	qsar-biodegr.	**5.45**	7.38	6.42
appendicitis	**5.55**	5.68	5.62	qualit.-bankruptcy	**3.50**	4.60	4.05
artif.-charact.	**1.17**	1.23	**1.2****	ringnorm	**5.81**	8.15	6.98
autoU.-au1-1000	9.78	**5.49**	7.63	robot-failure-lp4	**3.84**	6.80	5.32
autoU.-au4-2500	3.25	**2.93**	3.09	robot-failure-lp5	**2.67**	2.80	2.73
autoU.-au6-1000	5.64	**5.56**	5.60	saheart	2.93	**2.57**	2.75
autoU.-au6-250-dr.	**3.57**	5.92	4.75	seeds	6.52	**5.92**	6.22
autoU.-au6-cd1-400	**1.95**	3.08	2.52	seismic-bumps	10.34	**10.24**	10.29
banknote-auth.	**2.84**	3.80	3.32	spambase	**2.08**	2.31	2.20
breast-canc.-wisc.	**4.13**	4.22	4.17	spectf-heart	**4.90**	4.99	4.95
breast-tiss.-4class	**4.22**	4.99	4.61	spect-heart	4.53	**4.00**	4.27
breast-tiss.-6class	7.24	**6.31**	6.78	statlog-austr.-cr.	**5.06**	5.87	5.46
bupa	2.73	**2.69**	2.71	statlog-ger.-cr.	3.72	**2.74**	3.23
car-evaluation	**1.79**	2.02	1.9	statlog-ger.-cr.-num.	2.53	**2.33**	2.43
climate-sim.-crach.	6.84	**4.13**	5.48	statlog-heart	3.85	**3.58**	3.71
cloud	6.25	**5.86**	6.05	statlog-im.-segm.	**2.10**	2.34	2.22
cmc	2.61	**2.33**	2.47	statlog-land.-sat.	1.60	**1.23**	1.42
conn.-mines-vs-rocks	**2.87**	3.50	3.19	statlog-veh.-silh.	**1.94**	2.00	1.97
conn.-vowel	**2.17**	2.20	2.19	teaching-assist.-eval.	**3.05**	3.09	3.07
conn.-vowel-reduced	**2.28**	2.39	2.34	thoracic-surgery	**6.11**	12.21	9.16
contrac.-meth.-choice	2.17	**2.00**	2.08	thyroid-allhyper	4.62	**2.88**	3.75
dermatology	4.51	**4.13**	4.32	thyroid-allrep	**2.24**	2.34	2.29
ecoli	3.52	**3.27**	3.4	thyroid-ann	**1.79**	1.87	1.83
fertility-diagnosis	**5.74**	6.73	6.24	thyroid-dis	**2.55**	4.01	3.28
glass	**2.69**	3.32	3.00	thyroid-hypothyroid	**2.76**	3.71	3.23
habermans-survival	**5.14**	8.02	6.58	thyroid-newthyroid	**4.64**	5.18	4.91
hayes-roth	2.79	**2.27**	2.53	thyroid-sick	3.02	**2.70**	2.86
hepatitis	**4.15**	5.05	4.6	thyroid-sick-euthyr.	**6.83**	10.45	8.64
horse-colic-surgical	10.57	**9.31**	9.94	tic-tac-toe	6.93	**5.3**	6.12
indian-liver-patient	**5.85**	7.84	6.84	user-knowledge	2.47	**2.43**	2.45
ionosphere	8.49	**6.78**	7.63	voting	**3.74**	4.38	4.06
iris	**3.30**	4.08	3.69	wdbc	**3.08**	3.62	3.35
kr-vs-kp	**2.11**	4.69	3.4	wholesale-channel	**3.44**	3.78	3.61
leaf	3.22	**2.66**	2.94	wholesale-region	**3.55**	4.72	4.13
led7digit	4.60	**4.00**	4.30	wilt	**2.74**	2.90	2.82
leukemia-haslinger	4.40	**3.62**	4.01	wine	**5.58**	7.72	6.65
mammographic-mass	**4.67**	5.44	5.05	wine-quality-red	**11.49**	12.74	12.12
mfeat-fourier	**4.30**	4.63	4.46	wpbc	**4.34**	7.61	5.97
movement-libras	**2.96**	3.86	3.41	yeast	**2.79**	4.14	3.46
optdigits	**4.54**	6.04	5.29	yeast-4class	3.12	**2.78**	2.95

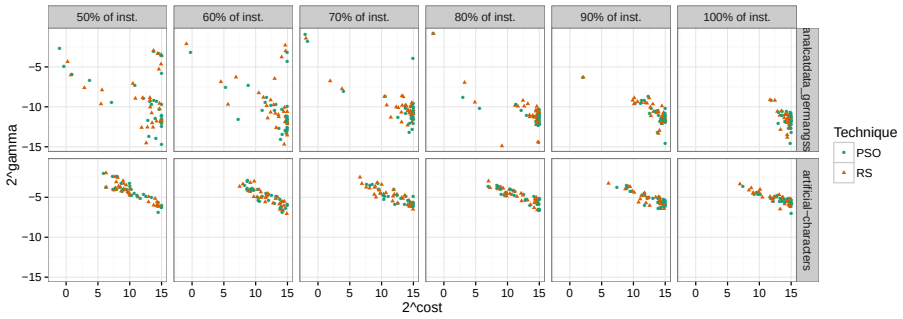

Fig. 1. The 30 best HP configurations found by RS (orange triangles) and PSO (green squares) for samples with 50%, 60%, 70%, 80%, 90% and 100% (1st, 2nd, 3rd, 4th, 5th and 6th columns, respectively) of the analcatdata_germangss (first row) and artificial-characters (bottom row) datasets.

However, Fig. 1 shows two different situations: For the artificial-characters dataset, the HP tuning results for different sample sizes are very similar, regardless whether RS or PSO was used (the six plots in the second row of charts). On the other hand, for the analcatdata_germangss data, the best HP configurations found differ more for distinct sample sizes. Besides, the 30 HP settings found are less dispersed across $\mathcal{H}$ for larger sample sizes. It means that tuning the C and γ HPs is less sensitive to sampling in the artificial-characters dataset than in the analcatdata_germangss dataset.

4.1 Hausdorff Distance

One way to measure the difference in the HP tuning results for various sample sizes is to use the Hausdorff distance [18], a commonly used dissimilarity measure between two sets $\mathcal{A}$ and $\mathcal{B}$ of points, defined as

$$d^H(\mathcal{A},\mathcal{B}) = max\{d(\mathcal{A},\mathcal{B}), d(\mathcal{B},\mathcal{A})\} \tag{2}$$

where

$$d(\mathcal{A},\mathcal{B}) = \underset{\mathbf{a}\in\mathcal{A}}{max}\{\underset{\mathbf{b}\in\mathcal{B}}{min}\{||\mathbf{a},\mathbf{b}||\}\} \tag{3}$$

and $||.,.||$ is any norm, e.g. the Euclidean distance. Two sets are close according to the Eq. 2 if every point of either set is close to some point of the other set.

According to the Alg. 1, for a $\mathbf{D}_i$ dataset and a fixed sample of size $p_1+\cdots+p_j$, such that $1 \leq j \leq n$, there are K HP settings $\mathbf{h}_{i_j}^{1\star}, \mathbf{h}_{i_j}^{2\star}, \ldots, \mathbf{h}_{i_j}^{K\star}$. Let $\mathcal{A} = \{\mathbf{h}_{i_j}^{1\star}, \mathbf{h}_{i_j}^{2\star}, \ldots, \mathbf{h}_{i_j}^{K\star}\}$ and $\mathcal{B} = \{\mathbf{h}_{i_l}^{1\star}, \mathbf{h}_{i_l}^{2\star}, \ldots, \mathbf{h}_{i_l}^{K\star}\}$, where $1 \leq j \neq l \leq n$, be two sets of HP settings computed by a HP tuning technique for two samples of different sizes sampled from the same dataset. Pairwise Hausdorff distances between various sets $\mathcal{A}$ and $\mathcal{B}$ can be calculated.

Table 2. Pairwise Hausdorff distances between HP tuning results for samples of different sizes averaged over all the datasets. Results for the RS and PSO HP tuning techniques are shown above and below the diagonal, respectively.

Sample size	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
50%	–	3.53	3.58	3.72	3.78	3.84	4.11	4.02	4.32	4.47	**5.03**
55%	4.01	–	3.68	3.64	3.62	3.83	4.01	4.01	4.27	4.58	**4.95**
60%	4.01	3.78	–	3.57	3.57	3.59	3.83	3.72	4.10	4.25	**4.58**
65%	4.24	3.82	3.78	–	3.52	3.57	3.88	3.75	3.75	4.27	**4.46**
70%	4.53	3.92	3.93	4.18	–	3.46	3.53	3.63	3.82	4.09	**4.53**
75%	4.24	4.14	3.88	4.01	4.00	–	3.66	3.57	3.62	4.01	**4.07**
80%	4.55	4.16	4.03	3.95	4.05	4.00	–	3.48	3.46	3.72	**4.16**
85%	4.54	4.36	4.00	4.01	3.88	3.86	3.72	–	3.49	3.69	**3.80**
90%	4.78	4.53	4.20	4.42	4.27	4.02	3.68	3.76	–	3.31	**3.57**
95%	5.27	4.94	4.60	4.72	4.52	4.24	3.94	3.84	3.58	–	**3.27**
100%	**5.61**	**5.30**	**4.85**	**4.85**	**4.76**	**4.73**	**4.22**	**4.10**	**4.14**	**3.78**	–

4.2 Pairwise Differences

The pairwise distances, averaged over all the datasets used in the experiment, are shown in Tab. 2. Although the results are very similar for the two HP tuning techniques evaluated, the RS results are less "distant" w.r.t. different sample sizes than the PSO results. A possible reason is that PSO is a more sophisticated optimization algorithm than RS and tends to narrow down the search space of candidate solutions. Thus, if $\mathcal{A}$ and $\mathcal{B}$ are less dispersed, then there is a higher chance that the overlap of the areas covered by these two sets is smaller.

4.3 Tuning based on Samples vs. the Whole Dataset

The distances between HP tuning results for samples of different size and the whole dataset, averaged over the 30 runs and all the datasets, are illustrated by bold in the last row and last column of Tab. 2. For both RS and PSO, the differences between the results (HP settings) found for the whole dataset and the results found for samples (abbreviated as average differences below) of size 60%, 65% and 70% are very similar. For RS in particular, average differences in results are similar for the samples of size of 50% and 55% and for samples of size 75% and 80%. Regarding PSO, similar average differences were recorded for sample of sizes 80%, 85% and 90%. Besides that, average differences for the samples of size 75% are similar to those of sizes 60%, 65% and 70%.

These results can support decisions about the size of samples to be used for HP tuning. For example, using a sample size of 60% would probably lead to similar results but in a shorter time as if the a sample size of 75% would be used for tuning the HP with PSO.

Tab. 1 contains, for each dataset, the averaged Hausdorff distances between the HP tuning results (averaged over the 30 runs) using the whole data and samples of different size, for RS (denoted as d^H_{RS}) and PSO (denoted as d^H_{PSO}) as well as their mean (denoted as d^H_{mean}). Datasets with the minimum and the maximum values for d^H_{mean} are highlighted.

4.4 Significant Differences Between Sample Sizes

Tab. 3 contains significant differences between HP tuning results for samples which size differ in 5%, 10%, ..., 45% and 50% of instances. The results were computed as follows: For each dataset, Hausdorff distances between its samples differing in 5%, 10%, ..., 50%, respectively, were averaged (e.g. in case of 10% difference, these are the distances between sample sizes of 50-60%, 60-70%, ..., 90-100%) resulting in ten 100-dimensional vectors. Wilcoxon test was applied to each pair of these vectors.

The results indicate that there are no significant differences in the results of HP tuning when the sample sizes differ in less than 25% or more than 35% of the size of the whole dataset.

Table 3. Statistically significant differences according to Wilcoxon test between the results of HP tuning for samples differing in 5%, ..., 50% of instances w.r.t. the size of the whole dataset. ○ denote p-value ≤ 0.05 and ● denote p-value ≤ 0.01. Results for RS and PSO are above and below the diagonal, respectively.

Difference in sample sizes	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%
5%	–					●	●	●	●	●
10%		–				○	○	●	●	●
15%			–			○		○	●	●
20%				–					○	●
25%					–				○	●
30%	●	○				–				
35%	○	○					–			
40%	●	●	●	○				–		
45%	●	●	●	●	●	○			–	
50%	●	●	●	●	●	○				–

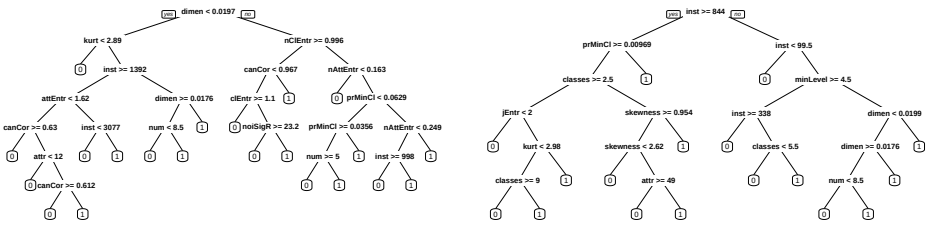

Fig. 2. Decision trees induced from dataset characteristics for a target variable d^H_{RS} (left) and d^H_{PSO} (right) from the Tab. 1. Leaves labeled with 1 mean that $d^H_{RS} > 3$ or $d^H_{PSO} > 3$, respectively, otherwise the leaves are labeled with 0.

4.5 Data Characteristics vs. Sensitivity of HP-Tuning on Sampling

In order to reveal relations between the sensitivity of HP tuning on sampling from a dataset and some characteristics of the dataset, a decision tree was in-

duced using the `rpart` package of `R`. As explanatory measures, some simple [12], statistical [17] and information-theoretic [15] measures were extracted from each dataset. These measures, known as meta-features, are well explored in meta-learning research [11]. A dataset was created where each dataset is represented by predictive attributes, a.k.a. the meta-features, and a target attribute. The target attribute can assume the values from d^H_{RS} or d^H_{PSO}, from Tab. 1, depending if the HP tuning for the dataset used RS or PSO, respectively. The continuous target values (d^H_{RS} or d^H_{PSO}) were transformed to binary values, as follows: values larger than 3 were encoded as 1 (True), otherwise as 0 (False), indicating that HP tuning on the given dataset is sensitive or not, respectively, to sampling. The resulting trees, illustrated in Fig. 2 were not pruned, in order to reveal all meta-features considered relevant to the classification.

For RS, the tree contains the following characteristics: dimensionality of the dataset (dimen), average kurtosis of continuous attributes (kurt), normalized class entropy (nClEnt), number of instances (inst), canonical correlation between attributes and labels (canCor), normalized attribute entropy (nAttEntr), attribute entropy (attEntr), class entropy (clEntr), probability of the minority class (prMinCl), noise-to-signal ratio computed from the average attribute entropy and the mutual information in the data (noiSigR), number of attributes (attr) and number of numeric attributes (num). Regrading PSO, the tree presents the following measures: number of classes (classes), minimum number of levels of nominal attributes (minLevel), joint entropy (jEntr), skewness and the inst, prMinCl, dimen, kurt, attr and num, measures introduced above.

In both cases, the number of attributes and the number of instances in the dataset were among the most important measures determining the sensitivity of HP tuning regarding the sampling.

5 Conclusions and Future Work

This paper investigated the effect of random sampling on SVM HP tuning. For such, several experiments were performed, where various samples were generated for 100 datasets using PSO and RS HP tuning techniques. The main goal of this study was to investigate how the size of the sample affects the HP tuning results, when compared with the results obtained for the whole dataset.

The presented research[8] is in its early stages and need to be further pursued but the experimental results obtained indicate some future research directions, which include: to investigate the use of the Hausdorff distance for HP tuning in data stream mining , where new instances arrive continuously, investigate how the results would be affected by the use of other HP tuning techniques [14,16] or some stratified sampling techniques [9]. To the best of the authors' knowledge, this research is the first to investigate the effects of sampling on the HP tuning process, is relevant for ML and worth further research.

[8] Supported by the Brazilian Funding Agencies CAPES, CNPq and São Paulo Research Foundation FAPESP (CeMEAI-FAPESP process 13/07375-0 and grant #2012/23114-9), and the Slovakian project VEGA 1/0475/14.

References

1. Claus Bendtsen. *pso: Particle Swarm Optimization*, 2012. R package version 1.0.3.
2. James Bergstra and Yoshua Bengio. Random Search for Hyper-parameter Optimization. *J. of Mach. Learn. Res.*, 13:281–305, 2012.
3. Igor Braga, Laís Pessine do Carmo, Caio César Benatti, and Maria Carolina Monard. A Note on Parameter Selection for Support Vector Machines. In *LNCC*, volume 8266, pages 233–244. 2013.
4. Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A Library for Support Vector Machines. *ACM Trans. on Intell. Syst. and Techn.*, 2(3):27:1–27:27, 2011.
5. William G. Cochran. *Sampling Techniques*. John Wiley, 3 edition, 1977.
6. Frauke Friedrichs and Christian Igel. Evolutionary Tuning of Multiple SVM Parameters. *Neurocomputing*, 64:107–117, 2005.
7. Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel Methods in Machine Learning. *The Annals of Statistics*, 36(3):1171–1220, 2008.
8. Frank Hutter, HolgerH. Hoos, and Kevin Leyton-Brown. Sequential Model-Based Optimization for General Algorithm Configuration. In *LNCS*, volume 6683, pages 507–523. 2011.
9. Kevin Lang, Edo Liberty, and Konstantin Shmakov. Stratified Sampling Meets Machine Learning, 2015.
10. R. G. Mantovani, A. L. D. Rossi, J. Vanschoren, B. Bischl, and A. C. P. L. F. de Carvalho. Effectiveness of Random Search in SVM hyper-parameter tuning. In *Int. Joint Conf. on Neur. Netw.*, pages 1–8, 2015.
11. R. G. Mantovani, A. L. D. Rossi, J. Vanschoren, B. Bischl, and A. C. P. L. F. de Carvalho. To tune or not to tune: Recommending when to adjust SVM hyper-parameters via meta-learning. In *2015 International Joint Conference on Neural Networks*, pages 1–8, 2015.
12. Rafael Gomes Mantovani, André L. D. Rossi, Joaquin Vanschoren, and André C. P. L. F. Carvalho. Meta-learning Recommendation of Default Hyper-parameter Values for SVMs in Classification Tasks. In *2015 Int. Worksh. on Meta-Learning and Algorithm Selection at ECML/PKDD*, pages 80–92, 2015.
13. Xiangrui Meng. Scalable Simple Random Sampling and Stratified Sampling. In *JMLR Worksh. and Conf. Proc.*, volume 28, pages 531–539, 2013.
14. Michinari Momma and Kristin P. Bennett. A Pattern Search Method for Model Selection of Support Vector Regression. In *SIAM Int. Conf. on Data Mining*. SIAM, 2002.
15. Matthias Reif, Faisal Shafait, Markus Goldstein, Thomas Breuel, and Andreas Dengel. Automatic Classifier Selection for Non-experts. *Pattern Analysis and Applications*, 17(1):83–96, 2014.
16. Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian Optimization of Machine Learning Algorithms. In *NIPS*, pages 2960–2968, 2012.
17. Carlos Soares and Pavel B. Brazdil. Selecting Parameters of SVM using Meta-learning and Kernel Matrix-based Meta-features. In *ACM Symp. on Applied computing*, pages 564–568. ACM, 2006.
18. Abdel Aziz Taha and Allan Hanbury. An Efficient Algorithm for Calculating the Exact Hausdorff Distance. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 37(11):2153–2163, 2015.
19. Yves Tillé. *Sampling Algorithms*. Springer, 2006.
20. Xin-She Yang, Zhihua Cui, Renbin Xiao, Amir Hossein Gandomi, and Mehmet Karamanoglu. *Swarm Intelligence and Bio-Inspired Computation: Theory and Applications*. Elsevier, 2013.