Contrasting the Profiles of Easy and Hard Observations in a Dataset

Camila Castro Moreno
Instituto Tecnológico de Aeronáutica (ITA)
Praça Marechal Eduardo Gomes, 50
São José dos Campos, SP, Brazil
camila.moreno@ga.ita.br

Pedro Yuri Arbs Paiva
Instituto Tecnológico de Aeronáutica (ITA)
Praça Marechal Eduardo Gomes, 50
São José dos Campos, SP, Brazil
paiva@ita.br

Gustavo H. Nunes
Universidade Federal de São Paulo (UNIFESP)
Av. Cesare Monsueto Giulio Lattes, 1201
São José dos Campos, SP, Brazil
gustavo.nunes@unifesp.br

Ana Carolina Lorena
Instituto Tecnológico de Aeronáutica (ITA)
Praça Marechal Eduardo Gomes, 50
São José dos Campos, SP, Brazil
aclorena@ita.br

Abstract

For supporting data-centric analyzes, it is important to identify and characterize which observations from a dataset are hard or easy to classify. This paper employs meta-learning strategies to describe the main differences between observations which are easy and hard to classify in a dataset. Intervals on significant meta-features values assessing the hardness levels of the observations are extracted and contrasted. This meta-knowledge allows for characterizing the hardness profile of a dataset and obtaining insights into the main sources of difficulty they pose, as shown in experiments using two super-classes of the CIFAR-100 dataset with different hardness levels.

1 Introduction

Knowing which observations in a dataset are hard to classify is a valuable tool for supporting data-centric analysis. These observations are usually worth a closer inspection for identifying potential data quality issues. They can also pinpoint a classification model's weaknesses, which can then be further explored. But what makes an instance hard to classify? There are multiple possible sources of difficulty in a classification problem (Lorena et al., 2019), such as class overlapping, feature and label noise, data sparsity, among others. For instance, one might expect observations in overlapping areas of the classes to be misclassified more often than observations placed in dense regions of instances sharing their label. Smith et al. (2014) define as hard an observation which gets systematically misclassified, regardless of the learning model employed in its analysis. They introduce an *instance hardness* (IH) measure which averages the probably of misclassification of an instance by a diverse pool of learning algorithms. In addition, they present a set of *hardness measures* (HM) intended to explain possible reasons as to why an instance is hard to classify.

Here we take advantage of such measures to perform a meta-analysis of a classification dataset with the intention of characterizing and describing its hardness profile. For such, a meta-dataset is first built where each observation is described by a set of HMs and is labeled according to a binarized IH value indicating if it is easy or hard to classify. Next, intervals contrasting the meta-features' values assumed by these subsets of observations are extracted. A methodology originally proposed to extract the domains of competence of classifiers based on datasets meta-characteristics from (Luengo and Herrera, 2013) is adapted for extracting such intervals. Joining these intervals allows us to build

NeurIPS Data-Centric AI Workshop 2021.

descriptive rules that characterize the profile of easy and hard instances from a dataset. Experiments with two super-classes of the CIFAR-100 image classification problem with different hardness levels evidence the utility of our methodology in identifying the hardness profile of a dataset and what are the main reasons for an instance being considered as hard.

2 Methodology

Meta-dataset Generation. The first step of the methodology involves describing the original dataset under analysis $\mathcal{D}$ by a set of HM, composing a meta-dataset $\mathcal{M}$. For this we used the PyHard[1] package. The meta-features used are summarized in Table 1, with name, acronym, minimum and maximum values achievable and the reference where they are defined. All measures were standardized so that higher values are indicative of a higher hardness level concerning the aspects they quantify.

Table 1: Hardness measures employed as meta-features in this work.

Measure	Acron.	Min	Max	Reference
k-Disagreeing Neighbors	kDN	0	1	(Smith et al., 2014)
Disjunct Class Percentage	DCP	0	1	(Smith et al., 2014)
Tree Depth (pruned)	TD_P	0	1	(Smith et al., 2014)
Tree Depth (unpruned)	TD_U	0	1	(Smith et al., 2014)
Class Likelihood	CL	0	1	(Smith et al., 2014)
Frac. features in overlapping areas	$F1$	0	1	(Arruda et al., 2020)
Frac. nearby instances of different class	$N1$	0	1	(Arruda et al., 2020)
Ratio of intra-extra class distances	$N2$	0	≈ 1	(Arruda et al., 2020)
Local set cardinality	LSC	0	1	(Arruda et al., 2020)
Local set radius	LSR	0	1	(Arruda et al., 2020)
Usefulness	U	≈ 0	1	(Arruda et al., 2020)
Harmfulness	H	0	≈ 1	(Arruda et al., 2020)

Next the observations in $\mathcal{D}$ must be labeled as either hard or easy. For such, we must compute their IH estimate. Given a classification dataset $\mathcal{D}$ with n pairs $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i$ is an observation and y_i is its class label, the IH of $\mathbf{x}_i$ is given by Equation 1 (Smith et al., 2014):

$$IH\big(\mathbf{x}_i, y_i\big) = 1 - \frac{1}{|\mathcal{A}|} \sum_{j=1}^{|\mathcal{A}|} p\big(y_i|\mathbf{x}_i, l_j\big), \tag{1}$$

where $p\big(c_i|\mathbf{x}_i, l_j\big)$ gives the probability a learning model l_j from a pool of models $\mathcal{A}$ induced from D assigns $\mathbf{x}_i$ to its expected class y_i. According to this formulation, observations that are frequently misclassified have a large IH value and are considered hard, whilst easy instances are likely to be correctly classified by any of the algorithms in $\mathcal{A}$. For composing the set $\mathcal{A}$ in this work, we employed seven classification models with different biases: Bagging (Bag), Gradient Boosting (GB), Support Vector Machines (SVM, with both linear and RBF kernels), Multilayer Perceptron (MLP), Logistic Regression (LR) and Random Forest (RF). The meta-dataset is labeled applying a threshold on the IH values. Herewith, observations with an IH value above a threshold (here set as 0.4) are labeled as hard. Otherwise, they are labeled as easy.

PyHard also includes a meta-feature selection step to keep in $\mathcal{M}$ only the meta-features which relate the most to the classification performance of the classifiers in the pool $\mathcal{A}$. Using a mutual information score, ordered lists of meta-features that best relate to the outputs of each of the classifiers in $\mathcal{A}$ are generated. These ranks are next aggregated and the 10 top-ranked meta-features are kept.

Automatic Extraction Method. Given the meta-dataset $\mathcal{M}$, our objective is to extract intervals of meta-feature values which characterize the profile of the easy and hard subsets of instances of the original base dataset $\mathcal{D}$. We adapt the descriptive method introduced in Luengo and Herrera (2013) for characterizing the domains of competence of different classifiers for such. There, given a classification technique and a pool of classification datasets described by meta-features assessing their complexity, they extract intervals of meta-feature values which better describe the predictive performance (binarized as good or bad) achieved for the datasets. Here, given a classification dataset

[1] `https://pypi.org/project/pyhard/`

whose observations are described by a set of hardness measures and a pool of classifiers with distinct biases, the intent is to extract similar intervals but related to the hardness level of the dataset's observations (binarized as easy or hard). This Automatic Extraction Method (AEM) first goes through each HM meta-feature and uses them to sort the observations in ascending hardness level according to that meta-features' values. Once sorted, the AEM finds continuous intervals containing either only easy or hard instances.

AEM has two main hyperparameters, percent_merge and percent_drop. Consecutive intervals of easy (hard) instances with gaps smaller than percent_merge $\times n$ are merged, where n is the size of the dataset, as small gaps between almost contiguous intervals may be anomalies or insignificant and can be disregarded. After this, intervals containing less than percent_drop $\times n_{\text{class}}$, where n_{class} is the size of the easy or hard class, are considered non representative and are dropped. To tune these hyperparameters we perform a grid search on validation sets separated from $\mathcal{M}$. percent_merge is varied from 0 to 0.01 with a step of 0.002 and percent_drop is varied from 0 to 0.25 with a step of 0.05. The hyperparameter values that result in the best weighted F1 score when the resulting intervals are used to classify the observations of the validation meta-dataset as easy or hard are chosen. Finally, we extract the intervals with the tuned hyperparameters. For each meta-feature we keep only the easy and the hard intervals with the largest support, that is, the easy/hard intervals that cover the greatest amount of instances, therefore being the most representative of easy/hard behavior. Compiling the meta-features with easy and hard intervals, the interval values and the interval support, we obtain a table that we can use as a ruleset describing easy or hard behavior.

The proposed methodology is implemented in Python and can be found in the *Instance Hardness Automatic Extraction Method* (IH-AEM) repository (Moreno, 2021).

3 Experiments

We present here case studies employing the proposed approach, using two different superclasses of the CIFAR-100 dataset (Krizhevsky, 2009). It has 100 classes that are grouped into 20 superclasses. Here we take the training sets of two of such superclasses, one with a high hardness level (people) and other with an easy profile (vehicles_2). These images were first supplied to an Inception network (Szegedy et al., 2016) pre-trained on the ImageNet dataset (Deng et al., 2009). The activation levels of the penultimate layer of the network are used as a feature vector. Meta-datasets are then produced using the resulting structured datasets.

Each meta-dataset was split into train and validation sets. The validation sets were 30% the size of the original sets. The experiments were all run on a laptop with an 2.40 GHz Intel i75500U processor using 8 GB of RAM, running Ubuntu version 20.04. The time it took run each step of the methodology is as follows: 200.2 minutes to generate the meta-dataset for the people dataset and 333.6 minutes for the vehicles_2 dataset; 12.3 minutes to tune the AEM hyperparameters for the people dataset and 12.25 minutes for the vehicles_2 dataset; and finally 36.2 seconds to extract the rules for the people dataset and 36.7 seconds for the vehicles_2 dataset.

People superclass. The people dataset has the following classes: baby, boy, girl, man, and woman. It has a hard profile and 81.2% of the dataset is composed of hard instances. Table 2 shows part of the intervals extracted for this dataset. In particular, only measures with defined intervals for both easy and hard classes are shown. The complete set of intervals is presented in the Supplementary material. Support values are higher for hard intervals, because most of the dataset is labeled as hard. Contrasting intervals, we can notice higher lower and upper bounds for the hard intervals, as expected. In particular, hard instances have a lower likelihood of belonging to their classes (as measured by CL), are surrounded by 20% to 100% of observations from another class (measured by kDN) and have a lower intra-class to extra-class distance ratio (N2). Other measures (LSC and Usefulness) are based on the distance of each observation to their nearest enemies, which corresponds to their nearest neighbor of another class, so that hard observations tend to be closer to these points. These results point that hard observations might be either borderline or noisy. But easy observations from this dataset have average to high values for most of the HMs and are not so easy. Figure 1 presents some examples of observations considered easy (top) and hard (bottom) by category of the people dataset. Most of the hard-to-classify observations are indeed borderline or poorly labeled, such as the cases of baby, boy and girl. Others are out of focus (woman) or pose (man).

Table 2: Part of the meta-feature intervals of easy and hard behavior of the people dataset

	Easy		Hard	
Meta-feat.	Interval	Support	Interval	Support
CL	[0.47, 0.47]	0.06	[0.51, 0.91]	0.63
LSC	[0.94, 0.99]	0.06	[0.99, 1.0]	0.59
N2	[0.45, 0.47]	0.05	[0.49, 1.0]	0.81
Usefulness	[0.92, 0.99]	0.07	[0.99, 1.0]	0.77
kDN	[0.0, 0.2]	0.06	[0.2, 1.0]	0.84

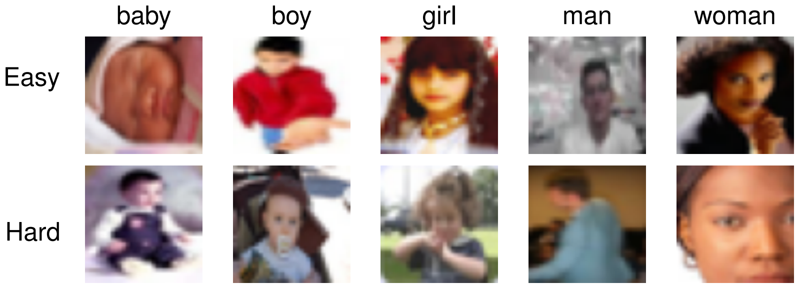

Figure 1: Easy and hard images for each label from the CIFAR-100 People Superclass dataset.

Vehicles_2 Superclass. The vehicles_2 dataset has an easy profile, with 87.9% of the dataset labeled as easy. Therefore, higher support values are verified here for the easy intervals, which are also more numerous. Part of the rules are shown in Table 3 (the complete table is within the Supplementary material). In general the meta-features values are much lower in the easy class than in the hard counterpart, evidencing they are really easy. Hard instances have a lower likelihood of belonging to their classes (CL), and are surrounded by neighbors of a different class (kDN and LSC). This indicates they may be similar to observations from other classes. Some representatives of the easy and hard classes per category of the vehicles_2 dataset are shown in Figure 2 from the Supplementary material. Hard representatives are out of focus or atypical.

Table 3: Part of the meta-feature intervals of easy and hard behavior of the vehicles_2 dataset

	Easy		Hard	
Meta-feature	Interval	Support	Interval	Support
CL	[0.19, 0.19]	0.14	[0.96, 0.96]	0.01
LSC	[0.48, 0.95]	0.41	[1.0, 1.0]	0.01
kDN	[0.0, 0.0]	0.23	[1.0, 1.0]	0.01

4 Conclusion

This short paper presents a meta-learning (MtL) methodology used to characterize the profile of observations easy and hard to classify in a dataset. The results prove the usefulness of such a descriptive approach to better understand the hardness profile of a dataset. A limitation of our work is that when regarding hardness, we frequently work with very unbalanced classes. Many datasets have either a greater amount of easy instances or a greater amount of hard instances. This hampers the extraction of representative intervals for the underrepresented group. Future work shall consider other meta-features and hardness levels beyond an easy/hard dichotomy.

Acknowledgments and Disclosure of Funding

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

References

Arruda, J. L., Prudêncio, R. B., and Lorena, A. C. (2020). Measuring instance hardness using data complexity measures. In *Brazilian Conference on Intelligent Systems*, pages 483–497. Springer.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report.

Lorena, A. C., Garcia, L. P., Lehmann, J., Souto, M. C., and Ho, T. K. (2019). How complex is your classification problem? a survey on measuring classification complexity. *ACM Computing Surveys (CSUR)*, 52(5):1–34.

Luengo, J. and Herrera, F. (2013). An automatic extraction method of the domains of competence for learning classifiers using data complexity measures. *Knowledge and Information Systems*, 42(1):147–180.

Moreno, C. (2021). Instance hardness automatic extraction method. `https://github.com/Camila44567/IH-AEM`.

Smith, M. R., Martinez, T., and Giraud-Carrier, C. (2014). An instance level analysis of data complexity. *Machine Learning*, 95(2):225–256.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

A Supplementary Material

This section presents the supplementary material of this work. Table 4 presents the complete set of intervals describing the easy and hard observations of the people dataset. Table 5 presents the same information for the vehicles_2 dataset. Figure 2 presents examples of easy and hard instances from the vehicles_2 dataset, per category of this classification problem.

Table 4: CIFAR-100 People Superclass dataset meta-feature intervals of easy and hard behavior

	Easy		Hard	
Meta-feat.	Interval	Support	Interval	Support
CL	[0.47, 0.47]	0.06	[0.51, 0.91]	0.63
F1	-	-	[0.96, 0.97]	0.23
Harmfulness	-	-	[0.0, 0.05]	1.00
LSC	[0.94, 0.99]	0.06	[0.99, 1.0]	0.59
LSR	-	-	[0.39, 0.51]	0.44
N2	[0.45, 0.47]	0.05	[0.49, 1.0]	0.81
TD_P	-	-	[0.6, 0.8]	0.74
TD_U	-	-	[0.38, 0.88]	0.88
Usefulness	[0.92, 0.99]	0.07	[0.99, 1.0]	0.77
kDN	[0.0, 0.2]	0.06	[0.2, 1.0]	0.84

Table 5: CIFAR-100 Vehicles_2 Superclass dataset meta-feature intervals of easy and hard behavior

	Easy		Hard	
Meta-feature	Interval	Support	Interval	Support
CL	[0.11, 0.12]	0.08	[0.96, 0.96]	0.01
DCP	[0.037, 0.084]	0.21	-	-
F1	[0.97, 0.97]	0.05	-	-
Harmfulness	[0.0, 0.0]	0.07	-	-
LSC	[0.48, 0.95]	0.41	-	-
N1	[0.0, 0.33]	0.07	[1.0, 1.0]	0.01
N2	[0.39, 0.44]	0.18	-	-
Usefulness	[0.39, 0.89]	0.28	-	-
kDN	[0.0, 0.0]	0.25	[1.0, 1.0]	0.01

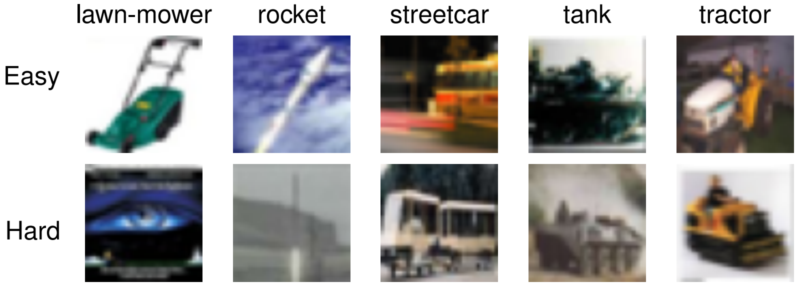

Figure 2: Easy and hard images for each label from the CIFAR-100 Vehicles_2 Superclass dataset.