|  |  |  |
| --- | --- | --- |
| ITU Logo | INTERNATIONAL TELECOMMUNICATION UNION**TELECOMMUNICATIONSTANDARDIZATION SECTOR**STUDY PERIOD 2017-2020 | FG-AI4H-B-027 |
| **ITU-T Focus Group on AI for Health** |
| **Original: English** |
| **WG(s):** | Plenary | New York, 15-16 November 2018 |
| **DOCUMENT** |
| **Source:** | Fraunhofer HHI |
| **Title:** | Draft: Selection of representative training data for evaluation of ML algorithms |
| **Purpose:** | Discussion |
| **Contact:** | Wojciech Samek Fraunhofer HHIGermany | Tel: +49 30 31002 417Email: wojciech.samek@hhi.fraunhofer.de  |

|  |  |
| --- | --- |
| **Abstract:** | This contribution describes different approaches for the selection of representative training data from a large dataset. The goal of these selection methods is to ensure that ML algorithms show similar generalization properties when trained on the training data and the full dataset. |

# Introduction

The FG-AI4H aims to collect large medical datasets for evaluating ML algorithms. Since this type of data is very valuable and it takes a lot of effort to collect it, only a small fraction of it will be made publicly available as *training data*. The other part will be held out for the evaluation of ML algorithms. In order to allow a fair evaluation and comparison, the training data need to be *representative*, i.e., reflect of the data distribution of the full dataset. If this is not ensured, e.g., training data may be biased or may ignore certain regions in data space (e.g., rare diseases), then ML methods will not be able to solve the learning task appropriately.

This contribution discusses several approaches for selecting a representative training dataset.

# Problem description

Let us assume we have a large dataset *X*containing *n* samples, e.g. histopathology images. Each data sample *x* contains an associated label *y*, e.g. specifying if the image contains cancerous cells or not. Assume that we can train a classifier *f* on this dataset, which is able to predict the labels with sufficient accuracy[[1]](#footnote-1).

Since we do not want to make the whole dataset *X* publically available but at the same time want to give other researchers the opportunity to train ML models for this classification task, we need to find a smaller, representative subset of *X* containing *m << n* samples. This dataset *Z* can then be published as training dataset. However, for a fair evaluation we need to ensure that a classifier *g* trained on *Z* has similar generalization properties than the classifier *f* trained on *X*. Note that this does not mean that both classifiers need to have the same generalization error. Since *f* is trained on much more data than *g*, it will in general have better overall performance (e.g., lower error). However, we want to ensure that there is no systematic bias in the dataset *Z*, which prevents *g* to learn about certain things (e.g. specific types of cancer cells) *f* has learned about.

One way to formalize this property is to analyze the distribution of errors that *f* and *g* make on a random subset *R* of dataset *X*. With that we can define *Z* to be *representative* of *X* if the errors that classifiers *f* and *g* make on a random (but large enough) dataset *R* (subset of *X*) is bounded with high probability, e.g. follows a Gaussian distribution. If this is not that case, then this means that we can find a dataset *R* (e.g. containing specific cancer cells), so that the performance of classifiers *f* and *g* largely deviate when applied to this dataset, e.g. *f* is able to correctly classify these cells, whereas *g* is not, because it has never seen this type of cancer in the training data.

In the following we discuss several (heuristic) approaches for constructing such a representative dataset *Z*.

# Selection based on sample similarity

If classifier *f* is not available, then the selection of training data needs to be based just on the dataset *X*. One way to find representative examples in *X* is to define a measure of similarity between individual data samples and to apply a clustering algorithm such as k-means, hierarchical k-means, spectral clustering, GMM etc. The cluster centers then would constitute the training dataset.

Note that the similarity measure does not necessarily need to be defined in the original input space (e.g. pixel space for images), but the clustering can also be performed in some feature space. The mapping to the feature space could be pre-specified or learned. For instance, one could apply a pre-trained neural network (e.g. AlexNet, VGG, GoogleNet) to the data and construct the feature vectors from the hidden activations. Also, one could train an autoencoder (unsupervised) on the dataset *X* and use the hidden representation as feature vectors.

There are many ways to perform the feature mapping and the clustering, however, all of them aim to find a good coverage of the input space, but do not take into account the similarity between *f* and *g*, i.e. do not solve the problem defined above. However, a good coverage may result in a representative dataset. An advantage of this approach is that it is fast and does not require the classifier *f*.

# Selection based on model information

In the following we assume that we have access to a classifier *f* which is able to classify our data which sufficient accuracy. Then there are several possibilities to construct a representative dataset.

## Selection based on clustering with discriminative similarity metric

As before we can select the representative samples as cluster centers. However, since we have access to the classifier *f*, we can define a similarity metric which takes into account discriminative information. There are many possibilities to do so. For instance, if *f* is a neural network, then we can e.g. take the activations of the last (convolutional) layer as features and perform clustering in this space using Euclidean distance. This approach has the advantage that it can be easily implemented and it would cover the different activation patterns of classifier *f*. However, it is not clear if classifier *g* is (in principle) able to learn the mapping from input to the activation space just from the training data *Z*.

## Selection based on Fisher Vector kernel

An alternative approach is based on Fisher kernels. Mathematically, a Fisher kernel can be defined as  with  and .

The Fisher kernel can be seen as a measure of similarity between two samples. However, the similarity depends on the model (here denoted as *f*(*x*) = *p(x |* θ*)*). Two samples are similar according to the Fisher metric if a slight perturbation of the fitted parameters of the model would impact the fit of the two samples similarly.

The paper [1] use the Fisher kernel to identify the most relevant data points to explain the prediction. For that the authors use Sequential Bayesian Quadrature to trade-off between selecting data points that are representative of the parametric fit and diversity of the selected points. In our case a similar procedure could be used to construct *Z*.

The disadvantage of the approach is its computational complexity, i.e., it may be not applicable to large dataset and large models.

## Selection based on influence function

Another approach which is related to the Fisher Vector based sampling has been presented in [2]. Also, here the authors aim to explain predictions by finding the most relevant samples. This is done by optimizing the following objective problem, where H is the Hessian and L the loss function. Once identified, the most influential data points could constitute the training dataset. The disadvantage of the approach is its computational complexity, i.e., it may be not applicable to large dataset and large models.

## Selection based on (reduced) support vectors

Another approach to select the most important samples is to train a support vector machine (SVM) and to use the samples, which correspond to the support vectors as training data. Note that also here the can be either trained on the original input or on some feature representation (e.g. neural network activations). The (linear) SVM optimization problem can be written in dual from as:



Note that 0 < *ci* specify the data samples *xi* which lie on the margin (i.e. support vectors). Analogously, one can identify support vectors in non-linear SVM classifiers.

There have been a lot of work (e.g. [3]) aiming to reduce the complexity of the SVM by reducing the number of support vectors. Here the idea is to randomly select a subset of training data considered as candidates of support vectors. The performance of these reduced SVMs has been shown to be comparable to SVMs trained on the full dataset.

## Selection based on Active learning

Finally, one can select the most relevant examples iteratively using an active learning approach. The goal of active learning is to actively identify the most relevant (unlabeled) sample in every iteration and to ask for the label of it. There are many strategies how to do that. These relevant samples could constitute the training data set *Z*.

# Theoretical guarantees

All the presented approaches subsample the full dataset in a more or less meaningful manner, but do not provide any guarantees on the generalization abilities of the classifier *g* trained on the subsampled data.

Approximation theory [4] provides bounds for the sampling error *εH,s(f) – εH(f)*, i.e., the deviation between the error of the best function in the hypothesis space *H* *εH(f)* and the best function in the hypothesis space which can be computed using the given samples *εH,s(f)*. One can prove that



samples suffice to guarantee that P[*εH,s(f) – εH(f)* ≤ *ε*] ≥ 1 – δ.

# Conclusion

Although various approaches exist to subsample a large dataset, these heuristics do not provide any guarantees that the training data will be representative. Some interesting theoretical work on bounding the sample error has been developed by the functional approximation community, which can potentially be extended to our problem setting. In summary, more research on the data selection problem is needed before it can be safely applied in the FG-AI4H setting.

# References

[1] R. Khanna, B. Kim, J. Ghosh, and O. Koyejo, “Interpreting Black Box Predictions using Fisher Kernels”, *arXiv:1810.10118*, 2018.

[2] P. W. Koh and P. Liang, “Understanding black-box predictions via influence functions”, *Proceedings of the 34th International Conference on Machine Learning*, 1885–1894, 2017.

[3] K.-M. Lin and C.-J. Lin, "A study on reduced support vector machines", *IEEE transactions on Neural Networks*, 14(6):1449-1459, 2003.

[4] F. Cucker and D. X. Zhou, *Learning theory: an approximation theory viewpoint*, Cambridge University Press, 2007.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. The generalization ability of the classifier *f* can be evaluated by cross validation. [↑](#footnote-ref-1)