Extended a Priori Probability (EAPP) tool (ITI)

Summary: ITI has worked in dataset bias discovery to develop machine learning solutions. As a result of this work, they have defined a semi-supervised baseline metric (EAPP). This metric is intended for binary classification tasks that consider not only the a priori probability but also some possible bias present in the dataset and other features that could provide a relatively trivial separability of the target classes. The procedure involves (1) multiobjective feature extraction and (2) a clustering stage in the input space with autoencoders, and (3) a subsequent combinatory weighted assignation from clusters to classes depending on the distance to the nearest cluster for each class.

Status : Containerized

Contacts : dsilveira@iti.es ; silviaruiz@iti.es

Table of Contents

I. Purpose

Method to calculate a more informative metric set than the simple *a priori* probability for estimating the difficulty or bias of the data in the context of a binary classification task.

II. Tool description for its conceptual validation

The conceptual validation aims to ensure that a tool is aligned in its purpose and functionality with the pre-processing tools specified in EUCAIM T5.3. For this, the following documentation will detail **the necessity of the tool and its functioning**.

An oral presentation has been given on June 26th 2023 during a focus group meeting dedicated to data quality assessment and data cleaning, for partners to understand the tool better as well as to provide feedback.
Link to the presentation of the tool :
https://drive.google.com/file/d/1Zi0SE3jpZQXLoApvV1Spue7Qo6JTQ7tm/view?usp=drive_link
Link to the video of the oral presentation :
https://drive.google.com/file/d/1LQS2rkTN-EEArnNdkEC3ajdJRw6LcEJ0/view?usp=drive_link

EUCAIM WP5 Task Force 2

Tools for data quality assessment and data cleaning

1. Name:

Extended a Priori Probability (EAPP) tool

2. Contributor:

ITI

3. Area:

Data quality assessment

4. Tool description:

The tool provides a semi-supervised metric for binary classification tasks that considers not only the a priori probability but also some possible bias present in the dataset, as well as other features that could provide a relatively trivial separability of the target classes. Therefore, it allows for evaluating the ease or complexity of the task or bias of the data beyond the well-established baseline for any binary classification.

5. Data:

The EAPP algorithm accepts both tabular data and images as input.
To test the EAPP behaviour for diverse binary classification tasks, a wide range of scenarios was covered so different well-known image datasets were analyzed:

a) since the EAPP deals with binary classification tasks, a subset of the handwritten digits dataset MNIST [1] was extracted. The images of '1' and '7', which are relatively similar, were compared to those of '8'.

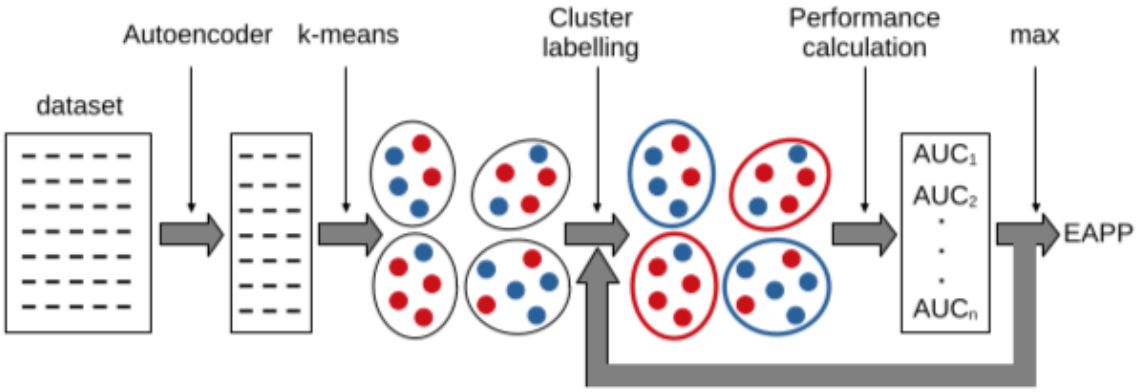
b) The ImageNet dataset [2] comprises 3.2 million images covering up to 20000 categories. Under the assumption that the categories mushroom and wedding may have different environmental elements, we selected the images of these two classes, looking at whether the background that does not contain the object could introduce bias to the dataset.

c) We also evaluated the metric using the BIMCV-PADCHEST [3] chest x-ray image dataset, a dataset with the presence of bias, and using (d) a subset of the nCOV2019 dataset [4] which contains potential bias sources.

6. Methodology/performance:

The approach is based on the area under the ROC curve (AUC ROC), known to be quite insensitive to class imbalance. The procedure involves multiobjective feature

extraction and a clustering stage in the input space with autoencoders and a subsequent combinatory weighted assignation from clusters to classes depending on the distance to nearest clusters for each class. Class labels are then assigned to establish the combination that maximizes AUC ROC for each number of clusters considered. To avoid overfit in the combined feature extraction and clustering method, a cross-validation scheme is performed in each case. EAPP is defined for different numbers of clusters, starting from the inverse of the minority class proportion, which is useful for a fair comparison among diversely imbalanced datasets. This metric represents a baseline beyond the a priori probability to assess the actual capabilities of binary classification models. The EAPP process is shown in the following figure:



7. Use: brief description of the tool's functioning (if it applies).

The process does not need any interaction from the user since it is fully automatic. The tool will provide the EAPP value for the provided input.

8. Input/output formats: description of the input/output of the tool at the validation stage (will help if some adaptation is needed).

Input : As previously commented, EAPP deals with binary classification tasks, so the tool receives two files, one for class 0 and the other for class 1. Tabular data must be in CSV format (using commas as separators and no header), while images should be in PNG or DICOM (either 2D or 3D) format. Images can be stored in a folder or compressed into a ZIP file.

Output**:** The tool generates a JSON file containing the numeric value of the EAPP metric, a description, and additional information about the process.

9. Quantitative results: performance obtained during training of the tool (if applies)

Quantitative results are shown using the EAPP metric. A high EAPP usually relates to an easy binary classification task, but it also may be due to a significant coarse-grained bias in the dataset, when the task is previously known to be difficult.

10. Qualitative results: Provide some visual results (if available) of applying such tools.

Not applicable

11. Additional information: successful use cases, external resources (open code, papers...) licence, certification … (if they apply).

There is an online version of the tool aimed for demonstration at request.

Publications

Extended a Priori Probability (EAPP): A Data-Driven Approach for Machine Learning Binary Classification Tasks. Ortiz V., Perez-Benito FJ., Del Tejo Catala O., Igual IS., January 2022; IEEE Access PP(99):1-1 ; DOI:10.1109/ACCESS.2022.3221936.

Datasets used to test the EAPP behaviour:

See the bibliography section below

Keywords for searching in databases: bias, clustering, autoencoder, combinatory, a priori probability, binary classification, semi-supervised, EAPP.

Bibliography

[1] Y. LeCun, C. Cortes, and C. J. C. Burges. (Jan. 2022). The MNIST Database of Handwritten Digits. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, ‘‘ImageNet A large-scale hierarchical image database,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255.

[3] A. Bustos, A. Pertusa, J.-M. Salinas, and M. de la Iglesia-Vayá, ‘‘PadChest: A large chest X-ray image dataset with multi-label annotated reports,’’ Med. Image Anal., vol. 66, Dec. 2020, Art. no. 101797.

[4] B. Xu, B. Gutierrez, S. Mekaru, K. Sewalk, L. Goodwin, A. Loskill, E. L. Cohn, Y. Hswen, S. C. Hill, and M. M. Cobo, ‘‘Epidemiological data from the COVID-19 outbreak, real-time case information,’’ Sci. Data, vol. 7, no. 1, pp. 1–6, 2020.

III. Technical specifications

This part of the documentation is dedicated to providing all relevant technical specifications to prepare for the tool's integration into the EUCAIM test environment.

It is possible that the tool's integration will require, among others, some modifications in the input/output, or the inclusion of monitoring mechanisms.

1. Data: In depth description of the data used to train the tool.

First of all, mention that the complete algorithm is semi-supervised, so it faces the problem of overfitting as do others of this kind. For this reason, a 10 cross-validation was established as a standard for all our experiments. To be precise, for each dataset, the whole process is split into two parts: the training process, in which 90% of the data is used to learn all clustering schemes for any *k* (centroids) and a particular internal representation, and the testing process, in which the remaining 10% is processed with all these parameters obtained by training. This procedure is repeated 10 times for each experiment to allow a fairer comparison throughout all datasets.

Four well-known datasets were used in this process:

(a) MNIST is a standard database of handwritten digits commonly used for training classifiers. It consists of the 10 different arabic numerals, but our approach is defined for binary classification. Therefore, we selected the set {'1', '7'} for the first class, and {'8'} for the second class.

(b) With MNIST experiments, a particular spatial distribution for bright pixels is easily noticeable. To overcome this, a slightly more complex dataset with more image richness is used: ImageNet. It is another well-established image dataset containing more than 20,000 categories. Still, again, we focused on two visually different categories (wedding and mushroom) to work with an appropriate subset for binary classification.

(c) The BIMCV-PADCHEST chest x-ray image dataset, a dataset with the presence of bias.

(d) The algorithm was also applied to the numerical, structured nCOV2019 dataset to evaluate the difficulty of a binary classification task that did not deal with image data and that, in addition, contains potential bias sources.

Finally, mention that, for nCov2019 dataset, the cross-validation process was 50-fold because of its reduced size, as it was not appropriate to suppress 10% of the data for training in each fold. Thus, we trained with 98% of the data in each iteration.

2. Methods: in depth description of the methodology used for its development including all data preprocessing.

The main objective of this methodology is to obtain a simple semi-supervised metric that allows evaluation of the ease or complexity of the task beyond the well-established baseline for any binary classification (namely, the a priori probability, that is, the proportion of examples belonging to the majority class). **If the task is previously known to be difficult but the EAPP is high, then the dataset is likely to have a heavy bias**, as a simple algorithm may be able to tell the difference between examples from two classes without supervision, using only the locality of the observations in the representation space.

For this purpose, a preprocessing step was performed as the analyzed image datasets contained images of multiple shapes. As Neural Networks are used and scale invariance is not our focus, the task is simplified, resizing all images to the same shape to perform feature extraction. Therefore, they were cropped as a square, keeping the same image center, and resized to 128 × 128 pixels. Regarding numerical datasets, all raw features were separately normalized (mean 0 and standard deviation 1).

EAPP aims to assess how well a non-supervised feature extraction method automatically splits classes into different clusters. The process is based on assigning the same class to all the examples that fall into the same cluster. This is done iteratively for various numbers of clusters and combinations of class assignments. A probability of belonging to a class is assigned to each observation depending on its distance to the centroids of the nearest positive and negative classes' clusters. The process is divided into three stages: feature extraction, clustering, and combinatory analysis.

Labels must not be used during the training phase to compute the EAPP metric. Therefore, feature extraction is performed using unsupervised learning methods only. Algorithms such as Convolutional AutoEncoders (CAEs) are valid candidates and are indeed used for this methodology.

Additionally, this algorithm should group observations that have similar features, as they are likely to be from the same class. Therefore, clustering algorithms such as K-means are helpful to find the inner clusters that group samples of the dataset. Even if only two classes are present within the data, we cannot assume that the latent representations of both classes are linearly separable. The optimal number of

clusters is unknown, so the algorithm should explore multiple values up to a certain limit.

Once a particular clustering scheme has been computed for a given feature representation, the aim is to assess the ease or complexity of the classification task by searching a higher baseline above the a priori probability, dependent only on the number of instances of each class. The underlying idea is, once the clustering process converges for a range of different cluster cardinalities *k* in the training stage, to save the model information (namely, the centroids) and apply this clustering to new test data but for each cluster assignation (for each *k* considered), assigning all the possible different combinations of binary labels to the clusters, leaving out the trivial ones (the extreme all negative and all-positive correspondences, since they would lead to a strongly unbalanced, useless classification). Moreover, we sort the set of observations based on the distance to the inferred clusters. The assignation to a binary class for each example in each combination is then performed depending on the distance between that instance and the centroids of the nearest positive and negative class clusters to make a continuous score available. Hence, to sum up, for each *k*, we compute all the possible binary assignations, leaving out both extreme configurations, and we obtain a similarity indicator depending on the distance to the nearest positive and negative clusters. Then, using this sorting procedure, we compute a performance index, in this case, the Area Under the ROC, and select the assignation that leads to the best score.

By using the previously mentioned datasets, this methodology has proven beneficial to preliminarily assess the difficulty of a binary classification task and suggest a certain level of bias in cases where the task is perceived to be easy and a high EAPP is found. Thus, our metric represents a baseline beyond the a priori probability to assess the actual capabilities of binary classification models.

3. Specific Technical information:

a. CPU: 8 cores (**GPU preferred** if available)
b. Programming language : Python
c. Expected RAM usage : at least 8 GB (depends on the amount of input data). 8GB allows for approximately 500 images (524 x 524).
d. Running mode (interactive/batch-based/case-based…): batch-based
e. Software version : v1.0.1
f. Libraries : docker
g. Security measures:
    - writing to host data restricted to a non-root user: Yes/**No**

- container does not require it to be executed in a privileged mode: Yes/**No**

4. Traceability and monitoring mechanism.

Not applicable

5. Unitary tests: description of the tests implemented to verify the correct functioning of the tool.

Yes, there are unitary tests for all the main functions of the tool.

6. Access restriction : do you have any access restriction to the source code or to the binaries of the tool?

The tool is dockerized and it can be executed in this environment; however, access to the source code or the binaries is not permitted.

7. Additional information for tool integration.

The tool has been adapted to accept images in DICOM format, either 2D or 3D. Due to this adaptation, some modifications have been made to the input/output. As previously mentioned, EAPP handles binary classification tasks, so the tool receives two files: one for class 0 and one for class 1. Tabular data must be in CSV format, while images should be in PNG or DICOM format. Images can be stored in a folder or compressed into a ZIP file. However, **mixing images in different formats is not allowed**.

The output has also been modified, and a JSON file is **generated** containing the numeric value of the EAPP metric, a description, and additional information about the process.

IV. Integration validation

1. Communication channel for the helpdesk

For any inquiries contact: dsilveira@iti.es, silviaruiz@iti.es

2. Most common errors

- **Invalid input format:** tabular data must be in CSV format (using commas as separators and no header), while images must be in PNG or DICOM format (2D or 3D).
- **Mixing image formats:** different image formats should not be combined.
- **Missing class-specific files:** the two files corresponding to each class must be provided.

3. FAQs

Q: What is the purpose of the tool?

A: The tool provides a semi-supervised metric for binary classification tasks that considers not only the a priori probability but also some possible bias present in the dataset, as well as other features that could provide a relatively trivial separability of the target classes. Therefore, it allows for evaluating the ease or complexity of the task or bias of the data beyond the well-established baseline for any binary classification.

Q: Which imaging format is accepted?
A: The tool expects CSV format for tabular data and PNG or DICOM format for images.

Q: What output does the tool generate?
A: A JSON file containing the numeric value of the EAPP metric, a description, and additional information about the process.

Q: How long does it take to compute the EAPP value?
A: The computation time varies depending on the size of the data and the available resources. For instance, processing CSV files takes around 8 seconds, while handling two folders containing 12 DICOM images per class takes around 50 seconds. If a GPU is used, the process is expected to be faster.

4. User Manual

a. Installation/configuration instructions

N/A

b. Usage instructions

The tool is dockerized and needs three arguments:

- **Input path for class 0 data.** It must be a CSV file, images stored in a folder or compressed into a ZIP file, or a 3D DICOM series file: **input_data_class0_path**
- **Input path for class 1 data.** It must be a CSV file, images stored in a folder or compressed into a ZIP file, or a 3D DICOM series file: **input_data_class1_path**
- **Output path.** The directory and filename (JSON format) where the results will be saved: **output_directory/results.json**

Run the following command:

- jobman submit -i image_batch_EAPP -r small-gpu -- -i0 input_data_class0_path -i1 input_data_class1_path -o output_directory/results.json

c. Additional considerations

Mandatory/optional data: the input paths for class 0 and class 1 data are mandatory to run the tool. The output path is not mandatory to run the tool but recommended to specify where the results will be stored and the filename.

Cases in which the tool should not be used: the tool must not be used with formats other than CSV, PNG, or DICOM.

5. Integration tests: Description of tests for assessing the correct integration of the tool