A probabilistic classification of X-ray sources

Classification of X-ray sources using Naive Bayes Optimized Inference (CLAXBOI)

This folder contains all the necessary to run the CLAXBOI code (Tranin et al. 2022, A&A 657, 138) to augment and classify your X-ray catalog. Requirements on the system:

1. Python >3.6
2. Ubuntu (for a few os.system commands)
3. Commonly-used Python packages in astrophysics (os, sys, numpy, scipy, astropy, yaml, tqdm)
4. NWAY (https://github.com/JohannesBuchner/nway) must be installed on your machine. Alternatively, you can skip the multiwavelength crossmatching step, but in that case it is preferable to have identified multiwavelength counterparts beforehand in your input catalog.

Requirements on the catalog:

1. first column is the identifier
2. there are columns for the coordinates (their name must be or contain "ra" and "dec" or "de")
3. there is a column for the X-ray flux

Usage:

Update the file `configfile.ini` with your filenames and data-dependent parameters (some default values are already provided). Run the following codes to transform your X-ray catalog into a value-added catalog (with multiwavelength and variability information).

1. `auto_nway.py` : identify multiwavelength counterparts
2. `auto_xlinks.py` : compute basic (multi-mission) X-ray variability information
3. `auto_gaiaglade.py` : add Gaia distance and proper motion for matching (Galactic) sources. Add GLADE distance and separation for matching (extragalactic) sources.
4. `auto_classes.py` : identify a training sample for each class using Vizier and Simbad databases.
5. `classify_new.py` : run the Naive Bayes Classification for probabilistic source identification in the whole X-ray catalog. Optimize this classification.

Documentation on `auto_nway.py`

NWAY is a Bayesian probabilistic cross-matching tool able to identify multiwavelength counterparts in large astronomical catalogs. It is used here in its simplest mode of correlating one primary catalog (the X-ray catalog) to another secondary catalog, based on geometrical (angular separations) and physical (magnitude priors) information.

`auto_nway.py` is used to identify multiwavelength counterparts of your X-ray sources. By default, the following set of wavedomain and CDS catalogues are used to search for counterparts:

- Optical: Gaia EDR3, PanSTARRS DR1, DES DR1, USNO-B1.0
- Infrared: 2MASS, AllWISE, UnWISE

This choice is motivated by the complementarity and combined completeness of these surveys. You can complement or update this set of catalogs by changing the `cds_table` list in `auto_nway.py` (and changing the `cds_cov`, `cds_ntot`, `cds_names` lists accordingly). The maximum angular separation is set by the `radius` parameter (default value: 8 arcsec). Since NWAY works on local files, eligible counterpart candidates are first searched and downloaded in cones of radius `1.25*radius` around each X-ray (true or fake) coordinates.

To obtain a good purity-completeness trade-off, NWAY features a code to calibrate a cutoff on the `p_any` quantity (i.e. probability that the source in the primary catalog has any counterpart in the secondary catalog). X-ray to optical (resp. infrared) associations having a `p_any` lower than this threshold are considered spurious matches. This calibration code is based on a comparison between matches with true and fake coordinates. Therefore, `auto_nway.py` generates fake coordinates for each X-ray source (random offset of up to 2 degrees in RA and DEC values) and run the NWAY calibration code, which computes the p_any threshold corresponding to different false positive rates. You can tune the false positive rate you need (default value 15%) by changing the `nwaycutoffs` list (the last value of the list is used as final calibration).

For each pair of X-ray / multiwavelength catalog, a file is saved containing the candidate associations along with their NWAY probabilities. Each X-ray source is then assigned a single (or no) optical (resp. infrared) counterpart, i.e. the best-matching reasonable counterpart (highest `p_i` and `p_any>threshold`) in the first multiwavelength catalog in which it was found (thus the order of `cds_tables` *does* count).

The output of the code is a catalog "..._with_counterparts.fits" containing the same number of entries as the input catalog.

Documentation on `auto_xlinks.py`

This program identifies common sources in different X-ray catalogues, assigning a unique MASTER_ID to each unique source, and computes the maximum-to-miminum flux ratios (with or without flux error). A special attention is given to the identification and removal of any ambiguous match, where a source is associated to several other sources at both 1 and 3 sigma (position errors). Spurious and flagged detections may be discarded as well.

Before running it, you need to change the variables `xmmdet`, `chadet` and `swidet` to match your local XMM, Chandra and Swift catalogs of detections.

The output of the code is a catalog "..._with_counterparts_x.fits" containing the same number of entries as the input catalog.

For more advanced X-ray-to-X-ray matches, please use the STONKS pipeline (https://github.com/ErwanQuintin/STONKS).

Documentation on `auto_gaiaglade.py`

Prerequisite: download the GLADE catalog from this directory and place it in the same folder.

`auto_gaiaglade.py` augments the X-ray catalog by adding information about the source distance and motion.

Proper motions and Gaia-based distances (from Bailer-Jones et al. 2021) are added to those sources having a Gaia counterpart identified in the previous step. The X-

ray mean luminosity based on this distance is then computed. This luminosity `Lx_2` should not be used for extragalactic sources.

Extragalactic sources are searched in nearby galaxies using the GLADE (2016 version, Dalya et al. 2018) catalog of galaxies, by performing an ellipse crossmatch using the D25 ellipse representing the visible area of the galaxy (or more exactly the ellipse of 1.26 times this radius, matching the Holmberg radius). The distance to the galaxy is assigned to X-ray matches to compute another luminosity (`Lx_1`). Another useful quantity is `SepToRadius` , the galactocentric radius, representing the ratio of the source angular separation (to the galaxy nucleus) to the radius of the galaxy at the source's position angle. `Lx_1` should not be used for Galactic sources.

At this stage, the quantities `logFxFb, logFxFr` and `logFxFw1, logFxFw2` are computed, representing the logarithm of the ratio between X-ray and optical (Gaia BP and RP bands) fluxes, and the ratio between X-ray and infrared (WISE W1 and W2 bands) fluxes, respectively.

The output of the code is a catalog "..._with_counterparts_x_loc.fits" containing the same number of entries as the input catalog.

Documentation on `auto_classes.py`

This code is used to build the training sample for each class. The present version of CLAXBOI is based on 7 classes:

- (distant) AGN and QSO
- Star
- Galactic X-ray binary
- Galactic CV
- Nearby AGN (or background QSO behind nearby galaxies)
- Extragalactic X-ray binary
- Extended source

Known sources of the first six types are identified using Vizier catalogs of AGN, stars, XRB and CV. You can find (and update) the dictionary of Vizier catalogs in the `vizcat` variable. The difference between nearby and distant AGN (resp. Galactic and extragalactic XRB) is the presence or absence of GLADE-association data.

As a result, columns `isAGN, isStar, isXRB` and `isCV` are added to the X-ray catalog, containing a bitwise flag encoding the reference(s) where the identification was (were) found. Example for `isAGN` : 1 means it is identified as AGN given its mid-infrared emission (Secrest+2015), 2 means it is in the Veron-Cetty & Veron catalog, and 3 means it is in both catalogs.

Columns `isAGN, isStar, isXRB` and `isCV` are also increased by 1024 for every source identified as the corresponding types in Simbad. Note that an identification in Simbad alone is not trusted as a robust identification and thus not used in the training sample (this is why e.g. isAGN=1024 does not produce class = 0).

When a source is unambiguously identified in the set of Vizier catalogs (meaning one of the `is...` column is non-zero and all other are zero), the value encoding the source type (which is by default, 0: QSO, 1: Star, 2: gal_XRB, 3: CV, 4: AGN, 5: ex_xrb, 6: extended) is stored in the column `class` , later used for training. Unidentified sources have no value (NULL / nan) in the `class` column.

The output of the code is a catalog "..._with_counterparts_x_loc_typ.fits" containing the same number of entries as the input catalog.

Documentation on `classify_new.py`

If you are not using the default CLAXBOI classes, you must update the `classnames` and `trueprop` variables in `configfile.ini` . The variable `trueprop` is a list of user-informed proportions of each class, expected to represent the underlying proportion of each class in the whole catalog (the sum must be one).

When you run `python3 classify_new.py` , instructions are given to help you fill missing data before running the Naive Bayes Classification. These data are (or will be after first run) contained in the files `configfile.ini` and `[name of the input catalog].in` .

Only after launching the previous command you can set `optimize_coeffs: 1` in `configfile.ini` , to perform the optimization if you wish to. This may take a few hours, but you may interrupt it before it (hopefully) converges (Ctrl+C). Then run again

```
python3 classify_new.py
```

to optimize the classification on the class of your choice. Finally, set `save: 1` and `optimize_coeffs: 0` in `configfile.ini` and run `python3 classify_new.py` one last time.

Input distributions and density estimations will be automatically plotted and saved to the directory `dirref` .

`python3 plotdistrib.py` can be used to plot these distributions again.

Eventually, to check the probability track (main interpretability tool) of a particular source:

`python3 Pbatrack.py` *`SourceIdentifier`*

For 4XMM-DR12, the output of the code can be accessed through this link: https://drive.google.com/drive/u/2/folders/13lO24K2QGayuX4GPapm2U7Xo4Zps-G-Y