Hyperclass: A Framework for Spectral Data Visualization and Analysis

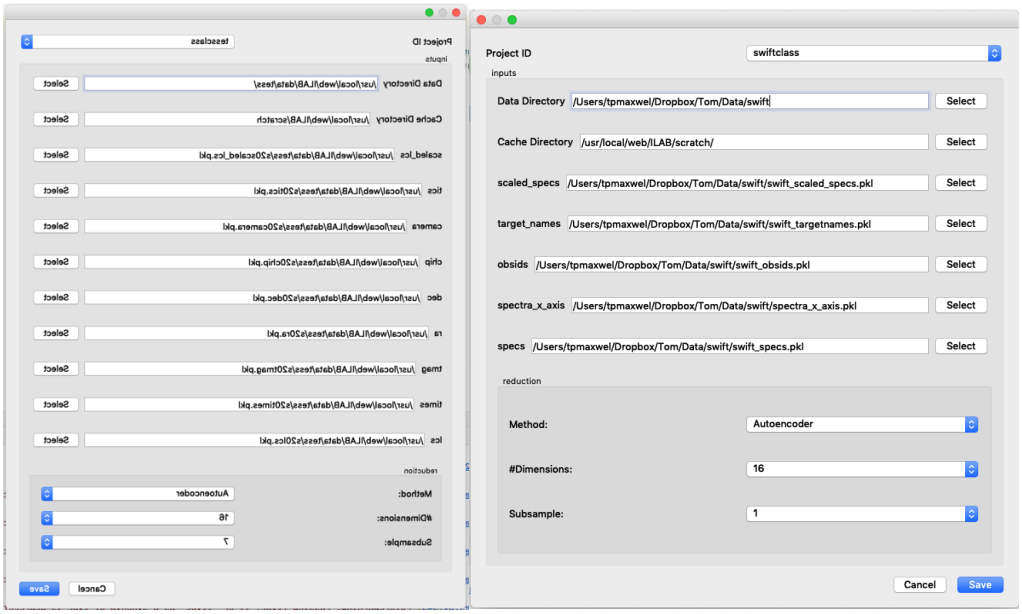
NASA Goddard Innovation Lab

Abstract

Hyperclass is an interactive workbench supporting visual data analysis of sensor data. It provides an extendable interactive interface and toolsuite which can be used to jumpstart the development of novel methods for addressing a wide range of data analysis challenges in both the earth and space sciences. We have chosen, as science drivers for the initial stage of development, the development of innovative semi-supervised machine learning methods for landscape classification using hyperspectral imagery and the visual exploration of astronomical x-ray & light curve data.

1. Introduction

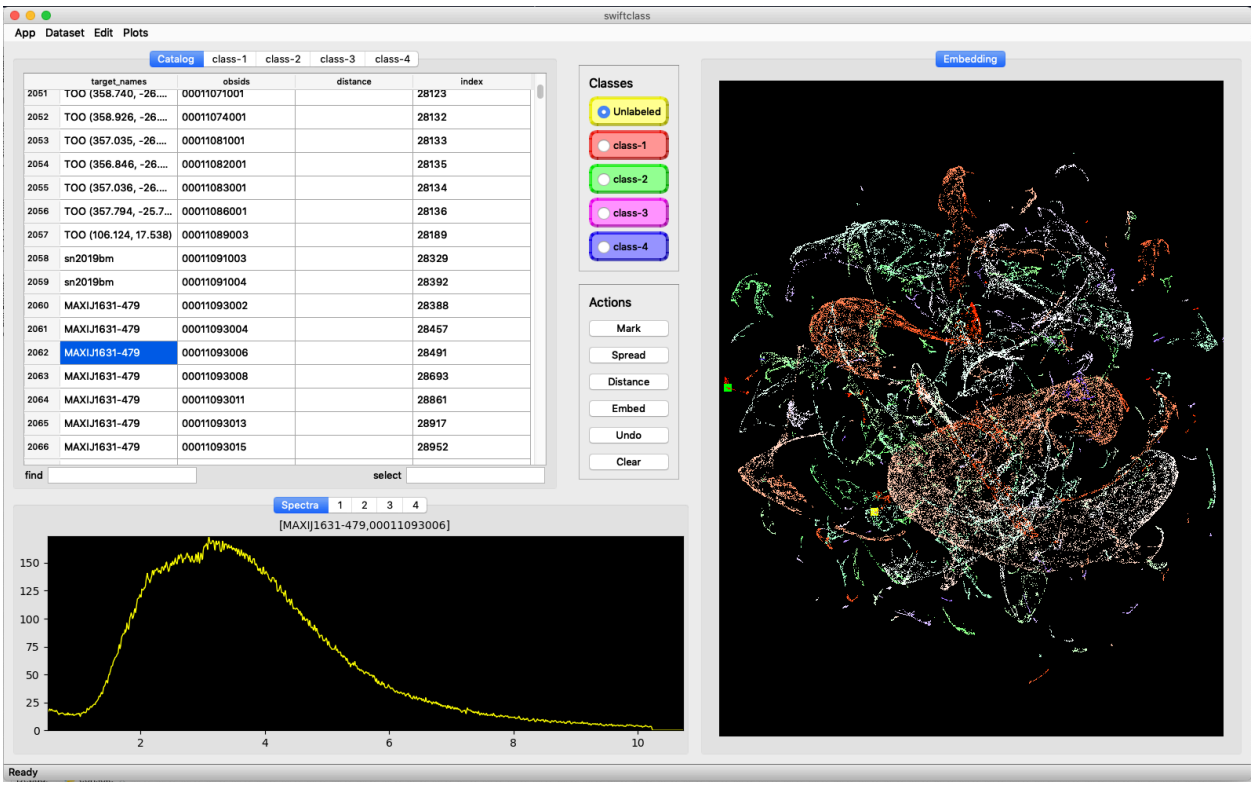
This document describes the applications of Hyperclass to visual exploration of astronomical x-ray & light curve data. It uses graph-based methods, including 3D UMAP visualization, to search for interesting items in a large database and track items which are similar to chosen items.



2. Data Preparation

Hyperclass provides dialogs for data preparation for the various supported data types (currently swift and tess, examples shown above). This dialog enables users to specify file paths for the various dataset components and parameters for dataset preprocessing (dimension reduction). When the “save” button is clicked, Hyperclass will read the various files, assemble a dataset, and apply the specified reduction operation to reduce the number of dimensions in the dataset from the number of input bands to the “# Dimensions” parameter specified in the dialog. It also enables optional subsampling of the data based on the value of the

"subsample" parameter, i.e. if the value of the subsample parameter is N, then every Nth item in the dataset will be retained and the rest will be discarded. Hyperclass then saves the dataset as a single file which can then be loaded in the Console GUI using the "Dataset" -> "Load" menu item.



3. Console GUI

The hyperclass console (shown above) provides the GUI for all operations performed using hyperclass.

The hyperclass console is composed of the following panels plus a toolbar:

- **Directory Panel**

  The Directory Panel, on the upper-left side of the console, is used to display a table of astronomical items and select items for further investigation. It allows users to search through items using any of the available metadata elements. It has the following features:

  - *Item table*: Provides a list of astronomical items with a column for each metadata element. Clicking on a table column header selects that column and sorts the table by its entries. Clicking on a row in the table selects that item in all of the panels.
  - *Class Grouping*: The group selector at the top enables the display of separate tables for each class of items. Items that are assigned a class label will be found in the table representing that class.
  - *Find/Select boxes:* The find and select boxes at the bottom of the Item table enable searches over items using the currently selected column. The find box selects the first row that matches text typed into the box. The select box selects all rows that match a regular expression typed into the box.

- **Embedding Panel**

The Embedding Panel, on the right side of the console, is used to display a UMAP embedding of the current dataset in three dimensions. The collection of points representing the current items can be viewed as a manifold in the (NB-dimensional) spectral space. Hyperclass captures the structure of this manifold by constructing a NN nearest-neighbor graph, where NN (the number of neighbors) ranges from 5 to 20 (typically 8). The UMAP mapping operation creates an embedding of this graph in three-dimensional space which endeavors to preserve the structure of the manifold. This embedding is visualized as a point cloud in the Embedding Panel. Each point in the cloud corresponds to a particular astronomical item. Visualizing the manifold structure of the hyperspectral dataset gives important insights to guide further analysis and exploration. The embedding panel has the following features:

- *Label Display*: The UMAP embedding of every label that is assigned in the Directory Panel is displayed as a marker in the Embedding Panel. As the labels are spread across the graph, the points in the point cloud are assigned the color of their inferred label.
- *Manifold Mapping*: Executing a Cmd-<right click> on a point in the point cloud places markers on both the selected point and the corresponding row in the Directory Panel.
- *Point size configuration*, available through the 'Layers' menu, allows the user to adjust the size of the point in Embedding Panel for customized visibility.
- *Navigation* in the Embedding Panel is enabled using click-and-drag operations: Left click -> pan/rotate, Right click -> zoom.

- **Point Spectral Panel**

The Point Spectral Panel is used to display a line plot of the spectrum of the currently selected item. A spectral plot is displayed in this panel whenever a point is selected in any of the other panels. The plot selector at the top enables the comparison of multiple spectra. Selecting a new plot, e.g. "2", presents a fresh canvas for the next item selection while preserving the previous spectrum plot on the previously selected canvas.

- **Labels Panel**

The Labels Panel, a buttonbox to the upper right of the Directory Panel, is used to assign class labels to selected items. The current list of classes and corresponding colors is currently specified in the Hyperclass startup script. The 'Unlabled' class is used to explore the structure of the data without assigning any labels. Selecting a label in this panel determines the currently selected class for operations in other panels.

- **Actions Panel**

The Actions Panel, a buttonbox to the lower right of the Directory Panel, is used to execute common data processing and selection operations. It provides the following action buttons:

- *Mark*: Assign the currently selected label to the item currently selected in the Directory panel.
- *Spread:* Given a set of user-specified labels, this action propagates the labels across the NN-graph, so that points that are similar to the current labeled points inherit the label of the most similar point. The spread action initiates 5 iterations of the spreading label algorithm (the algorithm is explained in more detail in Appendix 1). The inherited labels are displayed on the UMAP embedding of the NN-graph and in the Directory panel.
- *Distance:* Colors the points in the UMAP embedding by their distance to the currently labeled points. The colormap progresses from red through green to blue, with red representing "closest" and blue representing "farthest". This operation is very useful for visualizing the global structure of the embedding.

- *Embed:* Computes a UMAP embedding of the NN-graph representing the current dataset. If labels have been assigned to the data then a (semi-)supervised embedding will be computed. The results of the 3D embedding are displayed as a point cloud in the Embedding Panel (as explained in the section above).
- *Undo:* The action will undo the last operation. It is typically used to undo a label assignment.
- *Clear:* This action clears all the assigned and inherited labels, as well as their displays in the various panels, placing the dataset in its original state.

4. Appendix: Manifold-Aware Label Inference

Given a relatively small number of labeled points, this algorithm attempts to infer the labels of other similar points, thus greatly increasing the number of training samples for the classifier. It is based on the assumption that the collection of points representing the current image tile can be viewed as a manifold in the (NB-dimensional, NB=number of bands) spectral space, with an NB-dimensional Euclidean metric providing a distance measure. Hyperclass captures the structure of this manifold by constructing a NN nearest-neighbor graph, where NN (the number of neighbors) ranges from 5 to 20 (typically 8). Each edge in this graph is weighted by the distance between the connected vertices in spectral space. The graph distance between any pair of connected vertices is defined as the weight of the connecting edge, and the distance along any path through the graph is defined as the sum of the weights of all edges traversed in that path. We can then define the graph distance between any two vertices of the graph as the distance along the shortest path between them.

Using this framework we can define a simple method for inferring the label of any unlabeled vertex V. Assume we have a set of unlabeled vertices C. We can then compute the graph distance from V to all of the vertices in C and assign to V the class of the vertex in C which is closest (minimum graph distance). In order to effectively perform this operation we must construct an algorithm which efficiently computes the distance along the shortest path from any given point to each of the labeled points. The hyperclass incremental spreading activation algorithm, implemented using numpy arrays, provides a very efficient method for implementing this label inference method. If this algorithm is run to convergence, then every point in the graph will be assigned the label of the closest (minimum graph distance) labeled point. Alternately, one can iterate the algorithm a limited number of times to provide a partial labeling which covers those vertices that are closest to the preexisting labeled vertices.