GMRA Results / Data HDF5 file schema

HDF5 has the concepts of groups (which are like directories or folders), datasets (which are arrays of same-typed values or tables of mixed-type data in columns like a spreadsheet), and attributes (which are values like numbers or strings associated with either groups or datasets). Groups and datasets have names and can be hierarchically arranged like POSIX paths, but are considered "unordered" within the file. There are even the concepts of hard and soft links (references) to data, even in other files. Attributes are accessed through the object they're attached to, and are one of the mechanisms for making HDF5 data "self-describing". (Note: testing this with Sam's IPCA analysis and original data from the MNIST training 1s and 2s, the original data + labels + Sam's binary IPCA tree data dump is 220 Mb, and the uncompressed HDF5 file is 232 Mb.)

(g) group, (d) dataset, (a) attribute

/ (a) **GMRA_opts**: For now thinking of saving GMRA options as attributes attached directly to the root group of the file. They are so short, I don't think it makes sense to save them as datasets – attributes is more appropriate – but I don't know what else to attach them to...

/original_data (g)

/data (d) (d – [n_points, d_original] float)

Putting the following features as attributes of the original_data group, rather than the data dataset, because then they can be included even if the data itself isn't stored here (or referenced in another file).

/dataset_type (a – string)

This is where we'll say whether the original data points are images, text documents, etc, which will help us swap in and out different types of visualizations of the original data and feature vectors.

/image_n_rows (a – integer)

/image_n_columns (a – integer)

Attributes like this help us reconstruct the data for visualizations even when they're not needed for analysis.

/ambient_dimension (a – integer)

Used to have isCompressed, isDownsampled in Matlab...

…

/terms (d – [1, d_original] string)

This would only exist for document datasets. It is "term" or word associated with each dimension of the original data feature vectors.

/labels (g)

Vectors of numerical values or strings – one per original data point.

/first_label_name (d – [1, n_points] integer/float)

description (a – string)

This is part of the documentation. Just a description of what the label means or how it was generated. There could be other parameter attributes attached to the label which supported the label generation, including names, references or links to the models used to create it.

variable_type (a – string)

'categorical' or 'continuous'. (Perhaps in the future we'll make some distinction within categorical between nominal, ordinal and interval. Most of the data we use are nominal.)

key (a – array of (k,v) string arrays)

Specification of the correspondence between integers and more human-readable string names for each label. Only makes sense for categorical type labels. (Would be nice if this could be a group or dictionary with keys and values instead…)

- or -

key_keys (a – (n_cats,) array of integers or strings)

Ordered array of possible label values.

key_values (a – (n_cats,) array strings)

More human-readable variants of the label values (keys), ordered in the same way as key_keys.

...

/image_category (d – [1, n_points] integer)

/document_titles (d – [1, n_points] string)

…

/diffusion_graph (g)

Graph computed from Diffusion Geometry on the original data. Not saving all of the Matlab graph data computed at this point, but could be increased if it's useful.

Graph.Opts (a)

All of the options used to construct the diffusion graph. Storing them as attributes of the diffusion_graph node right now since they all tend to be small, but may find that Opts needs to be a group with some of these as attributes and some as datasets...

/eigenvectors (d – [n_points, n_eigenvals] float)

Original data point projected into the first n_eigenvals eigenbases (difusion embedding).

/eigenvalues (d – [1, n_eigenvals] float)

First n_eigenvalues of the original data

/full_tree (g)

Per-node data. The full tree Group name is node ID or covertree ID + level combination unique key. If there is overlap between information in these nodes and nodes in the covertree, links can be used to reference one from the other rather than duplicate storage

/n_nodes (a (of full_tree group) – integer)

Number of nodes in the tree. Trying not to build in any assumptions about node IDs being sequential anymore, partly because sometimes only storing a subtree that's been modified (pruned, aggregated, filtered) from the original...

Sam has thing like these in his header for the tree:
epsilon
d
m
minPoints

/full_tree (d – [n_fulltree_nodes, 5] integer)

This is the array in the covertree description / specification format: "It is an [n x 5] matrix A, where n is the number of points. A(i,1) is the level of point i in the cover tree, A(i,2) is the (index of the) parent of point i, A(i,3) is how many children i has, and A(A(i,4):A(i,4)+A(i,3)-1,5) are the (indices of the) children of point i."

/tree_root (d – hard link)

This will point to the tree root node, so can follow tree either by index or through these links down actual tree structure without having to build hierarchical tree with index paths

/node_ids (d – [n_fulltree_nodes, 5] integer)

Since there is no requirement that node IDs be sequential, need a vector of integers that will be used with the per-node labels data

/labels (g)

Vectors of numerical values or strings – one per full tree node. *(NOTE: Same sub-group format as original_data above, besides extra info about how the labels were generated)*

/first_label_name (d – [1, n_fulltree_nodes] integer/float)

description (a – string)

This is part of the documentation...

...

/diffusion_graph (g)

Graph computed from Diffusion Geometry on the tree nodes. This isn't implemented yet, but Mauro says this comes out of some of the covertree stuff. Currently using per-point diffusion embedding and then calculating approximate per-node embedding, but will switch over to this.

Graph.Opts (a)

All of the options used to construct the diffusion graph. Storing them as attributes of the diffusion_graph node right now since they all tend to be small, but may find that Opts needs to be a group with some of these as attributes and some as datasets...

/eigenvectors (d – [n_points, n_eigenvals] float)

Original data point projected into the first n_eigenvals eigenbases (difusion embedding).

/eigenvalues (d – [1, n_eigenvals] float)

First n_eigenvalues of the original data

/nodes (g)

Putting this group in just for ease of navigation in HDF5 browsers, so you can expand the full_tree group and not get a listing of all the nodes

/0 *(g)*

scaling_functions (d)
wavelets (d)
Probability_Measures *(g)*
gaussian (d)
mixture_of_gaussians (d)
Classifiers *(g)*
LDA (d)
regression (d)
parent_id (d – integer) : NOTE: may want to make this an attribute...
children (g)
This contains hard links to child nodes

Sam has things like these in each (full tree) node:
r
phi
sigma
dir
a
mse
center
indices
npoints

sigma2

/1 *(g)*
/2 *(g)*
...

/cover_tree (g)

Per-node data. Group name is node ID. Again, if there is overlap between information in these nodes and nodes in the full tree, links can be used to reference one from the other rather than duplicate storage

/cover_tree (d – [n_covertree_nodes, 5] integer)

/0 *(g)*

Same dataset types as in full tree?

/1 *(g)*
/2 *(g)*
...

/labels (g)

Vectors of numerical values or strings – one per covertree node.