Planet Nine's CS 207 Project: PersistentDB class

Sarah Wellons, Rodrick Kuate Defo, Harold Wang

5/13/16

Changes from dictdb.py: Binary Search Trees

- Implemented for numerical fields
  - Require declaration of type in schema
- Increases speed of select() function
- Modifications to standard BST:
  - Nodes keep a *list* of primary keys (for repeated values)
  - Deletion from tree requires value as well as key - deletes a single item from list rather than entire node
  - Added collect(key, op) function which traverses tree and collects contents of nodes which satisfy comparison operation

Changes from dictdb.py: Vantage points

- Incorporated into DB class via a new method add_vp(pk=None)
  - Allows one to specify pk of object to be made a vantage point
  - Chooses a random object if pk=None
  - Calculates distances to all existing objects and adds to tree
- Modified insert_ts to calculate distance to existing vps
- No need to specify vp distances in schema

Changes from dictdb.py: Similarity search

- Method simsearch(TimeSeries ts) finds the object in the database which is most similar, based on cross-correlation
- Identifies closest vantage point, finds all within 2*d, runs distance calculation on those and returns the pk of the closest object

Changes from dictdb.py: Persistence

- Added options to __init__:
  - load : bool, whether to load from existing db
  - dbname : str, filename to load/save
  - overwrite : bool, whether to overwrite existing file

Changes from dictdb.py: Persistence

- Storage method: very straightforward
  - timeseries: deconstruct into 2xN numpy array, save as .npy file dbname_ts/pk_ts.npy
  - metadata: save db log into dbname file
    - Each modification appends a line to the file of form pk:fieldname:val
    - DB can be reconstructed upon reading file
    - Only allow types int, float, bool, and str
    - Forbid ':' character
- Deletion: add line pk:DELETE:val
  - DELETE is forbidden fieldname