
synthit Documentation

Release 1.0.0

Jacob Reinhold

Dec 04, 2018

Contents:

1	Executables	3
2	Synthesis	7
3	Utilities	9
4	Indices and tables	11
	Python Module Index	13

synthit is an (alpha version) python package containing a variety of routines for MR image synthesis. This package was developed by [Jacob Reinhold](#) and the other students and researchers of [IACL](#).

1.1 Synthesis Trainer

train a patch-based regressor for MR image synthesis

```
usage: synth-train [-h] -s SOURCE_DIR [SOURCE_DIR ...] -t TARGET_DIR
                  [-o OUTPUT] [-m MASK_DIR] [-r {rf,xg,pr,mlr,mlp}] [-v]
                  [--cross-validate] [-ps PATCH_SIZE] [-fp] [-ns N_SAMPLES]
                  [-cr CTX_RADIUS [CTX_RADIUS ...]] [-th THRESHOLD]
                  [-pd POLY_DEG] [--mean] [--use-xyz] [-n N_JOBS]
                  [-msl MIN_SAMP_LEAF] [-nt N_TREES] [-mf MAX_FEATURES]
                  [-md MAX_DEPTH] [-nr NUM_RESTARTS] [-mi MAX_ITERATIONS]
                  [-hls HIDDEN_LAYER_SIZES [HIDDEN_LAYER_SIZES ...]]
                  [-rs RANDOM_SEED]
```

1.1.1 Required

- s, --source-dir** path to directory with domain images (multiple paths can be provided for multi-modal synthesis, put T1-w images first if they are not skull-stripped)
- t, --target-dir** path to directory with target images

1.1.2 Options

- o, --output** path to output the trained regressor
- m, --mask-dir** optional directory of brain masks for images
- r, --regr-type** Possible choices: rf, xg, pr, mlr, mlp
specify type of regressor to use
Default: “rf”

-v, --verbosity	increase output verbosity (e.g., -vv is more than -v) Default: 0
--cross-validate	do leave one out cross-validation on the provided dataset (e.g., if 5 datasets are provided, then 5 models are trained where all the data are used except one). Default: False

1.1.3 Synthesis Options

-ps, --patch-size	patch size extracted for regression [Default=3] Default: 3
-fp, --full-patch	use the full patch in regression vs a reduced size patch [Default=False] Default: False
-ns, --n-samples	use randomly sampled (with replacement) <i>n_samples</i> voxels for training regressor (None uses all voxels) [Default=None]
-cr, --ctx-radius	context radii to use when extracting patches [Default=(3,5,7)] Default: (3, 5, 7)
-th, --threshold	threshold for foreground and background (above is foreground) [Default=0] Default: 0
-pd, --poly-deg	degree of polynomial features derived from extracted patches (None means do not use polynomial features) [Default=None]
--mean	learn to take the mean value of input patch to the mean value of output patches Default: False
--use-xyz	use the x,y,z coordinates of voxels as features Default: False

1.1.4 Regressor Options

-n, --n-jobs	number of processors to use (-1 is all processors) [Default=-1] Default: -1
-msl, --min-samp-leaf	minimum number of samples in each leaf in rf (see min_samples_leaf) [Default=5] Default: 5
-nt, --n-trees	number of trees in rf or xg (see n_estimators) [Default=60] Default: 60
-mf, --max-features	proportion of features to use in rf (see max_features) [Default=1/3] Default: 0.3333333333333333
-md, --max-depth	maximum tree depth in rf or xg [Default=None (3 for xg)]
-nr, --num-restarts	number of restarts for mlr (since finds local optimum) [Default=8] Default: 8

- mi, --max-iterations** maximum number of iterations for mlr and mlp [Default=20]
Default: 20
- hls, --hidden-layer-sizes** number of neurons in each hidden layer for mlp [Default=(100,)]
Default: (100,)
- rs, --random-seed** set random seed for reproducibility [Default=0]
Default: 0

1.2 Synthesis Predictor

synthesize MR images via patch-based regression

```
usage: synth-predict [-h] -s SOURCE_DIR [SOURCE_DIR ...] -t TRAINED_MODEL
                    [-o OUTPUT_DIR] [-m MASK_DIR] [-v] [--cross-validate]
```

1.2.1 Required

- s, --source-dir** path to directory with domain images
- t, --trained-model** path to the trained model (.pkl)

1.2.2 Options

- o, --output-dir** path to output the synthesized images
- m, --mask-dir** optional directory of brain masks for images
- v, --verbosity** increase output verbosity (e.g., -vv is more than -v)
Default: 0
- cross-validate** do leave one out cross-validation on the provided dataset (e.g., if 5 datasets are provided, then 5 models are trained where all the data are used except one).
Default: False

2.1 Patch Based Synthesis

```
class synthit.PatchSynth(regr, patch_size=3, n_samples=100000.0, context_radius=(3, 5, 7),  
                        threshold=None, poly_deg=None, mean=False, full_patch=False, flat-  
                        ten=True, use_xyz=False)
```

provides the model for training and synthesizing MR neuro images via patch-based methods

Parameters

- **regr** (*sklearn model*) – an instantiated model class (e.g., `sklearn.ensemble.forest.RandomForestRegressor`) needs to have a fit and predict public method
- **patch_size** (*int*) – size of patch to use (patch_size x patch_size x patch_size)
- **n_samples** (*int*) – number of patches (i.e., samples) to use from each image
- **context_radius** (*tuple*) – tuple containing number of voxels away to get context from (e.g., (3,5) means get context values at 3 voxels and 5 voxels away from the patch center)
- **threshold** (*float*) – threshold that separated background and foreground (foreground greater than threshold) if None, then use the image mean as the threshold
- **poly_deg** (*int*) – degree of polynomial features to generate from patch samples
- **mean** (*bool*) – use the mean of the patch instead of the patch values
- **full_patch** (*bool*) – use a full patch instead of the 6-nearest neighbors
- **flatten** (*bool*) – flatten the target voxel intensities (needed in some types of regressors)
- **use_xyz** (*bool*) – use x,y,z coordinates as features

```
extract_patches_predict (source, mask=None)  
    extract patches and get indices for prediction/synthesis
```

extract_patches_train (*source, target, mask=None*)
get patches and corresponding target voxel intensity values for training

fit (*source, target, mask=None*)
train the model for synthesis given a set of source and target images

static image_list (*img_dir*)
convenience function to get a list of images in ANTsImage format

predict (*source, mask=None*)
synthesize/predict an image from a source (input) image

2.2 Mixture of Linear Regressors

class `synthit.LinearRegressionMixture` (*num_components, max_iterations=20, threshold=1e-10, num_restarts=1, num_workers=1, k=5, seed=1*)
Mixture of linear regressors model

3.1 Input/Output Functions

`synthit.util.io`

handle io operations for the synthit package

Author: Jacob Reinhold (jacob.reinhold@jhu.edu)

Created on: Jun 20, 2018

`synthit.util.io.split_filename(filepath)`

split a filepath into the directory, base, and extension

`synthit.util.io.glob_nii(path)`

grab all nifti files in a directory and sort them for consistency

3.2 Patch Extraction

`synthit.util.patches`

handle the extraction of patches and reconstruction from patches of 3d arrays (namely, 3d MR images)

Author: Jacob Reinhold (jacob.reinhold@jhu.edu)

Created on: Jun 20, 2018

`synthit.util.patches.extract_patches(data, idxs=None, patch_size=3, min_val=0, ctx_radius=(3, 5, 7), economy_patch=True, mean=False)`

extract patches (with or without context) from a 3D image

if the user sets `patch_size` equal to 1 and `ctx_radius` equal to (0,), then all (and only) the values of `idxs` will be directly pulled from the numpy array (very fast!)

if the user sets `patch_size` equal to 0, then the central pixel will not be extracted, only the six nearest neighbors and context features will be extracted

Parameters

- **data** (*np.ndarray*) – 3d data
- **idxs** (*tuple*) – tuple of np.ndarrays corresponding to indices (e.g., output from np.where)
- **patch_size** (*int*) – patch size (this cubed), must be odd
- **min_val** (*float*) – minimum value of extracted indices if idxs not provided
- **ctx_radius** (*tuple*) – tuple of positive integers greater than patch size ((0) if no context desired)
- **economy_patch** (*bool*) – return ‘economy-sized’ patches (not full patches, just the center and the 6-nearest neighbor voxels)
- **mean** (*bool*) – return mean value of patches

Returns array of patches

Return type patches (np.ndarray)

CHAPTER 4

Indices and tables

- `genindex`
- `modindex`

S

`synthit`, [7](#)
`synthit.util.io`, [9](#)
`synthit.util.patches`, [9](#)

E

`extract_patches()` (in module `synthit.util.patches`), 9
`extract_patches_predict()` (`synthit.PatchSynth` method), 7
`extract_patches_train()` (`synthit.PatchSynth` method), 7

F

`fit()` (`synthit.PatchSynth` method), 8

G

`glob_nii()` (in module `synthit.util.io`), 9

I

`image_list()` (`synthit.PatchSynth` static method), 8

L

`LinearRegressionMixture` (class in `synthit`), 8

P

`PatchSynth` (class in `synthit`), 7
`predict()` (`synthit.PatchSynth` method), 8

S

`split_filename()` (in module `synthit.util.io`), 9
`synthit` (module), 7
`synthit.util.io` (module), 9
`synthit.util.patches` (module), 9