Projet Codes

Ankita Naik
Indian Institute of Technology Madras
naik.ankita1210@gmail.com

May 2017

1 Data Creation

1.1 Packages available for 2-point correlations:

PyMKS [1] python package was used for 2-point correlations of 2-D microstructures. The code for the same is in file $Part0\_Checkerboard\_point\_stats.py$. Checkerboard microstructure spatial correlations was calculated using PyMKS but the 2-point correlations of 3-D microstructure gives an erroneous value at (0,0,0) for auto-correlations [2].

Matlab Spatial Correlations Toolbox [3] MATLAB package was used for 2-point correlations of 3-D microstructures.

1.2 Arranging data for classification and prediction

1. Calculate 2-point correlation (only auto-correlations used) of 3-D microstructure voxel (micros folder)
2. $63X63X63$ matrix converted into $1X250047$ matrix
3. Final dataframe has 400 rows for 400 samples. Thus, $400X250047$ is the dimension of final dataframe. Similar, restructuring done for microstructure as well using code $Part1\_microstructure.py$

Code $Matlab\_training\_dataset.m$ used for constructing the $400X250047$ dataset of 2-point correlation representation which is saved in file names $correlations\_final.csv$.

1.3 PCA on 2-point correlations

PCA is run to find 2-point correlations giving $[2, 3, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100]$ principal components. Code for the same is in file $Part2\_PCA\_correlations.py$.

[1] http://pymks.org/en/latest/rst/README.html

[2] (0,0,0) of auto-correlations should give the volume fraction which is approximately 0.25 for all the 8 material classes but that isn't the case for PyMKS package

[3] http://dx.doi.org/10.5281/zenodo.31329

1.4 Inter and Intra class PCA

According to the discussion of canonical form of PCA weights for microstructure representation using Mahalanobis distance measure carried out by Niezgoda in section on *Relational Statistics in Microstructure Space* the code for constructing the 2-D and 3-D representations in python is in file $Part3\_MD.py$. This code takes care of the PCA transformation of the original dataset and then the Mahalanobis representation.

1.5 Classification using SVM

Code $Part3\_tune\_multiple.py$ is used for calculating accuracy of SVM models on training dataset having varied number of principle components. The initial datasets (having varied number of PCs) required for training is present in $Corr\_datasets\_final\_rect\_classes$.

1.6 Property-Structure Linkages

1. Calculate the eigen value array ($Part4\_Eigenvalues\_weights.py$)
2. Structure the prediction values ($Part4\_Python - Stress.py$)
3. WLS, linear regression and svm using linear kernel for prediction of C_{11} and ys (Codes : All Part5 codes)

Code $Part6\_Dendogram.py$ constructs the dendogram for centroids of all 8 classes. $Corr\_datasets\_final\_rect\_classes$ has the PCA representations for different number of PCs.

Two more codes $Hull\_3D.m$ and $Plot\_correlations.m$ were used to make the 3-D hulls for all the 8 classes.
All datasets required for various codes are present in folder $Datasets$.