This package consists of the source codes of s2C3S, which was developed by Dr. Yunlong Mi for a semi-supervised concept-cognitive computing system. One can use them freely (for academic purpose only) at your own risk. For other purposes, please contact with Dr. Yunlong Mi directly.

Quickstart for s2C3S (work on Eclipse workstation):
JDK environment: jdk 1.8 or above.

Data format: The Data of the CSV format file looks like the following:
29.7037,21.3278,1.8359,0
30.4719,5.5551,36.8715,0
33.2494,-3.937,52.1075,0
…

Step 1: Set the file path in ParametersUtil.java

```
/** A demo example, a training and testing from Training 10% of SZCXR. */
public static String train_path = "./data/train[1].csv";
public static String test_path = "./data/test[1].csv";
```

Step 2: Set some related parameters in ParametersUtil.java, such as the parameters for Training 10% of SZCXR as follows:

```
/** Fixed Lambda(i): concept falling space, the $\lambda $ value and P */
public static int lambda  = 8; // it represents lambda = 8/10
public static double P = 1;// concept falling, P=1 or P=0.1
/** Epsion cocnept */
public static double e=0.9;// it means the similarity of two samples, CosineDistance [0,1].
/** MaxSize: The size of concept spaces for each class. */
public static int conceptSZ =100;
/** Chunk size: The size of each data chunk. */
public static int C =10;
```

```
/** Fixed Alpha: The concept similarity threshold. */
public static double distF = 0.5;
/** Fixed Delta: The range of the local $\alpha$-concept neighborhood. default radius=5. */
public static int radius = 5;
```

Step 3: run s2C3S in runMethod.java

```
/** Load datasets */
long s1 = System.currentTimeMillis();
Vector<Object> train_vec = LoadDataUtil
                .loadData(ParametersUtil.train_path.replace("indexNum", String.valueOf(index)));
Vector<Object> test_vec = LoadDataUtil
                .loadData2(ParametersUtil.test_path.replace("indexNum", String.valueOf(index)));
long e1 = System.currentTimeMillis();
System.err.println("Load dataset:  " + (e1 - s1) + "(ms)");

/** Instantiation system, 实例化系统 */
s2C3S C3S = new s2C3S(train_vec, train_vec);
/** Initial system, 系统初始化 */
long s2 = System.currentTimeMillis();
C3S.initialS();
long e2 = System.currentTimeMillis();
System.err.println("Initial system:  " + (e2 - s2) + "(ms)");

/** Learning for system, 系统学习 */
```

```
long s3 = System.currentTimeMillis();
C3S.trainS();
long e3 = System.currentTimeMillis();
System.err.println("Training system：  " + (e3 - s3) + "(ms)");

/** Evaluating and updating system, 系统动态更新与评估 */
long s4 = System.currentTimeMillis();
C3S.evaluateS(test_vec, index);
long e4 = System.currentTimeMillis();
System.err.println("Evaluating system：  " + (e4 - s4) + "(ms)");
```

NOTE: Please cite our work if you use these source codes in any way.
Yunlong Mi, et al. A semi-supervised concept-cognitive computing system for dynamic classification decision making with limited feedback information, EJOR.