BT-Gen: A Scalable Big Trajectory Generator

CODE Manual

Data science lab

2019

Table of Contents

[1 General Information 2](#_Toc25857320)

[1.1 System Overview 2](#_Toc25857321)

[1.2 Organization of the Manual 2](#_Toc25857322)

[2 Project Packages 2](#_Toc25857323)

[2.1 com.algorithm 2](#_Toc25857324)

Table of Figures

# General Information

## System Overview

BT-Gen is a trajectory generator that provides large scale generation. This application can read Digital Building Information (DBI) and use for data generation. BT-Gen also provide data visualization for validation purpose.

## Organization of the Manual

This user manual describes each packages and related class placed in projects. Each methods and attributes will explain clearly. This manual also includes some bugs that occurs during development phase.

# Project Packages

## com.algorithm

|  |  |
| --- | --- |
| Class name: | Algorithm |
| Class type: | Abstract Class |
| Description: | This class used to select positioning algorithm for RSSI data. |
| Class attributes: | inputPath: String  outputPath: String  props: Properties |
| Class methods: | public Algorithm(String proPropPath, String txtInputPath, String txtOutputPath)  public String getInputPath()  public void setInputPath(String inputPath)  public String getOutputPath()  public void setOutputPath(String outputPath)  public Properties getProps()  public void setProps(Properties props)  public abstract int calAlgorithmForAll(ExecutorService threadPool) |

|  |  |
| --- | --- |
| Class name: | AlgorithmForOne |
| Class type: | Abstract Class implements Runnable |
| Description: | This class used to embed positioning algorithm into each moving object. |
| Class attributes: | inputFileName: String  outputFileName: String  comments: String |
| Class methods: | public AlgorithmForOne()  public AlgorithmForOne(String inputFileName, String outputFileName)  public ArrayList<String> calAlgorithmForOneObj(String fileName)  public ArrayList<String> readRecord(String fileName)  public abstract ArrayList<String> calculateTrajectory(ArrayList<String> records);  protected void exportResult(ArrayList<String> result, String outputName)  public String getInputFileName()  public void setInputFileName(String inputFileName)  public String getOutputFileName()  public void setOutputFileName(String outputFileName)  public String getComments()  public void setComments(String comments) |

|  |  |
| --- | --- |
| Class name: | ClassificationTypeEnum |
| Class type: | enum |
| Description: | This class used to enumerate classification algorithm. |
| Class attributes: | WKNN(0, "algorithm\_weighted\_kNN"),  NB(1, "algorithm\_naive\_bayes"),  SVM(2, "algorithm\_svm")  classificationType: int  classificationName: String |
| Class methods: | ClassificationTypeEnum(int classificationType, String classificationName)  public int getClassificationType()  public void setClassificationType(int classificationType)  public String getClassificationName()  public void setClassificationName(String classificationName) |

|  |  |
| --- | --- |
| Class name: | DistanceMeasureTypeEnum |
| Class type: | enum |
| Description: | This class used to enumerate distance measurement algorithm. |
| Class attributes: | MAN(0, "distance\_measure\_manhattan"),  EUC(1, "distance\_measure\_euclidean")  distanceMeasureType: int  distanceMeasureName: String |
| Class methods: | DistanceMeasureTypeEnum(int distanceMeasureType, String distanceMeasureName)  public int getDistanceMeasureType()  public void setDistanceMeasureType(int distanceMeasureType)  public String getDistanceMeasureName()  public void setDistanceMeasureName(String distanceMeasureName) |

|  |  |
| --- | --- |
| Class name: | FPT |
| Class type: | Extends Algorithm class |
| Description: | This class is implementation of fingerprinting algorithm. |
| Class attributes: | outputFileName: String  seedSize: int  radioMap\_path: String  classifier: Classifier  grid\_size: double  margin\_left: double  margin\_right: double  maximum\_radiomap\_records\_for\_each: int  minimum\_radiomap\_records\_for\_each: int  k: int  positioningAlgorithmType: int  distanceMeasureType: int  dm: DistanceMeasure  outputFormatType: int  maximumSampleNumber: int  logarithmic: boolean  radiomap: Dataset |
| Class methods: | public FPT(String proPropPath, String txtInputPath, String txtOutputPath)  private Dataset buildRadioMap(double gs, int maxrr, int minrr, double ml, double mr)  private void exportRadioMap(Dataset data, String outputName)  private RadioMapRecord generateRadioMapRecord(ReferencePoint rp, Floor floor)  private void buildReferencePoints(double gs, double ml, double mr)  private List<ReferencePoint> generateReferencePointsForOneFloor(Floor floor, MBR2D mbr2d, double gs, double ml,double mr)  public int calAlgorithmForAll(ExecutorService threadPool)  public static String getRadioMap\_path()  public static void setRadioMap\_path(String radioMap\_path)  public Classifier getClassifier()  public void setClassifier(Classifier classifier)  public double getGrid\_size()  public void setGrid\_size(double grid\_size)  public double getMargin\_left()  public void setMargin\_left(double margin\_left)  public double getMargin\_right()  public void setMargin\_right(double margin\_right)  public int getMaximum\_radiomap\_records\_for\_each()  public void setMaximum\_radiomap\_records\_for\_each(int maximum\_radiomap\_records\_for\_each)  public int getMinimum\_radiomap\_records\_for\_each()  public void setMinimum\_radiomap\_records\_for\_each(int minimum\_radiomap\_records\_for\_each)  public int getK()  public void setK(int k)  public int getPositioningAlgorithmType()  public void setPositioningAlgorithmType(int positioningAlgorithmType)  public Dataset getRadiomap()  public void setRadiomap(Dataset radiomap)  public int getOutputFormatType()  public void setOutputFormatType(int outputFormatType)  public int getDistanceMeasureType()  public void setDistanceMeasureType(int distanceMeasureType)  public boolean isLogarithmic()  public void setLogarithmic(boolean logarithmic)  public int getMaximumSampleNumber()  public void setMaximumSampleNumber(int maximumSampleNumber) |

|  |  |
| --- | --- |
| Class name: | FPTForOne |
| Class type: | Extends AlgorithmForOne class |
| Description: | This class is implementation of fingerprinting algorithm for each moving object. |
| Class attributes: | classifier: Classifier  k: int  positioningAlgorithmType: int  outputFormatType: int  maximumSampleNumber: int |
| Class methods: | public FPTForOne(String inputFileName, String outputFileName)  public FPTForOne(String inputFileName, String outputFileName, int positioningAlgorithmType, Classifier classifier, int k, int outputFormatType, int maximumSampleNumber)  public void run()  public ArrayList<String> calculateTrajectory(ArrayList<String> records)  private IndoorLocation calculateLocationDeterWKNN(ArrayList<RSSIPackage> packs)  private IndoorLocation calculateLocationDeter(ArrayList<RSSIPackage> packs)  private SortedMap<Double, IndoorLocation> calculateLocationProb(ArrayList<RSSIPackage> packs)  public Classifier getClassifier()  public void setClassifier(Classifier classifier)  public int getOutputFormatType()  public void setOutputFormatType(int outputFormatType)  public int getPositioningAlgorithmType()  public void setPositioningAlgorithmType(int positioningAlgorithmType)  public int getMaximumSampleNumber()  public void setMaximumSampleNumber(int maximumSampleNumber) |

|  |  |
| --- | --- |
| Class name: | IndoorLocation |
| Class type: | Public class |
| Description: | This class is blueprint for indoor location object. |
| Class attributes: | x: double  y: double  floor: Floor  partition: Partition |
| Class methods: | public IndoorLocation()  public IndoorLocation(Point2D.Double point)  public IndoorLocation(Point2D.Double point, Floor floor)  public IndoorLocation(double x, double y)  public IndoorLocation(double x, double y, Floor floor)  public double getX()  public void setX(double x)  public double getY()  public void setY(double y)  public Floor getFloor()  public void setFloor(Floor floor)  public Partition getPartition()  public void setPartition(Partition partition)  public String toString() |