

CDC04 TeXDoclet

Christopher McKee

10 May 2017

Contents

Class Hierarchy	2
1 Package cdc04	3
1.1 Class StateAnalyser	3
1.1.1 Declaration	3
1.1.2 Constructor summary	3
1.1.3 Method summary	3
1.1.4 Constructors	3
1.1.5 Methods	4
2 Package weka.classifiers.meta	5
2.1 Class ProjectClassifier	5
2.1.1 Declaration	5
2.1.2 Constructor summary	6
2.1.3 Method summary	6
2.1.4 Constructors	6
2.1.5 Methods	7
2.1.6 Members inherited from class SingleClassifierEnhancer	13
2.1.7 Members inherited from class AbstractClassifier	13

Class Hierarchy

Classes

- `java.lang.Object`
 - `cdc04.StateAnalyser` (in [1.1](#), page [3](#))
 - `weka.classifiers.AbstractClassifier`
 - `weka.classifiers.SingleClassifierEnhancer`
 - `weka.classifiers.meta.ProjectClassifier` (in [2.1](#), page [5](#))

Chapter 1

Package cdc04

<i>Package Contents</i>	<i>Page</i>
Classes	
StateAnalyser	3
Class written to record a series of Weka instances, and to determine differences between them.	

1.1 Class StateAnalyser

Class written to record a series of Weka instances, and to determine differences between them.

1.1.1 Declaration

```
public class StateAnalyser
    extends java.lang.Object implements java.io.Serializable
```

1.1.2 Constructor summary

StateAnalyser() Constructs a new instance with no recorded instances

1.1.3 Method summary

addInstances(Instances) Adds an Instances object to be tracked
getNumberDifferences() Returns an integer representation of the number of rows which have changed between the previous two iterations of the classifier A negative return values shows that the number of differences could not be calculated, since there are not at least two iterations present.
getNumberIterations() Returns the number of instances which are currently contained in the tracker for analysis

1.1.4 Constructors

- **StateAnalyser**

```
public StateAnalyser()
```

- **Description**

Constructs a new instance with no recorded instances

1.1.5 Methods

- **addInstances**

```
public void addInstances(weka.core.Instances toAdd)
```

- **Description**

Adds an Instances object to be tracked

- **Parameters**

* toAdd – Instances object to be added

- **getNumberDifferences**

```
public int getNumberDifferences()
```

- **Description**

Returns an integer representation of the number of rows which have changed between the previous two iterations of the classifier. A negative return value shows that the number of differences could not be calculated, since there are not at least two iterations present.

- **Returns** – int representing number of differences between previous two iterations

- **getNumberIterations**

```
public int getNumberIterations()
```

- **Description**

Returns the number of instances which are currently contained in the tracker for analysis

- **Returns** – the current number of recorded instances

Chapter 2

Package weka.classifiers.meta

Package Contents

Page

Classes

ProjectClassifier	5
A classifier which is iteratively trained, imputing missing values into copies of the training data until no further change is observed.	

2.1 Class ProjectClassifier

A classifier which is iteratively trained, imputing missing values into copies of the training data until no further change is observed. Builds one learner per attribute, and therefore can take quite a while to run. Valid options are:

-W classifier

Full path to the target classifier to use, e.g. `weka.classifiers.trees.J48`

-S

Defines whether or not the classifier will impute a value for the class attribute as it trains.

-R

If set, the classifiers will be trained with any missing arguments filled in by random data. The classifier will then only iterate once.

-M integer

Sets the maximum number of times that a particular classifier will iterate before determining that it is trained.

Options after `-` are passed to the currently selected classifier.

2.1.1 Declaration

```
public class ProjectClassifier
```

extends `weka.classifiers.SingleClassifierEnhancer` **implements** `weka.classifiers.IterativeClassifier`

2.1.2 Constructor summary

ProjectClassifier() Constructor

2.1.3 Method summary

buildClassifier(Instances) Builds a set of classifiers based on the training data.
classifierOptionsTipText() Tip text to be displayed in the GUI for this property
classifyInstance(Instance) Classifies an instance.
defaultClassifierString() String describing default classifier.
distributionForInstance(Instance) Returns class probabilities for an instance.
done() Method called when iteration has terminated.
getCapabilities() Returns default capabilities of the classifier.
getClassifierOptions() Gets classifier options
getMaxIterations() Get the value of `m_MaxIterations`
getNumHiddenVariables()
getOptions() Gets the current settings of the Classifier.
getRandomData() Get the value of `m_RandomData`
getSupervised() Get the value of `m_Supervised`
globalInfo() Global information about the class
initializeClassifier(Instances) Makes copies of the training data which can be mutated, and initialise the array of Classifier objects
listOptions() Returns an enumeration describing the available options.
main(String[]) Main method for testing this class.
maxIterationsTipText() Tip text to be displayed in the GUI for this property
next() Retrains each of the classifiers, then attempts to impute missing data in a copy of the training data.
numHiddenVariablesTipText()
randomDataTipText() Tip text to be displayed in the GUI for this property
setClassifierOptions(String[]) Sets classifier options
setMaxIterations(int) Set the value of `m_MaxIterations`.
setNumHiddenVariables(int)
setOptions(String[]) Parses a given list of options.
setRandomData(boolean) Set the value of `m_RandomData`
setSupervised(boolean) Set the value of `m_Supervised`
supervisedTipText() Tip text to be displayed in the GUI for this property

2.1.4 Constructors

- **ProjectClassifier**

public ProjectClassifier()

- **Description**
 Constructor

2.1.5 Methods

- **buildClassifier**

```
public void buildClassifier(weka.core.Instances instances)
    throws java.lang.Exception
```

- **Description**

Builds a set of classifiers based on the training data. These are iteratively trained on copies of the data.

- **Parameters**

- * **instances** – the Instances object which comprises the training data

- **Throws**

- * **java.lang.Exception** – exception thrown is raised to a Weka error handler

- **classifierOptionsTipText**

```
public java.lang.String classifierOptionsTipText()
```

- **Description**

Tip text to be displayed in the GUI for this property

- **Returns** – tip text to be displayed in the GUI

- **classifyInstance**

```
public double classifyInstance(weka.core.Instance instance)
    throws java.lang.Exception
```

- **Description**

Classifies an instance.

- **Parameters**

- * **instance** – the instance to classify

- **Returns** – the classification for the instance

- **Throws**

- * **java.lang.Exception** – if instance can't be classified successfully

- **defaultClassifierString**

```
protected java.lang.String defaultClassifierString()
```

- **Description**

String describing default classifier.

- **distributionForInstance**

```
public double [] distributionForInstance(weka.core.Instance  
instance) throws java.lang.Exception
```

- **Description**

Returns class probabilities for an instance.

- **Parameters**

- * **instance** – the instance to calculate the class probabilities for

- **Returns** – the class probabilities

- **Throws**

- * **java.lang.Exception** – if distribution can't be computed successfully

- **done**

```
public void done() throws java.lang.Exception
```

- **Description**

Method called when iteration has terminated. Imputes class values if `m_Supervised` is set.

- **getCapabilities**

```
public weka.core.Capabilities getCapabilities()
```

- **Description**

Returns default capabilities of the classifier.

- **Returns** – the capabilities of this classifier

- **getClassifierOptions**

```
public java.lang.String [] getClassifierOptions()
```

- **Description**

Gets classifier options

- **Returns** – array of String objects to be passed to each classifier

- **getMaxIterations**

```
public int getMaxIterations()
```

- **Description**

Get the value of m_MaxIterations

- **Returns** – value of m_MaxIterations

- **getNumHiddenVariables**

```
public int getNumHiddenVariables()
```

- **getOptions**

```
public java.lang.String [] getOptions()
```

- **Description**

Gets the current settings of the Classifier.

- **Returns** – an array of strings suitable for passing to setOptions

- **getRandomData**

```
public boolean getRandomData()
```

- **Description**

Get the value of m_RandomData

- **Returns** – value of m_RandomData

- **getSupervised**

```
public boolean getSupervised()
```

- **Description**

Get the value of m_Supervised

- **Returns** – value of m_Supervised

- **globalInfo**

```
public java.lang.String globalInfo()
```

- **Description**

Global information about the class

- **Returns** – information about the classifier which is displayed in the CLI/GUI

- **initializeClassifier**

```
public void initializeClassifier(weka.core.Instances instances)
    throws java.lang.Exception
```

– **Description**

Makes copies of the training data which can be mutated, and initialise the array of Classifier objects

– **Parameters**

* **instances** – the training data

• **listOptions**

```
public java.util Enumeration listOptions()
```

– **Description**

Returns an enumeration describing the available options.

– **Returns** – an enumeration of all the available options.

• **main**

```
public static void main(java.lang.String[] args)
```

– **Description**

Main method for testing this class.

– **Parameters**

* **args** – the options

• **maxIterationsTipText**

```
public java.lang.String maxIterationsTipText()
```

– **Description**

Tip text to be displayed in the GUI for this property

– **Returns** – tip text to be displayed in the GUI

• **next**

```
public boolean next() throws java.lang.Exception
```

– **Description**

Retrains each of the classifiers, then attempts to impute missing data in a copy of the training data. Does not iterate again if the results of current iteration match the results of the previous iteration, or the max number of iterations has been reached.

- **Returns** – true if another iteration should be performed, otherwise false.

- **numHiddenVariablesTipText**

```
public java.lang.String numHiddenVariablesTipText()
```

- **randomDataTipText**

```
public java.lang.String randomDataTipText()
```

- **Description**

Tip text to be displayed in the GUI for this property

- **Returns** – tip text to be displayed in the GUI

- **setClassifierOptions**

```
public void setClassifierOptions(java.lang.String []  
    classifierOptions)
```

- **Description**

Sets classifier options

- **Parameters**

* **classifierOptions** – array of String objects to be passed to each classifier

- **setMaxIterations**

```
public void setMaxIterations(int maxIterations)
```

- **Description**

Set the value of m_MaxIterations. Defaults to Integer.MAX_VALUE if value less than 0 is supplied.

- **Parameters**

* **maxIterations** – new value of m_MaxIterations

- **setNumHiddenVariables**

```
public void setNumHiddenVariables(int numHiddenVariables)
```

- **setOptions**

```
public void setOptions(java.lang.String [] options) throws java.  
    lang.Exception
```

- **Description**

Parses a given list of options. Valid options are:

-W classifier

Full path to the target classifier to use, e.g. weka.classifiers.trees.J48

-S

Defines whether or not the classifier will impute a value for the class attribute as it trains.

-R

If set, the classifiers will be trained with any missing arguments filled in by random data. The classifier will then only iterate once.

-M integer

Sets the maximum number of times that a particular classifier will iterate before determining that it is trained.

Options after – are passed to the currently selected classifier.

- **Parameters**

- * options – The list of options as an array of Strings

- **Throws**

- * java.lang.Exception – if an option is not supported

- **setRandomData**

```
public void setRandomData(boolean randomData)
```

- **Description**

Set the value of m_RandomData

- **Parameters**

- * randomData – new value of m_RandomData

- **setSupervised**

```
public void setSupervised(boolean supervised)
```

- **Description**

Set the value of m_Supervised

- **Parameters**

- * supervised – the new value of m_Supervised

- **supervisedTipText**

```
public java.lang.String supervisedTipText()
```

- **Description**

Tip text to be displayed in the GUI for this property

- **Returns** – tip text to be displayed in the GUI

2.1.6 Members inherited from class SingleClassifierEnhancer

weka.classifiers.SingleClassifierEnhancer

- public String classifierTipText()
- protected String defaultClassifierOptions()
- protected String defaultClassifierString()
- public Capabilities getCapabilities()
- public Classifier getClassifier()
- protected String getClassifierSpec()
- public String getOptions()
- public Enumeration listOptions()
- protected m_Classifier
- public void postExecution() throws java.lang.Exception
- public void preExecution() throws java.lang.Exception
- public void setClassifier(Classifier arg0)
- public void setOptions(java.lang.String[] arg0) throws java.lang.Exception

2.1.7 Members inherited from class AbstractClassifier

weka.classifiers.AbstractClassifier

- public static BATCH_SIZE_DEFAULT
- public String batchSizeTipText()
- public double classifyInstance(weka.core.Instance arg0) throws java.lang.Exception
- public String debugTipText()
- public double distributionForInstance(weka.core.Instance arg0) throws java.lang.Exception
- public double distributionsForInstances(weka.core.Instances arg0) throws java.lang.Exception
- public String doNotCheckCapabilitiesTipText()
- public static Classifier forName(java.lang.String arg0, java.lang.String[] arg1) throws java.lang.Exception
- public String getBatchSize()
- public Capabilities getCapabilities()
- public boolean getDebug()
- public boolean getDoNotCheckCapabilities()
- public int getNumDecimalPlaces()
- public String getOptions()
- public String getRevision()
- public boolean implementsMoreEfficientBatchPrediction()
- public Enumeration listOptions()
- protected m_BatchSize
- protected m_Debug
- protected m_DoNotCheckCapabilities
- protected m_numDecimalPlaces
- public static Classifier makeCopies(Classifier arg0, int arg1) throws java.lang.Exception
- public static Classifier makeCopy(Classifier arg0) throws java.lang.Exception
- public static NUM_DECIMAL_PLACES_DEFAULT
- public String numDecimalPlacesTipText()
- public void postExecution() throws java.lang.Exception
- public void preExecution() throws java.lang.Exception
- public void run(java.lang.Object arg0, java.lang.String[] arg1) throws java.lang.Exception
- public static void runClassifier(Classifier arg0, java.lang.String[] arg1)
- public void setBatchSize(java.lang.String arg0)
- public void setDebug(boolean arg0)
- public void setDoNotCheckCapabilities(boolean arg0)
- public void setNumDecimalPlaces(int arg0)
- public void setOptions(java.lang.String[] arg0) throws java.lang.Exception