

Dokumentation Sugeno Klassifizierer

Sven Meyer

9. März 2021

Inhaltsverzeichnis

1	Installation	2
2	Anwendung	2
3	Beispiele	3
3.1	Einführendes Beispiel	3
3.2	Verwendung von Hyperparametern	3
3.3	Wahl von verschiedenen Klassenlabels	3

Der Sugeno Klassifizierer ist ein Algorithmus für binäre Klassifizierungsprobleme und wurde im Kontext des Überwachten Lernens entworfen. Dieser ist aufgrund der Definition, welche auf dem Sugeno Integral aufbaut, besonders für Attribute mit ordinalskalierten Wertebereichen geeignet, die eine Monotonie im Sinne von „Größere Werte sind bessere Werte“ voraussetzen. Die Implementierung ist kompatibel zu der Bibliothek *scikit-learn* und kann daher in verschiedenen Algorithmen wie der Rastersuche, welche in dieser Bibliothek implementiert ist, eingesetzt werden.

1 Installation

Der Sugeno Klassifizierer ist in der Programmiersprache Python 3.8 implementiert und getestet worden. Eine korrekte Nutzung von älteren Versionen kann daher nicht garantiert werden. Die Installation erfolgt über den Befehl

```
pip install <path_to_setup.py>
```

wobei `< path_to_setup.py >` der Pfad zu dem Skript `setup.py` ist.

2 Schnittstelle

Um den Sugeno Klassifizierer in eigene Projekte einbinden zu können, muss die Klasse `SugenoClassifier` aus dem Modul `sugeno_classifier` und dem Paket `classifier` importiert werden. Die Gestaltung der Schnittstelle folgt dann dem Design von *scikit-learn* und umfasst drei Funktionen:

SugenoClassifier(maxitivity=None, margin=0):

Der Sugeno Klassifizierer kann durch einen Konstruktor instanziiert werden. An dieser Stelle werden auch die Hyperparameter initialisiert. Der Standardwert der Maxitivität ist `None`, so dass eine m -maxitive Kapazitätsfunktion berechnet wird, wobei m die Anzahl der Attribute in einem Datensatz beschreibt. Der Standardwert des Margins ist 0. Untersuchungen haben gezeigt, dass deutlich bessere Ergebnisse erzielt werden können, wenn ein Wert für den Margin gewählt wird, der echt größer als 0 ist.

fit(X, y):

Die Funktion `fit` berechnet die Feature Transformation, die Kapazitätsfunktion und den Schwellwert für eine Menge von Beispieldaten `X` und den zugehörigen Klassenlabels `y`. `X` ist eine Array-ähnliche Datenstruktur mit den Dimensionen `(n_samples, n_features)`, wobei `n_samples` die Anzahl der Beispieldaten und `n_features` die Anzahl der Attribute angibt. `y` ist eine Array-ähnliche Datenstruktur mit den Dimensionen `(n_samples,)`, wobei `n_samples` der Anzahl der Beispieldaten von `X` entsprechen muss. Die Klassen können dabei sowohl durch ganzzahlige numerische Werte als auch textuell beschrieben werden. Die jeweilige kleinere Klasse bezüglich der Relation `<` bzw. der lexikographischen Ordnung wird intern der negativen und die größere Klasse der positiven Klasse zugeordnet.

predict(X):

Die Funktion `predict` bestimmt die Klassen für eine Menge von Testdaten `X`. `X` ist eine Array-ähnliche Datenstruktur mit den Dimensionen `(n_samples, n_features)`, wobei `n_samples` die Anzahl der Beispieldaten und `n_features` die Anzahl der Attribute angibt. Der Wert von

`n_features` muss dem Wert entsprechen, mit dem die Parameter des Sugeno Klassifizierers berechnet wurden. Zusätzlich muss vor der Anwendung von `predict` die Funktion `fit` aufgerufen worden sein.

3 Beispiele

3.1 Einführendes Beispiel

In einem ersten Beispiel wird die Verwendung des Sugeno Klassifizierers anhand eines Trainingsdatensatzes mit zwei Instanzen mit jeweils drei Attributen gezeigt:

```
>>> from classifier.sugeno_classifier import SugenoClassifier
>>> X = [[1, 3, 2],
...      [2, 1, 3]]
>>> y = [0, 1]
>>> sc = SugenoClassifier()
>>> sc.fit(X, y)
```

Nachdem die Parameter berechnet wurden, kann die Instanz dazu verwendet werden, um die Klassen für andere Daten zu bestimmen.

```
>>> X = [[3, 2, 1],
...      [1, 2, 3]]
>>> sc.predict(X)
array([0, 1])
```

3.2 Verwendung von Hyperparametern

Die Hyperparameter können durch den Konstruktor gesetzt werden:

```
>>> from classifier.sugeno_classifier import SugenoClassifier
>>> X = [[1, 3, 2],
...      [2, 1, 3]]
>>> y = [0, 1]
>>> sc = SugenoClassifier(maxitivity=2, margin=0.01)
>>> sc.fit(X, y)
```

Wie in dem einführenden Beispiel können mit der Instanz die Klassen für andere Daten bestimmt werden.

```
>>> X = [[3, 2, 1],
...      [1, 2, 3]]
>>> sc.predict(X)
array([1, 1])
```

3.3 Wahl von verschiedenen Klassenlabels

Die Klassen selbst müssen nicht durch die Labels 0 bzw. 1, sondern können durch beliebige ganzzahlige bzw. textuelle Werte beschrieben werden.

Das erste Beispiel umfasst die Klassen 2 und 4. Die Klasse 2 wird intern der negativen Klasse, die Klasse 4 wird der positiven Klasse zugeordnet, da $2 < 4$ gilt.

```

>>> from classifier.sugeno_classifier import SugenoClassifier
>>> X = [[1, 3, 2],
...      [2, 1, 3]]
>>> y = [2, 4]
>>> sc = SugenoClassifier()
>>> sc.fit(X, y)
>>> X = [[3, 2, 1],
...      [1, 2, 3]]
>>> sc.predict(X)
array([2, 4])

```

Das zweite Beispiel umfasst die Klassen „one“ und „two“. Die erst genannte Klasse wird der negativen Klasse, die zweit genannte Klasse wird der positiven Klasse zugeordnet, da der Begriff „one“ lexikographisch vor „two“ steht.

```

>>> from classifier.sugeno_classifier import SugenoClassifier
>>> X = [[1, 3, 2],
...      [2, 1, 3]]
>>> y = ['one', 'two']
>>> sc = SugenoClassifier()
>>> sc.fit(X, y)
>>> X = [[3, 2, 1],
...      [1, 2, 3]]
>>> sc.predict(X)
array(['one', 'two'], dtype='<U3')

```