Klassifikation von Messwertsequenzen

11.12.2022
Intelligente Systeme – 3

Karen Sardarjan
Tariq Al-Arashi

Vorwort

Bei der Aufgabenstellung handelt es sich um die Implementierung eines Klassifikators.

So soll ein Klassifikator implementiert werden, um die Problemstellung einer Maschine zu lösen. Hier soll eine Güterproduzierende Maschine auf Funktionstüchtigkeit geprüft werden. Es handelt sich darum, ein Kostenproblem zu lösen, bei der die Maschine gelegentlich kaputt geht, wenn sie in einer *gefährlichen* Phase nicht abgeschaltet wird. So entstehen Kosten von 5 wenn die Maschine kaputt geht für den Produktionsausfall und der Reparatur. Sollte die Maschine in einer *normalen* Phase abgeschaltet werden, entstehen Kosten der Höhe 2 für den Produktionsausfall. Ziel ist es die Kosten so niedrig wie möglich zu halten, indem *gefährliche* Phasen entdeckt werden und die Maschine abgeschaltet wird damit Kosten 2 und nicht 5 entstehen. Somit wissen wir, dass wir überprüfen müssen, dass die Maschine sich in einer *normalen* oder *gefährlichen* Phase befindet. Für das Analysieren der Problemstellung, kriegen wir durch einen Sensor für die Überwachung der Maschine, Messwerte. Die Messwerte werden in einer Sequenz von 20 Messwerten geliefert. Diese sind in einer Markov kette erster Ordnung aufgebaut. Unterschieden werden sie, durch niedrige(n), mittlere(m) und hohe(h) Werte. Diese Sequenzen werden genutzt, um mittels Überwachungsmechanismus überprüfen zu können, ob wir uns derzeit in einer *normalen* oder einer *gefährlichen* Phase befinden. Die Abfolge der Messwerte wird hierbei als Indikator für die Phasen genutzt.

Wichtig zu wissen ist, dass davon ausgegangen wird, dass in 90% der Fälle wir uns in einer *normalen* und zu 10% in einer *gefährlichen* Phase befinden.

Da wir wissen, dass es sich bei den Sequenzen um eine Markov Kette erster Ordnung handelt, wissen wir, dass jeder Wert in der Sequenz nur vom unmittelbar vorhergehenden abhängt.

Konkrete Aufgabenstellung und Implementierung

Durch die Verwendung des Klassifikators, wird überprüft, ob eine bestimmte Sequenz darauf hinweist, ob es sich um eine *normale* oder *gefährliche* Phase handelt. Die damit verbundenen Kosten hängen davon ab, ob das was uns der Klassifikator liefert, *normal* richtig, *normal* falsch, *gefährlich* richtig oder *gefährlich* falsch ist. Bei der Tabelle 1 sind die Unterschiede und die Kosten zwischen den Werten visualisiert.

Tabelle 1

Werte	Beschreibung	Entstehende Kosten
Normal richtig	Klassifikator liefert richtigerweise den Wert normal bei einer *normalen* Phase	0
Normal falsch	Klassifikator liefert fälschlicherweise Den Wert *gefährlich*, obwohl es sich um eine *normal* Phase handelt	2
Gefährlich richtig	Klassifikator liefert richtigerweise den Wert *gefährlich* bei einer gefährlichen Phase	2
Gefährlich falsch	Klassifikator liefert fälschlicherweise den Wert *normal*, obwohl es sich um eine *gefährliche* Phase handelt	5

Beschreibung aller Phasen

Das Optimum wäre, wenn der Klassifikator erst alle *normal* richtigen Phasen als *normal* klassifiziert und nur die *gefährlich* richtigen als gefährlich klassifiziert. Dadurch wird nur bei *gefährlich* richtigen Phasen ein Kostenwert von 2 entstehen statt von 5, da wir die Maschine rechtzeitig ausschalten.

Um den Klassifikator zu implementieren, müssen wir die wichtigste Eigenschaft, die Markovkette der ersten Ordnung betrachten. Wichtig zu wissen bei der Markovkette erster Ordnung ist, dass die Wahrscheinlichkeit für einen Zustand z.B. n, m oder h, abhängt vom vorherigen Wert. Somit machen wir die Markov-Annahme.

Zunächst einmal werden Tabellen für die beiden Phasen *normal* und *gefährlich* erstellt. Diese Tabellen werden erstellt, mit den Sequenzen aus den Testdaten Train N für die *normalen* und Train G für die Sequenzen, die eine *gefährliche* Phase erzeugen. Um die Tabellen zu befüllen, nehmen wir die Sequenzen aus den Trainingsdaten. Um aus diesen Sequenzen jetzt die absoluten Häufigkeiten zu bekommen, zählen wir Paare aufeinanderfolgender Messwerte in der jeweiligen Sequenz. Beispielsweise bei einer Sequenz (nnhh) kommt nn, nh und hh 1-mal vor. Danach wird die relative Häufigkeit anhand der Werte der absoluten Häufigkeit berechnet. Um nun die relative Häufigkeit zu berechnen haben wir die jeweiligen Zeilensummen berechnet und die Werte der absoluten Häufigkeit durch die jeweilige Zeilensumme geteilt. So kommt man beispielsweise bei der Zeile [4,6,10] = 20 auf die Werte [0.2,0.3,0.5] für die relative Häufigkeit. Diese Tabellen werden für die Berechnung einer Wahrscheinlichkeit einer Sequenz genutzt. In

Tabelle 2 sind die relative Häufigkeit für die *normale* Phase und in Tabelle 3 die relative Häufigkeit für die *gefährliche* Phase visualisiert.

Tabelle 2

	n	m	h
n	0.22	0.46	0.31
m	0.40	0.20	0.41
h	0.30	0.64	0.06

relative Häufigkeit normalen Tabelle

Tabelle 3

	n	m	h
n	0.11	0.80	0.09
m	0.37	0.31	0.32
h	0.29	0.21	0.50

relative Häufigkeit gefährlich Tabelle

Da wir wissen das es sich um eine Markovkette erster Ordnung handelt, können wir die Sequenzen in die Abfolgen teilen, die Wahrscheinlichkeit dieser Abfolgen aus den Tabellen auslesen und dann die einzelnen Wahrscheinlichkeiten miteinander multiplizieren. Dies ist möglich da die Wahrscheinlichkeiten stochastisch unabhängig voneinander sind. Diese werden mit 1/3 multipliziert, da wir am Anfang nicht vorhersagen können, welcher Messwert vor dem ersten Messwert in der Sequenz vorkam.

Der Klassifikator funktioniert in einer Art und Weise, bei der er die Wahrscheinlichkeit einer gegebenen Sequenz anhand der beiden Tabellen berechnet. Da wir wissen, dass 90% der Fälle in einer *normalen* Phase und 10% in einer *gefährlichen* Phase enden, multiplizieren wir die berechnete Wahrscheinlichkeit, die wir mit der *normalen* Tabelle bekommen haben, mit 0.9 beziehungsweise mit 0.1 für *gefährliche* Phasen mit der berechneten Wahrscheinlichkeit der *gefährlichen* Tabelle. Sollte der Wert der *normalen* Phase größer als der der *gefährlichen* Phase sein, so handelt es sich um eine *normale* Phase. Somit wäre es eine *gefährliche* Phase, wenn der Wert jener größer wäre.

Für die Evaluation unseres Klassifikators haben wir die gegebene Datei mit den Evaluationsdaten genutzt, um zu testen, wie effizient unser Klassifikator klassifizieren kann. Strukturiert ist die Evaluationsdatei, mit 1000 Sequenzen. Vor jeder Sequenz ist die tatsächliche Phase der Sequenz bereits vorgegeben und jeweils in einer Zeile mit der jeweiligen Sequenz gespeichert, um diese abzugleichen mit unseren Ergebnissen. Die Datei wird eingelesen und für jede Zeile ein Tupel erstellt, bei der als erster Eintrag die tatsächliche Phase und die entsprechende Sequenz gespeichert wird. Diese Tupel werden gespeichert. Eine Instanz des Klassifikators wird erstellt, die Iterativ immer eine Sequenz übergeben bekommt und uns einen Wert zurückliefert. Diesen Wert vergleichen wir mit dem tatsächlichen Wert. Um nun zählen zu können, wie viele Sequenzen richtig und falsch klassifiziert worden sind, haben wir *normal* richtig, *normal* falsch, gefährlich richtig und *gefährlich* falsch jeweils aufgezählt. Sollte der Wert richtig klassifiziert worden sein, wird entweder *normal* richtig oder *gefährlich* richtig hochgezählt, ansonsten *normal* falsch oder *gefährlich* falsch. Nach der Evaluation kamen wir zur folgenden Tabelle 4, bei der die Anzahl der Vorkommnisse für die jeweiligen Möglichkeiten in einer Konfusionsmatrix visualisiert sind. Außerdem wird in der letzten Spalte die dadurch entstehenden Kosten bereits berechnet gezeigt.

Tabelle 4

Wert	Anzahl der Vorkommnisse	Kosten
Normal richtig	8750	0
Normal falsch	250	500
Gefährlich richtig	841	1682
Gefährlich falsch	159	795

Konfusionsmatrix

Um die entstehenden Kosten zu berechnen, multiplizieren wir die Werte mit den entsprechenden Kosten, visualisiert in Tabelle 1. So kommt man beispielsweise bei *Normal* falsch auf die Rechnung 250 * 2.

Sollte die Maschine nicht abgeschaltet werden, kommen wir zu einem Kostenwert von 5000. Durch die Verwendung unseres Klassifikators haben wir eine Kostenverbesserung von 40.46% erreicht und haben Endkosten von 2977. Um den Klassifikator zu evaluieren, nutzen wir 2 unterschiedliche Bewertungsfunktionen, die Precision und Recall. Die Precision ist eine Metrik, die die Menge der korrekten Positiven klassifizierten Phasen berechnet und uns einen Wert zwischen 0.0 und 1.0 zurückgibt. Berechnet wird die Precision mittels TruePositives / (TruePositives + FalsePositives). 1.0 Wäre hierbei eine perfekte Precision. Recall hingegen ist eine Metrik, die die Menge der korrekten Positiven klassifizierten Phasen von allen hätten, möglich richtig getroffenen korrekten Positiven Phasen zählt. Somit wird hier außerdem die Fehlerrate betrachtet und nicht nur die Menge die richtig klassifiziert wurde. Berechnet wird sie mittels TruePositives / (TruePositives + FalseNegatives). In Tabelle 5 sind für die jeweiligen Phasen die Precision und Recall dargestellt.

Tabelle 5

Typ	Wert
Precision normal	0.97
Precision gefährlich	0.84
Recall normal	0.84
Recall gefährlich	0.77

Berechnete Effizienz mit Klassifikator