

Exposé für die Bachelorarbeit: Analyse von Zeitreihen-Klassifikatoren mit timeXplain

Birk Leppich

April 2021

Inhalt

1	Einleitung	2
2	Grundlagen	3
2.1	TimeXplain Framework	3
2.2	Klassifikatoren	3
2.2.1	Dictionary-basiert	3
2.2.2	Shapelet-basiert	3
2.2.3	Intervall-basiert	4
2.2.4	Ensemble Klassifikatoren	4
2.2.5	Rocket	5
3	Forschungsfragen	5
3.1	Arbeitshypothese I: Korrelation zwischen Klassifikatoren gleicher Kategorie	5
3.2	Arbeitshypothese II: timeXplain interpretiert Klassifikationsentscheidungen korrekt mittels Impact Vektoren	5
4	Umsetzung	6
4.1	Datensätze	6
4.2	Technische Realisierung	6

1 Einleitung

Als Zeitreihe bezeichnet man eine (meist zeitlich) geordnete Sequenz von reellen Zahlen. Beispiele sind hierfür Niederschlagsdiagramme, Aktienkurse und Elektrokardiogramme. Bei der Zeitreihen-Klassifikation wird ein Modell trainiert, um für neue Zeitreihen Vorhersagen aus der Menge an Klassenlabels zu treffen. Beispielsweise kann der Verlauf einer Aktie einer Klasse zugeordnet werden, welche angibt, ob die Aktie verkauft oder gehalten werden sollte. Es gibt verschiedene Kategorien von Lernverfahren, die Zeitreihen repräsentieren und klassifizieren, exemplarisch Deeplearning-, Shapelet-, Dictionary-, Intervall- oder Ensemblebasiert [6]. Selbst bei Ansätzen der gleichen Kategorie gibt es Unterschiede innerhalb der Funktionsweise und verschiedene Klassifikatoren produzieren auf den gleichen Datensätzen unterschiedlich genaue Ergebnisse. Die Art und Weise, wie eine Klassifikationsentscheidung entsteht, ist dabei meist nicht nachvollziehbar bzw. interpretierbar. Es ist wichtig zu verstehen, wie Lernalgorithmen zu den Ergebnissen gelangen. Viele Zeitreihen-Klassifikatoren erzeugen hunderte bis zehntausende statistische Features, auf denen der Klassifikator trainiert wird. Welche Features verwendet werden, hängt von der Kategorie des Klassifikators ab. Bei Zeitreihen ist üblicherweise ein bestimmtes Teilintervall wichtig für die Zuweisung zu einer Klasse. Ein oder mehrere gute Klassifikatoren würden diesem Teilintervall ein hohes Gewicht zuweisen.

In dieser Bachelorarbeit untersuchen wir Paare von Klassifikatoren einer Kategorie mit Hilfe des timeXplain Frameworks [13]. TimeXplain versucht eine Klassifikationsentscheidung zu einer Zeitreihe zu erklären. Durch gezieltes Erzeugen von Daten mit kleinen Änderungen an den Teilsequenzen, in der Nachbarschaft von einer zu erklärenden Klassifikationsentscheidung, kann timeXplain die für die Entscheidung relevanten oder irrelevanten Teilsequenzen erkennen und visualisieren. In einer Case-Study mit ausgewählten Datensätzen soll zum einen untersucht werden, inwiefern sich Klassifikatoren der gleichen Kategorie ähneln, aber auch wie sich Klassifikatoren verschiedener Kategorien unterscheiden. Zum anderen soll mittels eines Vergleichs der dokumentierten sowie der von timeXplain errechneten Interpretationen der Datensätze bestimmt werden, wie präzise Klassifikatoren als auch das timeXplain Framework sind. Die zu verwendeten Datensätze wurden im Bezug zur Zeitreihen-Klassifikation bereits untersucht und die Interpretation ausreichend dokumentiert, um die vom Framework erzeugten Interpretationen zu vergleichen.

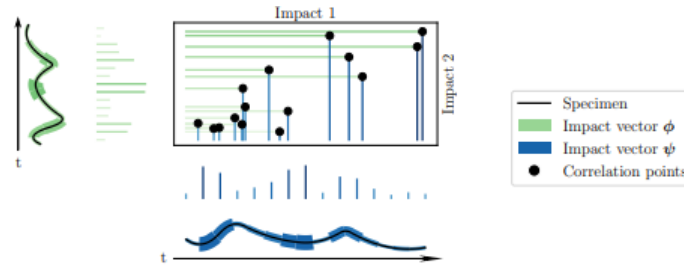


Figure 1: Beispielhafter Vergleich zweier Impact Vektoren [13]

Beide Impact Vektoren enthalten ähnlich hohe Gewichte für die Teilintervalle der Zeitreihe. Deshalb korrelieren sie stark miteinander. Die markierten Bereiche haben einen großen Einfluss auf die Entscheidung der Klassifikatoren.

2 Grundlagen

2.1 TimeXplain Framework

2.2 Klassifikatoren

2.2.1 Dictionary-basiert

Dictionary-basierte Ansätze reduzieren die Dimensionalität von Zeitreihen, indem die Daten in Histogramme repräsentativer Wörter umgewandelt werden und anschließend die Häufigkeitsverteilung der Wörter verglichen wird. Der Kernprozess von Dictionary-basierten Ansätzen besteht darin, Wörter durch das Verschieben eines sogenannten Sliding Windows zu bilden. Über dieses Fenster werden statistische Features (z.B. Mittelwert oder Fourier-Koeffizienten) aus Teilsequenzen berechnet und diskretisiert, um die Wörter zu bilden. Wir untersuchen den Algorithmus Bag of SFA Symbols (BOSS) [6], welcher eine diskrete Fourier-Transformation auf die Teilsequenzen anwendet und auf BOSS-Transformationen anschließend diese Features diskretisiert. Es werden mehrere Nearest Neighbour Klassifikatoren mit variablen Fenstergrößen gelernt, wobei die finale Klasse durch einen Mehrheitsentscheid der einzelnen Klassifikatoren entschieden wird.

WEASEL (Word ExtrAction for time SEries cLassification) [14] extrahiert Sliding Windows variabler Fenstergrößen aus der Zeitreihe, welche jeweils mit Hilfe der Fourier-Transformation approximiert werden. Hierbei werden Fourierkoeffizienten anhand eines statistischen Tests gewählt, die maximal repräsentativ für die Zeitreihe sind. Somit kann ein linearer Klassifikator gelernt werden.

2.2.2 Shapelet-basiert

Shapelets sind Teilsequenzen der Zeitreihe, welche maximal repräsentativ für eine Klasse sind. Der Klassifikator lernt die Klasse aufgrund einer oder

mehrerer solcher Teilsequenzen, indem die Zeitreihen anhand der Distanz der Shapelets zu anderen Zeitreihen berechnet werden. Diese Lernverfahren lassen sich gut auf verrauschte Daten anwenden, da der Klassifikator sich auf die Shapelets konzentriert und somit Rauschen, welches nur minimal repräsentativ für eine Klasse sein kann, ignoriert. Ein solcher Ansatz muss zuerst Shapelets finden, was, relativ zur Länge der Zeitreihe, eine hohe Laufzeitkomplexität hat. Shapelet Transform (ST) [10] erstellt Features, welche für jede Zeitreihe die minimale Distanz zu den repräsentativsten Shapelets darstellen. Nach dieser Transformation der Daten kann die jeweilige Klasse der Shapelets über einen oder mehrere beliebige Standard-Klassifikatoren gelernt werden. Mr-SEQL Classifier [11] transformiert ebenso die Zeitreihe, nutzt aber zuerst die symbolischen Repräsentationen im Feature-Raum von SAX und SFA um das Finden häufiger Teilsequenzen in Form von häufigen Wörtern zu beschleunigen. Anschließend filtert SEQL die charakteristischen Teilsequenzen der transformierten Zeitreihe, um ein lineares Modell zu lernen.

2.2.3 Intervall-basiert

In dieser Kategorie von Klassifikatoren werden statistische Features aus zufällig gewählten Teilsequenzen der Zeitreihe abgeleitet. Das Time Series Forest (TSF) [9] Verfahren nutzt Mittelwert, Standardabweichung und Steigung als Features der Teilsequenzen. Es wird eine vorher bestimmte Menge an Entscheidungsbäumen trainiert, die als Ensemble zusammen klassifiziert. Mittelwert, Standardabweichung und Steigung reichen für viele Klassifikationsprobleme aus, um korrekte Klassifikationsentscheidungen zu treffen und sind außerdem einfach zu berechnen. Das Canonical Interval Forest (CIF) [12] Lernverfahren nutzt catch22 (CANocial Time-series CHaracteristics) [7] als Ersatz für diese drei statistischen Maße. Catch22 besteht aus 22 statistischen Eigenschaften, die nachgewiesenermaßen in Zeitreihendaten große Relevanz haben.

Das Random Interval Spectral Ensemble (RISE) [6] basiert auf Entscheidungsbäumen ähnlich zu TSF. Zur Klassifikation wird eine Reihe an spektralen Features genutzt, indem die Intervalle fouriertransformiert werden. Auf diesen Features wird pro Intervall ein Entscheidungsbaum trainiert. Die Klassifikation geschieht über einen Mehrheitsentscheid.

2.2.4 Ensemble Klassifikatoren

Um genauere Ergebnisse zu produzieren, werden verschiedene Klassifikatoren zusammen gelernt. Dies hat einen höheren Rechenaufwand als wenn nur ein einzelner Klassifikator verwendet werden würde. Hierbei können unterschiedliche Klassifikatoren auf den gleichen Daten lernen, die dann zusammen eine Entscheidung treffen. Darüber hinaus können auch mehrere Klassifikatoren hintereinander lernen, damit die nachfolgenden Klassifikatoren die falsch klassifizierten Trainingsdaten richtig klassifizieren. Außerdem können auch die gleichen Klassifikatoren auf geteilte Datensätze verwendet werden,

welche letztendlich mit einem Klassifikator zusammengeführt werden, der die finale Entscheidung fällt.

Elastic Ensemble (EE) [6] nutzt mehrere Nearest-Neighbour Klassifikatoren, welche nach der Genauigkeit auf den Trainingsdaten gewichtet werden und anhand des maximalen Gewichts die Klasse wählen. EE nutzt verschiedene Distanzmaße, welche unter anderem Skalierung und Verschiebung der Zeitreihen sowie gleiche Teilsequenzen berücksichtigen, um die Zeitreihen zu vergleichen.

2.2.5 Rocket

RandOm Convolutional KErnel Transform (Rocket) [8] ist ein Lernverfahren, welches darauf ausgelegt ist, die Zeitreihen-Klassifikationen in kürzester Zeit zu berechnen. Dabei wird die Zeitreihe durch zehntausende zufällig gewählte Kernels transformiert und das Ergebnis jedes Kernels mittels einer Pooling Operation auf einen Wert reduziert. Die Eigenschaften (Größe, Gewicht, bias, dilation, padding) der Kernels werden zufällig initialisiert. Das Ergebnis aller Kernel bildet einen eindimensionalen Vektor pro Zeitreihe. Die transformierten Daten repräsentieren das Ausmaß der gelernten Merkmale jedes Kernels in der Zeitreihe. Zuletzt wird auf den transformierten Daten ein linearer Klassifikator trainiert.

3 Forschungsfragen

3.1 Arbeitshypothese I: Korrelation zwischen Klassifikatoren gleicher Kategorie

Wir vermuten, dass Klassifikatoren der gleichen Kategorie stark positiv in den Impact Vektoren korrelieren. Deshalb untersuchen wir Paare von Klassifikatoren der gleichen Kategorie und vergleichen die Impact-Vektor-Paare jeder Vorhersage, welche von timeXplain errechnet wurden. Um auswertbare Ergebnisse zu bekommen, wird jeder Klassifikator mehrfach Zeitreihen aus den jeweiligen Datensätzen klassifizieren, sodass eine ausreichende Anzahl an Impact Vektoren aufgebaut werden kann. Der Korrelationskoeffizient wird anschließend über den Mittelwert aller Korrelationskoeffizienten errechnet.

3.2 Arbeitshypothese II: timeXplain interpretiert Klassifikationsentscheidungen korrekt mittels Impact Vektoren

TimeXplain approximiert nicht-lineare Entscheidungsgrenzen in der Nachbarschaft einer Vorhersage durch eine lineare Entscheidungsgrenze. Daraus werden Impact Vektoren gewonnen. Die vorherige Analyse setzt voraus, dass timeXplains Impact Vektoren verlässlich sind. Um zu überprüfen, wie gut das Analysetool funktioniert, führen wir eine Case Study durch, in der wir die Vorhersagen für dokumentierte Datensätze untersuchen. Für diese Datensätze

ist dokumentiert, welche Datenpunkte ausschlaggebend für die Klassifizierung sind. Zu untersuchen sind demnach Impact Vektoren für die oben genannten Zeitreihen-Klassifikatoren mit dokumentierten Datensätzen.

4 Umsetzung

4.1 Datensätze

Wir benötigen gut dokumentierte Zeitreihendatensätze, um die Interpretationen zu vergleichen. Es muss dokumentiert sein, welche Intervalle der Zeitreihen charakteristisch für die jeweilige Klasse sind, sodass wir diese mit den von timeXplain erstellten Impact-Vektoren vergleichen können. Um Zeitreihen-Klassifikatoren und Analysetools zu testen, gibt es bereits eine große Menge an untersuchten Datensätzen, mit denen unter anderem auch timeXplain erstellt und getestet wurde. Wissenschaftliche Artikel und visuelle Darstellungen dieser Datensätze dienen als Dokumentation. Die Online-Bibliothek <http://www.timeseriesclassification.com/> enthält folgende detaillierte Zeitreihendatensätze, die wir in dieser Bachelorarbeit verwenden werden:

- GunPoint Dataset [3] - (2 Klassen)
- OSULeaf Dataset [4] - (6 Klassen)
- Coffee Dataset [2] - (2 Klassen)
- BinaryHeartbeat Dataset [1] - (2 Klassen)
- Rock Dataset [5] - (6 Klassen).

4.2 Technische Realisierung

Da die Klassifikatoren in sktime und timeXplain mit Python implementiert werden, wird diese Arbeit auch mit Python realisiert. In sktime sind die Klassifikatoren bereits implementiert, die wir zu der Analyse der Daten benötigen. In timeXplain müssen Adaptoren eingerichtet werden, um die Klassifizierung, welche in sktime stattfindet, zu untersuchen. Die Datensätze laden wir von timeseriesclassification.com. Dort befindet sich eine Dokumentation jedes Datensatzes, sodass wir die von timeXplain errechneten Impact Vektoren mit den in der Dokumentation vorhandenen Informationen abgleichen können.

References

- [1] Binaryheartbeat. <http://www.timeseriesclassification.com/description.php?Dataset=BinaryHeartbeat>.
- [2] Coffee. <http://www.timeseriesclassification.com/description.php?Dataset=Coffee>.
- [3] Gunpoint. <http://www.timeseriesclassification.com/description.php?Dataset=GunPoint>.
- [4] Osuleaf. <http://www.timeseriesclassification.com/description.php?Dataset=OSULeaf>.
- [5] Rock. <http://www.timeseriesclassification.com/description.php?Dataset=Rock>.
- [6] Anthony J. Bagnall, Aaron Bostrom, James Large, and Jason Lines. The great time series classification bake off: An experimental evaluation of recently proposed algorithms. extended version. *CoRR*, abs/1602.01711, 2016.
- [7] Lubba C.H., Sethi S.S., and Knaute P. catch22: Canonical time-series characteristics, 2019.
- [8] Angus Dempster, François Petitjean, and Geoffrey I. Webb. ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. *CoRR*, abs/1910.13051, 2019.
- [9] Houtao Deng, George C. Runger, Eugene Tuv, and Vladimir Martyanov. A time series forest for classification and feature extraction. *CoRR*, abs/1302.2277, 2013.
- [10] Ye L. and E. Keogh. Time series shapelets: a novel technique that allows accurate, interpretable and fast classification, 2010.
- [11] Thach Le Nguyen. Interpretable time series classification using linear models and multi-resolution multi-domain symbolic representations, 2019.
- [12] Matthew Middlehurst, James Large, and Anthony Bagnall. The canonical interval forest (cif) classifier for time series classification, 2020.
- [13] Felix Mujkanovic, Vanja Doskoč, Martin Schirneck, Patrick Schäfer, and Tobias Friedrich. timexplain – a framework for explaining the predictions of time series classifiers, 2020.
- [14] Patrick Schäfer and Ulf Leser. Fast and accurate time series classification with WEASEL. *CoRR*, abs/1701.07681, 2017.