5

Universität Stuttgart

Institut für Parallele und Verteilte Systeme
Anwendersoftware

Feature-Extraktion für Sensordaten zur Maschinenüberwachung

Seminararbeit

Advanced Topics in Data Management (WS 2018/2019)

Betreuer: Mathias Mormul

Niklas Kleinhans

Stuttgart, 05.11.2018

Feature-Extraktion für Sensordaten zur Maschinenüberwachung

Niklas Kleinhans

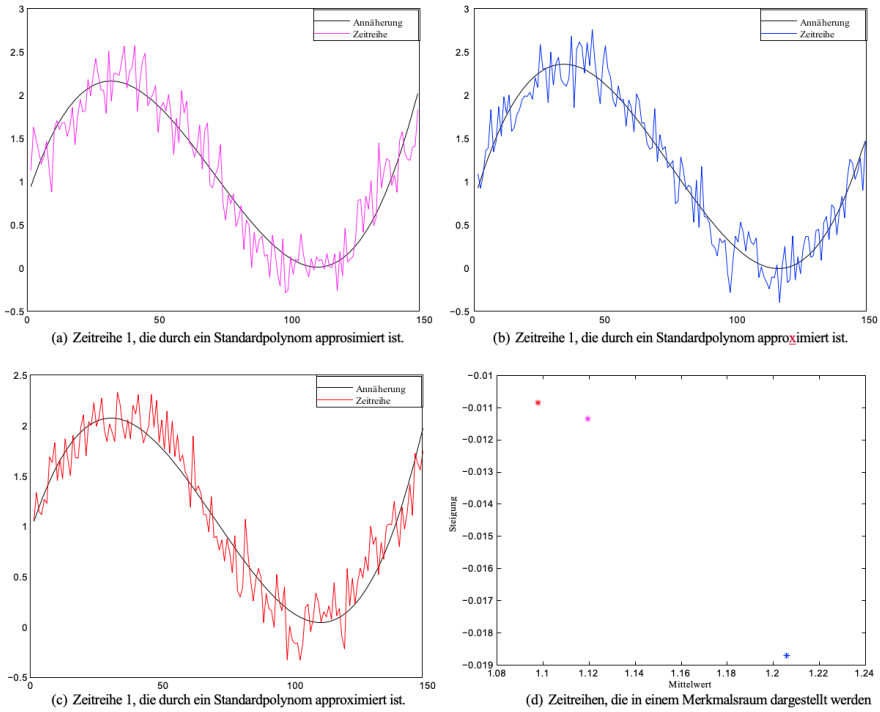
Zusammenfassung. Maschinen werden bis zum kleinsten Bauteil immer intelligenter und ermöglichen es Daten unterschiedlicher Struktur und Komplexität in großen Mengen aufzunehmen. Im industriellen Umfeld wird dabei von „Industrie 4.0“ gesprochen. Der Begriff ummantelt die Beschreibung einer neuen Industriegeneration, in der Systeme durch technologische Weiterentwicklungen intelligent werden und miteinander vernetzt sind. Ein Ergebnis dieser Intelligenz ist die Maschinenüberwachen. Die Maschinenen sollen mithilfe von Livedatenanalyse überwacht und dadurch der aktuellen oder auch zukünftige Maschinenzustand ermittelt werden. Um diese Intelligenz zu erreichen werden unteranderem Sensordaten aufgenommen und anschließend zur Analyse weiterverarbeitet. Die dadurch entstehenden Datenbilder lassen sich oft nicht mehr durch einfache, beispielsweise lineare, Analysmodelle abbilden, was eine rechenintensieveren Analyse zur Folge hat. Die Anforderung an die Sensordatenanalyse im Maschinellenumfeld sind vorallem eine schnelle Datenverarbeitung mittels geringer Rechenkapazität. Um dies zu gewährleisten wird versucht die komplexen Sensordaten so aufzubereiten, dass diese weiterehin durch weniger komplexe Analyseverfahren verarbeitet werden können. In dieser Arbeit werden Verfahren vorgestellt die mithilfe von „Feature Extraktion“ diese Anforderungen erfüllen. Dabei steht der Begriff „Feature“ für ein Merkmal, wodurch sich Daten unterscheiden lassen. Diese werden extrahiert und bilden die Grundlage zur Datendifferenzierung. Es werden die Herausforderungen in der Sensordatenanalyse, speziell in der Livedatenanalyse, beschrieben und Algorithmen vorgestellt.

Schlüsselwörter: Maschinenüberwachung, Feature Extraktion, Maschinelles Lernen, Sensordaten, Livedaten, Livedatenanalyse

1 Einleitung

Maschinen bestehen oft aus mehreren Komponenten und sind anschließend Teil eines großen Systems. Um zu gewährleisten, dass das System korrekt funktioniert und auch möglichst selten ausfällt, werden Daten über Kontrollsysteme und Sensoren jeder einzelnen Komponente aufgenommen. In einem großen System fallen durch kontinuierliche Datenaufnahme große Datenmengen an. Um schon preventiv Maschinenausfälle zu vermeiden werden diese Daten oft an Überwachungssysteme übertragen um anschließend mithilfe von Schwellwertüberschreitungen den Systemzustand darzustellen. Diese Datenbilder werden in Form von Diagrammen, Ampelsystemen, oder ähnlichem visualisiert und für den Menschen lesbar dargestellt. Es ist die Rede von „Condition-Monitoring“.

Um noch mehr Informationen aus diesen Daten zu erhalten werden unterschiedliche Analyseverfahren angewendet um Anomalien festzustellen und auf diese anschließend zu reagieren. Die Herausforderung dabei ist der kontinuierliche Datenfluss. Die Daten müssen, ähnlich zur Livedatenanalyse, direkt verarbeitet werden. Schwellwertüberschreitungen können diese Livedaten leicht verarbeiten, da jeweils nur ein Datenpunkt betrachtet werden muss. Nähere Zusammenhänge innerhalb eines Datenpakets oder sogar zwischen mehrere Datenpaketen zu erkennen ist Rechenaufwändiger. Betrachtt man mehere Daten innerhalb eines Zeitfensters können diese Datenpunkte mittels Approximationen wie in Abbilding 1 (a), (b) und (c) verglichen werden. Die Zeitreihendaten sehen in diesem Fall für Menschen nahezu gleich aus. Dennoch kann durch Anwenden von Analysverfahren ein gewisser Unterschied erkannt werden. In Abbildung 1 (d), wurden die Zeitreihen in einem Merkmalsraum, im englischen „Feature Space“, dargestellt. Zeireihe (a) und (c) sind weiterhin sehr ähnlich. Zeireihe (b) unterscheidet sich etwas im Vergleich zu (a) und (c).



(a) Zeitreihe 1, die durch ein Standardpolynom approsimiert ist.
(b) Zeitreihe 1, die durch ein Standardpolynom approximiert ist.
(c) Zeitreihe 1, die durch ein Standardpolynom approximiert ist.
(d) Zeitreihen, die in einem Merkmalsraum dargestellt werden

Abb. 1. Drei standard ploynom Approximationen mittes least-squared. Die Drei Approximationen werden in Abbildung (d) in einem Merkmalsraum dargestellt [6].

Dieser Merkmalsraum wird durch das Analysieren der Daten auf gewisse Merkmale, im englischen „Features“, erstellt. Der Merkmalsraum bietet eine

weitere Möglichkeit Daten miteinander zu vergleichen, Anomalien zu erkennen oder die Daten zu klassifizieren. Um solche Verfahren anwenden zu können müssen diese Merkmale zuvor extrahiert werden. Es ist die Rede von „Feature Extraktion“.

In diesem Fall handelt es sich um eine polynomielle Approximation der Zeitreihendaten. Diese muss in in jedem Zeitschritt für den gewählt Zeitraum neu berechnet werden. Dies kann je nach Dimension des zu approximierenden Polynoms sehr Rechenaufwändig und komplex werden. Bei einer hohen Abtastrate und einer großen Anzahl an Sensoren entstehen rießige Datenmengen, die unter harten Laufzeitbedingungen analysiert werden müssen. Ein weiterer Vorteil der „Feature Extraktion“ ist das Reduzieren der Komplexität der Daten. Die Daten müssen nicht mehr selbst über polynomielle Modelle analysiert werden, sondern die extrahierten Merkmalen können als Eingabe für die Analyseverfahren dienen. Somit können komplexe polynomielle Datenbilder mithilfe von beispielsweise linearer Verfahren analysiert werden.

In dieser Arbeit wird zu Beginn in Kapitel 3 die grundlegende Struktur von Sensordaten beschrieben, der Zusammenhang mit Livedaten diskutiert und anschließend die damit verbundene Verwendung von „Feature Extraktions“ Verfahren besprochen. In Kapitel 4 werden die Randbedingungen und Herausforderungen an Algorithmen zur kontinuierlichen Livedatenanalyse besprochen und Ansätze sowie Algorithmen vorgestellt, die sich mit desem Problemen auseinander setzen.

2 Verwandte Arbeiten

Viele Arbeiten beschäftigen sich damit Sensordaten mit Hilfe von analytischen Methoden zu verarbeiten. Ein Beispiel ist die Arbeit „Using Machine Learning on Sensor Data“ von Alexandra Moraru et al. [7]. In dieser wird die Anzahl von Mitarbeitern im Büro anhand von Sensordaten vorhergesagt. Dabei werden klassische Klassifikations- und Regressionsverfahren angewendet und validiert. Die Verfahren werden auf einen Trainingsdatensatz trainiert und anschließend auf weitere Datenpakete angewendet. Dabei wurden die Merkmale, an welchen die Anzahl der Mitarbeiter vorhergesagt werden sollen vordefiniert.

Bei komplexeren Problemen, wie bei der Maschinenüberwachung, müssen die Merkmale oft erst gefunden werden. Diese Merkmale können neben einfachen Schwellwertüberschreitungen auch beispielsweise gewisse Datenmuster sein. Diese Andre Gensler, Thiemo Gruber und Bernhard Sick beschreiben Verfahren in ihrer Arbeit „Fast Feature Extraction For Time Series Analysis Using Least-Squares Approximations with Orthogonals Basis Functions“, mit welchen sie diese Merkmale, unter harter Laufzeit und Speicheranforderungen, erkennen.

Die Analyseverfahren benötigen oft sehr viel Rechenleistung da die Proleme, welche einer nicht linearen Komplexität entsprechen, sehr aufwändig zu Berechnen sind. Man spricht dabei von Höherdimensionalen Problemen. Fabian Mörchen beschreibt in seiner Arbeit „Time series feature extraction for data mining using DWT and DFT“ eine Methode um die Dimension von Zeitreihendaten

zu reduzieren. Er Optimiert dabei die Auswahl der Koeffizienten und reduziert damit den Berechnungsaufwand für Analyseverfahren.

-TODO- Die weiteren Referenzen zusammenfassen

3 Grundlagen

In diesem Kapitel werden einige Grundlagen für die in Kapitel 4 diskutierten Algorithmen besprochen.

3.1 Was sind Sensordaten

Um den Ablauf einer Maschine zu koordinieren und den aktuellen Zustand zu Überwachen werden oft Sensoren an den Maschinen angebracht. Diese Daten werden zu definierten Taktzeiten aufgenommen und weiterverarbeitet. Es kann sich dabei um einfache Kontaktsensoren handeln, mit einer begrenzten Anzahl an Zuständen oder um Sensoren mit einem reellen Zustandsbereich, wie Umgebungssensoren (Luftdruck-, Temperatursensor, etc.). In Abbildung 2 sind Sensordaten in Form eines Datenplots dargestellt. Es handelt sich um Licht-, Temperatur- und Luftfeuchtigkeitssensordarten aus einem Büro im Zeitraum eines Arbeitstages. Abgebildet sind die Sensordaten zwischen 6 : 00 Uhr und 19 : 00Uhr.

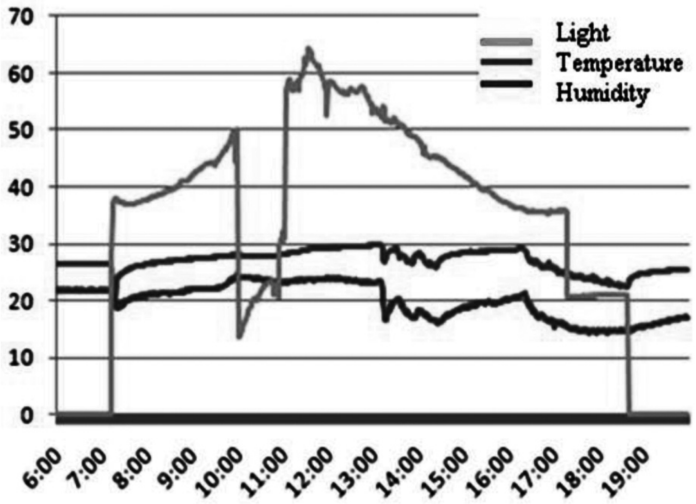


Abb. 2. Sensordatenabweichung anhand von Licht, Temperatur und Luftfeuchtigkeit im Buro innerhalb von einem Arbeitstag [7]

Um solche Sensordaten mathematisch weiterverarbeiten zu können, werden sie in Vektorform gebracht. Ein Datensatz zum Zeitpunkt 6 : 00 Uhr mit den

Werten Licht = 0, Temperatur=22 und Luftfeuchtigkeit = 25 könnte dargestellt werden als:

$$x_1 = \begin{pmatrix} 0 \\ 22 \\ 25 \end{pmatrix} \in \mathbb{R}^D \tag{1}$$

Ein Vektor enthält somit einen Sensordatensatz zu einem Aufnahmezeitpunkt. Um Zeiträume darzustellen werden die Daten in eine Matrixform gebracht. Beispielsweise erhält man bei einer stündlichen Abtastrate und einem Zeitraum von 8 : 00 Uhr bis 17 : 00Uhr folgende Matrix:

$$x_{1,10} \begin{pmatrix} 38\ 40\ 46\ 24\ 60\ 58\ 51\ 44\ 40\ 36 \\ 20\ 22\ 22\ 24\ 24\ 24\ 20\ 18\ 20\ 17 \\ 25\ 28\ 28\ 28\ 29\ 30\ 20\ 27\ 29\ 26 \end{pmatrix} \in \mathbb{R}^{D \times N} \tag{2}$$

Diese mathematische Darstellung ist nur ein Beispiel und kann beliebig strukturiert werden. Die erzeugte Matrix kann anschließend als Eingabe für Analysealgorithmen dienen. Schon bei diesen geringen Datenmengen entsteht eine $D \times N$ große Matrix mit:

- D = Anzahl der Paramter (Sensoren)
- N = Anzahl der Daten.

Sollen auch noch Daten Raumübergreifend analysiert werden, können diese in Form eines Tensors dargestellt werden. Bei M vielen Räumen entsteht ein $D \times N \times M$ großer Tensor. Schon anhand dieses simplen Beispiels wird die Datenmenge und Komplexität der Daten ersichtlich. In der Maschinenüberwachung entstehen dadurch schnell Datensätze im Millionenbereichen und da jede Komponente einer Maschine oft mit mehereren Sensoren ausgestattet ist, entstehen riesige Tensoren.

Neben klassischen Regressionsverfahren zur Datenanalyse, welche oft für Anomliedetektionen verwendet werden, gibt es auch verschiedene Klassifikationsverfahren. Dazu werden den Datensätzen manuell oder automatisiert Klassen hinzugefügt, welche jedem aufgenommenen Datenvektor eine Klasse zuordnet.

Mathematisch dargestellt erhalten wir dann einen Datensatz zum Zeitpunkt t_1 in Form eines Tupels $\tau_1 = (t_1, x_1)$. Die für den Zeitpunkt definierten Merkmale sind dann in x_k mit $(k = 1_d)$ und $d \in D$. Diesem Tupel wird abhängig von den verwendeten Verfahren eine Klasse y_1 zugewiesen [4]. Für den Zeitraum $(t_1, ..., t_n)$ mit $n \in \mathbb{N}$ vielen Daten erhalten wir den Datensatz

$$K = \{(\tau_{1_d}, y_1), ..., (\tau_{n_d}, y_n)\} \tag{3}$$

Das Ziel kann es sein das Tupel (t_{n+1}, y_{n+1}) voherzusagen.

Es werden auch nicht nur feste Zeiträume betrachtet. Durch dauerhaft laufende Maschinen entstehen kontinuierliche Datenströme. Daraus folgt ein kontiniuerlich wachsender Datenbestand. Um ressourcenschonend und möglichst in Echtzeit die Daten zu analysieren, werden harte Speicher- und Laufzeitbedinungen an Analysealgorithmen gestellt.

3.2 Feature Extraktion

Im maschinellen Lernen werden im Wesentlichen zwei Methoden zur Datenanalyse betrachtet. Bei der **Regression** werden die Inputdaten auf Datenwerte reduziert ähnlich zur Approximation in Abbildung 1. Zu dem Bürodatensatz in Kapitel 3.1 könnte mithilfe von Regression die Anzahl an Mitarbeitern im Büro berechnet werden. In der **Klassifikation** werden Daten bestimmten Klassen zugeordnet. Im Gegensatz zur Regression würden nicht die Anzahl der Mitarbeiter, sondern Beispielsweise die Klassen „Büro besetzt“ und „Büro unbesetzt“ ausgegeben werden.

In beiden Verfahren ist die Grundlage der Entscheidung der Dateninput und die dazu gehörigen Parameter. Somit ist die Einhaltung der Algorithmenschranken abhängig von den Parametern D. Es gibt unterschiedliche Gründe, wesshalb versucht wird die Anzahl der Inputparamter anzupassen:

- Die Komplexität eines Lernalgorithmus hängt von der Anzahl an Eingabe Dimensionen D und der Anzahl der Daten N ab. Wird die Anzahl an Dimensionen und somit Paramtern reudziert, dann reduziert sich auch die Komplexität [1].
- Wenn die richtigen Parameter ausgewählt werden, sind die Algorithmen teilweise sogar stabiler [8]
- Wenn eine Eingabe keinen Einfluss auf die Funktion und das Ergebnis des Algorithmus hat, können diese Kosten eingespart werden [1].
- Modelle aus Lernalgorithmen sind oft auf kleine Datensätze robuster, da diese eine geringere Variance und Rauschen aufweisen [1].
- Wenn Daten durch weniger Merkmale beschrieben werden könne, bekommt der Mensch ein besseres Verständnis über den gesamten Prozess [1].
- Je weniger Dimensionen, desto leichter die Visualisierung [1].

Aus diesen Gründen wird versucht die Anzahl der Eingabeparameter in einen Algorithmus möglichst zu reduzieren. Um so eine Dimensionsreduktion zu erreichen gibt es zweit Hauptansätze:

Feature Selektion ist ein Ansatz, indem die Dimension D auf eine Dimension L reduziert wird. Dabei werden interessante Parameter aus der Parametermenge entnommen und die restlichen $D - L$ Parameter werden verworfen. „Features“ sind dabei Merkmale, wodurch sich Daten unterscheiden lassen. In diesem Ansatz sind die Merkmale direkt die ausgewählten Eingabeparameter [1].

Feature Extraktion dagegen entnimmt nicht vorhandene Eingabeparameter sondern geniert bzw. extrahiert aus den vorhandenen Parameter neue Merkmale. Die Anzahl an Merkmalen ergibt die neue Dimension L [1]. Merkmale können einfache, auf die Parameter angewedete, Funktionen sein, wie in Abbildung 1 der Durchschnitt und die Steigung im Merkmalsraum (d). Oder komplexere Parameterkombinationen in höher Dimensionalenräumen, wie Verlaufsmuster von Raumflächen.

Auf Lernalgorithmen angewendet ist es letztlich immer das Ziel die Features so zu wählen und zu parametrisieren, dass der ermittelte Wert oder die ermittelten Zuordnung aus einem Lernalgorithmus dem tatsächlichen möglichst ähneln. Formal beschrieben anhand dem Datensatz 3. Sei $p(x_i)$ die Funktion, welche das Ergebnis des gewählten Algorithmus und des Trainingsdatensatzes ist. Es soll versucht werden den Fehlerunterschied

$$\sigma = p(x_i) - y_i \tag{4}$$

möglichst zu reduzieren [6]. Die konkreten Herausforderungen und Herangehensweisen in der Maschinenüberwachung werden in dem folgenden Kapitel besprochen

4 Feature Extraktion bei kontinuierlichen Livedaten

Die Sensordaten zur Überwachung von Maschinen erzeugen einen kontinuierlichen Datenfluss. Wie in den voherigen Kaptiteln beschrieben, stellt diese Eigenschaft eine hohe Anforderung an Lernalgorithmen. Einige Lernalgorithmen verwenden das Konzept des „Lazy Learnings“, im Deutschen „träges Lernen“. Bei diesen Lernalgorithmen wird das Modell auf Anfrage erstellt. Das bedeutet, jede Eingabe startet eine neue Modellbildung und fordet daher in kürzester Zeit eine hohe Rechenleistungen. Beim „Eeager Learning“, im Deutschen „Eifriges Lernen“, hingegen wird schon im Vorfeld mithilfe von Trainingsdaten ein Model bereitgestellt. Der Ressourcenverbrauch bei einer Anfrage von neuen Daten ist dann sehr gering [5].

Bei den kontinuierlichen Livedaten in der Maschinenüberwachung stehen die Daten meist auch zeitlich in korrelation. Daher ist die Rede von Zeitreihen Daten. Die Reihenfolge dieser Daten spielt bei der Analyse eine große Rolle. Zeitreihendatensätze werden mathematisch wie in dem Datensatz 3 dargestellt. Eines der Ziele für Lernalgorithmen kann es sein aus dem Datensatz 3 die Daten

$$\tau_{n+1}, \tau_{n+2}, ... \tag{5}$$

voherzusagen.

Um die Komplexität der Zeitreihendatensätze zu reduzieren wird die Dimension, entstehend durch die Parameter, versucht möglichst auf die notwendigen Merkmale zu reduzieren. Zeitreihendaten aus Sensoren sind oft hoch korrelierend, was zu einer sehr großen Datenredunanz führt. Eine bewerte Technik um Datenmengen aus Sensordaten fester größe darzustellen ist die „Fourier Transforamtion“. Dabei werden die Signale auf einen Frequenzbereich abgebildet und diese Abbildung mittels Koeffizienten dargestellt. Da es sich bei den kontinuierlichen Sensordaten meist um keinen vollständigen Datenbestand handelt, sondern nur Datenauszüge bietet sich die „Diskrete Fourier Transformation“ an.

$$F_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k W_N^{-kj} \quad \text{mit} \quad W_N = e^{\frac{2\pi i}{N}} \tag{6}$$

Durch die Diskrete Fourier Transformation in Formel 6 wird der Zeitreihenabschnitt in einer periodische Funktion durch Koeffizienten dargestellt [3]. Um die zu analysierdenen Merkmale zu reduzieren ist die Idee,dass nicht alle Koeefizienten als Parameter verwenden werden, sondern nur eine Auswahl von diesen. Bekannte Methoden sind dabei entweder die ersten k Koeffizienten zu verwenden. Man speichert quasi nur eine grobe Skizze der Kurven ab. Die ersten Koeffizienten abzuspeichern, behält die tieferen Frequenzen und ist eine sehr naive herangehensweise. Eine deutlich bessere Methode wäre es die größten Koeffizienten zu verwenden. Diese sind aber sehr aufwändig zu berechnen [8]. Fabian Mörchen stellt dagegen eine Methode in seiner Arbeit „Time series feature extraction for data mining using DWT and DFT“ vor, in der er eine Aggregatsfunktion verwendet, welche die Bedeutung der Koeffizienten misst. Es ist dadurch möglich mit seiner Aggregatsfunktion:

$$J_k^1(mean(c_j^2), C) \tag{7}$$

eine definierte Menge k an Koeffizienten als Merkmale für Lernalgorithmen als Eingabe zu geben. k entspricht dann der Dimension der zu verarbeitenden Datenmenge.

Eine weitere Arbeit von Dominique Gay et al. „Feature Extraction over Multiple Representations for Time Series Calssification“ stellt ein Verfahren vor in dem sie Zeitreihendaten so vorverarbeiten, dass in dem Verfahren extrahierte Merkmale die neuen Parameter fester Größe bilden. Dies geschieht in einem drei Schritteverfahren:

1. Der Datensatz wird in mehrere Datenrepräsentationen transformiert
2. Auf jede Repräsentation wird ein Co-Clustering Verfahren angewendet
3. Aus jeder Repräsentation wird eine Menge an Merkmalen erstellt und daraus ein neuer Datensatz generiert

Für den ersten Schritt schlagen sie verschiedene Transformationsverfahren vor. Beispielsweise Ableitungen oder comulatives Integrieren. In einem Beispiel sind die Vorteile einer solchen Vorverarbeitung erkennbar. In Abbildung 3 ist in (a) ein zwei Klassen ARSim Datensatz zu sehen. Dieser ist unverarbeitet und die Klassen sind farblich Markiert. Die Klassen sind in (a) nur sehr schwer separierbar. In (b) wurde der selbe Datensatz durch zweifaches Ableiten transformiert und wieder in einem Datenplot und farblicher markierung dargetsellt. Durch die Transformation sind die Klassen schon deutlicher erkennbar.

Der transformierte Datensatz bilde somit eine bessere Grundlage um mithilfe von Klassifizierungsalgorithmen die Daten zu Differenzieren. Der zweite Schritt ist das Co-Clustering. Dabei wird das Clustering als eine Vorverarbeitung für nachfolgende Lernalgorithmen verwendet. Die Idee ist es ähnliche Daten zu gruppieren und lokale Muster hervorzuheben [4]. Dabei stellen sie die Kurven in einer Menge (X, Y) dar und fügen jeder dieser Mengen eine Klasse Cid hinzu um sie der jeweiligen Kurve zuzuordnen. Es entsteht eine dreidimensinale Darstellung der Punkte. Das Ganze wird in eine dreidimensionale Gitterstruktur gebracht.

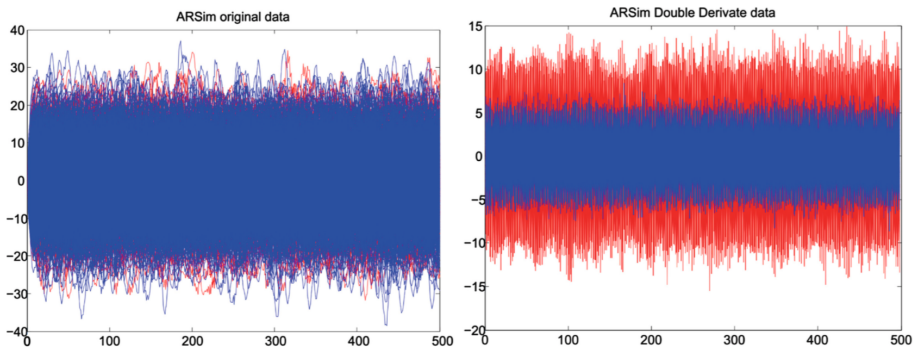


Abb. 3. In (a) ein unverarbeiteter original geplotteter Datensatz und in (b) der gleiche durch zweifache Ablteiungen transformierter Datensatz [5].

Das Endziel ist es Kurven- und Intervallcluster zu erhalten die Anschließend als Merkmalsgrundlage dienen.

Erreicht wird das durch das Anwenden des „Khiops Coclustering“ [2]. Dabei wird das optimale Gitterfeld durch die Optimierung des Bayes'schen Kriteriums, der sogenannen Kosten ermittelt.

$$cost(M) = -log(p(M) \times p(D|M)) \tag{8}$$

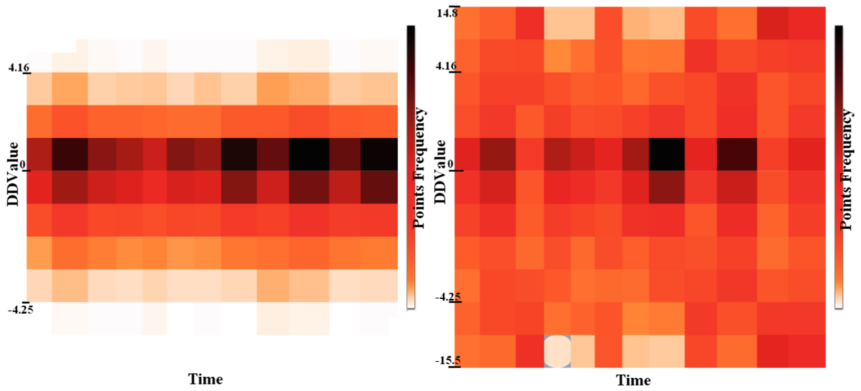
Als resultat lassen sich die Kosten so interpretieren, dass bei niedrigen Kosten eine hohe Kompression der Daten D auf das Modell M herrscht. Wobei das Modell in diesem Fall das optimale Gitterfeld ist.

Ein durchgeführtes Co-Clustering ist in Abbildung 4 zu sehen. Dabei ist die dritte Dimension Farblich dargestellt. Als Ergebnis sind 25 Cluster von Kurven entstanden.

Als letzten Schritten müssen noch die Merkmale extrahiert und ein Datensatz generiert werden. Es werden dre Merkmale definiert:

1. K_C ein numerisches Merkmal, welches die Unähnlichkeitswahrscheinlichkeit zu allen Kurvenclustern angibt.
2. Ein kategorisches Merkmal als index, welcher das nächste Kurvencluster angibt.
3. K_Y ein nummerisches Merkmal, welches die Anzahl an Punkten dieser Kurve in der jeweiligen Klasse angibt.

Durch dieses Verfahren wird die Dimension der Livedaten mit Hilfe von „Feature Extraktion“ auf Drei festgelegt und kann somit durch „Eager Learning“ Algorithmen analysiert werden.



(a) (b)

Abb. 4. Ergebniss eines CoCluserings mit Khiops Coclustering [5].

5 Fazit

Die Möglichkeiten Maschinen zu Überwachen und den aktuellen Maschinenzustand mittels Sensorik zu bestimmen werden immer größer. Nicht nur gegenwärtige Zustände, sondern auch Zustandsvohersagen sind mittlerweile möglich. Dadurch, dass schon kleinste Komponenten einer Maschine Sensordaten sammeln und diese zur Analyse bereitstellen, entsteht eine rießige Masse an Daten zur weiterverarbeitung. Doch diese kontinuierlichen Datenflüsse stellen Lernalgorithmen an große Herausforderungen. Die Lernanlgorithmen müssen unter harten Speicher- und Laufzeitanforderungen arbeiten und aus der Fülle an Daten die wichtigsten Informationen herausfiltern und diese auch möglichst noch für den Menschen verwendbar darstellen. Sensordaten fangen meist als einfache numerische Werte an, werden über die Anzahl an Paramter zu Vektoren zusammengefasst und ergeben über Vielfalt und Zeit rießig dimensionale Tensoren. In dieser Arbeit werden Verfahren behandelt, welche diese Dimensionen für Lernalgorithmen durch Vorverarbeitung reduzieren.

Dabei werden „Feature Extraktions“ Verfahren angewendet. Unter einem „Feature“ versteht man ein Merkmal, wodurch sich Daten unterscheiden lassen. Verschiedene Ansätze versuchen die relevantesten Merkmale auszuwählen (Feature Selektion) oder durch Berechnungen neuer Merkmale zu extrahieren (Feature Extraktion).

Einige dieser Ansätze bedienen sich bekannter Methoden, wordurch Datensätze mittels Transformationen parametrisiert werden und die koeffizienten als Merkmale weiterverwendet werden können. Die Darstellung der Sensordaten als Kurven in einem Datenplot bietet auch visuelle anssätze, indem die Daten Räumlich dargestellt werden und und dann durch Räumliche zuordnungen klassifiziert und durch Merkmale unterschieden werden können.

Andere Ansätze versuchen den Berechnungsaufwand so zu beschränken, dass zu Beginn die Berechnung unabhängig von der Datensatzgröße ist und nur noch von der durch die Anzahl der Paramter bestimmten Dimension abhängt. Das anschließende Anwenden von „Feature Extraktion“ leifert eine starke Laufzeiteinsparung.

Leider ist nie garantiert, dass durch die Extraktion ein Informationsverlust besteht und daher gibt es keine generelle Lösung für alle Datensätze. Die sogenannten Domänenexperten, die Personen welche sich Fachlich mit den Daten auskennen, sind vorerst Teil der Perfekten Analysekonfiguration.

Literatur

1. Alpaydin, E., Bach, F.: Introduction to Machine Learning. Adaptive Computation and Machine Learning series, MIT Press (2014), https://books.google.de/books?id=7f5bBAAAQBAJ
2. Boullé, M.: Functional data clustering via piecewise constant nonparametric density estimation. Pattern Recognition **45**(12), 4389–4401 (2012)
3. Butz, T.: Fouriertransformation für Fußgänger. Vieweg+Teubner Verlag | Springer Fachmedien Wiesbaden GmbH, Wiesbaden (2012)
4. Gay, D., Guigourès, R., Boullé, M., Clérot, F.: Feature extraction over multiple representations for time series classification. In: International Workshop on New Frontiers in Mining Complex Patterns. pp. 18–34. Springer (2013)
5. Gay, D., Guigourès, R., Boullé, M., Clérot, F.: Feature extraction over multiple representations for time series classification. In: International Workshop on New Frontiers in Mining Complex Patterns. pp. 18–34. Springer (2013)
6. Gensler, A., Gruber, T., Sick, B.: Fast feature extraction for time series analysis using least-squares approximations with orthogonal basis functions. In: Temporal Representation and Reasoning (TIME), 2015 22nd International Symposium on. pp. 29–37. IEEE (2015)
7. Moraru, A., Pesko, M., Porcius, M., Fortuna, C., Mladenic, D.: Using machine learning on sensor data. Journal of computing and information technology **18**(4), 341–347 (2010)
8. Mörchen, F.: Time series feature extraction for data mining using dwt and dft (2003)

Alle Links wurden zuletzt am 10.12.2018 geprüft.