Bestimmung der Reccurence Matrix für die Reccurence Quantification Analyse mittels Approximate Nearest Neighbor Search

Exposé zur Diplomarbeit

eingereicht von: David Salomon
geboren am: 07.08.1987
geboren in: Cottbus

Gutachter/innen: Prof. Dr. Ulf Leser

eingereicht am: 23.02.2016

1 Einleitung

Recurrence Quantification Analysis (RQA) ist eine Methode zur Analyse dynamischer Systeme. Hauptaufgabe der RQA ist das Auffinden von Wiederholungen (*Reccurences*) in einer Zeitreihe. Hierfür wird die Verteilung der vertikalen und diagonalen Linien in einer Ähnlichkeitsmatrix, der sogenannten *Reccurence Matrix*, bestimmt. Zur Konstruktion der Ähnlichkeitsmatrix werden die paarweisen Ähnlichkeiten zwischen mehrdimensionalen Vektoren berechnet, die aus einer gegebenen Zeitreihe extrahiert wurden. Als Ähnlichkeitsmaß dient z.B. die euklidische Norm. Unterschreitet die Distanz zwischen zwei Vektoren eine definierte Schranke, wird eine 1 in die Matrix eingetragen. Vertikale und diagonale Linien ergeben sich durch aufeinanderfolgende Einsen [1].

Anhand der *Recurrence* Matrix werden eine Reihe quantitativer Maße bestimmt. Zum Beispiel gibt die *Recurrence Rate (RR)* den Anteil der *Recurrence Points* (als ähnlich markierte Matrixelemente) in der Matrix an. Mit Hilfe der ermittelten diagonalen Linien kann ein weiteres Maß, der *Determinism* (DET), bestimmt werden. Dieser gibt für einen *Recurrence Point* die Wahrscheinlichkeit an, dass dieser Punkt zu einer diagonalen Linie mit mindestens der Länge *j* gehört [1].

Die RQA wird in vielen wissenschaftlichen Bereichen eingesetzt. Zum Beispiel wird in [2] die mögliche Anwendung in der Analyse des Herz-Lungen-System mittels der RQA genannt. Diese wird durchgeführt, um Einblicke in die Funktionsweise des Systems zu erhalten. Zusätzlich können Fehlfunktionen identifiziert werden, um lebensbedrohliche Zustände zu diagnostizieren [2].

Der Brute Force Algorithmus zur Konstruktion der Ähnlichkeitsmatrix hat eine quadratische Laufzeit, da dieser alle paarweisen Ähnlichkeiten berechnet und in der Matrix entsprechend kodiert.

Ein Ansatz zur Verbesserung der Performance ist die Parallelisierung des Algorithmus. Das Geoforschungszentrum Potsdam (GFZ) entwickelte eine Implementierung, die eine RQA auch für Zeitreihen mit mehr als 1 Millionen Datenpunkten in akzeptabler Zeit durchführt. Hierfür wurde ein stark parallelisierter Berechnungsprozess verwendet [2]. Die quadratische Laufzeit bleibt bei einem parallelen Ansatz allerdings erhalten. Hierdurch kommt es bei steigender Datenmenge zu einem drastischen Anstieg der Laufzeit.

Ein weiterer Ansatz ist die Reduktion der Anzahl an benötigten Ähnlichkeitsvergleichen mittels einer Indexstruktur. In [3] wurde dieser Ansatz im Kontext der k-Nächsten Nachbarsuche evaluiert und mit dem parallelen Ansatz des GFZ verglichen. Im Ergebnis wurde geschlussfolgert, dass durch Indexierung für bestimmte Datenmengen und Dimensionalitäten eine Verbesserung der Laufzeit erzielt werden kann. Die Dimensionalität der zugrunde liegenden Datenmenge hat den größten negativen Einfluss auf die Laufzeit der verwendeten Indexstruktur. Dies bedeutet: Für hinreichend große Datenmengen mit einer niedrigen Dimensionalität ($d < 10$) könnte eine echte Verbesserung der Konstruktionszeit gegenüber dem parallelen Ansatzes des GFZ erzielt werden. Für Dimensionalitäten größer als 10 übersteigt der Rechenaufwand den einer parallelisierten Brute Force Variante.

Sowohl der parallele Ansatz als auch die Verwendung einer Indexstruktur liefern immer exakte Ergebnisse. In dieser Diplomarbeit soll ein dritter Ansatz untersucht werden. Mit Hilfe approximativer Verfahren wird der Inhalt der *Reccurence Matrix* abgeschätzt. Dieser Ansatz hat potentiell einen geringeren Rechenaufwand als die beiden bereits untersuchten Ansätze. Im Gegensatz zu diesen wird aber keine exakte Ergebnismenge garantiert.

2 Related Work

In [4] wurde bereits ein approximativer Ansatz für die RQA untersucht. Dieser überspringt den Schritt der Konstruktion der Recurrence Matrix vollständig und bestimmt die *Recurrence Rate* und den *Determinism* näherungsweise anhand der zugrundeliegenden Zeitreihe. Die Genauigkeit der Ergebnisse hängt stark von der gewählten Ähnlichkeitsschwelle ab.

3 Bisherige Konstruktionsmethode der Reccurence Matrix

Es existieren zwei grundlegende Ansätze, eine *Reccurence Matrix* zu konstruieren. Diese nutzen zwei unterschiedliche Nachbarschaftsbedingungen: *fester Radius* und *feste Anzahl von nächsten Nachbarn*.

fester Radius: Alle Datenobjekte dϵD, welche eine definierte Ähnlichkeitsschranke zu dem Queryobjekt q nicht überschreiten – entspricht einer *Bereichssuche.*

feste Anzahl von nächsten Nachbarn: Genau die k Elemente, welche dem Queryobjekt q am ähnlichsten sind – entspricht einer *k-Nächsten Nachbarsuche.*

Um die *Reccurence Matrix* zu erzeugen wird pro Datenobjekt eine Suche durchgeführt. Die Ergebnismenge bildet eine Zeile bzw. Spalte der Matrix ab. Hierfür werden alle in der Ergebnismenge enthaltenen Datenobjekt mit einer “1“ kodiert, alle anderen werden dementsprechend mit “0“ kodiert.

Die in der Einleitung beschriebene *Reccurence Matrix* kann auf die Konstruktion mittels einer Bereichssuche zurückgeführt werden. Im Fokus dieser Arbeit steht genau diese Konstruktionsweise. Um Laufzeit zu sparen wird ein approximativer Ansatz zur Durchführung der Suche eingesetzt.

4 Approximative Konstruktionsmethode der Reccurence Matrix

Für die Bereichssuche existieren bereits approximative Ansätze, welche näherungsweise die Nachbarn zu einem Queryobjekt bestimmen [5,6,7,8,9]. Die Ergebnismenge eines approximativen Verfahrens kann zu wenige bzw. zu viele (und somit falsche) Nachbarn enthalten, welche auch als *false negative/positives* bezeichnet werden. Abhängig vom verwendeten Algorithmus können sowohl *false negatives* als auch *false positives* oder nur eines von beiden auftreten. Beim *Locality Sensitive Hashing* (LSH)[7] können beispielsweise beide Varianten auftreten, wohingegen beim *Randomized k-d-trees* [8] nur *false negatives* vorkommen.

Es werden zwei Kategorien der approximativen Bereichssuche untersucht: fehler- und ressourcenbasierte Algorithmen.

Ein fehlerbasierter Algorithmus im Kontext der Bereichssuche liefert beispielsweise zu einem Queryobjekt *q* alle Nachbarn, welche die Ähnlichkeitsbedingung näherungsweise erfüllen. Hierfür wird ein Fehler $\varepsilon > 0$ definiert, welcher angibt, wie stark die Ähnlichkeit der ermittelten Nachbarn von der geforderten Bedingung abweichen darf. Ist beispielsweise die Ähnlichkeitsbedingung die Distanz $d = 3$ und der definierte Fehlerfaktor $\varepsilon = 0.1$, dann sind ausschließlich Nachbarn erlaubt deren Distanz zum Queryobjekt kleiner als 3.3 ($d * (1 + \varepsilon)$) ist. Ein Vertreter dieser Kategorie ist das *Locality Sensitive Hashing* (LSH)[7], für welchen unter anderem die Implementierungen *lshash* in Python verfügbar ist.

Die ressourcenbasierten Suchalgorithmen hingegen hören mit der Suche auf, wenn eine bestimmte ressourcenabhängige Bedingung überschritten wurde. Eine Nachbarschaftssuche in einer baumartigen Indexstruktur würde beispielsweise nach einer definierten Anzahl von besuchten Blattknoten terminieren und das aktuelle Zwischenergebnis als Ergebnis liefern [5]. Für die Bereichssuche bedeutet dies, dass alle bis dahin gefundenen Datenobjekte, welche die Ähnlichkeitsbedingung erfüllen, das Ergebnis bilden. D.h. alle nicht besuchten Knoten werden dementsprechend ignoriert. Vertreter dieser Kategorie sind: *Randomized k-d-trees* [8] oder *Hierarchical k-means trees* [9]. Beide Algorithmen wurden in der ANN Bibliothek *FLANN* [6] implementiert.

5 Ziel der Diplomarbeit

Ziel dieser Diplomarbeit ist es, die beiden vorgestellten ANN Kategorien hinsichtlich des Einsatzes für die RQA zu evaluieren. Anschließend werden die geeigneten Algorithmen zur Konstruktion einer *Recurrence Matrix* implementiert oder, wenn möglich, bestehende Implementierungen verwendet und deren Laufzeiten mit der bestehenden parallelen Implementierung des GFZ verglichen [2].

Die implementierten Varianten sollen ebenfalls mit dem in [4] vorgestellten approximativen Ansatz verglichen werden. Hierfür müssen zusätzlich aus der näherungsweise bestimmten *Recurrence Matrix* die entsprechenden RQA-Maße extrahiert werden.

Es muss evaluiert werden mit welchen Einschränkungen eine RQA mittels einer approximativen Konstruktionsmethode für die *Reccurence Matrix* durchgeführt werden kann. Speziell die Auswirkung der *false positives/negatives* bei der Ähnlichkeitsabschätzung auf die RQA Maße muss untersucht werden.

Im Ergebnis soll festgestellt werden, unter welchen Rahmenbedingungen ein approximativer Ansatz sinnvoll ist. Solche Bedingungen sind unter anderem: verwendeter ANN-Algorithmus, Eigenschaften der Zeitreihe, die Ähnlichkeitsschwelle sowie die Genauigkeit der Ergebnisse.

Zusätzlich wird versucht den Folgefehler, welcher sich aus der Verwendung eines approximativen Verfahren ergibt, für ausgewählte RQA Maße analytisch abzuschätzen. Diese Analyse erlaubt voraussichtlich eine Abschätzung des Folgefehlers eben dieser RQA Maßen, anhand der Eigenschaften der zugrundeliegenden Datenmenge.

Im Ergebnis entsteht idealerweise eine eigens für die RQA optimierte approximative Methode. Denkbar ist, dass sich beim LSH[7] eine bestimmte Anzahl an Hashfunktionen

als optimal erweist. Ebenfalls könnte sich für die *Randomized k-d-trees* eine obere Schranke für die Anzahl der zu besuchenden Knoten im Kontext der RQA ergeben.

6 Datensets & Evaluation

Zur Evaluierung werden ausschließlich Zeitreihen verwendet, da diese die Grundlage einer RQA bilden. Am GFZ wurde im Rahmen einer Masterarbeit ein Zeitreihengenerator verwendet, welcher im Rahmen dieser Arbeit eingesetzt werden soll [11].

Die Evaluierung selbst findet auf einem zu definierenden Testsystem statt. Ein Testszenario vergleicht für eine Zeitreihe immer die Laufzeit eines ANN Verfahrens mit der parallelen Variante des GFZ. Zusätzlich werden die Recurrence Matrizen visualisiert, um die topologischen Unterschiede zu bewerten.

Es soll die Performance der RQA bei steigender Datenmenge/Dimensionalität untersucht werden. Zusätzlich wird untersucht, wie sich die Qualität der Ergebnismenge hierbei verändert. Im Ergebnis wird festgestellt, wie die verwendeten approximativen Verfahren bei steigender Datenmenge/Dimensionalität skalieren.

Die für ANN-Verfahren relevanten Parameter, wie der Fehlerfaktor ε bzw. die maximale Ressourcenmenge, werden ebenfalls untersucht. Es werden Tests mit einer identischen Datenmenge durchgeführt und nur diese Parameter variiert um den Effekt auf die RQA zu evaluieren.

Um die Auswirkung der näherungsweise bestimmten Recurrence Matrix auf die zu berechnenden RQA Maße zu analysieren, müssen in jedem Testszenario definierte RQA Maße ermittelt werden und diese mit den exakten Maßen verglichen werden.

7 Literaturverzeichnis

[1] N. Marwan, M.C. Romano, M. Thiel, J. Kurths: Recurrence Plots for the Analysis of Complex Systems, Physics Reports 438(2007) 237-329

[2] T. Rawald, M. Sips, N. Marwan, D. Dransch, Fast Computation of Recurrences in long Time Series

[3] D. Salomon: Massiv parallele kNN-Suche auf der GPU im Performancevergleich zur kNN-Suche über Indexstrukturen, Studienarbeit HU-Berlin

[4] D. Schultz, S. Spiegel, N. Marwan, S. Albayrak: Approximation of diagonal line based measures in recurrence quantification analysis, Physics Letters A, Elsevier, 2014

[5] J. Wang, N. Wang, Y. Jia, J. Li, G. Zeng, H. Zha, X.-S. Hua: Trinary-Projection Trees for Approximate Nearest Neighbor Search, Pattern Analyses and Machine Intelligence, IEEE Transaction on (Volume:36, Issue:2), 2013

[6] FLANN – Fast Library for Approximate Nearest Neighbors, http://www.cs.ubc.ca/research/flann/, Abgerufen: 26.11.2015

[7] P. Indyk, R. Notwani – Approximate nearest neighbor. Towards removing the curse of dimensionality, In Proceedings of the Symposium on Theory of Computing, 1998

[8] C. Silpa-Anan, R. Hartley - Optimised KD-trees for fast image descriptor matching. In CVPR, 2008

[9] K. Mikolajczyk, J. Matas - Improving descriptors for fast tree matching by optimal linear projection, In Computer Vision, 2007, IEEE 11th International Conference on, pages 1–8.

[10] J.L. Bentley - Multidimensional Binary Search Trees Used for Associative Searching. Communications of the ACM, Volume 18 Issue 9, 1975

[11] C. Witt – Masterarbeit: Clustering von Recurrence Plots, 2015