**Vergleich von Data-Mining-Verfahren zur Outlier Detection**

Jonas M. Brauer, 22.05.2023

**Abstract**

Beim Data-Mining sammelt man viele Daten und untersucht diese u.a. auf Abhängigkeitsbeziehungen zwischen Variablen. Man versucht, Korrelationen zu entdecken, da sich hinter diesen eine mögliche Kausalität befinden kann. Dabei ist es störend, wenn ein „Rauschen“ in Gestalt von Ausreißern vorhanden ist, das die Sicht auf diese Dinge trübt. In einem anderen Fall will man gerade an diese Ausreißer heran, um außergewöhnliche Vorgänge zu entdecken. Dies kann z.B. die Bekämpfung von Kreditkartenbetrug, die Angriffserkennung in Computer-Netzwerken oder die Fehlerdiagnose in Maschinen sein, indem man in den Daten nach ungewöhnlichen Vorgängen sucht. Ab wann ein Datenpunkt sich genügend von den restlichen Daten unterscheidet, um als Ausreißer zu gelten, ist Definitionssache und nicht immer eindeutig. Zwischen einem Punkt der intuitiv eindeutig als Ausreißer und einem der mit voller Gewissheit nicht als Ausreißer gewertet wird, gibt es oft unendlich mögliche Zwischenpunkte. Ob man jetzt Ausreißer loswerden oder genauer betrachten will, in beiden Fällen muss man sie irgendwie erkennen. Das ist bei Daten mit einer (univariat), zwei (multivariat) oder sogar drei Dimensionen auch von Menschenhand möglich, aber bei vieldimensionalen Daten muss, auch zur effizienten Automatisierung, ein Verfahren her. Davon gibt es einige, die über die Zeit hinweg entwickelt wurden. Doch sind sie alle gleich fähig? In diesem Vortrag sollen ausgewählte Verfahren vorgestellt und miteinander verglichen werden. Dazu wurden vordergründlich wissenschaftliche Veröffentlichungen zurate gezogen, um sich in das Thema der Verfahren zur Ausreißererkennung einzulesen, aus den existierenden Verfahren eine Auswahl zu treffen und sich mit den Verfahren tiefergehend zu beschäftigen. Hierzu und zum Zweck der Vorstellung wurde ein Python-Programm geschrieben, das den Benutzer interaktiv Datenpunkte in einen Graphen einfügen lässt und das mithilfe eines vom Benutzer angegebenen Verfahrens und nötiger Parameter Ausreißer unter diesen Datenpunkten findet und visuell darstellt. In dem Vortrag geht es zu Beginn erst einmal um eine kurze Vorstellung verschiedener Verfahrenstypen und Verfahren zur Ausreißer-Erkennung sowie im weiteren Verlauf um die beiden Verfahren DBSCAN und Isolation Forest im Genaueren sowie ein anschließender Vergleich dieser beiden Verfahren miteinander. Letzteres wurde anhand von Vergleichskriterien wie den Laufzeiten, der Speicherkomplexität und der „Receiver Operating Characteristic“ (ROC) mit der „Area Under the Curve“ (AUC) sowie anderen Vor- und Nachteilen vorgenommen. DBSCAN zählt zu den dichtebasierten Verfahren. Es ordnet jeden Punkt, abhängig von seiner Umgebungsdichte, entweder einem Cluster oder der Gruppe der Rauschpunkte zu. Dabei wird über zwei Parameter ein globaler Schwellenwert bzgl. Dichte festgelegt. Alle Punkte, die eine höhere Dichte haben, sowie ihnen naheliegende Punkte, gehören Clustern an. Alle anderen Punkte sind Rauschpunkte und damit Ausreißer. Das Verfahren des Isolation Forest baut hingegen eine festgelegte Anzahl zufällig generierter Isolation-Trees auf, indem es eine Stichprobe konstanter Größe der Datenpunkte nach zufällig gewählten Attributen und zufällig gewählten Werten aufteilt und so voneinander isoliert. Die Datenpunkte sind in den Blattknoten zu finden, wobei Ausreißer im Mittel eine kurze Pfadlänge haben. Nach diesem Kriterium wird für die Datenpunkte eine Wahrscheinlichkeit bestimmt, die angibt, wie wahrscheinlich es sich bei ihnen um Ausreißer handelt. Die Pfadlänge kann von Baum zu Baum schwanken, weswegen es einen Wald von Bäumen gibt, aus denen man das Mittel errechnen kann. Der Algorithmus besteht aus zwei Phasen: Dem Aufbau des Waldes und der Auswertung der Punkte. Allgemein lässt sich sagen, dass Isolation Forests mit linearer Laufzeit hocheffektiv und hocheffizient sind. Ihre Fähigkeit, genau die Ausreißer zu identifizieren, ist mit hohen AUC-Werten sehr gut. Außerdem ist die Parameterbestimmung im Gegensatz zu DBSCAN sehr einfach. Isolation Forests sind die richtige Wahl für große Datenmengen, mit vielen Attributen und Daten, über deren Verteilung und Ausreißeranteil man wenig weiß. DBSCAN identifiziert Ausreißer und teilt nebenbei alle normalen Datenpunkte automatisch in Cluster ein. Dabei spielt es keine Rolle, welche Form ein Cluster hat. Der Rechenaufwand ist quadratisch und der Algorithmus somit nicht für große Datenmengen nutzbar. Außerdem muss die Parameter angeben, die nicht trivial zu ermitteln sind.

**Literatur**

Schubert, E., Sander, J., Ester, M., Kriegel, H.-P., Xu, X. (2017). “DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN”. ACM Trans. Database Syst. 42,3, Article 19 (July 2017), 21 pages.

https://doi.org/10.1145/3068335

Hahsler, M., Piekenbrock, M., Doran, D. (2019). „dbscan: Fast Density-Based Clustering with R“. Journal of Statistical Software, October 2019, Volume 91, Issue 1.

https://doi.org/10.18637/jss.v091.i01

Hodge, V.J., Austin, J. (2004). “A survey of outlier detection methodologies.” Artificial Intelligence Review. pp. 85-126. ISSN 1573-7462.

https://doi.org/10.1023/B:AIRE.0000045502.10941.a9

Wang, H., Bah, M. J., Hammad, M. (2019). “Progress in Outlier Detection Techniques: A Survey”. IEEE, DOI: 10.1109/ACCESS.2019.2932769

F. T. Liu, K. M. Ting, Z.-H. Zhou, “Isolation Forest”. (2008). Eighth IEEE International Conference on Data Mining, Pisa, Italy, 2008, pp. 413-422, doi: 10.1109/ICDM.2008.17.

Ester, M, Kriegel, H P, Sander, J, and Xiaowei, Xu. (1996). "A density-based algorithm for discovering clusters in large spatial databases with noise".

scikit learn: Evaluation of outlier detection estimators, in: scikit learn, o. D., https://scikit-learn.org/stable/auto\_examples/miscellaneous/plot\_outlier\_detection\_bench.html (abgerufen am 22.05.23)

scikit learn: 3.3.2.15. Receiver operating characteristic (ROC), in: scikit learn, o. D., https://scikit-learn.org/stable/modules/model\_evaluation.html#roc-metrics (abgerufen am 22.05.23)