Seminar-Vortrag „Vergleich von Data-Mining-Verfahren zur Outlier Detection“ - Skript

Verbesserungsvorschläge

* Ein Distanz- (evtl. eine aus {Control Chart, Linear Regression, Manhattan Distance Techniques} (Q3)) und ein dichtebasiertes Verfahren, diese vergleichen. Nicht gut und schlecht, sondern nach Anwendungsfall.
* Inhalt u. a.: Einführung zu Outlier-Detection-Verfahren, grobe Kategorien, ein Beispiel
* Vergleichskriterium, z. B. ob oder wie nominal

# DBSCAN

* “Density-Based Spatial Clustering of Applications with Noise”
* Ein beliebtes dichtebasiertes Gruppierungsverfahren
* Ausreißer sind diejenigen Punkte, die in einem Radius von epsilon (**ε-Umgebung**) weniger als p (**Dichte**) Nachbarn haben
* **ε**: Ein festgelegter Radius, der die Reichweite der ε-Umgebung bestimmt
* **minPts**: Eine Anzahl an Punkten, die eine Mindestdichte festlegt
* **Kernpunkt („Core Point“)**: Ein Datenpunkt gilt als Kernpunkt, wenn sich in einem Umkreis von ε um ihn herum, mindestens minPts Datenpunkte (inkl. sich selbst) befinden.
* **Direkt-dichteerreichbar („Direct Density Reachable“)**: Ein Punkt q ist von einem Punkt p direkt-dichteerreichbar genau dann, wenn p in seiner ε-Umgebung mindestens minPts Nachbarn hat und q einer dieser Nachbarn ist (wenn q weniger als minPts Nachbarn in seiner ε-Umgebung hat gilt die direkte Dichteerreichbarkeit nur in eine Richtung: von p nach q).
* **Dichteerreichbar („Density Reachable“)**: Ein Punkt q ist von einem Punkt p dichteerreichbar, wenn zwischen p und q Punkte liegen, die der Reihe nach, von p nach q führend, direkt-dichteerreichbar sind.
* **Dichteverbunden („Density Connected“)**: Zwei Punkte p und q sind dichteverbunden, wenn sie von einem dritten Punkt o dichteerreichbar sind.
* **Randpunkt („Border Point“)**: Ein Punkt, der in der ε-Umgebung eines anderen Kernpunktes liegt, aber selbst kein Kernpunkt ist.
* **Rauschpunkt („Noise“)**: Ein Punkt, der weder Kern- noch Randpunkt ist.
* OPTICS, LSDBC und HDBSCAN (Border Points gehören hierbei nicht automatisch zu einem Cluster) sind DBSCAN-Varianten, die Gruppen auf verschiedenen Hierarchie-Ebenen finden können

Quellen:

https://sci-hub.se/10.1145/3068335

file:///C:/Users/Jonas%20Brauer/Downloads/v91i01.pdf

Fragen:

* Data-Mining hypothesenfrei – wir sagen dem Verfahren nicht, wonach es suchen soll. Inwiefern ist die Bestimmung der DBSCAN-Parameter hypothesenfrei?
* Wie bestimmt man die Parameter?

# Alt:

# Einleitung

Ich halte heute meinen Vortrag über „Vergleich von Data-Mining-Verfahren zur Outlier Detection“. Wir fangen einfach an, aber später wird es sehr mathematisch, daher ist eure volle Aufmerksamkeit vonnöten.

***[🡺Gliederung]***

Stell dir vor, du arbeitest beim Hamburger Verkehrsministerium und sollst aus **Geschwindigkeitsmessungen** an einem Standort **aussagekräftige Kennzahlen** ermitteln.

Dich interessiert im ersten Schritt, ob die Autofahrer im **Durchschnitt** das Tempolimit einhalten und wie groß der **Unterschied** bei den gefahrenen Geschwindigkeiten ist. Dazu willst du den **Mittelwert** bzw. die **empirische Standardabweichung** der Geschwindigkeit berechnen.

Hierbei sind Messungen ein Problem, die stark vom Rest der Messpunkte **abweichen[[1]](#endnote-1)**, da diese sowohl den Mittelwert als auch die empirische Standardabweichung stark **verfälschen** können. Diese **Ausreißer** können z.B. vereinzelte extreme **Raser** oder **Polizei- und Rettungswägen** im Einsatz auf der einen oder **Stau** auf der anderen Seite sein.

# Univariate Statistik

In der **univariaten Statistik**, in der die Beobachtungen nur eine Variable haben, beschreibt man die Eigenschaft des **Mittelwerts** und der **empirischen Standardabweichung**, von Ausreißern verfälscht werden zu können, als **nicht robust**. Der **Breakdown-Value** ist der **minimale Anteil** an Ausreißern an den Messdaten, ab dem eine Kennzahl verfälscht wird undliegt für beide bei **0%**. Weiterhin ist ihre **Influence-Function** unbegrenzt, d.h. ein Ausreißer kann beide Kennzahlen unbegrenzt beeinflussen. Abhilfe schaffen können ähnliche, aber jeweils mit einem Breakdown-Value von **50%** erheblich robustere Kennzahlen wie der Median statt des Mittelwerts und dem MAD, statt der empirischen Standardabweichung.[[2]](#endnote-2)

Der **Median** ist bei sortierten Elementen der Wert, der an der mittleren Stelle steht. Selbst wenn knapp 50% der Geschwindigkeitsmessungen extreme Raser sein würden, verändert dies den Median überhaupt nicht. Der **MAD** ist der Median aller Abstände zum Median, multipliziert mit einem Korrekturfaktor von 1,483 bei Normalverteilungen.[[3]](#endnote-3)

Um Ausreißer zu identifizieren, kann bei symmetrischer Verteilung z.B. **Tukey‘s Boxplot** verwendet werden. Als Ausreißer werden alle Datenpunkte gewertet, die außerhalb des „**Fence**“ von mehr als 1.5\*IQR außerhalb der mittleren Hälfte der Datenpunkte liegen. Die IQR, die **Interquartile Range**, ist die Differenz zwischen dem größten und dem kleinsten Datenpunkt dieser mittleren Hälfte. Sie gibt die *Spannweite* der Werte der mittleren Hälfte der Beobachtungen an und hat einen Breakdown-Value von **25%**.[[4]](#endnote-4)

Ausreißer können auch u. a. mithilfe von Regeln bestimmt werden, z. B. seien alle Datenpunkte mit einem **Z-Score** > 1 Ausreißer.

* Ein Z-Score ist das Verhältnis des Abstandes eines Datenpunktes vom Mittelwert zur Standardabweichung.
* Eine **robuste Variante** der Z-Score-Berechnung nimmt statt des unrobusten Mittelwerts den robusteren **Median** und statt der unrobusten Standardabweichung den robusteren **MAD**.[[5]](#endnote-5)

# Multivariate Statistik

In unserem Beispiel haben wir bisher nur die Geschwindigkeit betrachtet und **andere Dimensionen**, wie z.B. die Tageszeit oder den Wochentag **ignoriert**. Betrachten wir nun Datenpunkte mit mehr als einer Variablen, bewegen wir uns im Bereich der **Multivariaten Statistik**. Beim Data-Mining untersucht man Daten, die üblicherweise mehr als eine Variable besitzen und bei denen man **Beziehungen zwischen Variablen** entdecken will, so dass **mindestens eine Variable von mindestens einer anderen Variable abhängig ist**.[[6]](#endnote-6)

Man versucht, **Korrelationen** zu entdecken, da sich hinter diesen eine mögliche **Kausalität** befinden kann.

Ausreißer zu finden kann nützlich sein, einmal, um gewonnene Daten nicht zu verfälschen und somit die falschen Schlüsse aus ihnen zu ziehen (--> man will die Ausreißer los werden), aber auch, um abnormales Verhalten festzustellen, z.B. um Kreditkartenbetrug zu erkennen und dagegen vorzugehen (--> man will an die Ausreißer ran).[[7]](#endnote-7)

Im weiteren Verlauf meines Vortrags will ich auf **fünf Verfahren zur Ausreißer-Erkennung** eingehen und diese miteinander **vergleichen**.

## Mahalanobis-Distanz & Robuste Distanz

Die **Mahalanobis-Distanz** ist ein Maß für die Entfernung eines Datenpunkts vom **Zentrum** der Punktwolke. Man definiert damit solche Punkte als **Ausreißer**, die eine bestimmte Mahalanobis-Distanz **überschreiten**. Das Problem hierbei ist aber, dass die Mahalanobis-Distanz anfällig für den **Maskierungseffekt** ist. Dabei werden klassische Kennzahlen wie Mittelwert und Kovarianzmatrix **von Ausreißern beeinflusst** und infolgedessen auch die Mahalanobis-Distanz, die mithilfe dieser Kennzahlen berechnet wird. Sie erkennt dadurch Ausreißer weniger gut. An ihrer Stelle kann die **Robuste Distanz** verwendet werden, die wie die Mahalanobis-Distanz berechnet wird, mit dem Unterschied, dass ein Mittelwert und eine Kovarianzmatrix aus, durch **MCD** von Ausreißern **bereinigten Daten**, ermittelt wird.[[8]](#endnote-8)

Die MCD ist die **Minimum-Covariance-Determinante**. Hierbei handelt es sich um einen **robusten** Schätzer **multivariater** Beobachtungen, der **Ausreißer aussortiert**, indem eine **festgelegte Anzahl** von Beobachtungen ausgewählt wird, deren **Kovarianz-Matrix** die **kleinstmögliche Determinante** besitzt.[[9]](#endnote-9)

Die **Kovarianz-Matrix** wiederum ist eine **symmetrische** Matrix, die den **Zusammenhang** jeweils zweier **Zufallsvariablen** beschreibt.[[10]](#endnote-10)

Die MCD kann nur bei **unimodalen**, **symmetrischen Verteilungen** verwendet werden.[[11]](#endnote-11) Dies würde eine Verwendung in unserem Verkehrsbeispiel wahrscheinlich ausschließen, da es nicht nur eine Hauptverkehrszeit, sondern wochentags mindestens zwei gibt – eine vor und eine nach der Arbeit.

## Local-Outlier-Factor

Der **Local-Outlier-Factor (LOF)** gehört zu den **dichtebasierten** Verfahren. Für alle Punkte wird der LOF berechnet und die Datenpunkte als Ausreißer definiert, die größer oder gleich einem **Schwellenwert** (engl. „threshold“) δ sind. Der LOF eines Punktes x wird wie folgt ermittelt: Die **k nächsten Nachbarn** von x werden ermittelt (bei mehreren Datenpunkten mit der **gleichen maximalen Distanz** zu x, dürfen es entsprechend mehr als k Nachbarn sein). Von diesen Nachbarn wird der **Durchschnitt ihrer Local-Reachability-Density** genommen und durch die von x geteilt. Die Local-Reachability-Density eines Punktes ist wiederum der **reziproke Durchschnitt** der **Maxima** aus den **Entfernungen des Punktes zu seinen Nachbarn** und **deren Entfernungen zu deren weitesten Nachbarn**.[[12]](#endnote-12)

Der Vorteil dieses Verfahrens gegenüber der **robusten Distanz** ist, dass es mit **multimodalen** Verteilungen umgehen kann. Während die Robuste Distanz nur **globale Ausreißer** finden kann, kann das LOF-Verfahren, auch **lokale Ausreißer**, wie z.B. einzelne Datenpunkte zwischen zwei Häufungspunkten, finden.

## k-nächste-Nachbarn-Methode

Die **k-nächste-Nachbarn-Methode** gehört zu den **entfernungsbasierten** Verfahren der Ausreißererkennung. Hierbei wird für jeden Punkt seine k-nächsten-Nachbarn und die einzelnen Entfernungen zwischen ihm und diesen bestimmt. Die n Punkte mit der **größten Maximalentfernung** werden als Ausreißer definiert.[[13]](#endnote-13)

## Stahel-Donoho-Outlyingness

Die **Stahel-Donoho-Outlyingness** ist ein robuster Schätzer für Lage und Streuung, der im Gegensatz zu manch anderem Verfahren **nicht diskret** Datenpunkte ganz oder gar nicht in die Berechnung von statistischen Kennzahlen einbezieht, sondern der die Datenpunkte **kontinuierlich gewichtet**, je nach Stärke ihrer „**Ausreißerhaftigkeit**“. Wie abseits ein Datenpunkt liegen muss, um **als Ausreißer zu gelten**, ist nämlich **Definitionssache**. Aus den gewichteten Datenpunkten wird anschließend ein **gewichtetes Mittel** und eine **gewichtete Kovarianzmatrix** berechnet.[[14]](#endnote-14)

# Fazit

Nicht jedes Verfahren zur Erkennung von Ausreißern lässt sich in jedem **Anwendungsfall** nutzen. Die **Robuste Distanz** funktioniert nur bei **unimodalen, symmetrischen Verteilungen**. Auch ist nicht jedes Verfahren gleich gut. Bei der Nutzung der Mahalanobis-Distanz werden tendenziell weniger Ausreißer, da erstere von letzteren abhängt. Stattdessen sollte man besser die Robuste Distanz verwenden.

Für multimodale Verteilungen funktionieren diese distanzbasierten Verfahren nur eingeschränkt. Lokale Ausreißer können mit dem k-nächste-Nachbarn-Verfahren oder, noch besser, mit dem Local-Outlier-Factor-Verfahren erkannt werden.

# Codebeispiel

*(Veranschaulichung am Programm durchführen)*

# Rohmaterial

Vgl. Q3S2 - Grobe Einteilung von Verfahren zur Erkennung von Ausreißern in drei Ansätze: statistisch (Control Chart, Linear Regression Techniques), entfernungsbasiert (Manhattan distance technique), abweichungsbasiert.

Q1S4 - **Breakdown-Values** einiger multivariater Schätzer:

* M-, R-, L-Estimator: 0% (due to vulnerability to bad leverage points)
* MCD: (n − h + 1)/n (highest when h= ⌊(n+p+1)/2⌋)

Q1S4 - **Lineare Regression**: Methoden zur Schätzung der **Koeffizientenmatrix θ**:

* **LS (least squares)**: Eine Gerade wird so durch vorhandene normalverteilte Punkte gelegt, dass die Summe der Residuenquadrate minimal sind. Sehr anfällig für Regressionsausreißer.
* **LTS (least trimmed squares)**: Wie LS, nur dass wahlweise ein größter Anteil an Residuenquadraten ignoriert wird. Ein schneller Algorithmus, namens FAST-LTS, ist unter der Fußnote 22 zu finden.

Q1S4 - **Standardisierte LTS-Residuen**: Residuum teilen durch die Wurzel aus dem Produkt des Quadrats einer Konstante und des Durchschnitts der Quadrate der ausgewählten unteren Residuen.

Vgl. Q6S2 – {Die **Regressionsanalyse** ist eine statistische Methode. Bei dieser gibt es zwei Herangehensweisen. In beiden wird vorausgesetzt, dass man **vorab** über die **Fehlerverteilung** und die **Abhängigkeiten** der variablen Bescheid weiß.

In der **ersten Herangehensweise**, der **Reverse Search**, wird die Regression mit allen Punkten durchgeführt und die mit den z.B. größten Residuen-Quadraten als Ausreißer definiert.

In der **zweiten Herangehensweise**, der **Direct Search**, wird die Regression mit einer Teilmenge der Punkte durchgeführt und nach und nach die Punkte hinzugefügt, die die geringsten Abweichungen aufweisen. Die zuletzt hinzugefügten Punkte werden als Ausreißer definiert.}

Vgl. Q6S3 – {Die DB(p, D)-Methode ist eine entfernungsbasierte Herangehensweise, bei der Punkte als Ausreißer gelten, die weiter als die entfernt sind von einem Anteil p von allen Punkten.

Ein Nachteil bei dieser Methode ist ein relativ hoher Rechenaufwand wegen der Betrachtung aller Punkte und dass man p und D vorher wissen muss. Außerdem werden nicht am Rande liegende Ausreißer oft ignoriert.}

1. Vgl. Zuriana, A. B., Rosmayati, M., Akbar,  
   A., Mustafa, M. D.. (2006). „A Comparative Study for Outlier Detection Techniques in Data Mining“. IEEE, 1-4244-0023-6/06, S. 1 (Originalquelle, zitiert nach gelesener Quelle) [↑](#endnote-ref-1)
2. Vgl. Hubert, M., Rousseeuw, P. J.. (2011). “Robust statistics for outlier

   detection”. John Wiley & Sons, Inc., WIREs Data Mining Knowl Discov, 2011, 1, 73–79, DOI: 10.1002/widm.2, S. 1-2. [↑](#endnote-ref-2)
3. Vgl. Hubert, M., Rousseeuw, P. J.. (2011). “Robust statistics for outlier

   detection”. John Wiley & Sons, Inc., WIREs Data Mining Knowl Discov, 2011, 1, 73–79, DOI: 10.1002/widm.2, S. 1-2. [↑](#endnote-ref-3)
4. Vgl. Hubert, M., Rousseeuw, P. J.. (2011). “Robust statistics for outlier

   detection”. John Wiley & Sons, Inc., WIREs Data Mining Knowl Discov, 2011, 1, 73–79, DOI: 10.1002/widm.2, S. 2.

   Vgl. Ismiguzel, Idil: Outlier Detection with Simple and Advanced Techniques: A tutorial on how to detect outliers using standard deviation, interquartile range, isolation forest, DBSCAN, and local outlier factor, in: Towards Data Science, 2022, https://towardsdatascience.com/detecting-outliers-with-simple-and-advanced-techniques-cb3b2db60d03, (abgerufen am 30.04.23) [↑](#endnote-ref-4)
5. Vgl. Hubert, M., Rousseeuw, P. J.. (2011). “Robust statistics for outlier

   detection”. John Wiley & Sons, Inc., WIREs Data Mining Knowl Discov, 2011, 1, 73–79, DOI: 10.1002/widm.2, S. 2. [↑](#endnote-ref-5)
6. Vgl. Petrovskiy, M. I.. (2003). “Outlier Detection Algorithms in Data Mining Systems”. Programming and Computer Software, Vol. 29, No. 4, 2003, pp. 228–237. Translated from Programmirovanie, Vol. 29, No. 4, 2003. 0361-7688/03/2904-0228. S. 2 [↑](#endnote-ref-6)
7. Vgl. Zuriana, A. B., Rosmayati, M., Akbar,  
   A., Mustafa, M. D.. (2006). „A Comparative Study for Outlier Detection Techniques in Data Mining“. IEEE, 1-4244-0023-6/06, S. 2 [↑](#endnote-ref-7)
8. Vgl. Hubert, M., Debruyne, M., Rousseeuw, P. J.. (2017). “Minimum Covariance Determinant and Extensions”. arXiv:1709.07045v1, S. 3-4. [↑](#endnote-ref-8)
9. Vgl. Hubert, M., Rousseeuw, P. J.. (2011). “Robust statistics for outlier

   detection”. John Wiley & Sons, Inc., WIREs Data Mining Knowl Discov, 2011, 1, 73–79, DOI: 10.1002/widm.2, S. 3. [↑](#endnote-ref-9)
10. Vgl. Hubert, M., Rousseeuw, P. J.. (2011). “Robust statistics for outlier

    detection”. John Wiley & Sons, Inc., WIREs Data Mining Knowl Discov, 2011, 1, 73–79, DOI: 10.1002/widm.2, S. 3. [↑](#endnote-ref-10)
11. Vgl. scikit learn: sklearn.covariance.MinCovDet, in: scikit learn, o. D., https://scikit-learn.org/stable/modules/generated/sklearn.covariance.MinCovDet.html (abgerufen am 01.05.23) [↑](#endnote-ref-11)
12. Vgl. Ming-Chao Guo, Shi-Jie Pan, Wen-Min Li, Fei Gao, Su-Juan Qin, Xiao-Ling Yu, Xuan-Wen Zhang, Qiao-Yan Wen. (2023). “Quantum Algorithm for Unsupervised Anomaly Detection”. arXiv:2304.08710v1, S. 1-2 [↑](#endnote-ref-12)
13. Vgl. Petrovskiy, M. I.. (2003). “Outlier Detection Algorithms in Data Mining Systems”. Programming and Computer Software, Vol. 29, No. 4, 2003, pp. 228–237. Translated from Programmirovanie, Vol. 29, No. 4, 2003. 0361-7688/03/2904-0228. S. 3 [↑](#endnote-ref-13)
14. Vgl. Hubert, M., Rousseeuw, P. J.. (2011). “Robust statistics for outlier

    detection”. John Wiley & Sons, Inc., WIREs Data Mining Knowl Discov, 2011, 1, 73–79, DOI: 10.1002/widm.2, S. 3. [↑](#endnote-ref-14)