Seminar - Actuarial Machine Learning

Machine Learning Basics

Tugba Yilmaz

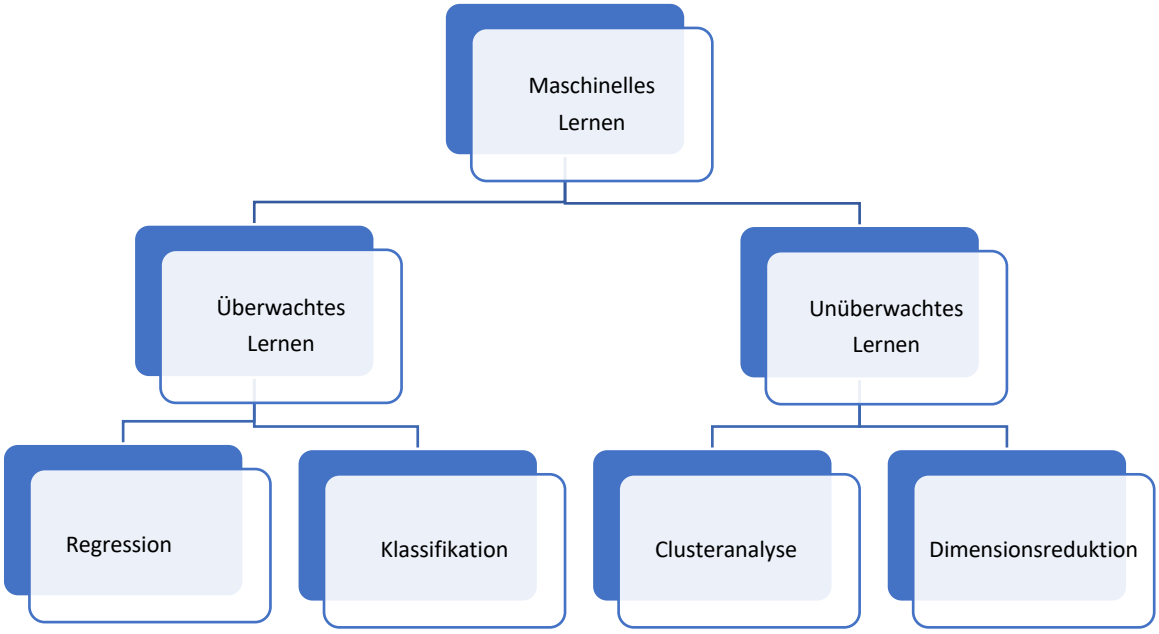
18. Dezember 2020

Dozent: Dr. Zoran Nikolic

Mathematisches Institut
Universität zu Köln

Definition:
Maschinelles Lernen ist ein Teilgebiet der künstlichen Intelligenz. Dabei wird “Wissen” aus “Erfahrung” generiert. D.h. Lernalgorithmen entwickeln aus vorgegebenen Testdaten Modelle, welche später unbekannte Daten so genau wie möglich abschätzen.

Grobe Übersicht des Maschinellen Lernens (ML)



Überwachtes Lernen (Supervised)

Zunächst betrachten wir wie überwachtes Lernen überhaupt abläuft. Es werden Trainingsdaten vorgegeben, von denen die Ergebnisse bereits bekannt sind. Ausgehend von den Trainingsdaten werden mit Hilfe von Lernalgorithmen Modelle erstellt, die die Testdaten möglichst genau abschätzen.
Dieses Modell wird in den Testdaten angewendet, welche im Allgemeinen einen größeren Datensatz als Trainingsdaten beinhalten. Die Abschätzung der Testdaten wird daraufhin anhand der realen Ergebnisse kontrolliert. Wenn das Modell eine gute Abschätzung liefert, wird dieses an unbekannten Daten angewendet, andernfalls muss das Modell korrigiert werden und der Kreislauf fängt von vorne an.

Überwachtes Lernen teilt sich in Regression und Klassifikation auf. Die Ergebnisvariablen können als qualitative oder quantitative Variablen bezeichnet werden. Dabei definieren quantitative Variablen numerische Ergebnisse (bspw. Alter, Gewicht, Größe, …) und qualitative definieren Ergebnisse die in K verschiedene Kategorien aufgeteilt werden können (Geschlecht, Marke, Diabetes 1 oder 2).

Regression	Klassifikation
Quantitative Variablen: • Numerische Variablen (Alter, Gewicht, Größe, …) • Lineare Regression; Least squares	Qualitative Variablen • In K verschiedene Kategorien aufteilbare Variablen (Geschlecht, Marke, Diabetes 1 oder 2, …) • Logistische Regression

Vorgehensweisen:

Angenommen unsere Zielfunktion ist gegeben als $Y = f(X) + \varepsilon$ mit $f(X)$ als unbekannte Funktion von Merkmalen (bzw. Beobachtungen) X und ε als Fehlerterm.
Die Vorgehensweisen teilen sich in zwei Kategorien auf.

Prediction

Für diese Vorgehensweise ist die reale Form von der Datenfunktion irrelevant. Wir suchen lediglich eine Funktion, die unsere Daten so genau wie möglich abschätzt. Dabei tauchen natürlich Fehler auf, wobei man zwischen reduzierbaren und irreduzierbaren Fehlern unterschiedet. Fehler können in der Abschätzung selber liegen oder auch bei der Messung der Daten auftauchen. Wenn wir uns den Erwartungswert der quadrierten Differenz ansehen können wir diese Aufteilung recht schnell identifizieren.

$$E\left[Y - \hat{Y}\right]^2 = E\left[f(X) + \varepsilon - \hat{f}\right]^2 = \underbrace{[f(X) - \hat{f}(x)]^2}_{\text{Reduzierbar}} + \underbrace{Var(\varepsilon)}_{\text{Irreduzierbar}} \quad (1)$$

Das Ziel ist es die reduzierbaren Fehler zu minimieren und somit eine genaue Abschätzung der Daten zu erhalten.

Inference

Die Inferenz im Gegenzug zur Prognose (Prediction) will f abschätzen und gibt in der Regel keine guten Näherungen für Y. Dabei interessiert uns eher die Relation zwischen X und Y. Die bekanntesten Fragestellungen dieser Vorgehensweise sind:

- Was sind die wichtigsten Merkmale?
- Wie reagiert Y auf Veränderungen dieser Merkmale?
- Wie kann ich diese Merkmale zusammenfassen?

Methoden zur Modell Erstellung

Parametrisch	Nicht Parametrisch
Erste Überlegungen für f: $\hat{f}(X) = \beta_0 + \sum_{j=1}^{p} \beta_j X_j$ $\hat{f}$ anpassen	• Sucht keine spezifische Funktion für f • Schätzt f anhand der gegebenen Daten ab • Hat das Potenzial einen größeren Bereich möglicher Formen für f genau anzupassen

Nachteil: • Modell schwankt von tatsächlicher Form von f ab • Fehler können priorisiert werden	Nachteil: • Braucht viel mehr Daten für eine optimale Abschätzung

Fehlererkennung:

Jedes Modell des überwachten Lernens bringt Approximationsfehler mit sich. Die Genauigkeit dieser Abschätzungen wird anhand der *mittleren quadratischen Abweichung (MSE)* bemessen.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2 \qquad (2)$$

Wir können die erwartete Abweichung wie folgt Zerlegen.

$$E\left[(y - \hat{f}(x))^2\right] = \underbrace{Bias[\hat{f}(x)]^2}_{\text{Verzerrung}} + \underbrace{Var\left[\hat{f}(x)\right]}_{\text{Varianz}} + \underbrace{\sigma^2}_{\text{Fehlerterm}} \qquad (3)$$

Varianz: Der Betrag, um den sich $\hat{f}$ verändern würde, wenn ein anderer Datensatz genutzt wird.

Bias: Bezieht sich auf den Fehler, der durch die Annäherung eines echten Problems eingeführt wird.

Least squares

Die meisten Modelle sind von linearem Ursprung der Form $\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j\hat{\beta}_j$ auch geschrieben als $\hat{Y} = X^T\hat{\beta}$. Für die Anpassung des Modells an die Daten existieren verschiedene Methoden. Hier betrachten wir die *least squares* Methode, womit die Koeffizienten β mit *residual sum of squares(RSS)* minimiert werden.

$$\begin{aligned} \text{RSS}(\beta) &= \sum_{i=1}^{N}(y_i - x_i)^2 \\ &\Leftrightarrow (y - X\beta)^T(y - X\beta) \qquad |\text{Differenziere nach } \beta \\ &\Longrightarrow X^T(y - X\beta) = 0 \end{aligned} \qquad (4)$$

falls XX^T nicht singulär ist, folgt $\hat{y}_i = x_i^T\hat{\beta}$ als Abschätzung der i-ten Inputvariable.

K- nearest-neighbors

K-nearest-neighbors schätzt mit den Beobachtungen der Trainingsdaten die nächsten Nachbarn des Inputs X ab.
Dabei definieren wir das Modell wie folgt:

$$\hat{Y}(X) = \frac{1}{k}\sum_{x_{i\in N_k(x)}} y_i \qquad (5)$$

$N_k(x)$ ist definiert als die Nachbarschaft von x mit den k nächsten Nachbarn, wobei k die Anzahl der Nachbarn darstellt. Die Distanz wird hierbei mit der euklidischen Distanz berechnet

Unüberwachtes Lernen (Unsupervised)

Beim unüberwachten Lernen werden Daten genutzt, die keine bekannten Ergebnisse liefern. Daher arbeiten wir so gesehen blind und können den Algorithmus nicht wie beim überwachten Lernen trainieren.
Wir können aber Algorithmen finden, die die Struktur der Daten sinnvoll zusammenfassen. Eine Strukturierung ist jedoch nicht immer erfolgreich.
Das unüberwachte Lernen unterscheidet zwischen zwei Methoden: Clusteranalyse und Dimensionsreduktion.

Clusteranalyse

1. *K-Means Clusteranalyse*

K-Means Clusteranalyse weist die Daten zu gegebenen K-Clustern zu.
Wichtig ist, dass die Cluster nicht überlappen und jede Observierung maximal einem Cluster zugehörig ist. Die Idee hinter der K-Means Funktion ist, dass die Varianz W(c_K) innerhalb der Cluster so klein wie möglich gehalten wird.

$$\mathrm{W}(C_K) = \frac{1}{|C_K|}\sum_{i,i' \in C_K}\sum_{j=1}^{p}(x_{ij} - x_{i'j})^2 \qquad (6)$$

Algorithmus:
1. Weise jeder Beobachtung zufällige Zahlen von 1 bis K zu.
2. Wiederhole folgendes bis keine Veränderungen auftauchen:
    1. Berechne für jeden der K -Cluster einen Schwerpunkt
    2. Ordne jede Beobachtung dem Cluster zu, dessen Schwerpunkt am nächsten liegt. (Euklidischer Abstand)

Wichtig ist zu wissen, dass dieser Algorithmus das lokale Optimum bestimmt und nicht das globale Optimum. Daher ist es von äußerst großer Bedeutung den Algorithmus öfter anzuwenden, um eine Optimale Clusterverteilung zu finden.
Vorteil dieser Methode ist der schnelle Algorithmus, der geringe Speicherbedarf und die Anwendbarkeit verschiedener Datentypen.
Nachteilig ist jedoch, dass die vordefinierte Clusteranzahl K nicht immer den tatsächlichen Gruppierungen entspricht.

2. *Hierarchische Clusteralanyse*

Hierarchische Clusteranalyse entwickelt eine Baumartige Visualisierung der Daten (Dendogramm).
Da der Algorithmus eine sehr hohe Interpretierbarkeit liefert, betrachten wir die Informationen, welche aus einem Dendogramm interpretiert werden können.

- Je früher sich die Blätter zusammenfügen, desto ähnlicher sind sie zueinander.
- Cluster finden wir, indem wir das Dendogramm horizontal schneiden.
- Zusammengefügte Zweige können vertauscht werden, ohne die Bedeutung des Dendogramms zu verändern.

Es existieren 2^{n-1} mögliche Zusammensetzungen von Dendogrammen.

Algorithmus:

1. Beginne mit n Beobachtungen und einem Maß aller $\binom{n}{2} = n(n-2)/2$ paarweise Unterschiede und betrachte diese als Cluster.
2. $i = n$
    1. Betrachte alle paarweise Unterschiede der i Cluster. Finde ein Clusterpaar der sich am wenigsten unterscheidet und verknüpfe diese. (Die Unähnlichkeit zwischen diesen beiden Clustern gibt die Höhe im Dendrogramm an, an dem die Verknüpfung platziert werden soll.)
    2. Berechne die neuen paarweisen Intercluster-Unterschiede zwischen den verbleibenden $i - 1$ Clustern.

Als Unähnlichkeitsmaß, wird oft die euklidische Distanz verwendet. Es existieren aber auch viele andere Unähnlichkeitsmaße.
Manchmal weiß man nicht, wie die Cluster miteinander verknüpft werden sollen, deshalb werden diese in folgender Tabelle definiert.

Verknüpfung	**Beschreibung**
Complete	Maximaler Abstand aller Elementpaare aus den Clustern A und B $D(A,B) \coloneqq \max_{a\in A, b\in B}\{d(a,b)\}$
Single	Minimaler Abstand aller Elementpaare aus den Clustern A und B $D(A,B) \coloneqq \min_{a\in A, b\in B}\{d(a,b)\}$
Average	Durchschnittlicher Abstand aller Elementpaare aus den Clustern A und B $D(A,B) \coloneqq \frac{1}{\lvert A\rvert\lvert B\rvert}\sum_{a\in A, b\in B} d(a,b)$
Centroid	Abstand der Zentren der beiden Cluster $D(A,B) \coloneqq d(\bar{a},\bar{b})$ $\bar{a}$ ist Zentrum des Clusters A und $\bar{b}$ st Zentrum des Clusters B

Complete, Average und Single sind die meistgebrauchten Verknüpfungen.

<u>Hauptkomponenten Analyse PCA</u> (Dimensionsreduktion)

Angenommen wir wollen n Beobachtungen und p Merkmale $X_1, \dots, X_p$ visualisieren. Wir suchen eine niedrig dimensionale Repräsentation der Daten mit möglichst wenig Informationsverlust.
PCA liefert genau das.

Jede gefundene Dimension ist eine Linearkombination von P Merkmalen.
Es stellt sich die Frage, wie diese Dimensionen (principal components) gefunden werden. Die Berechnung der Hauptkomponenten hilft uns hierbei.

1. Erste Hauptkomponente

Wir suchen eine normierte Linearkombination Z_1 mit größter Varianz

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{p1}X_p;\ \sum_{j=1}^{p} {\phi_{j1}}^2 = 1 \tag{7}$$

Dabei bezeichnen wir $\phi_1 = \begin{pmatrix}\phi_{11}\ \phi_{21}\ \dots\ \phi_{p1}\end{pmatrix}$, als loading Vektor.

Zur Berechnung der ersten Hauptkomponente betrachten wir Stichproben Merkmale

$$z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + \cdots + \phi_{p1}x_{p1} \tag{8}$$

Ziel ist es,

$$\underset{\phi_{11},\dots,\phi_{p1}}{\text{maximiere}} \left\{\frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j1}x_{ij}\right)^2\right\} \text{ mit } \sum_{j=1}^{p}{\phi_{j1}}^2 = 1 \tag{9}$$

zu lösen, welches mit Methoden aus der Linearen Algebra lösbar ist.

Geometrisch gesehen können wir annehmen, dass Z_1 eine Koordinaten-Achse aufspannt, dessen Richtung mit ϕ_1 gegeben ist.

2. Zweite Hauptkomponente

Die Berechnung der zweiten Hauptkomponente ist recht einfach, da wir eine Linearkombination von $X_1, \dots, X_p$ suchen, die gleichzeitig auch unkorreliert zu Z_1 ist. Diese Einschränkung ist äquivalent zu $\phi_2 \perp \phi_1$.

Für $p > 2$ lösen wir ein (9) ähnliches Problem, wobei wir ϕ_1 mit ϕ_2 ersetzen und zusätzlich annehmen, dass zu $\phi_2 \perp \phi_1$ ist.
Wenn alle Hauptkomponenten berechnet wurden, können wir sie gegeneinander Plotten, um niedrigdimensionale Visualisierungen zu erhalten.

Informationsverlust

Um uns zu vergewissern, dass bei der Dimensionsreduktion keine Informationen verlorengegangen sind, können wir das PVE (Proportion of Variance Explained) berechnen.
Dafür benötigen wir die totale Varianz der Daten und die Varianz der m-ten Hauptkomponente.

totale Varianz:

$$\sum_{j=1}^{p} Var(X_j) = \sum_{j=1}^{p}\frac{1}{n}\sum_{i=1}^{n}x_{ij}^2 \tag{10}$$

Varianz der m-ten Hauptkomponente:

$$\frac{1}{n}\sum_{i=1}^{n} z_{im}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p} \phi_{j1} x_{ij}\right)^2 \tag{11}$$

$$PVE = \frac{\text{die totale Varianz der Daten}}{\text{Varianz der m−ten Hauptkomponente}} = \frac{\sum_{i=1}^{n}\left(\sum_{j=1}^{p} \phi_{j1} x_{ij}\right)^2}{\sum_{j=1}^{p}\sum_{i=1}^{n} x_{ij}^2} \tag{12}$$

Literatur

- http://faculty.marshall.usc.edu/gareth-james/ISL/
- https://web.stanford.edu/~hastie/ElemStatLearn/
- https://www.iais.fraunhofer.de/content/dam/bigdata/de/documents/Publikationen/Fraunhofer_Studie_ML_201809.pdf
- http://statweb.stanford.edu/~tibs/stat315a/LECTURES/chap2.pdf
- https://cs.nju.edu.cn/zlj/Course/DM_15_Lecture/Lecture_12.pdf