Mai 2017 / by MIOsoft

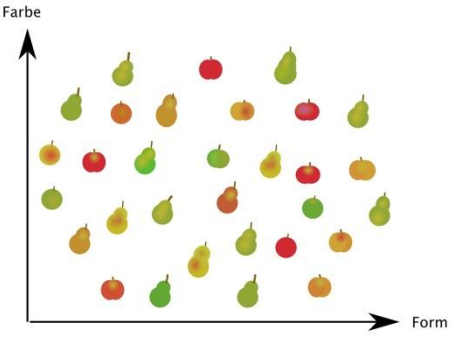
Machine Learning: der Weg zu echter Computer-Intelligenz?

Glaubt man dem, was vor dem Hintergrund des digitalen Wandels in der Presse über Maschinenintelligenz kolportiert wird, dann sind Computer schon jetzt die besseren Wahrsager: Offenbar können Computerprogramme allein anhand des Einkaufsverhaltens von Menschen vorhersagen, wann und wie sich die Lebensumstände dieser Menschen ändern (in den USA prognostizierte die Software eines Discounters bei einer jungen Frau eine baldige Schwangerschaft, die auch tatsächlich eintrat), und angeblich können sie Polizisten noch vor dem Stattfinden eines Verbrechens an den voraussichtlichen Tatort schicken. Das Geheimnis dieser Wahrsagerei ist maschinelles Lernen. Genauer: maschinelles Lernen und Big Data. Dank Machine Learning können Computer auf der Basis von ausreichend großen Datenmengen recht zuverlässig Zukunftsprognosen erstellen. Was aber ist Machine Learning?

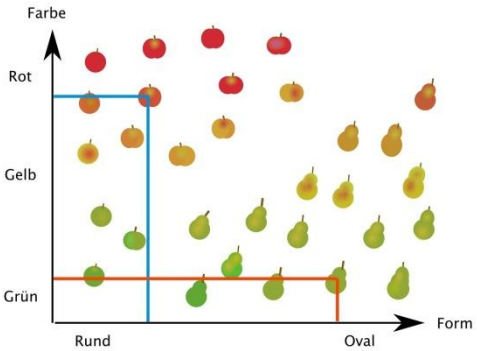
Wie lernt eine Maschine?
Wie jedes Lernen ist auch maschinelles Lernen die Generierung von Wissen aus gemachter Erfahrung. Wie aber kann eine Maschine Erfahrungen machen? Und wie generiert sie daraus Wissen? Betrachten wir ein Beispiel: In einem Obstkorb befinden sich Äpfel und Birnen. Es soll nun die Frage beantwortet werden: „Was genau enthält der Obstkorb?“ Für einen Menschen wäre die Sache einfach: Er würde die Früchte zählen, sie den Kategorien „Apfel“ bzw. „Birne“ zuordnen und am Ende sagen: „Der Obstkorb enthält x Äpfel und y Birnen!“ Eine Maschine hätte zunächst einmal das Problem, dass sie gar nicht wüsste, was ein Apfel und was eine Birne ist. Sie muss diesbezüglich erst instruiert werden – und zwar über Merkmale.

Wir brauchen also zuerst eine maschinenfreundliche Beschreibung der Früchte. Sehen wir uns zwei Früchte im Obstkorb an:

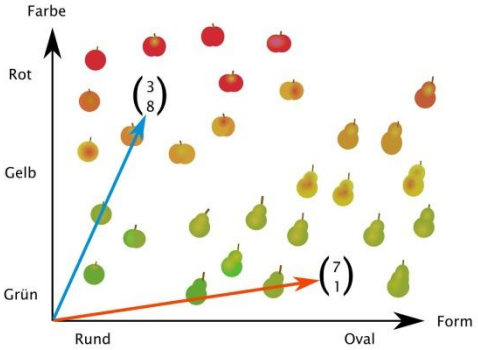
Eine Frucht ist grün, eine ist rot. Ein offensichtliches Unterscheidungsmerkmal (engl.: *Feature*) ist also die Farbe. Ein weiteres Unterscheidungsmerkmal ist die Form – sie kann rund sein oder eher oval. Mit diesen beiden Features – Farbe und Form – können wir einen zweidimensionalen Merkmalsraum (engl.: *Feature Space*) aufspannen:



Da es im Obstkorb nicht nur rote und grüne Früchte gibt, sondern auch Früchte, die zwischen Rot und Grün angesiedelt sind, positionieren wir die Früchte im Feature Space in der Dimension „Farbe“ aufsteigend von Grün über Gelb bis Rot und in der Feature-Dimension „Form“ von Rund bis Oval:



Die Klassifizierung jeder Frucht lässt sich nun durch einen Feature-Vektor darstellen. Zudem führen wir eine numerische Skala für die beiden Feature-Dimensionen ein: Die Dimension „Farbe“ soll von 1 (= Grün) bis 9 (= Rot) verlaufen, und auch die Dimension „Form“ soll von 1 (= Rund) bis 9 (= Oval) verlaufen. Nun lässt sich die Position jeder Frucht im Feature Space durch einen doppelten Zahlenwert darstellen:

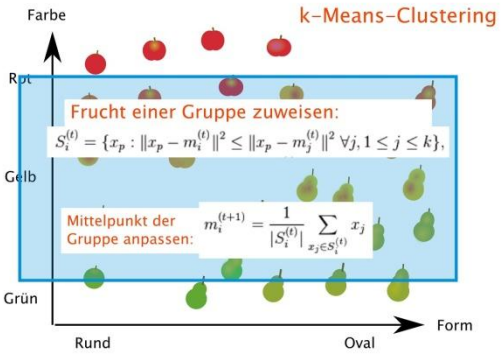


Noch maschinenfreundlicher wird die Beschreibung, wenn wir alle Feature-Vektoren in eine Tabelle schreiben:

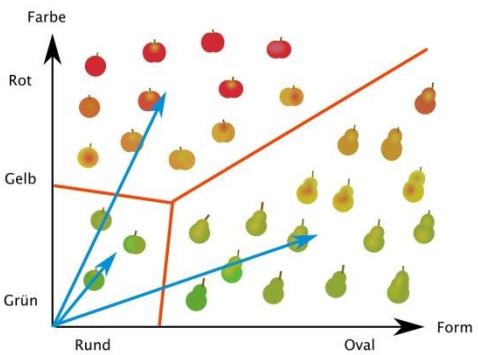
Frucht	Form	Farbe
1	4	9
2	3	7
3	3	2
4	5	1
5	1	1
6	6	3
7	7	4
…	…	…

Indem wir der Machine nun eine Reihe von Klassifizierungsmerkmal an die Hand gegeben haben, haben wir sie mit „Erfahrungen“ gespeist.

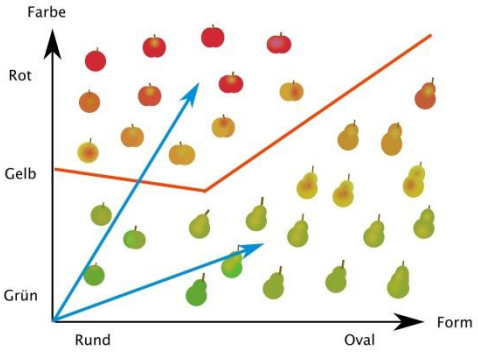
Wie aber generiert die Maschine aus dieser Erfahrung Wissen? Dazu benötigt sie einen Systematisierungs-Algorithmus wie z. B. Decision Tree, k-Means oder Support Vector Machine. Nehmen wir den Cluster-Algorithmus k-Means. Den genauen Aufbau des k-Means-Algorithmus darzulegen, würde den Rahmen dieses Textes sprengen, deshalb nur so viel: k-Means führt Objekte mit gleichen Merkmalen zu Clustern zusammen. Jedes Cluster wird dabei durch einen Feature-Vektor repräsentiert, der den Schwerpunkt des Clusters beschreibt. Um in unserem Fall zu bestimmen, welchem Cluster eine Frucht zuzuordnen ist, wird der geometrische Abstand zwischen den Clusterschwerpunkt-Vektoren und dem Feature-Vektor der jeweiligen Frucht gemessen. Die Frucht wird schließlich demjenigen Cluster zugeordnet, bei dem dieser Abstand am kleinsten ist. Je häufiger man diesen Vorgang wiederholt, desto besser wird die Clusterqualität.



Der Algorithmus beantwortet die Frage „Was genau enthält der Obstkorb?“ dann wie folgt:



Er hat drei Cluster für drei Arten von Früchten gebildet – seiner Ansicht nach enthält der Obstkorb also zum einen Äpfel, zum anderen Birnen und außerdem ein paar Drittfrüchte. Die Schwerpunkte der Cluster sind durch die blauen Vektoren markiert. Warum hat der Algorithmus nicht nur zwei Cluster – also ein Apfel-Cluster und ein Birnen-Cluster – gebildet? Weil bei der Verwendung von k-Means die Anzahl der Cluster vorgegeben werden muss. Wir setzen den Parameter „Cluster-Anzahl" also auf 2 und wiederholen die maschinelle Obstkorb-Analyse. Wir erhalten folgendes Ergebnis:



Die Antwort der Maschine lautet nun: „Es gibt zwei Gruppen von Früchten – eine Gruppe von eher roten Früchten und eine Gruppe von eher grünen Früchten!. Das Ergebnis ist also wiederum unbefriedigend. Warum? Zum einen haben wir der Maschine nicht genügend Informationen gegeben – wir haben ihr nämlich nicht gesagt, dass das Unterscheidungsmerkmal „Form" im Zweifel wichtiger ist als das Unterscheidungsmerkmal „Farbe". Zum anderen haben wir das Lernziel nicht klar formuliert: Wir hätten ganz konkret erklären müssen, dass wir eine Unterteilung der Früchte in die zwei Kategorien „Äpfel" und „Birnen" erwarten. Es gibt also zwei Möglichkeiten, das Ergebnis zu optimieren: Entweder geben wir der Maschine mehr Informationen, oder wir geben ihr ein Lernziel vor. In letzterem Fall spricht man von „überwachtem Lernen".

Lernen mit Ziel: überwachtes Lernen

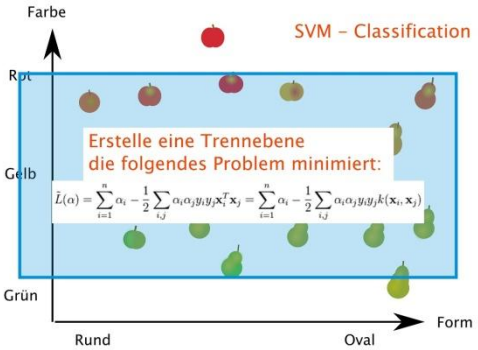
Was ändert sich, wenn wir zu überwachtem Lernen wechseln? Zunächst teilen wir die Daten in Testdaten und in Trainingsdaten auf. Die Trainingsdaten werden um eine Spalte „Spezies" erweitert – diese Spalte soll die Information enthalten, um welche Art von Frucht (Apfel oder Birne) es sich bei jedem Objekt handelt. Für die Testdaten muss diese Spalte durch den Machine-Learning-Algorithmus gefüllt werden.

Frucht Nr.	Form	Farbe	Spezies
1	4	9	Apfel
2	3	7	Apfel
3	3	2	
4	5	1	Birne
5	1	1	
6	6	3	
7	7	4	Birne
...	...	...	...

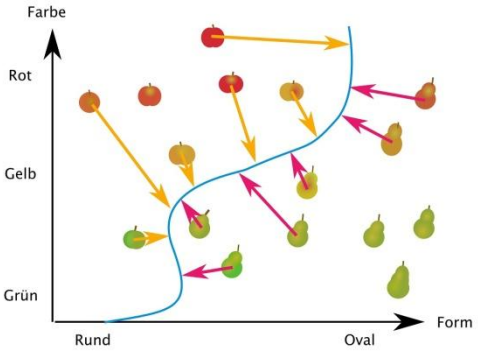
Außerdem haben wir nun zwei Lernphasen: eine Trainingsphase und eine Testphase.

Trainingsphase

In der Trainingsphase „üben" wir mit der Maschine, und zwar mithilfe des „Support-Vector-Machine"-Algorithmus. Der konzeptionelle Aufbau dieses Algorithmus ist noch komplizierter als der Aufbau des k-Means-Algorithmus, daher auch hier nur die Kurzfassung: In der Trainingsphase wird eine Hyperebene definiert, die den Feature Space in zwei Teile unterteilt. Jeder Teil bekommt eine Spezies zugewiesen.



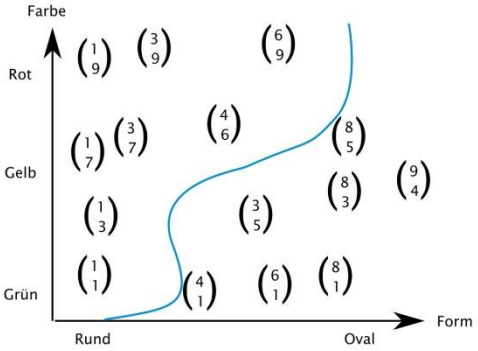
Die Darstellung einer Ebene erfordert eigentlich ein dreidimensionales Feld. Da unser Feature Space aber nur zweidimensional ist, müssen wir die Hyperebene als Linie darstellen:



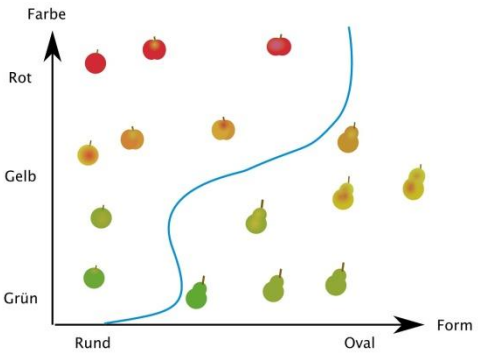
Die Hyperebene bzw. Hyperebenenlinie wird durch Stützvektoren (engl.: *Support Vectors*) definiert, die auf den Feature-Vektoren der einzelnen Früchte basieren.

Testphase

Die Trennung zwischen der Basisebene und der Hyperebene schafft die Grundlage für das Einleiten der Testphase. In der Testphase soll nun für jede Frucht dahingehend überprüft werden, ob sie auf die Basiseben oder auf die Hyperebene gehört bzw. ob sie links oder rechts von der Hyperebenenlinie liegen muss. Dazu wird für jede Frucht ein Skalarprodukt gebildet – dieses errechnet sich aus den Feature-Vektor-Zahlen der Frucht und dem Vektorwert der Hyperebene.



Je nachdem, ob die Frucht links oder rechts von der Hyperebenenlinie positioniert ist, ist das Skalarprodukt positiv oder negativ. Legt man nun fest, dass alle Früchte mit positivem Skalarproduktwert Äpfel sind und alle Früchte mit negativem Skalarproduktwert Birnen, dann kann die Spalte „Spezies“ gefüllt werden. Die Maschine ist nun endlich in der Lage, die Frage „Was genau enthält der Obstkorb?“ korrekt zu beantworten:



In Wirklichkeit sind die Abläufe natürlich wesentlich komplizierter. Eines der größten Probleme besteht z. B. darin, dass die Datenqualität oft unzureichend ist. Bezogen auf das Obstkorb-Beispiel bedeutet das, dass der Obstkorb in der Realität auch einige verfaulte Früchte enthalten würde, für die eine Kategorisierung obsolet wäre (weil niemand verfaultes Obst essen will, egal, ob es sich um verfaulte Äpfel oder um verfaulte Birnen handelt), die aber das Obstkorbanalyse-Ergebnis verfälschen würden. Daten, die als Grundlage für maschinelles Lernen dienen sollen, müssen also zunächst qualitativ bereinigt werden. Außerdem müssen sie auch noch so transformiert werden, dass sie als Input für Machine-Learning-Algorithmen taugen. Im Enterprise Umfeld kommt auch noch das Mengenproblem hinzu: Die Anzahl der Daten ist oftmals so riesig, dass spezielle Software-Architekturen nötig sind, um sie bewältigen zu können. Hier treten Spezialisten wie etwa das Unternehmen MIOsoft auf den Plan – sie bieten Tools und Lösungen an, mithilfe derer maschinelles Lernen als Schlüssel zu echter Computer-Intelligenz reibungslos funktioniert.