TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Informatik Institut für Systemarchitektur, Professur für Datenbanken

Diplomarbeit

MERKMALSAUSWAHL ZUR OPTIMIERUNG VON PROGNOSEPROZESSEN AUF VERKAUFSDATEN

Marcel Spranger
Matr.-Nr.: 3305508

Betreut durch:
Prof. Dr.-Ing. Wolfgang Lehner

Eingereicht am 30. November 2014

ERKLÄRUNG

Ich erkläre, dass ich die vorliegende Arbeit selbständig, unter Angabe aller Zitate und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Dresden, 30. November 2014

ABSTRACT

Das Ziel der Merkmalsauswahl ist es, von einer Menge von Attributen (Merkmalen) eine Untermenge von Attributen zu bestimmen, die für ein vorgegebenes Problem relevant sind. In dieser Arbeit werden die Einflüsse von Merkmalsauswahl-Algorithmen auf die Qualität von Verkaufsdatenvorhersagen mittels Regression untersucht. Dabei kommen neben Wrapper- und Filter-Methoden auch zwei hybride Varianten zum Einsatz. Durch diese Methoden konnte für alle untersuchten Datensätze eine Steigerung der Prognosegenauigkeit erreicht werden. Besonders gute Ergebnisse lieferten im Regelfall jeweils eine der beiden Wrapper-Methoden mittels Forward Selection und Backward Elemination, sowie durch der genetische Algorithmus, der zudem schneller terminiert. Doch auch die Filter-Methoden konnten in manchen Fällen den Prognosefehler besonders stark mindern. Jedoch konnte kein Algorithmus ausgemacht werden, der durchgängig für alle Datensätze überdurchschnittliche Ergebnisse erzielt.

INHALTSVERZEICHNIS

1 EINLEITUNG

Die Prognose von Zeitreihenwerten spielt in vielen wirtschaftlichen Bereichen eine wichtige Rolle. Insbesondere zur Planung zukünftiger Investitionen und Vermeidung von Engpässen hat sich die Zeitreihenvorhersage als wichtiges Instrument in Wirtschaft, Wissenschaft und Politik etabliert. Diese Arbeit konzentriert sich auf den Bereich der Verkaufsdomäne. Eine wichtige Methode zur Vorhersage von Verkaufszahlen stellt die Regression dar. Zur Berechnung einer solchen Vorhersage auf der Basis regressiver Modelle, steht in der Regel eine große Anzahl von Attributen zur Verfügung. Nicht immer kann ohne weiteres bestimmt werden, welche Attribute zu einer Steigerung der Vorhersagegenaugikteit führen. Um relevante Attribute zu bestimmt und irrelevanten auszusortieren, wurden eine Reihe von Merkmalsauswahl-Algorithmen entwickelt. Diese lassen sich in zwei Klassen unterteilen. Die Filter-Methode, welche die relevanten Merkmale anhand von statistischen Eigenschaften bestimmt, und die Wrapper-Methode, bei der der eigentliche Zielalgorithmus zur Anwendung kommt.

In dieser Arbeit wird untersucht, wie sich die Wrapper-Methode in Verbindung mit Backward Elimination, Forward Selection und mit genetischen Algorithmen, sowie der korrelationsbasierte Filter-Ansatz, verhalten. Zudem werden hybride Varianten vorgestellt. Zunächst wird in Kapitel 2 auf die Grundlagen der Zeitreihenvorhersage, der verwendeten Programmiersprache und die in der Evaluation verwendete Prognosetechnik, welche speziell für die Verkaufsdomäne entwickelt wurde, eingegangen. In Kapitel 3 wird zum besseren Verständnis des Filter-Ansatzes zunächst der Korrelationskoeffizient erläutert, der im korrelationsbasierten Filter zum Einsatz kommt. Nach der Beschreibung des Wrapper-Ansatzes werden zudem zwei hybride Ansätze vorgestellt. Insbesondere die Wrapper-Methode kommt nicht ohne effektive Suchalgorithmen aus. Aus diesem Grund werden einige grundlegende Suchalgorithmen vorgestellt, die in der Evaluation zur Anwendung kommen. In der Evaluation (Kapitel 4) werden sechs Verkaufsdatensätze auf einige Eigenschaften untersucht, bevor auf ihnen die vorgestellten Merkmalsauswahl-Algorithmen zur Anwendung kommen. Die aus der Evaluation gewonnenen Erkenntnisse werden in Kapitel 5 zusammengefasst.

2 GRUNDLAGEN DER ZEITREIHENVORHERSAGE

Bevor in Kapitel 3 auf die Merkmalsauswahl eingegangen wird, werden zunächst andere Grundlagen dieser Arbeit erläutert. Die Methode der Merkmalsauswahl, die in dieser Arbeit behandelt wird, dient der Auswahl relevanter Attribute zur Zeitreihenvorhersage. Diese soll in Abschnitt 2.1 erklärt werden. Da sämtliche Algorithmen dieser Arbeit auf der Programmiersprache R basieren, wird diese kurz in Abschnitt 2.2 vorgestellt. Das Cross-Sectional Forecasting und der im Rahmen des SIS-Projetes entstandene Algorithmus SIS.predict, welcher in R programmiert wurde und die Grundlage dieser Arbeit bildet, wird in Abschnitt 2.3 erläutert.

2.1 GRUNDLAGEN DER ZEITREIHENVORHERSAGE

In diesem Abschnitt wird auf die Grundlagen der Zeitreihenvorhersage eingegangen. Dazu wird erläutert, was Zeitreihen sind, wie der Vorgang ihrer Vorhersage abläuft und wie das Ergebnis einer solchen Vorhersage bewertet werden kann.

2.1.1 Zeitreihen

Als Zeitreihe wird eine Sammlung von Daten beschrieben, die in zeitlicher Folge beobachtet wurde [Cha82]. Obwohl die Zeitintervalle zwischen zwei aufeinanderfolgenden Elementen in solch einer Reihe unterschiedlich sein können, werden diese Intervalle im Allgemeinen (und auch in dieser Arbeit) als konstant angenommen. Beispiele für solch eine Zeitreihe sind die Einwohnerzahl eines Landes zu verschiedenen Jahren, monatliche Verkaufszahlen oder der tägliche Energieverbrauch einer Stadt oder eines ganzen Landes.

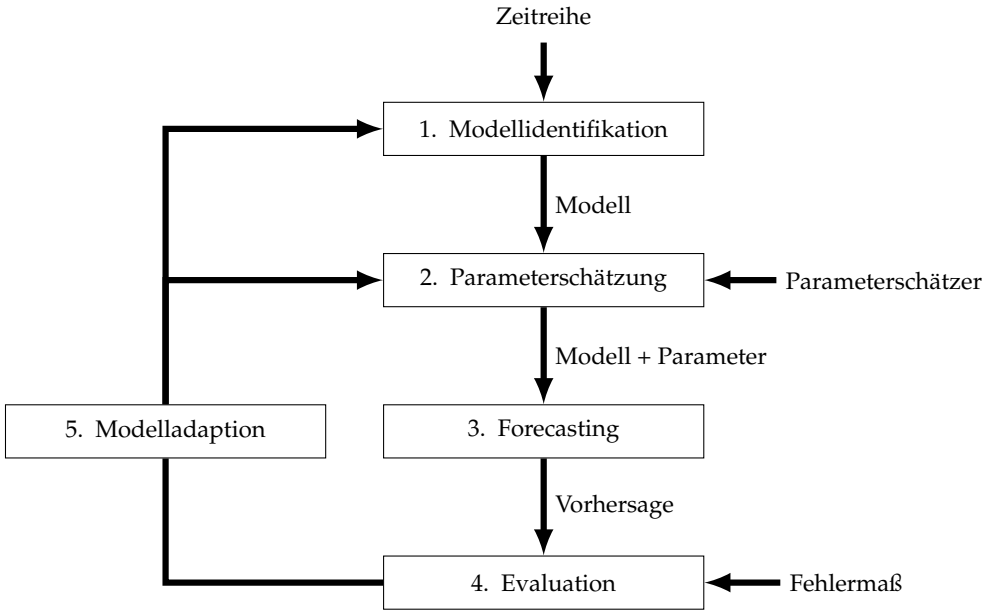


Abbildung 2.1: Vorgang einer Zeitreihen-Vorhersage

2.1.2 Prognose von Zeitreihen

In vielen Bereichen von Wirtschaft, Wissenschaft und Technik ist es notwendig, eine zuverlässige Aussage über die zukünftige Entwicklung einer Zeitreihe zu treffen. Die Vorhersage von zukünftigen Ereignissen und Zuständen wird Zeitreihenvorhersage, bzw. im Englischen Forecasting, genannt [Bow93]. So muss die Regierung eines Landes die zukünftige Bevölkerungsentwicklung kennen, um z.B. Schulen oder Altersheime zu planen, ein Produktionsbetrieb möchte die richtige Menge eines Produkts herstellen und ein Stromkonzern muss Energie-Engpässe vermeiden. Zum Erstellen zuverlässiger Prognosen wurden eine Vielzahl von Verfahren entwickelt, die auf der Analyse der vorhergegangenen Zeitreihe basieren. Die in dieser Arbeit verwendete Form der Zeitreihenvorhersage basiert auf der linearen Regression, die in Abschnitt 2.3.1 erläutert wird.

2.1.3 Vorgang eines Prognoseprozesses auf eine Zeitreihe

Beim Vorgang einer kompletten Zeitreihen-Vorhersage handelt es sich um einen rekursiven Prozess, der aus fünf Schritten besteht (Abbildung 2.1). Zunächst wird manuell oder semi-automatisch für eine gegebene Zeitreihe ein Vorhersage-Modell ausgesucht (1). Für dieses Modell müssen Parameter abgeschätzt werden (2), mit denen das Modell nun Vorhersagen für die gegebene Zeitreihe erstellt (3). Mit einem geeigneten Fehlermaß werden die von dem Modell vorhergesagten Werte mit den tatsächlich eingetretenen verglichen und bewertet (4). Ist der gemessene Fehler zu groß, kommt es zu einer Modelladaption (5), das heißt, dass die Parameter angepasst werden oder sogar der Modelltyp geändert und an entsprechender Stelle der gesamte Vorgang wiederholt werden muss. Der Schwerpunkt dieser Arbeit liegt auf der Modellidentifikation und Parameterschätzung.

2.1.4 Modellbildung

Ein Modell hat die Aufgabe, die systematische Veränderung einer beliebigen Zeitreihe darzustellen [Mar73]. Ausgehend von der bisherigen Entwicklung einer Zeitreihe, soll es die zukünftige Entwicklung vorhersagen. Obwohl Modelle auch mehrere zukünftige Werte prognostizieren können, wird in dieser Arbeit lediglich die Vorhersage des nächsten Zeitreihenwerts betrachtet. Wie gut die Vorhersage eines Modells mit einem tatsächlich eingetroffenen Wert übereinstimmt, hängt nicht nur von dem gewählten Modell ab, sondern auch von Parametern, die für das Modell bestimmt werden müssen. In SIS.predict wird für die Vorhersage auf die lineare Regression zurückgegriffen.

2.1.5 Bewertung der Vorhersage

Um die Güte eines Modells zu bestimmen wird von dem Modell eine Reihe von bereits bekannten bekannten Zeitreihenwerten berechnet. Mithilfe einer Metrik werden die durch ein Prognosemodell vorhergesagten Werte mit den bereits bekannten Zeitreihenwerten miteinander verglichen. Diese Metrik ist die Bewertungs- oder Fehlerfunktion, die die Güte des Modells beschreibt. Je kleiner das Ergebnis, desto besser stimmt die Vorhersage mit dem tatsächlich eingetroffenen Wert überein. Ist das Ergebnis 0, konnte das Modell alle Werte exakt vorhersagen.

Ein einfaches Fehlermaß ist der mittlere absolute Fehler (MAE, Mean Absolute Error).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - y_i| \tag{2.1}$$

Er beschreibt den mittleren absoluten Unterschied zwischen den einzelnen Vorhersagen und dem eingetroffenen Ereignis. Dieses Fehlermaß eignet sich jedoch nicht für Ereignisse, deren eingetroffene Werte sich stark unterscheiden, da Ereignisse mit hoher Amplitude stärker bewertet werden, als solche mit kleiner. Um dieses Problem zu umgehen, wurde der Mittlere prozentuale Fehler ($MAPE$, Mean Absolute Percentage Error) eingeführt.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|x_i - y_i|}{y_i}\cdot 100\% \tag{2.2}$$

Dies führt jedoch ebenfalls zu Problemen. Zum einen führt ein eingetroffenes Ereignis von 0 zu einer Nulldivision, zum anderen ist dieses Maß nicht symmetrisch. Vorhersage und Ereignis lassen sich also nicht ohne Weiteres austauschen. Der symmetrische mittlere prozentuale Fehler ($sMAPE$) vermeidet auch diese Probleme.

$$sMAPE = \frac{1}{n}\sum_{i=1}^{n}sAPE(x_i, y_i) \tag{2.3}$$

mit

$$sAPE(x_i, y_i) = \begin{cases} 0 & \text{, wenn } x_i = y_i = 0 \\ \frac{2|x_i - y_i|}{|x_i| + |y_i|}\cdot 100\% & \text{sonst.} \end{cases} \tag{2.4}$$

Vorhersage und reales Ergebnis werden gleichberechtigt behandelt. Die gesonderte Behandlung, wenn Prognose und tatsächlicher Wert mit Null übereinstimmen, verhindert eine Nulldivision.

Das Ergebnis dieser Formel liegt immer zwischen 0, also absoluter Übereinstimmung, und 200 Prozent, wenn keine Übereinstimmung zwischen Vorhersage und Ereignis vorhanden ist. Im Bereich der Zeitreihenvorhersage gibt es noch viele weitere Fehlermaße, jedoch ist der $sMAPE$ für die nachfolgenden Experimente ausreichend.

2.2 DIE PROGRAMMIERSPRACHE R

R ist eine für die gängigsten Plattformen frei verfügbare Umgebung und Programmiersprache zur Datenanalyse und -darstellung [Hat11]. Da R vieles, was von einem modernen Werkzeug zur Datenanalyse erwartet wird, anbietet und durch zahlreiche Pakete für alle möglichen Anwendungen erweitert werden kann, hat es sich im Bereich der angewandten Statistik sowohl im kommerziellen Bereich als auch an Universitäten etabliert [Lig07].

R fasst standardmäßig alle Datentypen zunächst als Vektoren (oder eindimensionale Arrays) auf. Dies ermöglicht eine einfache Handhabung selbst großer Datenstrukturen. Während bei vielen Rechenoperationen mit Arrays oder Vektoren in anderen Programmiersprachen wie C, C++ oder Java, erst Code zur Iteration geschrieben werden muss, ist dies in R folgendermaßen möglich:

```
> array <- c(1,2,3,4,5)      # Erstellen und Zuweisen eines Vektors
> array                      # Ausgabe des Vektors
[1] 1 2 3 4 5

> array * 2                  # Multiplikation auf alle Elemente einzeln
[1]  2  4  6  8 10

> sum(array)                 # Aggregation auf gesamten Vektor
[1] 15
```

Auch mehrdimensionale Vektoren, sogenannte Data Frames, sind unter R möglich. In dieser Arbeit wird eine Erweiterung von Data Frame, data.table, verwendet, die relationale Algebra auf großen Datensätzen ermöglicht. R wurde unter den Programmiersprachen Fortran, welche vor allem für numerische Berechnungen eingesetzt wird, und C programmiert. Dadurch kann R, obwohl es sich dabei um eine Interpretersprache handelt, selbst komplexe Berechnungen mit großen Datensätzen in geringen Zeitumfang durchführen. Die Algorithmen, die in dieser Arbeit vorgestellt werden, wurden entweder selbst unter R implementiert oder durch von R zur Verfügung gestellte Pakete verwirklicht. Auch die Diagramme wurden unter R erstellt.

2.3 CROSS-SECTIONAL FORECASTING

Das Ziel des SIS-Projektes (Self-Adjusting Imputation System) ist eine möglichst exakte Vorhersage von Zeitreihen aus großen Datensätzen der Verkaufsdomäne („Big Data“). Im Rahmen dieses Projektes wurde unter der Programmiersprache R die Funktion SIS.predict erstellt, die die Grundlage dieser Arbeit darstellt und im Folgenden erläutert wird.

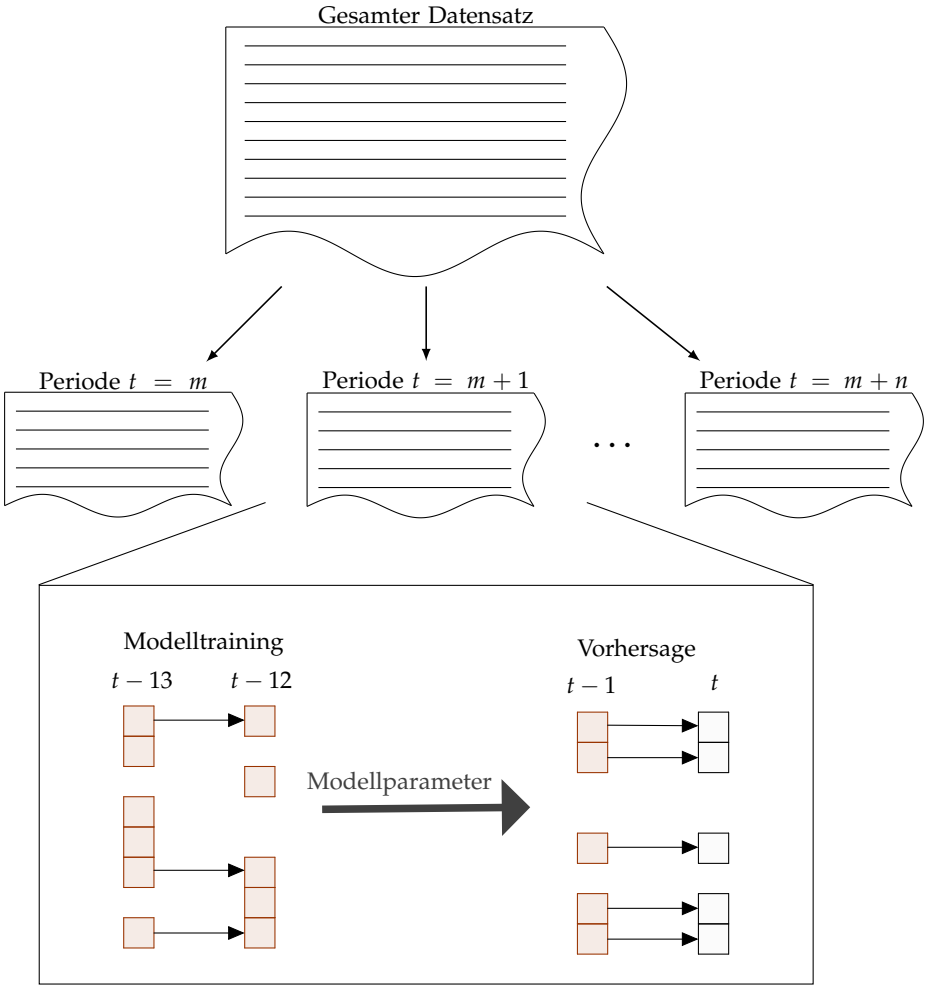


Abbildung 2.2: Die Funktionsweise von SIS.predict

Zur Verdeutlichung der Vorgehensweise von SIS.predict dient die Abbildung 2.2. Zusammen mit einigen Parametern erhält das Programm die gesamten vorhandenen Verkaufsdaten. Diese werden in Perioden aufgeteilt, deren Prognose verlangt wird. Dazu wird für jede Periode t die Verkaufsdaten des vorhergehenden Monats $t - 1$, des Monats vor einer Saisonlänge $t - 12$, sowie dessen Vormonats $t - 13$ benötigt. Sofern vorhanden, können zusätzlich auch die Verkaufsdaten von Perioden, die um ein vielfaches der Saisonlänge zurückliegen (also $t - 24$ und $t - 25$, $t - 36$ und $t - 37$ usw.), einbezogen werden. Anhand der Verkaufszahlen des Monats $t - 12$ und der Regressionsattribute des Vormonats $t - 13$ werden in der Trainingsphase die Parameter des Regressionsmodells bestimmt. In der Abbildung wird deutlich, dass nicht für alle Artikel eines Monats auch die Attribute des Vormonats vorhanden sind oder zwar Attribute im Vormonat vorhanden sind, aber keine Verkaufszahlen für den folgenden Monat. Diese Fälle bleiben bei der Modellbildung unbeachtet. Mit dem nun konfigurierten Modell werden anhand der Regressionsattribute des Vormonats $t - 1$ in der Vorhersagephase eine Verkaufsprogonose für jeden Artikel erstellt, sofern Regressionsattribute für den jeweiligen Artikel vorhanden sind. Die Vorhersagen wird zum Schluss für jede Periode und, sofern angegeben, nach weiteren Evaluationsattributen zusammengefasst.

2.3.1 Lineare Regression

Bei dem für die Vorhersage verwendeten Modelle in SIS.predict handelt es sich um die lineare Regression. Bei der linearen Regression, genauer dem linearen multiplen Regressionsmodell, wird davon ausgegangen, dass zwischen der Zielvariable y und den den Ausgangvariablen $x_1, \ldots, x_n$ ein linearer Zusammenhang besteht, der sich durch die Linearkombination in der Formel 2.5 beschreiben lässt [Bam01].

$$y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + u \tag{2.5}$$

Darin handelt es sich bei $\beta_0, \ldots, \beta_n$ um die Regressionskoeffizienten und bei u um die Störvariable, die nicht weiter vorhersagbar ist.

Die Regressionskoeffizienten werden in der Modellbildungsphase anhand der Daten der letzten Saison ermittelt. x_i entspricht dabei den Verkaufsattributen der Periode $t - 13$, während es sich bei y um die Verkaufsanzahl der Periode $t - 12$ handelt. Bei der eigentlichen Vorhersage dienen dann die Verkaufsattribute der Periode $t - 1$ als Eingabe, um die Verkaufszahl der Periode t zu ermitteln.

Im Folgenden soll gezeigt werden, wie die Erstellung und Anwendung eines linearen Modells unter R geschieht. Zunächst muss die Gleichung für das Modell mittels formula() erstellt werden. Dazu muss das Zielattribut und die verwendeten Regressionsattribute angegeben werden. Mit der Funktion lm(), der das Modell und die Trainingsdaten, übergeben werden, wird ein auf die Trainingsdaten angepasstes Modell bestimmt. Die Trainingsdaten müssen sowohl Ziel- als auch Regressionattribute besitzen. Mittels predict(), welches das Modell und die Regressionsattribute des Vormonats erhält, wird nun eine Vorhersage erstellt.

```
> formel     <- formula(zielattribut ~ attribut1 + attribut2)
> modell     <- lm(formel, trainingsdaten)
> vorhersage <- predict(modell, regressionsdaten)
```

2.4 ZUSAMMENFASSUNG

In diesem Kapitel wurden die Grundlagen der Zeitreihenvorhersage erläutert, sowie die Programmiersprache R vorgestellt und das Cross-Sectionale Forecasting beschrieben. Die unter R geschriebene Funktion SIS.predict verwendet das Cross-Sectionale Forecasting zur Vorhersage von Verkaufsdaten. Diese wird mittels der linearen Regression verwirklicht. Zur Ermittlung der richtigen Regressionsattribute wird eine verlässliche Auswahl-Methode benötigt. Eine Reihe möglicher Methoden wird in dem folgenden Kapitel vorgestellt.

3 MERKMALSAUSWAHL

Die Berechnung zukünftiger Verkaufszahlen wird bei der linearen Regression auf der Basis kardinaler[1] Merkmalsausprägungen oder Attributen der Vorperiode ausgeführt. Im einfachsten Fall liegt nur ein auswertbares Attribut vor, welches als Eingabevariable der Vorhersagefunktion dient. Dabei kann es sich jedoch auch um eine sehr viel größere Anzahl von Attributen handeln. Im Falle der für diese Arbeit verfügbaren Daten handelt es sich um überschaubare zwölf, in anderen Fällen können es auch mehrere hundert sein. Jedes zusätzliche Attribut bedeutet nicht nur einen Mehraufwand bei der Vorhersage, sondern kann unter Umständen sogar die Ergebnisqualität negativ beeinflussen. Aus diesem Grund ist es sinnvoll eine Untermenge der zur Verfügung stehenden Attribute auszuwählen, welche zur Berechnung verwendet werden sollen. Diese Auswahl wird Merkmalsauswahl genannt. Die zwei verbreitetsten Ansätze zur Merkmalsauswahl sind der Filter-Ansatz, der anhand von Heuristiken eine Attributsauswahl trifft, und der Wrapper-Ansatz (zu Deutsch etwa Hülle), der zur Auswahl den später zu verwendenden Algorithmus auf den Testdaten ausführt [HSL99, Inz04].

3.1 KORRELATIONSKOEFFIZIENT

Bevor auf die Filter-Methode eingegangen wird, muss zunächst der Korrelationskoeffizient erläutert werden. Ziel ist es, ein geeignetes Maß zu finden, welches den Zusammenhang zweier Reihen von Merkmalsausprägungen ausreichend beschreibt. Bevor dies geschieht, muss jedoch erst eine einzelne Reihe aussagekräftig beschrieben werden.

[1] Kardinale Attribute (genauer: Merkmalsausprägungen) sind solche, die nicht nur in eine Reihenfolge gebracht werden können, sondern auch noch bestimmt werden kann, in welchem Maß sich zwei kardinale Attribute unterscheiden. Sie unterscheiden sich dadurch von Nominalzahlen (jedes Attribut erhält eine Nummer, die lediglich zur Identifikation dient) und Ordinalzahlen (eine ordnungserhaltende Abbildung auf Zahlen) [Bam01]. Weiterhin werden kardinale Attribute lediglich als *numerisch* bezeichnet.

3.1.1 Varianz

Um die Ausprägung eines Merkmals möglichst kompakt zu beschreiben, bedient man sich unter anderem den drei sogenannten Lageparametern Modalwert, Median, arithmetisches und geometrisches Mittel. Häufig genügt diese Beschreibung nicht um die Verteilung einzelner Merkmale zu beschreiben, weshalb man sich der Streuungsparameter bedient. Einer der wichtigsten Streuungsparameter ist die Varianz. Die Varianz σ^2 oder mittlere quadratische Abweichung beschreibt das Streuverhalten eines Merkmals. Sie ist das arithmetische Mittel der quadrierten Abweichung aller Beobachtungswerte x von einem Lageparameter, in diesem Fall dem Mittelwert $\overline{x}$.

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2 \tag{3.1}$$

Da, vor allem bei der Zeitreihenvorhersage, nicht die Grundgesamtheit der Daten vorhanden ist, ist σ^2 zu optimistisch und wird deshalb, unabhängig von dem Umfang der Daten, um den Korrekturfaktor $\frac{1}{n-1}$ erweitert. Damit erhält man die korrigierte Stichprobenvarianz s^2.

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2, \tag{3.2}$$

3.1.2 Standardabweichung

Die Standardabweichung ergibt sich aus der positive Wurzel der mittleren quadratischen Abweichung und stellt einen Erwartungswert dar, wie stark eine Reihe von Werten von dem verwendeten Lageparameter abweichen.

$$s = \sqrt{s^2} \tag{3.3}$$

3.1.3 Kovarianz

Um eine Vorstellung von der Stärke des statistischen Zusammenhangs zwischen zweier Merkmale X und Y zu vermitteln, dient die Kovarianz, bzw die korrigierte Stichprobenkovarianz.

$$Cov = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y}) \tag{3.4}$$

Wenn es keinen Zusammenhang zwischen X und Ygibt, ergibt sich eine Kovarianz von 0. Für vergleichende Aussagen eignet sich dieses Maß jedoch nicht, da die Kovarianz nicht beschränkt sein muss. Je größer der Betrag der Merkmale, desto größer ist auch der Betrag der Kovarianz. Dies zeigt Tabelle 3.1. Zwischen dem Zielattribut und den ersten drei Attributen lässt sich ein eindeutiger Zusammenhang ausmachen, während die Zahlen für das Attribut 4 zufällig gewählt sind. Dennoch ist die Kovarianz des letzten Attributs mit dem Zielattribut höher als die des ersten Attributs. Zur vergleichenden Beschreibung des Zusammenhangs bedient man sich daher des Korrelationskoeffizienten.

Zielatt.	Att. 1	Att. 2	Att. 3	Att. 4
1	1	-1	100	38
2	2	-2	200	7
3	3	-3	300	1
4	4	-4	400	11
5	5	-5	500	43
6	6	-6	600	86
7	7	-7	700	97
8	8	-8	800	49
9	9	-9	900	48
10	10	-10	1000	2
Kovarianz:	9,17	-9,17	916,67	28,00
Korrelation:	1,00	-1,00	1,00	0,27

Tabelle 3.1: Beispiel für die korrigierte Stichprobenkovarianz und Korrelation

3.1.4 Der Korrelationskoeffizient nach Bravais und Pearson

Der Korrelationskoeffizient nach Bravais und Pearson erweitert die Kovarianz um eine Normalisierung und macht somit Vergleiche möglich.

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{3.5}$$

Man erkennt im Zähler die bereits vorgestellte Kovarianz, während sich im Nenner das Produkt der beiden Standardabweichungen befindet. (Der vorgestellte Faktor $\frac{1}{n}$ bzw. $\frac{1}{n-1}$ kann gekürzt werden.) Die Formel zur Berechnung des Korrelationskoeffizienten lässt sich somit zu der folgenden Formel zusammenfassen.

$$r = \frac{\mathrm{Cov}(X, Y)}{s(X)s(Y)} \tag{3.6}$$

Der Korrelationskoeffizient ist ein Wert zwischen -1 und 1. Wie bei der Kovarianz ergibt sich ein Korrelationskoeffizient von $r = 0$, wenn keine Korrelation zwischen zwei Merkmalen vorliegt. Je stärker die Korrelation, desto höher ist der Betrag des Koeffizienten. Dabei kann eine positive ($r = 1$) oder eine negative ($r = -1$) Korrelation vorliegen.

Zum besseren Verständnis des Bravais-Pearson-Korrelationskoeffizienten dient das aus [Bam01] entnommene Streuungsdiagramm (Abbildung 3.1). Die gestrichelten Linien entsprechen den jeweiligen Mittelwerten der Merkmale X und Y. Die Abweichprodukte $(x_i - \overline{x})(y_i - \overline{y})$, die Teil des Korrelationskoeffizienten sind, sind jeweils im linken unteren und im rechten oberen Quadranten positiv, während sie in den anderen beiden Quadranten ein negatives Ergebnis liefern. Die Wertepaare, die mengen- und größenmäßig überwiegen, bestimmen somit darüber, ob sich ein positiver oder ein negativer Koeffizient ergibt.

Neben den Beispielen in Tabelle 3.1 dienen die Diagramme in der Abbildung 3.3 als visuelle Bei-

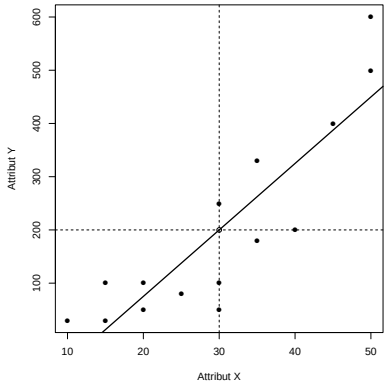


Abbildung 3.1: Streudiagramm zweier Merkmale mit einem Bravais-Pearson-Korrelationskoeffizient von $r = 0.88$

spiele für den Korrelationskoeffizienten. In den ersten beiden Diagrammen kann man deutlich erkennen, dass eine Korrelation zwischen den Merkmalen X und Y existiert, während sie in den nächsten beiden weniger stark ist. In dem fünften Diagramm sind die Wertepaare rein zufällig verteilt, sodass keine Abhängigkeit vorhanden ist. Im letzten Diagramm ist zwar offensichtlich eine Korrelation vorhanden, jedoch wird diese nicht erkannt, da nur lineare Abhängigkeiten vom Bravais-Pearson-Korrelationskoeffizienten erkannt werden. Das heißt, dass ein niedriger Korrelationskoeffizient nicht bedeutet, dass keine Korrelation vorhanden ist. Der Korrelationskoeffizient nach Bravais und Pearson kann nur lineare Korrelationen abbilden. Möchte man nicht-lineare Korrelationen bestimmen, müssen die Daten vorbehandelt werden, indem sie linealisiert werden. In dem gezeigten Beispiel könnte dies durch das Ziehen der Wurzel geschehen. Ein solches Vorgehen ist in dieser Arbeit jedoch nicht nötig.

Zwar kann ein hoher Korrelationskoeffizient durchaus bedeuten, dass zwei Merkmale einander beeinflussen, dennoch bedeutet dies nicht, dass sie auch tatsächlich voneinander abhängig sind. Ein hoher Koeffizient kann stattdessen ein Hinweis auf eine gemeinsame Ursache sein (Abbildung 3.2). Ein häufig genanntes Beispiel für solch eine sogenannte Scheinkorrelation ist der Zusammenhang zwischen Geburtenrate und Storchpopulation. Obwohl das verbreitete Kindermärchen eine Ursache-Wirkung-Beziehung behauptet, ist eine gemeinsame Ursache (zum Beispiel der Grad der Industrialisierung) eine wahrscheinlichere Erklärung. Auch in dieser Arbeit werden die Regressionsattribute des Vormonats nicht als Ursache betrachtet, sondern lediglich als Indizien für das zukünftige Verhalten der Verkaufszahlen. Die Funktionen zur Berechnung der korrigierten Varianz, Kovarianz und der Korrelation nach Bravais und Pearson sind bereits in dem Standardpaket von R enthalten und lassen sich mittels var(), cov() und cor() aufrufen.

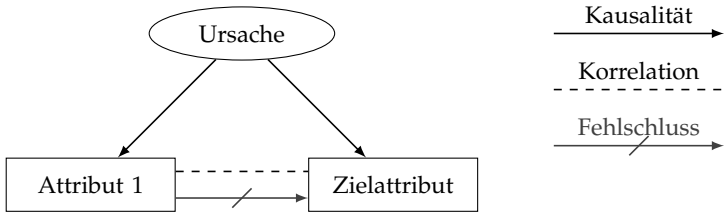
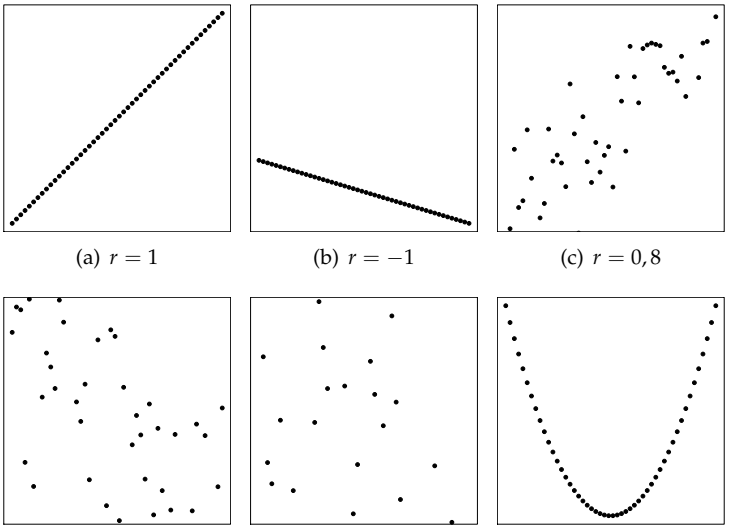


Abbildung 3.2: Korrelation bedeutet nicht Abhängigkeit



(a) $r = 1$ (b) $r = -1$ (c) $r = 0,8$

(d) $r = -0.6$ (e) $r = 0$ (f) $r = 0$

Abbildung 3.3: Abhängigkeit des Bravais-Pearson-Korrelationskoeffizienten von der Form des Streuungsdiagramms

3.2 FILTER-ANSATZ

Die Filter-Methode bestimmt die Güte der vorgeschlagenen Attribute oder Attributskombination anhand statistischer Eigenschaften der Daten [Inz04]. Dabei verwendet der Filter für gewöhnlich die gesamten verfügbaren Trainingsdaten [HSL99]. Irrelevante Attribute werden durch eine schlechte Bewertung herausgefiltert [Seb02]. Die Filter-Methode agiert dabei unabhängig von dem später verwendeten Algorithmus [HSL99], in dem in dieser Arbeit vorliegenden Fall also der Zeitreihenvorhersage.

Eine Möglichkeit des Filter-Ansatzes ist es, Attribute auszuschließen, deren Informationsgehalt von anderen Attributen bereits abgedeckt wird [Kol96]. Eine weitere Möglichkeit besteht darin, eine Rangliste der Attribute anhand einer Relevanz-Punktevergabe zu erstellen [Kir92, Hol95]. Eine Kombination dieser Methoden bildet die in [HSL99] vorgestellte korrelations-basierte Merkmalsauswahl auf Basis des im Abschnitt 3.1.4 erläuterten Korrelationskoeffizienten, die in dieser Arbeit näher betrachtet wird.

Die Korrelation zwischen Zielattribut und den potentiellen Regressionsattributen kann schon von sich aus als Relevanz-Bewertung für die einzelnen Attribute betrachtet werden. In einer Erweiterung, die in Abschnitt 3.2.3 näher erläutert wird, dient die Korrelation zudem auch dem Verwerfen redundanter Attribute. Damit vereint der Korrelationskoeffizient bereits die beiden vorgenannten Prinzipien des Filter-Ansatzes. Ein weiterer Vorteil des Korrelationskoeffizienten ist die Tatsache, dass er, wie die spätere Zielfunktion, also der Regressionsalgorithmus, für lineare Zusammenhänge ausgelegt ist.

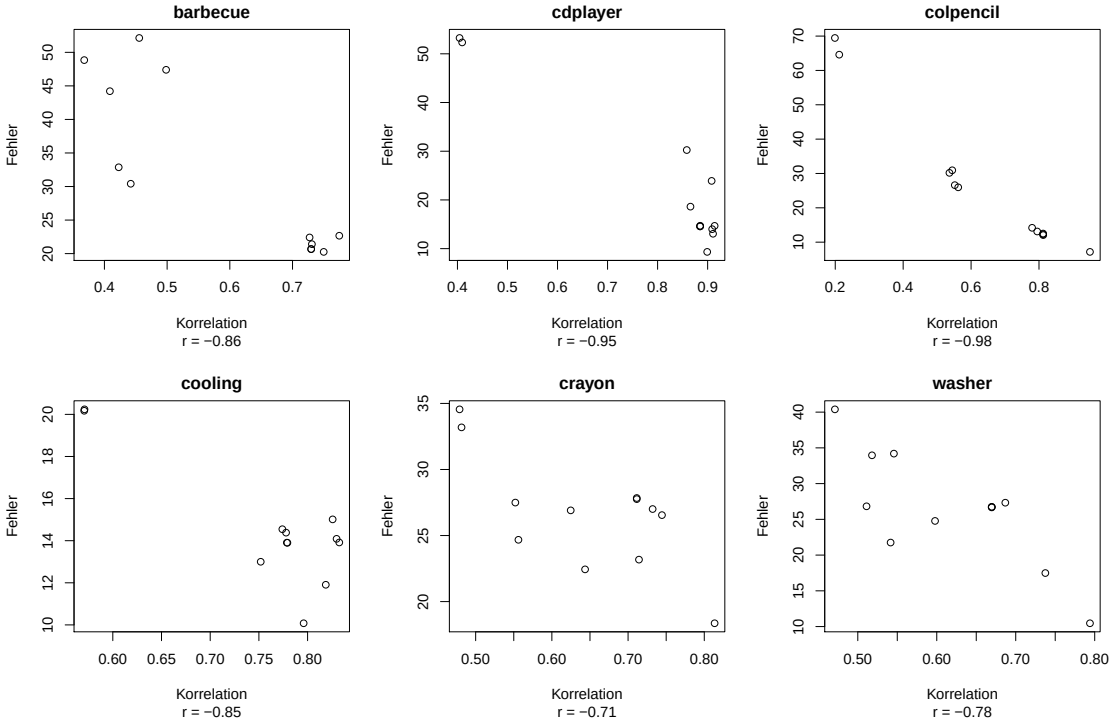


Abbildung 3.4: Einfluss der Korrelation auf die Vorhersagen

3.2.1 Korrelation einzelner Attribute mit dem Zielattribut

Die simpelste Methode der Merkmalsauswahl ist die Bewertung aller Attribute, um dasjenige auszuwählen, welches die beste Bewertung erhalten hat. Als Grundlage der Bewertung der einzelnen Attribute dient hier der in Abschnitt 3.1.4 vorgestellte Korrelationskoeffizient nach Bravais und Pearson. Als Eingabe werden die Werte des Zielattributs der Trainingsperiode und die jeweiligen Werte des zu bewertenden Regressionsattributs des Vormonats verwendet.

Abbildung 3.4 zeigt, wie sich die Korrelation der Trainingsdaten und der mittlere Vorhersagefehler der Testdaten zueinander verhalten. Dabei wurde zum Einen jeweils der Korrelationskoeffizient des Zielattributs mit einem numerischen Attribut des Vormonats berechnet. Zum Anderen wurde dieses Attribut als Regressionsattribut für die in Abschnitt 2.3 erläuterte Vorhersage-Funktion SIS.predict verwendet. Zur Verfügung stehen sechs Datensätze, mit 12 numerischen Merkmalen. Von den 36 zur Verfügung stehenden Monaten, wurden die ersten 24 zum Training der monatlichen Modelle verwendet, um die letzten 12 Monate vorherzusagen. Die so entstandenen Prognosen wurden mit den eingetroffenen Werten verglichen und der Prognosefehler anhand des $sMAPE$ aus der Formel 2.3 ermittelt. Eine nähere Beschreibung der Daten und der Vorhersageumgebung findet in Abschnitt 4.1 statt. In Abbildung 3.4 wird deutlich, dass sich die Regressionsattribute sowohl in ihrer Korrelation als auch in ihrer Fähigkeit zur Vorhersage unterscheiden. Obwohl deutlich wird, dass innerhalb kleiner Grenzen ein Attribut mit höherer Korrelation nicht zwingend ein besseres Ergebnis erzielt, führt im Allgemeinen eine hohe Korrelation zu einem niedrigen Fehler und umgekehrt. Genauer lässt sich für die Korrelation zwischen Regressions- und Zielattribut und dem entstehenden Vorhersagefehler des Regressionsattributs für das Zielattribut wiederum eine Korrelation feststellen, die für die gezeigten Datensätze jeweils zwischen

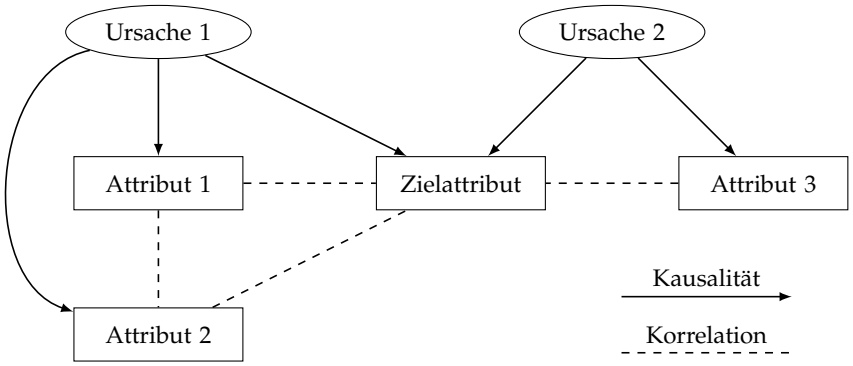


Abbildung 3.5: Korrelation mehrerer Attribute

$-0,71$ und $-0,98$ liegt.

Diese Form der Filterung ist zwar einfach, jedoch verschwendet sie durch die Nutzung lediglich eines Attributs unter Umständen wertvolles Datenmaterial. Im folgenden Abschnitt soll daher erläutert werden, wie durch die Korrelation auch mehrere Attribute von der Merkmalsauswahl zurückgegeben werden können.

3.2.2 Korrelation mehrerer Attribute mit dem Zielattribut

Ein naiver Ansatz zur Ausgabe mehrerer Attribute besteht darin, eine bestimmte Anzahl der Attribute mit der besten Bewertung auszugeben. Diese Anzahl kann sowohl vorher festgelegt sein oder durch die Einhaltung eines Mindestmaßes durch die Attributsbewertung bestimmt werden. Bei der Umsetzung dieser Methode stößt man, vor allem bei der Verwendung des Korrelationskoeffizienten, jedoch auf ein schwerwiegendes Problem. Werden zum Beispiel die zwei Regressionsattribute mit der höchsten Korrelation zum Zielattribut ausgewählt, ist zu befürchten, dass auch zwischen diesen beiden Attributen eine hohe Korrelation besteht. Dies ist besonders in den hier untersuchten Datensätzen der Fall, da einige Attribute lediglich die von äußeren Einflüssen bereinigten Daten der anderen Attribute enthalten. Eine hohe Korrelation zwischen den Regressionsattributen bedeutet, dass eines der beiden Attribute nur geringe zusätzlichen Informationen beinhaltet und eventuell darauf verzichtet werden kann (vergleiche Abbildung 3.5). Es ist sinnvoller, ein weiteres Attribut aufzunehmen, welches keine nennenswerte Korrelation mit den bereits aufgenommenen Attributen vorweist.

Um zu zeigen, wie sich die Korrelation zweier Attribute untereinander auf die Qualität der Vorhersage, also den Vorhersagefehler, auswirkt, kann wie folgt vorgegangen werden:

1. Die Vorhersagequalität der beiden Attribute allein wird jeweils empirisch bestimmt. Das heißt, zur Modellbildung wird jeweils nur eines der beiden Attribute verwendet.

2. Danach wird die Korrelation zwischen den beiden Regressionsattributen berechnet.

3. Nun wird die Vorhersagequalität für die Attribute gemeinsam bestimmt. In das Modell fließen also beide Attribute.

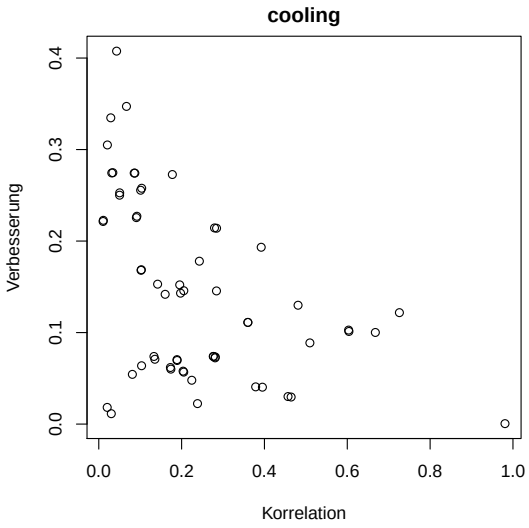


Abbildung 3.6: Verbesserung der Vorhersage im Verhältnis zur Korrelation zweier Regressionsattribute

4. Zum Schluss wird die Qualität des Modells aus Punkt 3 ins Verhältnis zu dem besseren Ergebnis der beiden Modelle aus Punkt 1 gesetzt. Es wird also bestimmt, ob und um wie viel das ohnehin schon bessere Modell mit dem Attribut des schlechteren Modells sich noch einmal verbessern konnte.

Das Ergebnis dieser Vorgehensweise ist in dem Datensatz aus Abbildung 3.6 besonders deutlich. Auf der X-Achse sind jeweils die Korrelationen der beiden Attribute eingetragen, während auf der Y-Achse, unabhängig von der eigentlichen Qualität der Vorhersage, lediglich die Verbesserung des schlechteren der beiden verwendeten Modelle eingetragen sind.

In dem Diagramm wird deutlich, dass mit zunehmender Korrelation der Regressionsattribute, die mögliche Verbesserung abnimmt. Eine niedrige Korrelation der beiden Attribute kann hingegen zu einer starken Verbesserung der Vorhersagequalität führen.

Guyon et al. kommen zu dem Ergebnis, dass perfekt korrelierende Attribute zwar redundant sind und somit keinen Informationsgewinn erreichen, eine lediglich sehr hohe Korrelation aber nicht bedeutet, dass sich die beiden Attribute nicht trotzdem ergänzen können [Guy03]. Auch in dem Diagramm zeigt sich, dass selbst hohe Korrelationen zumindest zu leichten Verbesserungen führen können.

Weiterhin kommen Guyon et al. zu dem Schluss, dass Attribute, die für sich genommen nutzlos sind, in Verbindung mit anderen Attributen zu einer starken Verbesserung führen können. Zwei für sich nutzlose Attribute können in Kombination zu nutzbaren Ergebnissen führen.

Die Tatsache, dass schlechte Attribute, also solche mit niedriger Korrelation zu dem Zielattribut, zumindest in Verbindung mit weiteren Attributen hilfreiche Merkmalsträger darstellen, führt zu dem Schluss, dass die Nutzung der lediglich besten Attribute nicht sinnvoll ist. Es müssen alle Korrelationen beachtet und zu einem sinnvollen Maß zusammengefasst werden. Ein solches Maß stellt der im Folgenden beschriebene korrelationsbasierte Filter dar.

3.2.3 Der korrelationsbasierte Filter

Hall et al. fasst die eben beschriebenen Erkenntnisse so zusammen, dass die gewählten Merkmale eine hohe Korrelation zu dem Zielattribut haben soll, jedoch eine geringe Korrelation untereinander [HSL99]. Basierend darauf, schlägt er als Heuristik die Funktion $Merit_S$ vor, die als Maß für die Leistung einer Attributskombination dient.

$$Merit_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{3.7}$$

Dabei entsteht ein hoher Wert, wenn die gewählten Attribute eine hohe Korrelation zu dem Zielattribut besitzen, jedoch eine geringe Korrelation untereinander. S stellt dabei eine Kombination von Attributen dar, k ist die Anzahl der Attribute in S, r_{cf} ist die durchschnittliche Korrelation der verwendeten Regressionsattribute $f \in S$ mit dem Zielattribut c, und r_{ff} die durchschnittliche Korrelation der Regressionsattribute untereinander.

Der Zähler bildet die Summe der Korrelationen mit dem Zielattribut. Hohe Korrelationen erhöhen somit $Merit_S$ stärker, als schwache. $k(k-1)\overline{r_{ff}}$ kann als Redundanz-Indikator betrachtet werden. Da er sich im Nenner befindet, senken hier hohe Korrelationen – und damit Redundanz – den Wert von $Merit_S$, während niedrige Korrelationen den Wert erhöhen. Auch wenn allgemein von Korrelationen gesprochen wird, ist der Absolutbetrag gemeint. Andernfalls würde das Ergebnis verfälscht.

Abbildung 3.7 zeigt, wie sich der $Merit_S$ einer Merkmalskombinationen auf den Vorhersagefehler auswirken kann. Dazu wurde jeweils für alle möglichen Kombinationen der $Merit_S$ berechnet und danach die Funktion SIS.predict ausgeführt, die als Regressionsattribute eben jene Kombination erhielt. Jeder Punkt stellt jeweils den Vorhersagefehler einer Merkmalsauswahl mit dem dazugehörigen $Merit_S$ dar. Eine hohe Bewertung von $Merit_S$ führt im Allgemeinen zu einem niedrigen Fehler. Jedoch wird auch deutlich, dass große Schwankungen für einen bestimmten $Merit_S$-Wert exisitieren und dass die Attributskombination mit der besten Bewertung nicht den niedrigsten Fehler verursacht. Dennoch soll in der Evaluation zwei Filter-Ansätze vorgestellt werden, die anhand des $Merit_S$ eine Attributsauswahl treffen.

3.3 WRAPPER-ANSATZ

Im Gegensatz zum Filter-Ansatz, welcher die Trainingsdaten mit statistischen Methoden und anhand von Heuristiken untersucht, greift der Wrapper-Ansatz auf den eigentlichen Zielalgorithmus zurück, für den auch die Attribute ausgesucht werden, und bewertet daran die Vorhersagequalität der Attributskombination [HSL99]. Der Zielalgorithmus wird als Blackbox betrachtet, die als Eingabe die Trainingsdaten und eine gewählte Attributskombination erhalten und deren Vorhersagefehler als Wertung für diese Kombination dienen [Koh97, Guy03]. Mit dem Zielalgorithmus als Heuristik führt er dann eine Suche auf den möglichen Kombinationen durch. Aus diesem Grund sind Wrapper-Algorithmen in der Regel nur in Verbindung mit Such-Algorithmen anzutreffen. Eine Auswahl von Suchalgorithmen wird in Abschnitt 3.5 vorgestellt. Die Wrapper-Methode gibt im allgemeinen die bessere Attributskombination zurück, läuft jedoch aufgrund der in der Regel aufwendigen Zielfunktion wesentlich langsamer als der Filter-Ansatz [HSL99].

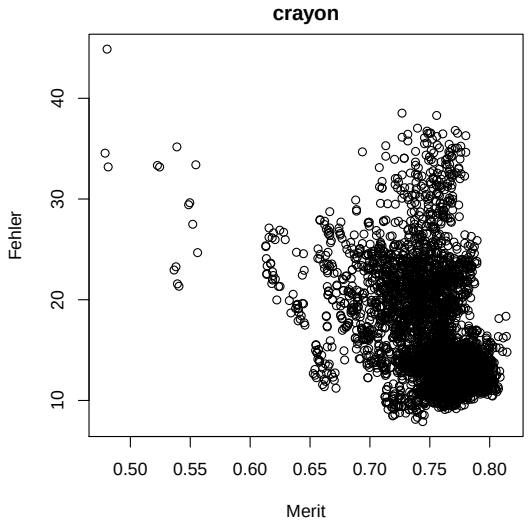


Abbildung 3.7: Verhältnis von $Merit_S$ und Vorhersagefehler

3.3.1 Kreuzvalidation

Wenn der Zielalgorithmus, im vorliegenden Fall also die Modellbildung und die Vorhersage, auf die Trainingsdaten angewendet werden soll, müssen diese wiederum in interne Trainings- und Validierungsdaten aufgeteilt werden (Abbildung 3.8). Tut man dies nicht und testet auf den selben Daten, auf denen das Modell auch trainiert wurde, droht eine sogenannte Überfittung. Das bedeutet, dass das Modell so stark auf die Trainingsdaten angepasst wurde, dass es für die externen Testdaten keine ausreichend guten Ergebnisse mehr liefern kann. Bei der einfachen Kreuzvalidation werden die verfügbaren Daten zufällig in k möglichst gleich große Teilmengen aufgeteilt [Koh95]. Für jede Teilmenge T_i mit $i \in \{1, \ldots, k\}$ wird nun mit den anderen Teilmengen das Modell trainiert, dieses Modell auf T_i getestet und die Güte des Modells zurückgegeben. Aus den k Bewertungen wird der Durchschnitt gebildet, der die Gesamtbewertung der vorgegebenen Attributskombination darstellt.

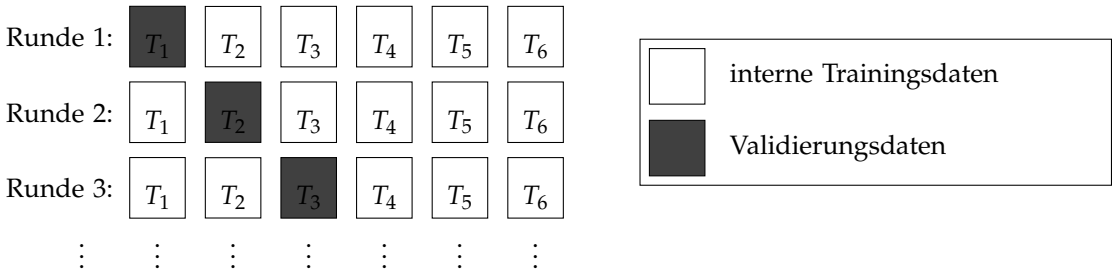


Abbildung 3.8: Kreuzvalidation

3.4 HYBRIDE ANSÄTZE

Die Tatsache, dass die Filter-Methoden schnell, jedoch ungenau, die Wrapper-Methode dafür genau, jedoch langsam ist, legt nahe, hybride Varianten zu implementieren, die die Vorteile von beiden Seiten zusammenführen. Im Folgenden werden zwei einfache Hybrid-Ansätze vorgestellt.

3.4.1 Filter für Wrapper

In der Abbildung 3.7 wird deutlich, dass die Kombinationen mit dem höchsten $Merit_S$ im Vergleich zu anderen Kombinationen durchaus auch nur mittelmäßig abschneiden können. Dennoch sind viele der Kombinationen mit hohem $Merit_S$ vielversprechend. Denkbar ist daher, mit der Filter-Methode eine bestimmte Anzahl von Attributskombinationen auszuwählen, für die die Wrapper-Methode ausgeführt wird. Die Anzahl kann fest oder durch eine Bewertungsminimum vorgegeben sein. So lässt sich ein Kompromiss zwischen Geschwindigkeit und Qualität der Ausgabe bestimmen. Je mehr Kombinationen der Filter-Teil durchlässt, desto mehr Kombinationen werden im Wrapper-Teil überprüft. Dies geht zwar auf die Kosten der Zeit, aber zu Gunsten der Qualität. Lässt der Filter nur wenige Lösungen durch, nimmt die Geschwindigkeit zu, die Qualität jedoch ab.

3.4.2 Korrelationsbasierte Filter in Verbindung mit Wrapper

Die Formel 3.7 zur Berechnung von $Merit_S$ benötigt nicht zwingend die Korrelationskoeffizienten als Eingabe, auch andere Koeffizienten sind denkbar, sofern sie eine normierte Wertung ausdrücken können. In Abschnitt 2.1.5 wurde erläutert, dass die Vorhersagebewertungsfunktion $sMAPE$ (Formel 2.3) nur Werte zwischen 0 und 200% annehmen kann. Durch eine einfache Umwandlung, lässt sich dieses Ergebnis auf Werte zwischen 0 und 1 normalisieren.

$$w = 1 - \frac{sMape}{200\%} \tag{3.8}$$

Der Gedanke dahinter ist, dass mit dieser Formel eine qualitativ hochwertigere Korrelation zustande kommt, die auf die einzelnen Ausgangsattribute angewendet werden kann. Fügt man diese, statt den Korrelationskoeffizienten in $Merit_S$ ein, erhält man die leicht modifizierte Gleichung

$$Merit_S^+ = \frac{k\overline{w_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}. \tag{3.9}$$

Für die Korrelation der Regressionsattribute untereinander bleibt der Korrelationskoeffizient erhalten, da dieser lediglich zur Beschreibung der Redundanz dient. Mit $Merit_S^+$ steht so mit linearen Mehraufwand eine qualitativ bessere Bewertung der Attributskombinationen zur Verfügung.

3.5 SUCHE

Der Wrapper-Ansatz bedient sich einer informierten Suche innerhalb der möglichen Merkmalskombinationen. Von einer informierten Suche wird gesprochen, wenn diese mithilfe einer Bewertungsfunktion stattfindet, welche die (Zwischen-)Ergebnisse bewertet [Rus04]. Das Durchsuchen der Merkmalskombinationen kann entweder deterministisch (Backward Elimination, Forward Selection, Abschnitt 3.5.2) oder zufällig (Genetische Algorithmen, Abschnitt 3.5.3) erfolgen. In beiden Fällen hängt das Ergebnis entscheidend von der Bewertungsfunktion ab. Die Bewertungsfunktion ist bei der Wrapper-Methode die Zielfunktion, auf die die Trainingsdaten angewendet werden.

3.5.1 Brute Force

Die Brute-Force-Methode oder die erschöpfende Suche wertet alle möglichen Merkmalskombinationen aus und wählt aus diesen diejenige mit dem besten Ergebnis. Stehen also n Attribute zur Verfügung, wertet die Brute-Force-Methode 2^n Merkmalskombinationen aus. Die 12 verfügbaren Regressionsattribute in den verwendeten Datensätzen bedeuten, dass es $2^{12} = 4096$ Kombinationsmöglichkeiten gibt. Da die leere Menge jedoch nicht betrachtet wird, wird des Weiteren jedoch von 4095 Kombinationen die Rede sein. Aufgrund des exponentiellen Wachstums und der aufwendigen Berechnung der Zielfunktion kommt diese Methode für den Wrapper-Ansatz nicht in Frage. Bei der Filter-Methode handelt es sich jedoch tatsächlich um eine erschöpfende Suche, da die verwendete Heuristik auf alle Attributskombinationen angewendet wird. Dies ist möglich, solange die Heuristik keinen zu hohen Aufwand birgt und der Suchraum nicht zu groß ist. Andernfalls können auch für Filter-Methoden die folgenden Suchalgorithmen angewendet werden.

3.5.2 Backward Elimination und Forward Selection

Die Backward Elimination und Forward Selection gehören zur Klasse der Greedy-Algorithmen. Die Backward Elimination bewertet zunächst die Kombination aller Attribute und entfernt jeweils das Attribut, dessen Entfernung die größte Verbesserung verursacht. Dieser Vorgang wird solange fortgesetzt, bis kein Attribut mehr vorhanden ist oder keine Verbesserung mehr feststellbar ist. Ähnlich geht die Forward Selection vor. Sie beginnt mit einer leeren Auswahl und erweitert diese jeweils um das Attribut, dessen Kombination mit der bisherigen Auswahl die Bewertung am stärksten verbessert. Der Algorithmus endet, wenn keine zusätzlichen Attribute mehr zur Verfügung stehen oder eine weitere Hinzunahme keine Verbesserung mit sich bringt [Joh94].

Bei n Attributen, terminiert der Algorithmus spätestens nach $n(n+1)/2$ Berechnungen, für gewöhnlich jedoch eher. Ein Nachteil dieser Form der Merkmalsauswahl ist die Tatsache, dass unter Umständen gute Lösungsansätze verworfen werden, indem bei der Backward Elimination Attribute vorschnell entfernt oder bei der Forward Selection aufgenommen werden, dessen positive (bzw. negative) Wirkung sich erst mit einem weiteren Attribut zeigt. Dennoch ist diese Form der Merkmalsauswahl häufig anzutreffen [Joh94, Koh97, Seb02, HSL99, Inz04, Guy03].

Bei der Evaluation in Abschnitt 4.4 wird deutlich, dass weder Backward Elimination, noch Forward Selection allein ein akzeptables Ergebnis für alle Datensätze erreicht. Aus diesem Grund soll bereits jetzt die Kombination der beiden Methoden vorgeschlagen werden. Der Forward-Backward-Ansatz sucht mit beiden Varianten ein jeweiliges Optimum. Dasjenige der beiden, welches ein besseres Ergebnis liefert, also den kleineren Fehler bei der Kreuzvalidierung zurückgibt, wird als Lösung zurückgegeben. Da somit zwei Suchen durchgeführt werden müssen, wirkt sich diese Variante negativ auf die Geschwindigkeit, jedoch voraussichtlich positiv auf die Güte der dadurch vorgeschlagenen Vorhersagemodelle aus.

3.5.3 Genetische Algorithmen

Genetische Algorithmen ahmen den Vorgang der natürliche Auslese in der Evolution nach [Rus04]. Eine einzelne Parameterbelegung, im vorliegenden Fall die Verwendung oder der Ausschluss eines Attributs, wird dabei Individuum genannt und in Form einer Bitkette repräsentiert. Eine Gruppe von Individuen nennt man Population. Die einzelnen Parameter, also die Entscheidungen für je ein Attribut, entsprechen den Genen. Zu Beginn werden zufällige Parameterbelegungen erzeugt. Im Laufe einer Iteration tauschen verschiedene Individuen Teile ihrer Bitketten (Crossing Over) aus. Gelegentlich, wird auch ein Bit einer Bitkette zufällig verändert (Mutation). Die neu entstandenen Individuen werden der Bewertungsfunktion unterzogen. Die Parameterbelegungen mit den besten Bewertungen, werden in die nächste Generation (sprich: die nächste Iteration) übernommen und der Vorgang wiederholt sich von Neuem. Der Algorithmus terminiert entweder nach einer bestimmten Zeit oder wenn eine Parameterbelegung gefunden wurde, die einer gegebenen Endbedingung genügt. Die Parameterbelegung mit dem besten gefundenen Bewertungergebnis wird zurückgegeben.

Der Genetische Algorithmus ist erst effektiv, wenn er mehrere Iterationen durchlaufen kann. Deshalb ist er nicht für kleine Suchräume geeignet. Dafür zeigt er seine Mächtigkeit vor allem in sehr großen Suchräumen, also bei einer hohen Anzahl von Attributen. Sein Vorteil gegenüber den vorher genannten Algorithmen ist, dass durch die Mutation auch ursprünglich verworfene Ideen wieder ins Spiel gebracht werden und diese sozusagen eine zweite Chance erhalten.

3.6 ZUSAMMENFASSUNG

In diesem Kapitel wurden Grundlagen und Methoden der Merkmalsauswahl behandelt. Eine Merkmalsauswahl stellt eine Suche innerhalb der Kombinationsmöglichkeiten der zur Verfügung stehenden Attribute dar. Durch das exponentielle Wachstum der Möglichkeiten, kann die Brute-Force-Methode nur dann zur Anwendung kommen, wenn die verwendete Bewertungsfunktion wenig Aufwand bedeutet und die Attributanzahl nicht zu groß ist. Ersteres ist meist bei Filter-Methoden der Fall, wenn zum Beispiel der Korrelationskoeffizient oder $Merit_S$ als Bewertung herangezogen werden. Ist der Suchraum auch für die Filter-Methode zu groß oder die Bewertungsfunktion zu aufwendig, was für die Wrapper-Methoden immer zutrifft, da sie die eigentliche Ziel-Funktion ausführen müssen, kommen informierte Suchalgorithmen zum Einsatz. Beispiele dafür sind die Forward Selection, Backward Elimination und der genetische Algorithmus.

In dem folgenden Kapitel wird untersucht, welche Form der Merkmalsauswahl sich am besten für die vorhandenen Datensätze eignen.

4 EVALUATION

In den beiden bisherigen Kapiteln wurde auf die Zeitreihenvorhersage und die Merkmalsauswahl eingegangen. Ziel dieser Arbeit ist es, die Vorhersage auf Verkaufszahlen durch eine automatisierte Merkmalsauswahl signifikant zu verbessern gegenüber Modellen mit statischen Regressionsattributen. Dazu wird empirisch untersucht, welche der in Kapitel 3 vorgestellten Algorithmen den Vorhersagefehler am stärksten senken. Vorbereitend darauf werden jedoch zunächst die Experimentaldaten beschrieben und einige Voruntersuchungen unternommen. Es wird bestimmt, welche Saisonlänge am besten für die Vorhersage des statischen Modells geeignet ist und wie sich die Qualität der Vorhersage zwischen verschiedenen statischen Modellen unterscheidet. Nach der Untersuchung der Auswahl-Algorithmen, findet abschließend eine zeitliche Betrachtung der untersuchten Algorithmen statt.

4.1 DIE EXPERIMENTALDATEN

Bei der Datenbasis der folgenden Experimente handelt es sich um private Datensätze eines Marktforschungsunternehmens, welche die Verkaufsdaten für unterschiedliche Produktgruppen enthält. Jeder Eintrag beschreibt unterschiedliche Attribute für einen Artikel bei einem Verkäufer zu einem bestimmten Zeitpunkt. Diese drei Attribute bezeichnen den Primärschlüssel. Obwohl die einzelnen Datensätze jeweils zwischen mehreren hunderttausend bis mehreren Millionen Einträge besitzen, sind die Daten darin dünn besetzt, das heißt, dass nicht für jede Artikel-Verkäufer-Zeitpunkt-Kombination auch Daten vorhanden sind. Bei den Attributen handelt es sich sowohl um Daten des Verkäufers (Ort, Verkaufskanal, Umsatzklasse), als auch um Attribute des Produkts (Marke, Preis, Produkteigenschaften). Neben der Verkaufszahl (sales_units), die in dieser Arbeit jeweils vorhergesagt werden soll, existieren noch 11 weitere kardinale Verkaufsmerkmale, die zur Vorhersage genutzt werden können. Dazu gehören der Preis, die Neubestellungen des Verkäufers, die Bestandsdaten, sowie deren von besonderen Einflüssen (Rabattaktionen, besonders günstiger Verkaufsplatz, usw.) bereinigten Werte.

Eine Grundannahme dieser Arbeit besteht darin, dass externe Einflüsse unbeachtet bleiben und kein Vorwissen über die Daten existiert (sofern diese nicht bereits in die Datensätze eingearbeitet

wurden). Das heißt, dass nur die vorhandenen Daten zur Analyse und Vorhersage dienen und auch Wissen über mögliche Korrelationen zwischen den Merkmalen erst während der automatisierten Analyse entsteht.

Die Zeitgranularität beträgt einen Monat über 3 Jahre. Das bedeutet, dass 36 Monate vorhanden sind. Zur Analyse, Merkmalsauswahl und später zur Modellbildung dienen jeweils die ersten 24 Monate, während die letzten zwölf anhand ihrer jeweiligen direkten Vormonaten vorhergesagt werden. Die Verkaufsdaten, bzw. die numerischen Attribute, der einzelnen Monate werden auf Artikelebene aggregiert. Die Daten werden also je Artikel zusammengefasst. Für jede Artikel-Monat-Kombination wird somit eine Vorhersage erstellt, die mit den tatsächlich eingetroffenen Werten verglichen wird. Der Fehler wird anhand des $sMAPE$s bestimmt. Bei den in den Darstellungen verwendeten Fehler wird der durchschnittliche $sMAPE$ der vorgeschlagenen Attributsauswahl verwendet.

4.2 DIE SAISONLÄNGE

In einem ersten Versuch soll überprüft werden, ob die festgelegte Saisonlänge von 12 Monaten eine geeignete Wahl der Vorhersage darstellt, oder ob zum Beispiel die beiden Vormonate des vorherzusagenden Monats repräsentativer für den aktuellen Übergang sind. Dazu wurde SIS.predict mit unterschiedlichen Saisonlängen ausgeführt. Als statische Vorhersagevariable wurde *sales_units*, also die Verkaufsanzahl des Vormonats, verwendet.

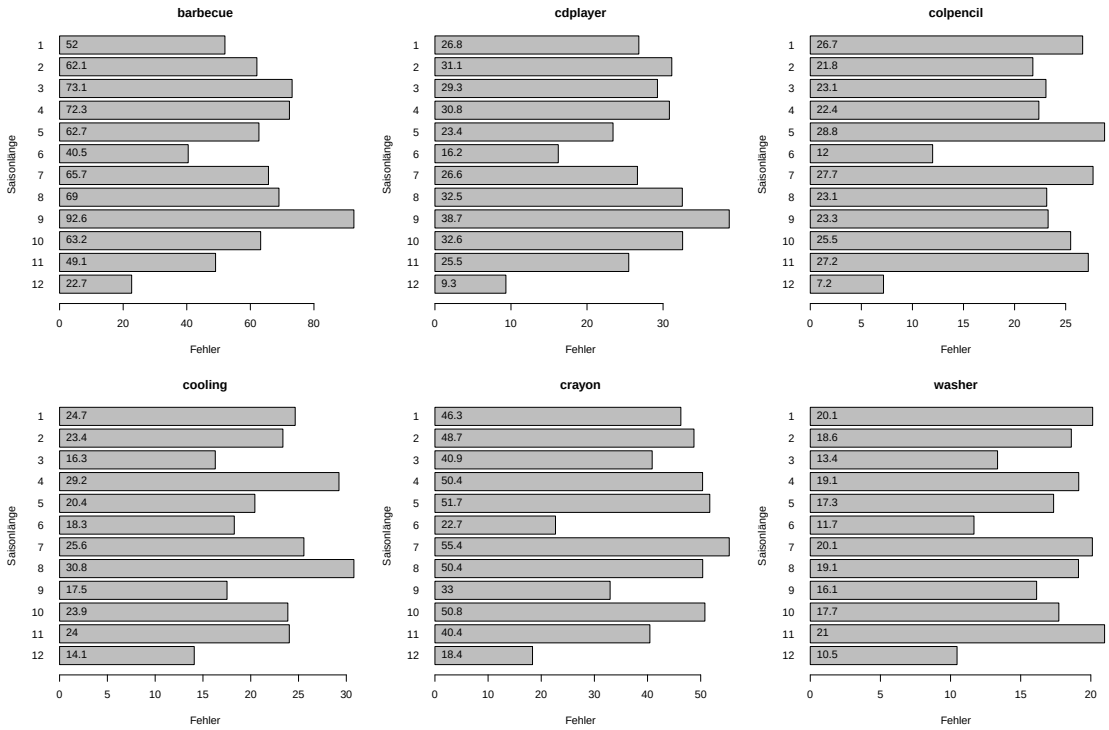


Abbildung 4.1: Einfluss der Saisonlänge auf die Vorhersagen

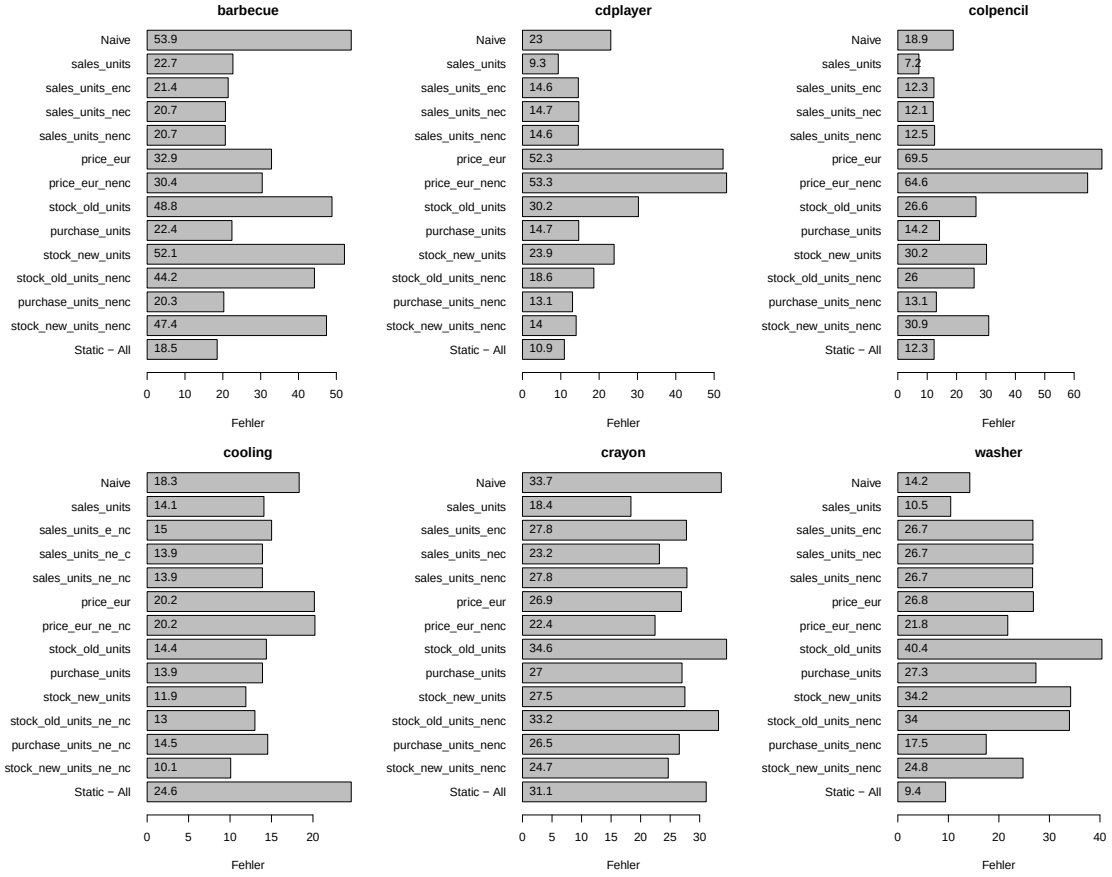


Abbildung 4.2: Einfluss eines statischer Modelle auf die Vorhersagen

In Abbildung 4.1 sind die durchschnittlichen relativen Fehler für die unterschiedlichen Saisonlängen abgebildet. Obwohl bei einigen Datensätzen eine Saisonlänge von sechs Monaten ebenfalls ein gutes Ergebnis liefern kann, zeigen die Ergebnisse eindeutig, dass die Saisonlänge von zwölf Monaten die besten Werte liefert. Auch der Übergang der beiden Vormonate (Saisonlänge 1) ist offensichtlich keine gute Wahl. Im Verlauf dieser Arbeit wird also zur Modellbildung – und damit auch zur Merkmalsauswahl – die Monate und Vormonate von vor genau einem Jahr verwendet werden.

4.3 STATISCHE MODELLE

In dem zweiten Experiment soll gezeigt werden, wie wichtig die richtige Wahl eines Attributs ist. Dazu wurde jedes der möglichen numerischen Attribute als einzelnes Regressionsattribut für die Vorhersage festgesetzt und eine komplette Vorhersage über das letzte Jahr der Datensätze erzeugt und verglichen. Die Ergebnisse sind in Abbildung 4.2 zu sehen. Neben den Ergebnissen für die einzelnen Attribute ist außerdem noch das Ergebnis für ein Modell aufgeführt, dass alle zwölf Attribute in das Vorhersage-Modell aufnimmt, sowie die naive Vorhersage zum Vergleich. Die naive Vorhersage geht davon aus, dass die vorherzusagenden Verkaufszahlen für den Monat t

den Zahlen des Vormonats entsprechen, sodass gilt

$$\hat{x_t} = x_{t-1} \tag{4.1}$$

In allen Diagrammen wird deutlich, dass die richtige Wahl des Regressionsattributs entscheidend für die Vorhersage ist. Mit der richtigen Wahl lässt sich der Fehler gegenüber der naiven Vorhersage signifikant senken. Eine falsche Wahl kann jedoch zu einer Vervielfachung des Vorhersagefehlers führen. Zudem wird deutlich, dass die Aufnahme aller Attribute in das Vorhersage-Modell ein gutes Ergebnis liefern kann (barbecue, cdplayer, washer), jedoch auch zu einem höheren Fehler führen kann (cooling, crayon). Bei dem Datensatz cooling liefert die Aufnahme aller Attribute sogar das schlechteste der gezeigten Ergebnisse.

Eine gute Wahl für ein Regressionsattribut bildet sales_units, welches gleichzeitig das Zielattribut ist. Dieses Verhalten ist die Motivation für autoregressive Modelle, die eine Vorhersage nur anhand der historischen Entwicklung des Zielattributs bestimmt. Lediglich bei dem barbecue- und bei dem cooling-Datensatz finden sich bessere Einzelattribute. Der Zusammenhang zwischen Korrelation von Regressions- und Zielattribut und der Vorhersagefähigkeit des Regressionsattributs wurde bereits in Abschnitt 3.2 beschrieben (Abbildung 3.4).

4.4 EVALUATION DER MERKMALSAUSWAHL-ALGORITHMEN

Nachdem die Eigenschaften der Datensätze beschrieben wurde und erste Vorexperimente unternommen wurden, kann eine Evaluation der Merkmalsauswahl-Algorithmen stattfinden. Im vorherigen Kapitel konnte bereits gezeigt werden, dass selbst bereits die Wahl eines einzelnen Attributs einen großen Einfluss auf die Vorhersagequalität der Vorhersage besitzt. Zudem wurde gezeigt, dass die Wahl aller Attribute ebenfalls zu hohen Fehlerraten führen kann. In diesem Abschnitt wird untersucht, welche Methoden die günstigste Attributsauswahl zurückgibt und damit in besonderem Maße für die Verkaufsdatenvorhersage geeignet ist.

4.4.1 Implementation der Merkmalssuche

Die in Kapitel 3 erläuterten Algorithmen wurden mit den folgenden Spezifikationen unter R implementiert. Als Eingabe dienen die Daten der ersten 24 Monate der insgesamt 3 Jahre langen Datensätze. Sofern eine Kreuzvalidation nötig ist, wird diese mit $k = 3$ durchgeführt. Das bedeutet, dass die Trainingsdaten zufällig in drei gleichgroße Fragmente aufgeteilt wurden, für die jeweils anhand der anderen beiden Teile eine Vorhersage durchgeführt wird. Um die Algorithmen vergleichbar zu halten, wurde der Zufallsgenerator für die Verteilung der Daten für jeden Datensatz mit dem gleichen Startwert initialisiert.

Best $Merit_S$ errechnet nach der Formel 3.7 den $Merit_S$ für jede mögliche Attributskombination aus und gibt die Kombination zurück, die den höchsten $Merit_S$ erhielt. Es handelt sich also um die Brute-Force-Methode.

Best $Merit_S^+$ geht nach dem selben Prinzip vor. Jedoch wird zunächst anhand der Kreuzvalidation die voraussichtliche Vorhersage-Fehler der einzelnen Attribute bestimmt und mit der Formel 3.9 w bestimmt. Auch hier wird zum Schluss die Brute-Force-Methode angewendet.

Filter-Wrapper bestimmt zunächst den $Merit_S$ für alle möglichen Attributskombinationen. Die besten 15 Ergebnisse werden herausgefiltert. (Genauer: Die Kombinationen innerhalb des höchsten $\frac{15}{2^n-1}$-Quantil werden herausgefiltert. So wird verhindert, dass das Ergebnis an 16. Stelle herausgefiltert wird, wenn dieses die selbe Bewertung erhielt, wie die Kombination an 15. Stelle.) Anhand der Kreuzvadidation wird der voraussichtliche Fehler aller Kombinationen bestimmt und diejenige mit dem niedrigsten Fehler zurückgegeben.

Forward Selection und Backward Elemination wurden exakt so implementiert, wie in Abschnitt 3.5.2 erläutert.

Der Genetische Algorithmus besitzt folgende Konfiguration:
Mit sechs Individuen wurde eine relativ geringe Populationsgröße bestimmt. Jedoch werden in jeder Iteration zwölf neue Individuen erzeugt. Die Mutationsrate beträgt zu Beginn zwei je neuem Individuum, sinkt jedoch mit jeder Iteration. Von den somit 18 vorhandenen Individuen gelangen nur die besten sechs in die nächste Iteration. Zur Bewertung wird wieder der voraussichtliche Fehler mittels der Kreuzvalidation bestimmt. Zur Zeitersparnis werden die Bewertungen für jede gefundene Kombination zwischengespeichert und bei Bedarf neu abgerufen. Bei den anderen Algorithmen ist dieses Vorgehen nicht nötig, da diese keine der untersuchten Belegungen zwei mal überprüfen müssen. Hat sich über zwei Generationen das Individuum mit der besten Bewertung nicht verändert, wird dessen Kombination durch den genetischen Algorithmus zurückgegeben. Da, wie bereits festgestellt wurde, das Zielattribut fast immer einen positiven Einfluss auf den Vorhersagefehler besitzt, befindet sich in der Ausgangspopulation zunächst ein Individuum, welches der Belegung dieses Attributs allein entspricht.

4.4.2 Versuch

Der Versuch besteht darin, dass für jeden Datensatz jeder der im vorigen Abschnitt genannten Algorithmen durchgeführt wird. Als Eingabe dienen die Daten der ersten 24 Monate der insgesamt 3 Jahre langen Datensätze. Die Wrapper-Ansätze erhalten zudem noch als Bewertungsfunktion die in die Kreuzvalidation eingebettete Zielfunktion SIS.predict. Die von den eben genannten Algorithmen ermittelten Attributskombinationen werden als Modell zur Vorhersage bestimmt. Dessen Parameter werden für jeden Monat festgesetzt und die Vorhersage berechnet (siehe Abschnitt 2.1). In der Abbildung 4.3 sind die Bewertungen der so entstandenen Modelle aufgeführt.

Bei Static handelt es sich um das statische Modell, in dem zur Vorhersage lediglich das Zielattribut, also *sales_units* verwendet wird. Die senkrechte Linie stellt das bestmögliche Ergebnis dar, welches mit Zukunftswissen erreicht werden kann. Dazu wurden sämtliche Attributskombinationen auf die Zielfunktion ausgewertet und diejenige mit dem besten Ergebnis ausgewählt. Diese Linie dient lediglich dem Vergleich. Für gewöhnlich lässt sich ein solches Ergebnis nicht anhand der historischen Daten erreichen, da es in solchen Daten immer Unsicherheitsfaktoren gibt und externe Einflüsse unmöglich abgedeckt werden können.

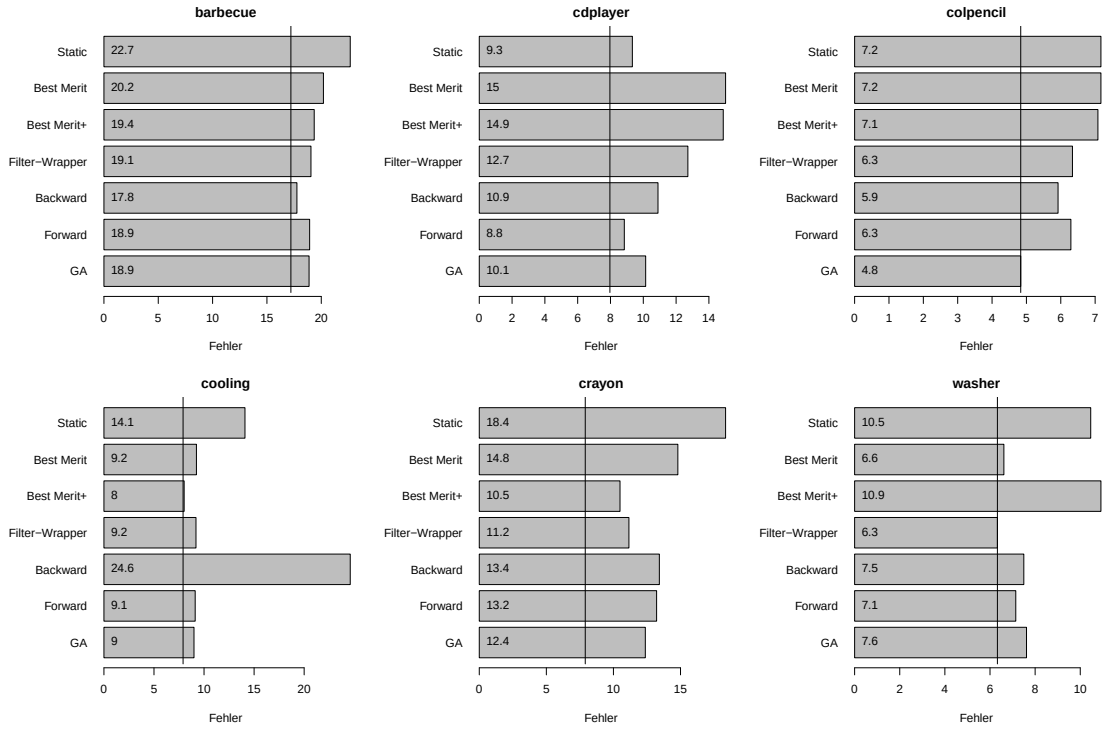


Abbildung 4.3: Fehler für die Modelle, deren Attribute von den jeweiligen Algorithmen festgelegt wurden

Welche Merkmale von den jeweiligen Algorithmen vorgeschlagen wurden, kann der Tabelle auf Seite 4.1 entnommen werden. Die fett gedruckten Algorithmen sind jeweils diejenigen, die das beste der sechs Ergebnisse geliefert haben. Ausgegraut sind die Attributskombinationen, die zu einem schlechteren Ergebnis als die statische Variante geführt haben.

4.4.3 Auswertung

Eine eindeutige Aussage zu treffen ist anhand der in Abbildung 4.3 dargestellten Ergebnisse nur schwer möglich. Beinahe jeder Algorithmus liefert mal besonders gute, mal weniger gute Lösungen.

In den meisten Fällen konnte ein niedrigerer Fehler als beim statischen Modell erreicht werden, jedoch tritt in einigen Fällen eine drastische Verschlechterung auf. Der Datensatz cdplayer stellt eine besondere Herausforderung dar. Bereits das statische Modell liefert einen niedrigen Fehler, der nahe am bestmöglichen Ergebnis liegt. Lediglich der Forward Selection-Algorithmus konnte dessen Wert unterbieten. Auch in den anderen Datensätzen, liefert der Forward Selection-Algorithmus gute Ergebnisse. Der Backward Elemination-Algorithmus bietet für einen Datensatz die beste Lösung, versagt jedoch schwer im cooling-Datensatz. Der genetische Algorithmus führt – von dem cdplayer-Datensatz abgesehen – stets zu einer Verbesserung des Prognosefehlers. Für colpencil erreicht dieser sogar das Optimum.

Die beiden reinen Filter-Ansätze sind einige male schlechter als die Wrapper-Ansätze, jedoch nicht immer. Die Verbesserung von $Merit_S^+$ hat seine Wirkung nicht verfehlt und liefert bis auf

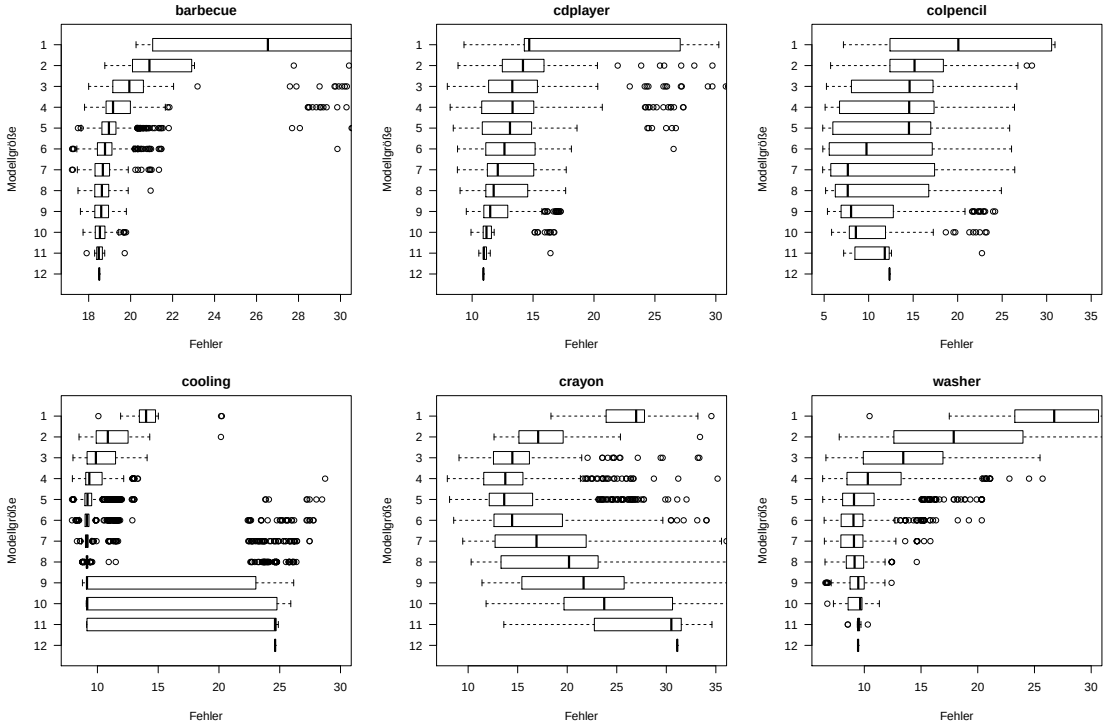


Abbildung 4.4: Verteilung der Fehler in Abhängigkeit von der Modellgröße

eine Ausnahme regelmäßig bessere Ergebnisse als $Merit_S$. Bei dem washer-Datensatz gibt der Filter-Wrapper-Ansatz das Optimum aus. Insgesamt liefern die beiden Wrapper-Ansätze Forward Selection und Backward Elemination gute Ergebnisse. Eine Ausnahme bildet lediglich der Backward-Algorithmus für die cooling-Datensätze.

Erste Erklärungsansätze für die stark schwankenden Verhalten bietet die Tabelle auf Seite 41. So erkennt man, dass der Backward Elemination-Algorithmus bei den cooling- und cdplayer-Daten in einem lokalen Optimum bei den Trainingsdaten hängen geblieben sein muss, nachdem er gerade mal ein Attribut entfernt hat. Der Forward Selection-Algorithmus erkennt bei den cdplayer-Daten zwei der drei Attribute für das bestmögliche Modell. Der Algorithmus hat die Kombination mit dem dritten Attribut zwar in Betracht gezogen, diese jedoch nach der Bewertung mittels Kreuzvalidation wieder verworfen. Weiterhin zeigt die Tabelle, dass die Algorithmen innerhalb eines Datensatzes häufig die gleichen Attribute ausgeben. Auch die Wahl der Attribute zwischen verschiedenen Datensätzen ist ähnlich, jedoch nicht gleich. So werden fast immer das Zielattribut sales_units und das Attribut price_eur_nenc in die Ergebnismenge aufgenommen.

4.4.4 Nähere Untersuchung

Bisher konnte noch nicht eindeutig erklärt werden, wie das Verhalten der Algorithmen zustande kommt. Aus diesem Grund wurde ein weiter Versuch durchgeführt. Zur Bestimmung der bestmöglichen Attributskombination wurden bereits alle möglichen Kombinationen ausgewertet. Diese Daten wurden für die Abbildung 4.4 verwendet. Abgebildet ist jeweils die Verteilung der Fehler in Abhängigkeit von der Modellgröße, also der Anzahl der verwendeten Regressionsattributen.

Dabei ergeben sich für jede Modellgröße k und maximal möglicher Attributanzahl n je $\binom{n}{k}$ mögliche Kombinationen.

Hauptaugenmerk liegt dabei auf dem Minimum und dem Median, weshalb der Übersichtlichkeit halber der rechte Rand nicht vollständig gezeigt wird. In allen Boxplots wird deutlich, dass der Median und das Minimum einen hohen Fehler aufweisen. Mit zunehmender Attributanzahl, verringert sich der Fehler. Umgekehrt weißt auch ein Modell mit allen Attributen meist einen hohen Fehler auf. Nicht immer verbessert sich der Median, jedoch immer der minimale Fehler mit abnehmender Attributsanzahl. Das Optimum, also der geringste Fehler über alle Modelle, wird dabei von unterschiedlichen Datensätzen an unterschiedlichen Punkten erreicht.

Vergleicht man diese Abbildung mit der Abbildung 4.3 auf Seite 38, bzw. mit der Tabelle auf Seite 41, wird deutlich, dass der Forward Selection-Algorithmus sehr gute Ergebnisse liefert, wenn das Optimum bei einer niedrigen Modellgröße liegt. Umgekehrt führt der Backward-Elemination-Algorithmus zu sehr guten Ergebnissen, wenn das Optimum bei einer relativ hohen Attributsanzahl liegt. Eine Ausnahme bildet der washer-Datensatz, bei dem der Filter-Wrapper-Ansatz sogar das Optimum findet. Jedoch liefert hier der Forward Selection-Ansatz ein besseres Ergebnis als der Backward Elemination-Ansatz.

Dieses Verhalten lässt sich damit erklären, dass beide Algorithmen eine Best-First-Baumsuche durchführen und auf dem Weg zum globalen Optimum Gefahr laufen, in einem lokalen Optimum zu landen. Je länger der Weg ist, desto höher ist die Wahrscheinlichkeit, dass sie in einem lokalen Optimum stehen bleiben. Der Algorithmus mit dem voraussichtlich kürzeren Weg zum Optimum erreicht mit höherer Wahrscheinlichkeit eine Lösung nahe dem Optimum.

4.5 EIN KOMBINIERTER FORWARD-BACKWARD-ANSATZ

Aus der Motivation heraus, dass mindestens einer der beiden Algorithmen Forward Seleciton und Backward Elimination sehr gute Ergebnisse erzielt, wurde eine Kombination der beiden Algorithmen implementiert, wie sie in dem letzten Absatz von Abschnitt 3.5.2 bereits angekündigt wurde. Die Suche wurde sowohl mit der Forward Selection, als auch der Backward Elimination ausgeführt. Am Ende liegen zwei mögliche Merkmalskombinationen vor, von denen diejenige ausgegeben wird, die bei der Kreuzvalidierung eine bessere Bewertung erfahren hat. Bei der Wrapper-Methode ist dies der Fall, wenn der prognostizierte Vorhersagefehler kleiner ist. Wie in dem vorangegangenen Experiment wurde die Suche für alle Datensätze durchgeführt. Als Trainingsdaten wurden die ersten 24 Monate verwendet. Das von der Suche vorgeschlagene Attributskombination wurde dann für das dritte Jahr ausgewertet. Das Ergebnis ist in Abbildung 4.5 dargestellt. An den Ergebnissen erkennt man, dass in der Regel jeweils die Attributskombination gewählt wurde, die einen niedrigeren Fehler verursacht. Lediglich bei dem crayon-Datensatz wählt der Algorithmus die schlechtere Variante. Das lässt sich damit Begründen, dass sich die Trainingsdaten von den darauffolgenden Daten unterscheiden, was jedoch nicht verhindert werden kann. In allen Datensätzen führte diese Variante zu einem niedrigeren Fehler der Vorhersage gegenüber dem statischen Modell.

Datensatz	Algorithmus	sales_units	sales_units_enc	sales_units_nec	sales_units_nenc	price_eur	price_eur_nenc	stock_old_units	purchase_units	stock_new_units	stock_old_units_nenc	purchase_units_nenc	stock_new_units_nenc
barbecue	**Backward**	×			×	×	×	×	×	×	×		×
	Filter-Wrapper	×	×						×				×
	Forward	×				×	×	×	×		×	×	
	GA			×		×		×	×	×	×	×	×
	Best $Merit_S$	×										×	
	Best $Merit_S^+$					×		×				×	
	Best			×	×	×		×	×		×		×
cdplayer	Backward	×		×	×	×	×	×	×	×	×		×
	Filter-Wrapper	×									×	×	
	Forward	×											×
	GA	×			×		×					×	
	Best $Merit_S$								×	×			
	Merit2			×		×	×	×	×			×	
	Best	×				×							×
colpencil	Backward	×	×	×		×		×	×	×		×	
	Filter-Wrapper	×	×		×								
	Forward	×	×		×		×	×					
	GA	×	×		×		×	×	×		×		
	Best $Merit_S$	×											
	Best $Merit_S^+$	×					×	×	×				
	Best	×	×		×		×		×	×	×		
cooling	Backward	×	×	×		×	×	×	×	×	×	×	×
	Filter-Wrapper	×	×						×	×			
	Forward	×	×		×	×	×		×		×	×	×
	GA	×	×	×	×		×		×	×			×
	Best $Merit_S$		×					×	×	×		×	
	Best $Merit_S^+$					×		×				×	
	Best					×	×		×	×	×	×	×
crayon	Backward	×		×	×	×	×	×		×		×	
	Filter-Wrapper	×					×		×	×			×
	Forward	×				×	×		×				×
	GA	×	×	×		×			×	×			×
	Best $Merit_S$	×					×		×				
	Best $Merit_S^+$	×					×			×			
	Best		×		×		×				×		
washer	Backward	×	×	×	×		×	×	×	×	×		
	Filter-Wrapper	×					×					×	×
	Forward	×	×	×	×		×		×				×
	GA	×	×	×	×	×		×		×			
	Best $Merit_S$	×					×					×	
	Best $Merit_S^+$						×		×		×		
	Best	×					×					×	×

Tabelle 4.1: Die von den jeweiligen Algorithmen ausgewählten Attribute

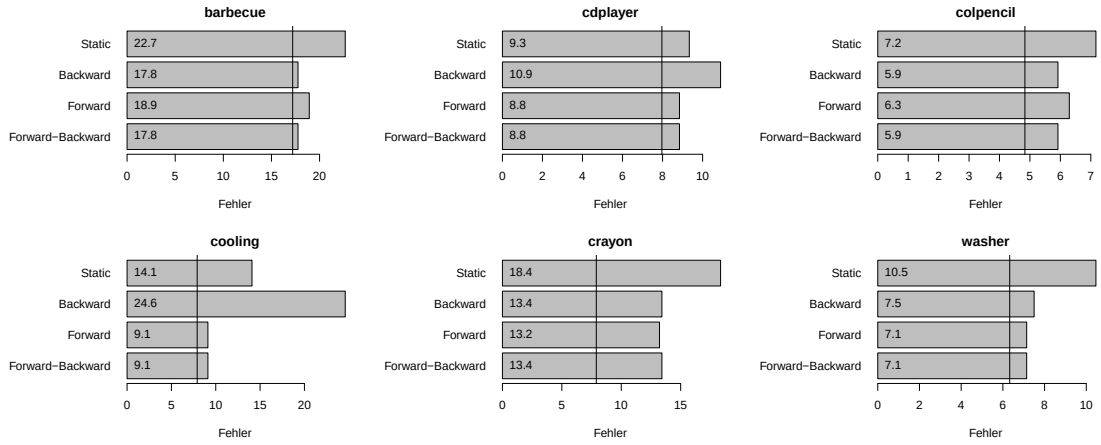


Abbildung 4.5: Fehler für die Forward-Backward-Kombination

4.6 ZEITLICHE BETRACHTUNGEN

Zum Abschluss sollen noch die Zeitkosten in Betracht gezogen werden. Neben der Zeit, die ein Merkmalsauswahl-Algorithmus zur Ergebnisfindung benötigt, ist in Abbildung 4.6 auch die Anzahl der Aufrufe von SIS.predict aufgeführt. Dabei bedeuten 300 Aufrufe, dass 100 Attributkombinationen ausgeführt wurden, da die Kreuzvalidierung für jede Kombination je 3 mal ausführt wird. Wie lang SIS.predict zur Berechnung der Daten braucht, hängt von der größe des Datensatzes ab. So benötigte der verwendete Rechner für den Datensatz crayon mit nur wenigen Hunderttausend Einträgen etwa 2 Sekunden, während für den cooling-Datensatz mit mehreren Millionen Einträgen das zwanzigfache der Zeit nötig ist.

In den Diagrammen wird deutlich, dass ein starker Zusammenhang zwischen der Anzahl der Ausführungen von SIS.predict und der verstrichenen Zeit existiert. Dies liegt daran, dass die Zielfunktion der aufwendigste Teil der Suche darstellt. So terminiert der Best $Merit_S$-Ansatz, der kein einziges Mal die Zielfunktion aufruft, bereits nach wenigen Sekunden, während Best $Merit_S^+$ mit 36 Aufrufen ($= 3 \cdot 12$) bereits wesentlich mehr Zeit benötigt.

Der Filter-Wrapper-Aufruf ist mit konstanten 45 Aufrufen geringfügig langsamer. Die Wrapper-Ansätze Forward Selection und Backward Elimination benötigen bereits wesentlich mehr Aufrufe und die Kombination verständlicherweise noch mehr. Die beiden Baumsuchen verursachen zwischen 100 und 200 Aufrufe, was bedeutet, dass sie jeweils um die 30 bis 70 Kombinationen der 4095 Möglichkeiten überprüft wurden. Überrascht hat der genetische Algorithmus, der bereits mit vergleichsweise wenig Aufrufen akzeptable Ergebnisse erreicht.

Neben der absoluten Zeit, bzw. der Anzahl der Aufrufe der Zielfunktion, spielt auch die Skalierbarkeit eine große Rolle. Diese soll anhand der möglichen Extremfälle bewertet werden. In Tabelle 4.2 sind die jeweiligen minimalen und maximalen Zeitkosten der jeweiligen Algorithmen aufgeführt. n steht dabei für die Problemgröße, also die zur Verfügung stehenden Attribute.

Best $Merit_S$ benötigt immer 2^n Bewertungen, da bei diesem Ansatz alle möglichen Kombinationen bewertet werden. Wie man in Abbildung 4.6 sehen kann, stellt das zumindest für geringe Problemgrößen kein Problem dar. Jedoch kann sich dies durch das exponentielle Wachstum än-

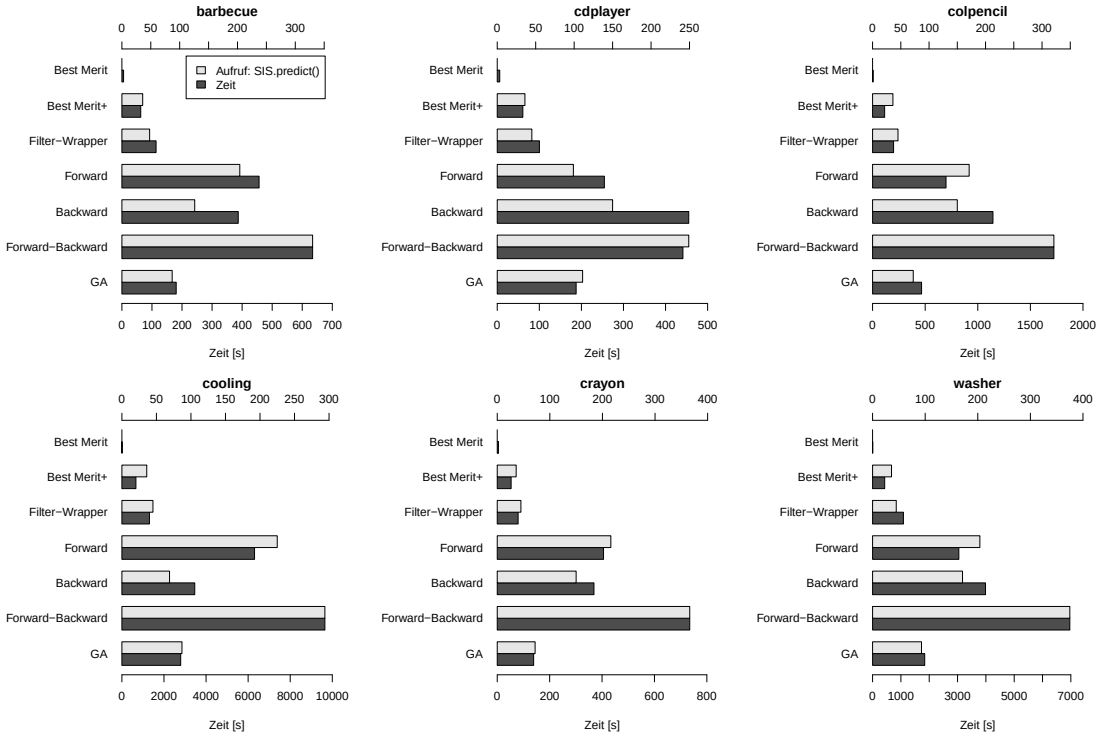


Abbildung 4.6: Zeitkosten der Algorithmen für die Datensätze

dern. Ist dies der Fall, muss auch für die Filter-Ansätze eine Suche stattfinden. Ebendies gilt auch für Best $Merit_S^+$ und Filter-Wrapper. Die linearen Kosten für den Wrapper-Anteil benötigen zwar zunächst den größten Zeitanteil. Doch auch hier kann schnell mit steigendem n schnell ein Umschwung stattfinden, der eine Suche auf den Filter-Teil nötig macht. Die beiden Baumsuchen müssen in der vorgestellten Implementation mindestens die ersten beiden Ebenen durchsuchen, bevor sie die Suche abbrechen. Im schlechtesten Fall führen sie diese Tiefensuche bis zum Schluss durch. Damit gehören sie zu den Algorithmen mit polynomieller Zeitkomplexität. In der vorgestellten Konfigurierung benötigte der genetischen Algorithmus mindestens eine Konstante Anzahl von k Versuchen. k steht in der hier vorgestellten Konfiguration für 6. Dieser Fall kommt lediglich zustande, wenn die Ursprungspopulation zufällig keine neuen, unbekannten Individuen erzeugt. Die Extrema des genetischen Algorithmus zu bestimmen, ist hingegen nicht so einfach, da die Zeitkomplexität von vielen Faktoren, wie Mutationsrate, Populationsgröße, Komplexität und Qualität der Fitness-Funktion und der Beschaffenheit des Suchraums abhängig sind. In der Literatur wird jedoch in der Regel eine polynomielle Zeitkomplexität $\mathcal{O}(n^2)$ angegeben [Tsa14, Lob00].

Anzumerken ist, dass in dieser Arbeit lediglich sequentielle Algorithmen betrachtet wurden. Sämtliche vorgestellte Algorithmen bieten Möglichkeiten zur Parallelisierung und damit Raum zur Beschleunigung. Siehe dazu auch die Anmerkungen in der Zusammenfassung.

Algorithmus	Minimum		Maximum		Zeitkomplexität	
Best $Merit_S$	2^n		2^n		$\mathcal{O}(2^n)$	
Best $Merit_S^+$	n	2^n	n	2^n	$\mathcal{O}(n)$	$\mathcal{O}(2^n)$
Filter-Wrapper	2^n	kn	2^n	k	$\mathcal{O}(2^n)$	$\mathcal{O}(1)$
Forward Selection	$2n-1$		$\frac{n(n+1)}{2}$		$\mathcal{O}(n^2)$	
Backward Elimination	$2n-1$		$\frac{n(n+1)}{2}$		$\mathcal{O}(n^2)$	
Forward-Backward	$4n-2$		$n(n+1)$		$\mathcal{O}(n^2)$	
GA	k		(n^2)		$\mathcal{O}(n^2)$	

Tabelle 4.2: Laufzeit und Komplexität der Merkmalsauswahl-Algorithmen

5 ZUSAMMENFASSUNG

5.1 ERGEBNIS

Die richtige Wahl der Modellattribute ist für die Zeitreihenvorhersage mittels Regression unerlässlich. In dieser Arbeit konnte gezeigt werden, dass eine automatisierte Merkmalsauswahl potentiell in der Lage ist, die Qualität einer solchen Vorhersage erheblich zu verbessern gegenüber Modellen mit einer statischer Attributsauswahl. Dabei konnten sowohl Filter- als auch Wrapper-Modelle den Prognosefehler verbessern und die Genauigkeit steigern. Nur in Einzelfällen kann es zu Ergebnissen kommen, die schlechter als die des statische Modells sind. Gelingt es jedoch, diese Fälle zu vermeiden, können insbesondere in Anbetracht der Rechenzeit folgende Schlüsse gefasst werden:

Für zeitkritische Anwendungen kommt nur eine Filter-Merkmalsauswahl wie der Best $Merit_S$-Ansatz in Frage, jedoch birgt dieser das größte Risiko einer Fehlentscheidung und führt am seltensten zu dem besten Ergebnis. Bei weniger zeitkritischen Anwendungen sollten der Wrapper-Ansatz gewählt werden. Insbesondere die Kombination des Forward Selection- und Backward Elemination-Algorithmus können hervorragende Ergebnisse liefern. Ähnliche gute Ergebnisse liefert der genetische Algorithmus. Für eine hohe Attributszahl bietet sich die Filter-Methode als Vorfilter für die Wrapper-Methode an.

5.2 AUSBLICK

Die in dieser Arbeit verwendeten Algorithmen wurden lediglich in ihrer sequentiellen Form betrachtet. Dabei bieten alle dieser Algorithmen ein hohes Potential zur Parallelisierung und damit sowohl zur Beschleunigung, als auch zur Verbesserung des Vorhersagefehlers. So müssen die Suchalgorithmen mehrere Kombinationen unabhängig voneinander überprüfen, was sich ideal zur Verteilung anbietet. SIS.predict bietet bereits die Möglichkeit der parallelen Abarbeitung. Vorsicht ist jedoch geboten, wenn alle Recheneinheiten auf den selben Arbeitsspeicher zugreifen müssen, da SIS.predict bei großen Datensätzen sehr platzintensiv ist. Wie groß die Auswirkung

der parallelen Umsetzung der genannten Algorithmen ist, ist zu untersuchen.

Die vorliegenden Datensätze besitzen lediglich zwölf kardinale Merkmalsausprägungen. Eine Validation der Ergebnisse dieser Arbeit auf die Merkmalsauswahl mit einer großen Anzahl von Attributen blieb somit bisher aus. Insbesondere in Anbetracht der Tatsache, dass die Forward Selection und die Backward Elemination ihre Suche jeweils mit einer leeren bzw. vollen Attributsauswahl beginnen und sich dann einem lokalen Optimum nähern, könnte bei tiefen Suchbäumen zu schweren Fehlern führen.

In Einzelfällen verursachten beinahe alle Algorithmen ein Ergebnis, dass schlechter war als das statische Modell ist. Es ist nicht gelungen, die Ursachen für dieses Verhalten eindeutig zu bestimmen oder solche Ausreißer automatisch während der Trainingsphase zu erkennen.

Die Kreuzvalidation verhindert die Überanpassung eines Modells an seine Trainingsdaten. Aus Gründen der Zeit- und Platzersparnis wurden die Trainingsdatensätze lediglich in drei Teile aufgespalten. Bei der Wiederholung einiger Experimente zeigte sich, dass eine unterschiedliche Verteilung der Trainingsdaten bei der Kreuzvalidation gelegentlich zu Ausreißern bei der Evaluation des Ergebnissets führt. Dies betraf vor allem die kleinen Datensätze, während die Ergebnisse der größeren Datensätze davon unbeeinflusst blieben. Da auch in den vorgestellten Daten, die Algorithmen gelegentlich sehr schlechte Ergebnisse erzielt haben, muss Untersucht werden, ob diese allein durch eine ungünstige Verteilung der Trainingsdaten zustande kamen oder ob andere Ursachen in Frage kommen. Es wird ein Mechanismus benötigt, der solche Ausreißer verhindert.

In dieser Arbeit wurde lediglich auf die Wirkung des $sMAPE$ auf die Gesamtfehler der einzelnen Monate eingegangen. Jedoch können die Vorhersagen auch nach anderen Attributen, wie zum Beispiel der Marke oder dem Bundesland der Verkaufsstandorte, aggregiert werden. Erste Experimente haben gezeigt, dass eine Verbesserung des Gesamtfehlers häufig auch zur Verbesserung des Fehlers auf Bundesland-Ebene führen, jedoch selten zur Verbesserung auf Marken-Ebene.

Eine weitere Möglichkeit zur Verbesserung der Vorhersage bietet die getrennte Untersuchung von Partitionen der Datensätze nach nicht-numerischen Attributen. Zu Untersuchen ist, ob unterschiedliche Partitionen unterschiedliche Modelle präferieren. Prinzipiell wurde bereits eine Partitionierung vorgenommen, da die Datensätze nach Produktgruppen aufgeteilt wurden. So stellt jede Produktgruppe eine eigene Partition dar, für die je eine eigene Modellauswahl getroffen wurde. Aufgabe ist es, die Granularität zu erhöhen und zu überprüfen, ob auch innerhalb der Produktgruppen unterschiedliche Modelle angewendet werden können. Gleiches gilt für unterschiedliche Monate. Bereits jetzt werden in SIS.predict jeden Monat andere Modellparameter festgelegt. Zu überprüfen ist, ob nicht nur die unterschiedliche Parameter, sondern sogar unterschiedliche Modelle zu noch genaueren Vorhersagen führen können. Erste Experimente dazu haben gezeigt, dass die Modellattribute einzelner Monate sich tatsächlich unterscheiden und dieses Vorgehen zu einer Verbesserung der Vorhersage führen kann.

5.3 VERWANDTE ARBEITEN

Im Bereich der Merkmalsauswahl konzentriert sich der Großteil der Literatur auf die Verwendung von Merkmalsauswahl-Algorithmen zur Klassifizierung [Joh94, Koh97, Seb02, HSL99, Hall99,

Inz04]. Nur wenige Werke widmen sich jedoch auch der Merkmalsauswahl zur Regression, z.B. [Bro97].

Die Korrelationsbasierte Merkmalsauswahl beschreibt Hall in [HSL99] und vergleicht sie dort mit Wrapper-Ansätzen. Ausführlicher geht er auf diese Methode in seiner Dissertation ein [Hall99]. Die Funktion $Merit_S$ führt er dort u.a. auf [Ghi64] zurück. Komplexere Filter-Wrapper-Hybride als in dieser Arbeit werden in [Seb02] und [Xin01] vorgestellt. Die Backward Elimination wurde zuerst von [Mar63] vorgestellt und von [Kit86] verallgemeinert wurde und so unter anderem auch zur Feature Selection führt. In [Bro97] kommt der Genetische Algorithmus ebenfalls zur Merkmalsauswahl in Regressionsmodellen zum Einsatz.

Eine Einführung in die statistischen Grundlagen bietet [Bam01], sowie [Hat11]. Auch der Korrelationskoeffizient wird, als wichtiges Maß in der Statistik, in beiden Büchern beschrieben. [Hat11] ist zudem eine gute Anleitung zum Umgang mit R als Werkzeug zur Datenanalyse und -darstellung. [Lig07] geht hingegen auf R als Programmiersprache ein.

DANKSAGUNG

An dieser Stelle möchte ich mich bei allen Personen bedanken, die mich bei der Erstellung dieser Arbeit und auf dem Weg dahin unterstützt haben.

Besonderer Dank gilt Claudio Hartmann, der mir nicht nur durch seine moralische Unterstützung, sondern mit seinen kritischen Fragen und guten Ratschlägen eine große Hilfe war.

LITERATURVERZEICHNIS

[Bam01] BAMBERG, Günter; BAUR, Franz *Statistik.* R. Oldenbourg Verlag München Wien (2001) (11. Auflage)

[Bow93] BOWERMAN, Bruce L.; O'CONNELL, Richard T. *Forecasting and time series: an applied approach.* Duxbury Press, Belmont, California (1993)

[Bro97] BROADHURST, D.; GOODACRE, R.; JONES, A.; ROWLAND, J. J.;KELL, D. B. *Genetic algorithms as a method for variable selection in multiple linear regression and partial least squares regression, with applications to pyrolysis mass spectrometry.* Analytica Chimica Acta, 348(1, 1997): 71-86.

[Cha82] CHATFIELD, Christopher *Analyse von Zeitreihen: Eine Einführung.* Teubner Verlagsgesellschaft, Leipzig (1982)

[Ghi64] GHISELLI, E. E. *Theory of psychological measurement* McGraw-Hill, New York (1964).

[Guy03] GUYON, I.; ELISSEEFF, A. *An introduction to variable and feature selection.* The Journal of Machine Learning Research, 3 (2003): 1157-1182.

[Hall99] HALL, M. A. *Correlation-based feature selection for machine learning* Doctoral dissertation, The University of Waikato (1999).

[Hat11] HATZINGER, R.;HORNIK, K.; NAGEL, H. *R: Einführung durch angewandte Statistik.* Pearson Deutschland GmbH (2011).

[Hol95] HOLMES, G.; NEVILL-MANNING, C. G. *Feature Selection via the Discovery of Simple Classifcation Rules.* Proceedings of the International Symposium on Intelligent Data Analysis (1995).

[HSL99] HALL, Mark A.; SMITH, Lloyd A. . *Feature Selection for Machine Learning: Comparing a Correlation-Based Filter Approach to the Wrapper.* FLAIRS conference. (1999)

[Inz04] INZA, I.; LARRAÑAGA, P.; BLANCO, R.; CERROLAZA, A. J. *Filter versus wrapper gene selection approaches in DNA microarray domains.* Artificial intelligence in medicine, 31(2) (2004): 91-103.

[Joh94] JOHN, G. H.; KOHAVI, R.; PFLEGER, K. *Irrelevant Features and the Subset Selection Problem.* ICML (Vol. 94, 1994): 121-129.

[Kir92] KIRA, K.; RENDELL, L. *A Practical Approach to Feature Selection.* Machine Learning: Proceedings of the Ninth International Conference, Morgan Kaufmann (1992)

[Kit86] KITTLER, J. *Feature Selection and Extraction.* Academic Press, New York (1986): Chapter 3, 59-83.

[Koh95] KOHAVI, R. *A study of cross-validation and bootstrap for accuracy estimation and model selection.* IJCAI (Vol. 14, No. 2, 1995): 1137-1145).

[Koh97] KOHAVI, R.; JOHN, G. H. *Wrappers for feature subset selection.* Artificial intelligence, (1, 1997): 273-324.

[Kol96] KOLLER, D.; SAHAMI, M. *Towards Optimal Feature Selection.* Machine Learning: Proceedings of the Thirteenth International Conference, Morgan Kaufmann, (1996): 294-292.

[Lig07] LIGGES, U. *Programmieren mit R* (Vol. 2). Springer. (2007).

[Lob00] LOBO, F. G.; GOLDBERG, D. E.; PELIKAN, M. *Time complexity of genetic algorithms on exponentially scaled problems.* Urbana, 51, 61801 (2000).

[Mar63] MARILL, T.; GREEN, D.M. *On the effectiveness of receptors in recognition systems.* IEEE Trans. Infirm. Theory 9 (1, 1963): l-17.

[Mar73] MARTENS, Peter. *Prognoserechnung.* Physica-Verlag, Rudolf Liebing KG, Würzburg (1973)

[Rus04] RUSSEL, Stuart; NORVIG, Peter. *Künstliche Intelligenz: Ein Moderner Ansatz.* Pearson Studium, München (2004)

[Seb02] SEBBAN, Marc; NOCK, Richard. *A hybrid filter/wrapper approach of feature selection using information theory.* Pattern Recognition 35.4 (2002): 835-846.

[Tsa14] TSAI, C. W.; TSENG, S. P.; CHIANG, M. C.; YANG, C. S.; HONG, T. P. *A High-Performance Genetic Algorithm: Using Traveling Salesman Problem as a Case.* The Scientific World Journal (2014).

[Xin01] XING, E. P., JORDAN, M. I., KARP, R. M. *Feature selection for high-dimensional genomic microarray data* ICML (Vol. 1, 2001): 601-608.

ABBILDUNGSVERZEICHNIS