TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Informatik Institut für Systemarchitektur, Professur für Datenbanken

Diplomarbeit

LASTGESTEUERTE MODELLWARTUNG FÜR PROGNOSEANFRAGEN

Lars Kegel
Matr.-Nr.: 3514043

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

Eingereicht am 31. März 2015

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, 31. März 2015

ZUSAMMENFASSUNG

Im Rahmen des Projekts Flash-Forward Query Framework wird die Zeitreihenprognose in ein Datenbankmanagementsystem (DBMS) integriert. Nutzer können Prognosemodelle für Zeitreihen in der Datenbank erstellen und für Prognoseanfragen verwenden. Wenn die Zeitreihen fortgeschrieben werden, müssen Wartungsoperationen die Anpassung der Prognosemodelle durchführen. Dies ist entscheidend, da die Modelle sonst nicht mehr aktuell sind und an Genauigkeit verlieren. Die vorliegende Arbeit stellt die Integration von Ableitungsmodellen und deren automatische Erstellung durch eine Wartungsoperation vor. Zeitreihen eines Datensatzes sind oft hierarisch organisiert, sodass direkte Prognosemodelle referenziert von Ableitungsmodellen auf mehreren Hierarchieebenen eingesetzt werden können. Es wird gezeigt, dass Ableitungsmodelle die Prognosegenauigkeit erhöhen und die Anzahl benötigter direkter Prognosemodelle reduzieren können.

Des Weiteren untersucht die Arbeit das DBMS bei hoher Systemlast, die zur Verzögerung von Prognoseanfragen führen kann. Um dem Nutzer eine Anfrage mit geringer Verzögerung zu ermöglichen, wird eine Laststeuerung entwickelt, mit der das System angefragte Prognosemodelle bevorzugt bedient. Es wird gezeigt, dass ihr Einsatz auch bei hoher Systemlast eine schnelle Bedienung der Prognoseanfragen ermöglicht. Dabei wird die Prognosegenauigkeit berücksichtigt, die der Nutzer fordert.

INHALTSVERZEICHNIS

ABBILDUNGSVERZEICHNIS

TABELLENVERZEICHNIS

1 EINLEITUNG

Wir blicken so gern in die Zukunft, weil wir das Ungefähre,
was sich in ihr hin und her bewegt, durch stille Wünsche
so gern zu unsern Gunsten heranleiten möchten.
(J. W. von Goethe in „Die Wahlverwandtschaften")

Dem Sinnspruch folgend ist es seit jeher Wunsch der Menschen, einen Blick in die Zukunft zu wagen, um daraus einen Vorteil zu erzielen. In der Praxis versuchen Experten und computerunterstützte Systeme, diesen Wunsch zu erfüllen. Sie führen die Zeitreihenprognose durch, die mithilfe der statistischen Analyse der Vergangenheit und Gegenwart eine Vorhersage für die Zukunft ermöglicht. In vielen Geschäftsfeldern ist dieses Verfahren eine wesentliche Entscheidungshilfe. Im Kontext von Business Intelligence werden komplexe wirtschaftliche Zusammenhänge erfasst, um Vorhersagen über ihre künftige Entwicklung zu treffen, vgl. [CGS12]. Bei der Planung eines industriellen Lagers muss der künftige Lagerbestand abgeschätzt werden, um Grenzen wie den Minimal- und Maximalbestand nicht zu verletzen, vgl. [MR05]. Auch bei der Prognose der Energieverteilung findet das Verfahren Einsatz. Die Stromproduktion durch Erneuerbare Energien hat in Deutschland stark zugenommen, die Produktion aus Solar- oder Windenergie unterliegt jedoch naturgemäß hohen Schwankungen. Daher ist eine Bedarfsprognose unentbehrlich, um eine gleichmäßige Stromversorgung zu gewährleisten, vgl. [BB13].

Experten können aufgrund ihrer Erfahrung häufig eine bessere Prognose als computergestützte Systeme erstellen. Bei großen Datensätzen jedoch gelingt dies nicht mehr; wenn beispielsweise 500 verschiedene Produkte in einem Lager gehalten werden und deren Zukunftsbedarf zu prognostizieren ist, ist die genaue Analyse des Experten nur unter sehr hohem zeitlichem Aufwand oder mit Verallgemeinerungen möglich. Hier beweisen sich computerunterstützte Systeme, die jedes Produkt mit der gleichen Akkuratesse analysieren und insgesamt genauer und schneller als Experten arbeiten, vgl. [MR05].

Das Projekt *Flash-Forward Query Framework* untersucht die Integration der Zeitreihenprognose in ein Datenbankmanagementsystem (DBMS). Dadurch kann die Trennung zwischen der Speicherung der Zeitreihen und der statistischen Analyse überwunden werden, die bei bisherigen computerunterstützen Systemen bestand. Zudem ergeben sich zahlreiche Vorteile: Prognosemodelle

können zum Beispiel von Experten erstellt und im DBMS abgespeichert werden, sodass beliebige Nutzer sie für eine Prognoseanfrage wiederverwenden können. Zudem können Prognoseanfragen mit Verbundoperationen in komplexe Anfragen eingebettet werden.

Das Fortschreiben von Zeitreihen erfordert eine Modellwartung, das heißt eine Anpassung der Prognosemodelle, sodass deren Prognose weiterhin in der Zukunft liegt und so genau wie möglich ist. Im Rahmen der vorangegangenen Belegarbeit wurde eine asynchrone Wartungskomponente entwickelt, die Wartungsoperationen entgegen nimmt und parallel unter Ausnutzung der zur Verfügung stehenden Betriebsmittel ausführt, vgl. [Keg14]. Bei einer mittleren Systemlast kann die Verzögerung einer Prognoseanfrage minimiert werden, da die Wartung weder das Fortschreiben der Zeitreihe noch die Prognoseanfrage belastet.

Datensätze bestehen oft aus mehreren Zeitreihen, die sich in eine oder mehrere Hierarchien einordnen. So lassen sich Produkte aus einem Lager in Produktfamilien und höhere Aggregate hierarchisieren. Dabei kann das Prognosemodell für eine Produktfamilie beispielsweise auch gute Prognosen für die einzelnen Produkte der Familien liefern oder für das Aggregat aller Produkte. Diese Beobachtung aus [Fli01] motiviert zur Integration von Ableitungsmodellen, die die Prognose anderer Modelle nutzen. Mit ihnen lassen sich Wartungsoperationen definieren, welche eine zusätzliche Erhöhung der Prognosegenauigkeit und die Einsparung von Prognosemodellen ermöglichen.

Darüber hinaus steht die Gestaltung von Prognoseanfragen im Fokus der vorliegenden Arbeit. Anfragen sollen künftig auf die Anforderungen des Nutzers angepasst werden. Wünscht der Nutzer eine Prognose mit hoher Genauigkeit, muss er eine Verzögerung der Anfrage akzeptieren, weil zunächst Wartungsoperationen abzuschließen sind. Andererseits kann er eine gering verzögerte Anfrage fordern, muss jedoch einen höheren Prognosefehler in Kauf nehmen. Insbesondere bei einer hohen Systemlast müssen Konzepte integriert werden, welche die Verzögerung eines angefragten Prognosemodells minimieren und dabei die Genauigkeitsanforderungen des Nutzers berücksichtigen. Dies motiviert zur Entwicklung einer lastgesteuerten Wartungskomponente.

Die vorliegende Arbeit ist wie folgt gegliedert: Kapitel 2 gibt einen Überblick über die Grundlagen der Zeitreihenprognose und der Prognosemethode, die für die Evaluation eingesetzt wird. Die Vorarbeit des Projekts, der Prototyp *Flash-Forward Database System* (F²DB), wird kurz vorgestellt, da er zur Evaluation der eigenen Konzepte dient. Kapitel 3 stellt die Integration von Ableitungsmodellen vor. Hierfür werden zunächst Ableitungsmodelle definiert. Anschließend ermöglichen neu eingeführte Wartungsoperationen die automatische Erstellung und Wartung von Ableitungsmodellen. Kapitel 4 befasst sich mit der Erweiterung der Prognoseanfrage um eine nutzerdefinierte Genauigkeitsklasse, sodass eine Anfrage besser auf die Anforderungen des Nutzers eingehen kann. Dies erfordert mehrere Optimierungen, sodass angefragte Prognosemodelle bevorzugt bedient werden können. Kapitel 5 stellt eine Laststeuerung vor, die das Verhalten der Wartungskomponente bei hoher Systemlast verbessert. Das Kapitel 6 fasst die Resultate der Arbeit zusammen und gibt einen abschließenden Ausblick.

2 ZEITREIHENPROGNOSE IN RELATIONALEN DATENBANKEN

In diesem Kapitel werden Grundlagen und Vorarbeiten zusammengefasst, auf denen die Arbeit aufbaut. Abschnitt 2.1 führt das Prognosemodell und die technischen sowie mathematischen Grundlagen der kurzfristigen Zeitreihenprognose ein. Abschnitt 2.2 präsentiert das Holt-Winters-Verfahren, die Prognosemethode, die bei der Evaluation der Arbeit zum Einsatz kommt. Abschnitt 2.3 stellt den Prototypen F²DB vor, in dem die Konzepte für die Zeitreihenprognose in relationalen Datenbanken implementiert werden. Abschnitt 2.4 nennt die Ziele der Arbeit.

2.1 ZEITREIHENPROGNOSE IM ÜBERBLICK

Die Zeitreihenprognose ist ein Verfahren zur statistischen Analyse von Zeitreihen und zum Herausfinden von Informationen über ihre zukünftige Entwicklung, die *Prognose*. Das Verfahren wird durch ein computerunterstütztes System, dem *Prognosesystem*, durchgeführt.

Eine *Zeitreihe* ist eine zeitabhängige Folge von Werten. Für die Arbeit wird angenommen, dass sie in gleichem, diskretem Abstand vorliegen, z. B. ein Monat oder ein Quartal. Dieser Abstand heißt *Granularität*. Die statistische Analyse erfolgt über eine Reihe von *Messwerten*, die *Historie* x_t $(t = 1, 2, \ldots, T)$. Der Zeitpunkt $t = T$ wird als Gegenwart interpretiert. Die Informationen der statistischen Analyse verdichten sich in einem *Prognosemodell*, das die systematischen Eigenschaften der Historie zusammenfasst. Es besteht aus vier Komponenten:

- Die *Prognosemethode* bestimmt die wesentliche Klasse und die Charakteristika des Prognosemodells. Typische Vertreter sind die linearen stochastischen Prozesse der Box-Jenkins-Methode [BJ70] und das Exponentielle Glätten nach dem Holt-Winters-Verfahren [Win60]. Das Verfahren wird vom Nutzer oder vom Prognosesystem ausgewählt.
- Die *Metaparameter* spezifizieren die Prognosemethode hinsichtlich der Komponenten der Zeitreihe. Dazu gehört der *Trend* und die *Saison*. Der Trend ist die lineare Veränderung der

Zeitreihe, während die Saison die zyklische Veränderung darstellt. Metaparameter, d. h. das Vorkommen dieser Komponenten in der Zeitreihe, können nicht geschätzt werden, sondern werden vom Nutzer oder vom Prognosesystem festgelegt.

- Nach Festlegung der Prognosemethode und der Metaparameter dienen die *Modellparameter* der Anpassung eines Modells auf eine spezifische Zeitreihe. Sie werden durch das Prognosesystem geschätzt.
- Der *Zustand* stellt Parameter dar, die das Prognosemodell zum aktuellen Zeitpunkt beschreiben. Zu Beginn werden sie mit geeigneten Werten initialisiert. Beim Fortschreiben der Zeitreihe werden sie auf den neuen Zeitpunkt angepasst.

Als Zusammenfassung systematischer Eigenschaften ermöglicht das Prognosemodell eine Vorhersage der zukünftigen Entwicklung der Zeitreihe. Ausgehend vom gegenwärtigen Zeitpunkt $t = T$ wird eine Prognose $\hat{x}_{T,\tau}$ $(\tau = 1, 2, \ldots, H)$ für die zukünftigen Messwerte x_t $(t = T + 1, T + 2, \ldots, T + H)$ ermittelt. Die geschätzten Werte $\hat{x}_{t,\tau}$ werden im Folgenden als *Prognosewerte* bezeichnet. Die Länge H gibt den Prognosezeitraum, den *Horizont*, an. Nach oben ist dieser Zeitraum durch die geforderte Genauigkeit der Prognose begrenzt. Je größer der Zeitraum ist, desto größer ist die Gefahr einer Fehlprognose, vgl. [Sch12a].

Solche Prognosemodelle finden vor allem in der kurzfristigen Planung eines Unternehmens Einsatz. Beispielsweise werden sie bei der Bedarfsplanung in einem industriellen Lager genutzt: Hier müssen kurzfristige Prognosen in kurzen Zeitabständen und für eine große Anzahl an Produkten verfügbar sein. Die vorliegende Arbeit konzentriert sich im Folgenden auf die kurzfristige Zeitreihenprognose.

2.1.1 Anforderungen

Schröder stellt in [Sch12a] vier Anforderungen an die Zeitreihenprognose. Wesentlich ist die *Prognosegenauigkeit*, die über die Abweichung des Prognosewerts vom Messwert beschrieben wird. Da ein Prognosewert in der Regel nicht exakt dem Messwert entspricht, ist die Ursache der Abweichung zu bestimmen. Die *Stabilität* sowie *Reagibilität* des Verfahrens sind wichtig, um zufällige von regelmäßigen Abweichungen zu unterscheiden. Der Einfluss zufälliger Abweichungen wird durch die Berücksichtigung einer langen Historie minimiert (Stabilität). Das Verfahren muss regelmäßige Abweichungen erkennen und das Prognosemodell gegebenfalls anpassen (Reagibililät). Eine weitere Anforderung ist die *Eingriffsmöglichkeit* des Menschen. Das Expertenwissen eines Nutzers ist im Prognosesystem zu berücksichtigen, da es zu einer zusätzlichen Verbesserung der Prognose führt. Schließlich fordert Schröder, dass die Prognosemethode die *effiziente Nutzung* von Rechenzeit und Speicherplatz berücksichtigt. Hierzu zählen die schnelle Berechnung der Prognose und der Umgang mit häufigen Prognoseanfragen.

2.1.2 Komponenten

Der Zeitreihenprognose geht eine Vorverarbeitung voraus, in der Daten ausgewählt und analysiert werden müssen, vgl. [Sch12a]. Die Auswahl untersucht, welche Daten und in welchem

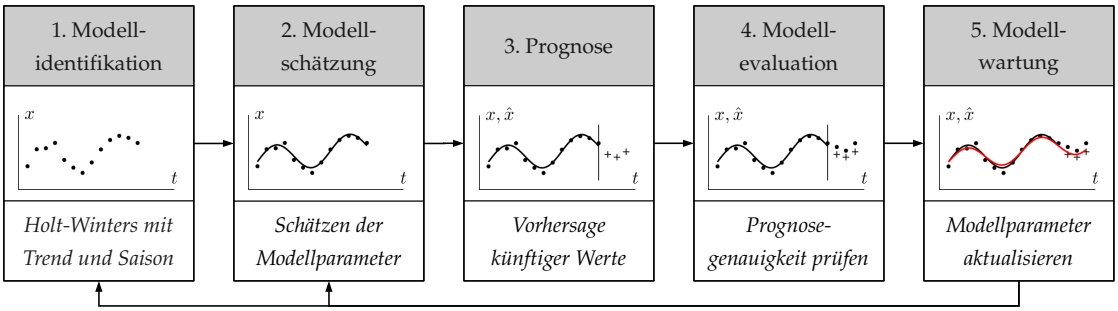


Abbildung 2.1: Komponenten der Zeitreihenprognose und Beispiel

Umfang sie zur Bildung des Prognosemodells verwendet werden. Die Datenanalyse führt anschließend eine Bereinigung der Daten von Ausreißerwerten und ungültigen Werten durch. Als Resultat geht die Historie hervor.

Die Zeitreihenprognose schließt sich der Vorverarbeitung an und wird in mehrere Komponenten aufgeteilt, die in Abbildung 2.1 dargestellt sind. Sie übernehmen folgende Aufgaben:

1. Die *Modellidentifikation* ist die Auswahl einer Prognosemethode und der Metaparameter durch den Nutzer oder durch das Prognosesystem.
2. Bei der *Modellschätzung* werden die Modellparameter für die gegebene Historie geschätzt und bezüglich eines Fehlermaßes evaluiert. Die Schätzung kann mittels verschiedener Optimierungsfunktionen stattfinden. Ein bekannter Vertreter ist der Simplex-Algorithmus nach Nelder und Mead [NM65].
3. Für den Gegenwartszeitpunkt T wird eine Prognose $\hat{x}_{T,\tau}$ $(\tau = 1, 2, \ldots, H)$ für den Horizont H bestimmt.
4. Nach Einfügung neuer Messwerte in die Historie wird der Zustand des Prognosemodells gewartet und die Genauigkeit im Rahmen der *Modellevaluation* überprüft. Hierfür werden ein *fehler-* und ein *zeitbasierter Schwellwert* eingesetzt. Der fehlerbasierte Schwellwert ergibt sich durch Mittelwertbildung über mehrere *Prognosefehler*. Ein Prognosefehler drückt die Abweichung des Prognosewerts vom Messwert aus. Der *zeitbasierte Fehler* ist die Anzahl eingefügter Zeitpunkte seit der letzten Modellschätzung. Wird der Schwellwert von einem der Fehler überschritten, so wird eine Modellwartung veranlasst. Die Schwellwerte werden vom Nutzer festgelegt und gelten für alle Prognosemodelle.
5. Das Überschreiten des fehler- oder zeitbasierten Schwellwerts impliziert, dass die Prognosewerte nicht mehr den Messwerten der Zeitreihe entsprechen. Im Rahmen der *Modellwartung* werden *Wartungsoperationen* angefordert, die eine Anpassung des Prognosemodells durchführen. Für die Modellparameter findet eine neue Modellschätzung statt. Die Metaparameter können nicht durch Schätzung bestimmt werden, sondern sie werden festgelegt. Nach anschließender Modellschätzung der Modellparameter kann die Eignung der Metaparameter überprüft werden.

Diese Komponenten beschreiben die Zeitreihenprognose als Prozess, der durch die Modellwartung eine Feedback-Funktion hat, um das Prognosemodell bestmöglich der gegebenen Historie anzupassen.

2.2 DAS HOLT-WINTERS-VERFAHREN

Als Beispiel für eine Prognosemethode stellt der folgende Abschnitt das *Holt-Winters-Verfahren* [Win60] vor. Zudem wird begründet, inwieweit es für Zeitreihenprognose in relationalen Datenbanken geeignet ist.

2.2.1 Komponentenanalyse

Das Holt-Winters-Verfahren baut auf dem Komponentenansatz der Zeitreihenanalyse auf, der eine Zeitreihe als Verknüpfung einer Trendkomponente r_t mit einer Saisonkomponente q_t und einer Restkomponente u_t beschreibt, vgl. [Sch12b]. Die Restkomponente ist die Zusammenfassung aller zufälligen Effekte, für die angenommen wird, dass sie einen Erwartungswert Null, eine konstante Standardabweichung σ hat und einer Normalverteilung gehorcht, vgl. [Sch12a]. Sie entspricht damit einer $\mathcal{N}(0, \sigma)$-verteilten Zufallsvariable.

Drei verschiedene Verknüpfungen der Komponenten sind typisch: Falls die Saisonamplitude unabhängig vom Grundwert ist, so ist sie additiv mit der Trendkomponente verknüpft:

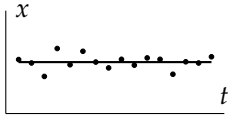
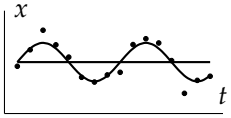
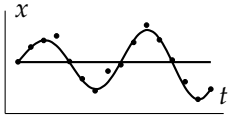
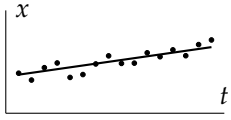
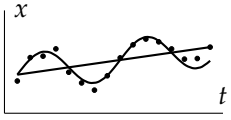
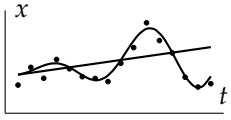
$$x_t = r_t + q_t + u_t \tag{2.1}$$

Wenn die Saisonamplitude zeitabhängig ist, kann von einer gemischt multiplikativ-additiven Überlagerung ausgegangen werden. Dabei ist die Restkomponente weiterhin additiv verknüpft:

$$x_t = r_t \cdot q_t + u_t \tag{2.2}$$

Die rein multiplikative Verknüpfung kann durch Anwendung des Logarithmus in eine additive Verknüpfung überführt werden, vgl. [Sch12b]. Eine graphische Veranschaulichung ist in Tabelle 2.1 gegeben, vgl. [Peg69]. Wenn eine Trendkomponente existiert, so wird von einem linearen, additiven Trend ausgegangen. Die Entscheidung für oder gegen eine Komponente und die Art der Verknüpfung werden vom Nutzer oder vom System getroffen. Hinzu kommt die Angabe der *Saisonlänge* L. Zum Beispiel drückt $L = 12$ die Jahressaison bei einer Zeitreihe mit monatlichen Messwerten aus. Zusammen bilden diese Angaben die Metaparameter des Prognosemodells.

Tabelle 2.1: Verknüpfungen der Komponenten einer Zeitreihe

	Ohne Saison	**Saison additiv**	**Saison multiplikativ**
Ohne Trend			
Mit Trend			

2.2.2 Exponentielles Glätten

Das Prognosemodell muss während der Modellschätzung für eine gegebene Prognosemethode und für gegebene Metaparameter auf die Zeitreihe angepasst werden. Ziel ist es, eine Menge von Modellparametern zu bestimmen, die die Historie genau beschreibt, um so eine Prognose zu ermöglichen.

Grundlage für das Holt-Winters-Verfahren ist das *Exponentielle Glätten*. Es berücksichtigt alle vergangenen Messwerte der Historie mit exponentiell fallendem Einfluss. Dafür nutzt es *Koeffizienten*, die rekursiv bestimmt werden, und *Glättungparameter*, die für die gesamte Zeitreihe geschätzt werden. Für Modelle mit und ohne Trendkomponente und für die additive und multiplikative Saisonverknüpfung kann Exponentielles Glätten eingesetzt werden.

Das Holt-Winters-Verfahren entspricht dem Exponentiellen Glätten mit drei Glättungsparametern. Es unterstellt eine additive bzw. gemischt multiplikativ-additive Verknüpfung der Komponenten, sodass die sechs Verknüpfungen in Tabelle 2.1 durch das Verfahren verarbeitet werden können. Die Prognosefunktion im additiven Fall lautet:

$$\hat{x}_{T,\tau} = \hat{a}_T + \hat{b}_T \cdot \tau + \hat{c}_{T+\tau-\lceil \tau/L \rceil \cdot L} \tag{2.3}$$

Dabei ist $\hat{x}_{T,\tau}$ der τ-Schritt-Prognosewert mit Kenntnis der Historie bis zum Gegenwartszeitpunkt $t = T$; $\hat{a}$, $\hat{b}$ und $\hat{c}$ sind die Koeffizienten des Verfahrens. Der Anteil $\hat{a}_T + \hat{b} \cdot \tau$ schätzt die Trendkomponente r_t und der Anteil $\hat{c}_{T+\tau-\lceil \tau/L \rceil \cdot L}$ die Saisonkomponente q_t. L ist die Saisonlänge. Der Index der Saison-Koeffizienten $\hat{c}$ zeigt an, dass alle Saisons der Prognose die gleichen Saison-Koeffizienten nutzen:

$$\begin{aligned} \hat{c}_{T+\tau-L} &= \hat{c}_{T+\tau-\lceil \tau/L \rceil \cdot L} \\ &= \hat{c}_{T+(\tau+L)-\lceil (\tau+L)/L \rceil \cdot L} \\ &= \hat{c}_{T+(\tau+2 \cdot L)-\lceil (\tau+2 \cdot L)/L \rceil \cdot L} \\ &= \dots \end{aligned}$$

Die Koeffizienten werden zu Beginn geeignet initialisiert und danach rekursiv durch Updategleichungen ermittelt, vgl. [Sch12b]. Tabelle 2.2 stellt sie für beide Verknüpfungsvarianten vor.

Die Glättungsparameter α, β, γ sind konstant und jeweils im Intervall $[0, 1]$ wählbar, vgl. [Win60]. Sie werden durch die Modellschätzung ermittelt. Zusammenfassend besteht das Prognosemodell im Fall des Holt-Winters-Verfahrens aus den in Tabelle 2.3 genannten Komponenten.

Tabelle 2.2: Rekursionsform des Holt-Winters-Verfahrens nach [Sch12b]

Additive Saisonkomponente	**Multiplikative Saisonkomponente**
$\hat{a}_t = \alpha \cdot (x_t - \hat{c}_{t-L}) + (1-\alpha) \cdot (\hat{a}_{t-1} + \hat{b}_{t-1})$ $\hat{b}_t = \beta \cdot (\hat{a}_t - \hat{a}_{t-1}) + (1-\beta) \cdot \hat{b}_{t-1}$ $\hat{c}_t = \gamma \cdot (x_t - \hat{a}_t) + (1-\gamma) \cdot \hat{c}_{t-L}$ $\hat{x}_{T,t} = \hat{a}_T + \hat{b}_T \cdot \tau + \hat{c}_{T+\tau-\lceil \tau/L \rceil \cdot L}$	$\hat{a}_t = \alpha \cdot \frac{x_t}{\hat{c}_{t-L}} + (1-\alpha) \cdot (\hat{a}_{t-1} + \hat{b}_{t-1})$ $\hat{b}_t = \beta \cdot (\hat{a}_t - \hat{a}_{t-1}) + (1-\beta) \cdot \hat{b}_{t-1}$ $\hat{c}_t = \gamma \cdot \frac{x_t}{\hat{a}_t} + (1-\gamma) \cdot \hat{c}_{t-L}$ $\hat{x}_{T,t} = (\hat{a}_T + \hat{b}_T \cdot \tau) \cdot \hat{c}_{T+\tau-\lceil \tau/L \rceil \cdot L}$

Zwei Vereinfachungen gehen aus dem Modell hervor: Setzt man in der multiplikativen Form $\hat{c} = 1$, so entsprechen die Updategleichungen der Methode von Holt für nichtsaisonale Zeitreihen, vgl. [Hol57]. Setzt man zusätzlich $\hat{b} = 0$, findet auch keine Trendkorrektur statt. Dies entspricht dem Exponentiellen Glätten erster Ordnung.

Tabelle 2.3: Einordnung des Holt-Winters-Verfahrens als Prognosemodell

Komponente	**Ausprägung**
Prognosemethode	Holt-Winters-Verfahren
Metaparameter	Trendkomponente vorhanden (ja/nein) Saisonkomponente vorhanden (ja/nein) Wenn Saisonkomponente: Saisonlänge L Wenn Saisonkomponente: additiv oder multiplikativ
Modellparameter	Glättungsparameter α, β, γ
Zustand	Koeffizienten $\hat{a}$, $\hat{b}$, $\hat{c}$

Zusammenfassend wird festgehalten, dass das Holt-Winters-Verfahren eine robuste, automatisierbare Prognosemethode ist, die auf dem Verfahren des Exponentiellen Glättens aufbaut. Dabei ist seine Genauigkeit vergleichbar mit Prognosemethoden, die auf komplexen, rechenaufwändigen Verfahren basieren, was in einer empirischen Studie untersucht wurde, vgl. [Arm01]. Ferner kommt die Prognosemethode mit einer geringen Anzahl von Parametern aus und hat einen geringen Speicherbedarf. Somit ist sie für die Zeitreihenprognose in relationalen Datenbanken geeignet.

2.3 DAS FLASH-FORWARD DATABASE SYSTEM

Im Rahmen des Projekts *Flash-Forward Query Framework* wurde der Prototyp *Flash-Forward Database System* (F²DB) entwickelt. Er stellt eine Lösung zur Integration von Prognosemethoden in ein Datenbankmanagementsystem (DBMS) dar und bietet eine Reihe von Vorteilen verglichen zu anderen Prognosesystemen, vgl. [FRL12]:

- Durch die Integration in ein DBMS ist der Export von Zeitreihen zu externen Systemen wie R oder MATLAB nicht mehr nötig, vgl. [Hyn14, MAT14].
- Prognosemodelle können wiederverwendet werden. Beispielsweise speichern Experten Prognosemodelle im System, sodass fachfremde Nutzer auf diese Informationen zugreifen können, ohne genauere Details der Modellkomponenten zu kennen.
- Zeitreihen mit Prognose können in anderen Anfragen eingesetzt werden, bspw. durch eine Verbundoperation.
- Die Trennung von physischem und konzeptionellem Schema ermöglicht die Optimierung hinsichtlich des Speicher- und Rechenzeitbedarfs.

Die nachfolgenden Unterabschnitte erläutern den Aufbau und Komponenten des Prototyps.

2.3.1 Aufbau des Prototyps F²DB

Der gegenwärtige Aufbau des Prototyps F²DB ist in Abbildung 2.2 dargestellt. Sie zeigt einen Bestand an *Basistabellen*, auf den alle *Nutzer* des Systems zugreifen können. *Zeitreihen* gehen durch eine Zeit- und eine Messwertspalte aus einer Basistabelle hervor. Zusätzlich können sie durch *Kategorienattribute* spezifiziert werden, welche die *Selektionsbedingungen* bilden.

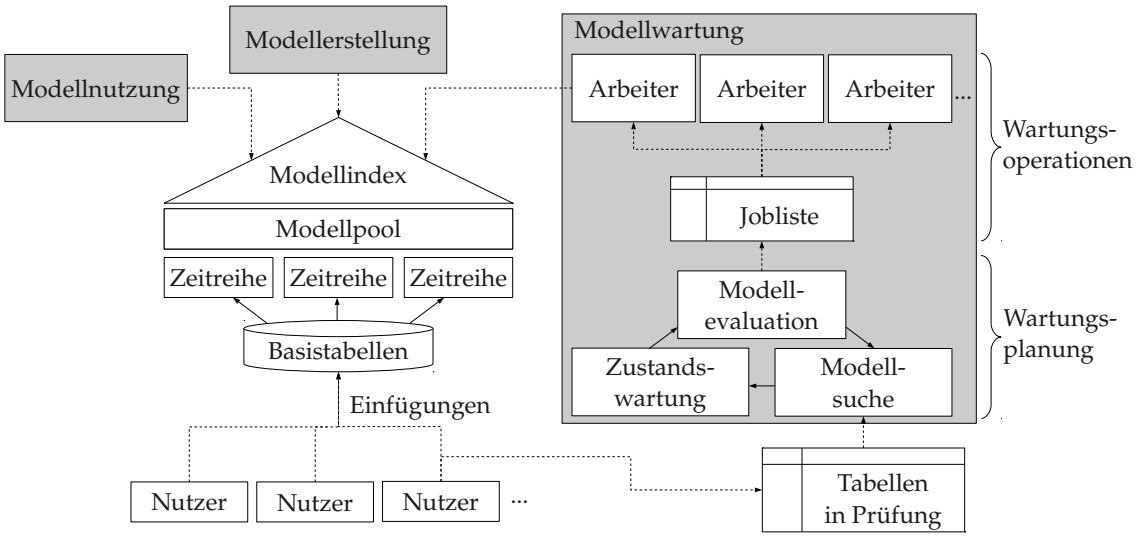


Abbildung 2.2: Aufbau des Prototyps F²DB

Für eine Zeitreihe wird ein Prognosemodell erstellt, das in einem Speicherbereich, dem *Modellpool*, abgelegt wird. Bei einer Anfrage muss es effizient wiedergefunden werden. Hierfür entwickelten Fischer et al. einen logischen Entscheidungsbaum, den *Modellindex*, vgl. [FRBL10].

Der Modellpool und Modellindex sind Strukturen, auf die alle Nutzer der Datenbank gemeinsam zugreifen, vgl. [Keg14]. Ein Nutzer kann das Prognosemodell erstellen und andere können es für ihre Prognoseanfragen verwenden.

2.3.2 Lebenszyklus von Prognosemodellen

In Analogie zum Lebenszyklus von materialisierten Sichten wird der *Lebenszyklus* von Prognosemodellen in die Prozesse Modellerstellung, -nutzung und -wartung eingeteilt, s. Abbildung 2.2.

Zur *Modellerstellung* gehört die Anforderung eines Nutzers, ein Prognosemodell für eine gegebene Zeitreihe zu erstellen. Für diesen Prozess wird die Angabe der Zeitreihe mit Zeit- und Messwertspalte benötigt; optional kann der Nutzer die Prognosemethode, die Meta- und Modellparameter vorgeben. Anschließend berechnet das Prognosesystem das Prognosemodell und legt es im Modellpool ab. Der Modellindex wird um einen Eintrag ergänzt.

Die *Modellnutzung* ermöglicht den Zugriff auf ein Prognosemodell bei einer *Prognoseanfrage*. Dafür wird der Modellpool mithilfe des Modellindexes durchsucht.

Die *Modellwartung* dient der Anpassung der Prognosemodelle, wenn sich die Zeitreihen fortschreiben. Die Komponenten Zustand, Modellparameter und Metaparameter sind wartbar. Entsprechende Wartungsoperationen werden asynchron ausgeführt, sodass die Einfügung von Tupeln und die Nutzeranfragen möglichst unverzögert durchgeführt werden. Die Zustandwartung ist schnell und findet unmittelbar nach Zuordnung eines Tupels zu einem Prognosemodell statt. Die Modellparameter- und Metaparameterwartung sind zeitaufwändig und werden daher in gesonderten Prozessen abgearbeitet. Die einzelnen Arbeitsschritte, s. Abbildung 2.2, führen folgendes durch:

- Die Pufferung von eingefügten Tupeln in den *Tabellen in Prüfung* dient der Entkoppelung der Modellwartung von der Einfügung der Tupel. Die Modellwartung wird regelmäßig aktiv, um die Prognosemodelle mit neuen Tupeln fortzuschreiben.
- Die *Wartungsplanung* führt die Aktualisierung der Prognosemodelle durch und entscheidet über den Einsatz von Wartungsoperationen. Sie besteht aus:
  - der *Modellsuche* zur Rückgabe von Prognosemodellen, die von der Einfügung eines Tupels betroffen sind,
  - der *Zustandswartung* betroffener Prognosemodelle und
  - der *Modellevaluation* zur Erstellung weiterer *Wartungsoperationen*.
- Ein *Job* ist ein Prognosemodell mit angeforderter Wartungsoperation, der in eine *Jobliste* eingefügt wird.
- *Arbeiter* nehmen Jobs entgegen und führen sie asynchron aus.

Dadurch ist es möglich, die zeitaufwändigen Wartungsoperationen von der Einfügung sowie der Prognoseanfrage zu entkoppeln und sie im dazwischenliegenden Zeitraum abzuarbeiten, sodass die Befehle möglichst unverzögert ausgeführt werden. Dies entspricht dem Konzept von Lazy Maintenance, wonach das Prognosemodell bei freier Rechenkapazität möglichst vor einer Anfrage gewartet wurde, vgl. [ZLE07]. Die Anzahl der Arbeiter kann auf die Anzahl der Prozessoren des Systems abgestimmt werden, wodurch die Skalierbarkeit der Komponente ermöglicht wird.

2.3.3 Aufbau des Modellindexes

Der Modellindex dient der Zuordnung von Prognoseanfrage zu Prognosemodell. Dies geschieht durch Prüfung der Selektionsbedingungen einer Anfrage. Eine beispielhafte Indexstruktur ohne Verbund ist in Abbildung 2.3 dargestellt: Hier wurde das Prognosemodell S für den Verkauf von Audio-Produkten in Deutschland erstellt und indexiert. Der Modellindex besteht aus *Tabellen-*, *Spalten-* und *Prädikatsknoten*. Für eine Prognoseanfrage an S wird geprüft, ob der Tabellenknoten für die angefragte Tabelle existiert. Dies ist im Beispiel der Fall. Vom Tabellenknoten ausgehend sind zwei Prädikate, $Land =$ „Deutschland“ und $Produktfamilie =$ „Audio“, gesucht, die durch die Spalten- und Prädikatsknoten erfasst wurden. *Konnektoren* verbinden Prädikatsknoten konjunktiv (AND) oder disjunktiv (OR). Da die Prognoseanfrage zwei Bedingungen stellt, muss das Prognosemodell am Konnektor der Prädikatsknoten gesucht werden. Dort befindet sich das angefragte Prognosemodell S. Zuletzt ist noch zuzusichern, dass die angefragte Zeit- und Messwertspalte auf das Prognosemodell zutreffen (nicht dargestellt).

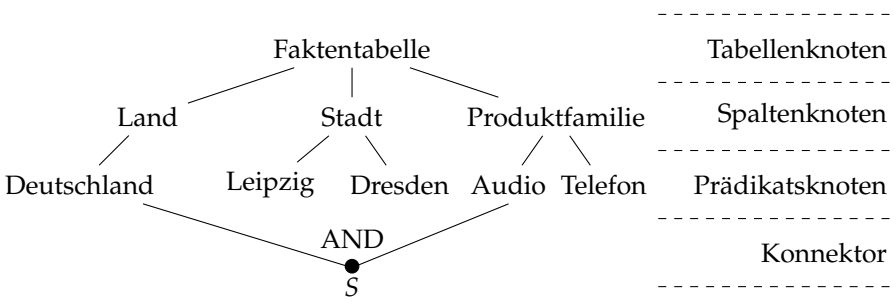


Abbildung 2.3: Modellindex mit Prognosemodell nach [Fis14]

Neben Prädikatsknoten existieren auch *Verbundknoten*, die eine Prognoseanfrage mit Verbund mehrerer Tabellen repräsentieren. Das derzeitige Prognosesystem unterstützt Anfragen für Sternschemata, eine Trennung in *Fakten-* und *Dimensionstabellen* ist somit möglich.

Knoten, an denen sich Prognosemodelle befinden (Tabellen-, Prädikats-, Verbundknoten sowie Konnektoren) werden als *Indexknoten* bezeichnet. An Spaltenknoten befinden sich Modelle nicht direkt, sondern stets am zugehörigen Prädikatsknoten oder Konnektor.

2.3.4 Aufbau des Modellknotens

Prognosemodelle werden als *Modellknoten* in den Modellindex eingefügt. Diese Knoten fügen dem Modell weitere Informationen hinzu, die für den Einsatz des Modells wichtig sind. Im Folgenden werden sie kurz zusammengefasst:

- Attribute für die Zeitreihe: Die *Zeitspalte* und *Messwertspalte* bestimmen, aus welchen Spalten der Tabelle die Zeitreihe x_t zusammengesetzt ist. Die *Granularität* ist der Abstand zwischen zwei Messwerten. Der *Zeitstempel* bestimmt den Gegenwartszeitpunkt $t = T$. Der *Aggregationstyp* signalisiert, ob für das Prognosemodell Messwerte nach Zeitpunkten gruppiert werden oder ob lediglich ein Messwert pro Zeitpunkt existiert. Die *Messwertsumme* ist dabei die Summe der Messwerte des gegenwärtigen Zeitpunkts.
- Attribute für die Prognose: Der Modellknoten verknüpft zum *Prognosemodell*, das die eigentliche Prognose ermöglicht. Zusätzlich kennt es den aktuellen *Prognosewert* $\hat{x}_{T,1}$.
- Attribute für die Wartung: Der *Modellfehler* enthält den zeitbasierten Fehler und den Prognosefehler. Bei Fortschreiben der Zeitreihe wird der zeitbasierte Fehler inkrementiert und der Prognosefehler um den Fehler vom Prognosewert $\hat{x}_{T,1}$ und Messwert x_{T+1} aktualisiert. Der Prognosefehler wird als symmetrischer mittlerer absoluter prozentualer Fehler (SMAPE) dargestellt; er ist somit zeitreihenunabhängig und liegt im begrenzten Intervall [0 %, 200 %], vgl. [CY04]:

$$SMAPE = 100 \cdot \frac{1}{J} \sum_{j=1}^{J} \frac{2 \cdot |x_{T+j} - \hat{x}_{T+j-1,1}|}{|x_{T+j}| + |\hat{x}_{T+j-1,1}|} \tag{2.4}$$

  Dabei bezeichne J die Anzahl fortgeschriebener Zeitpunkte. Bei Überschreiten der Schwellwerte wird die Modellparameter- und Metaparameterwartung veranlasst. Der *Wartungszustand* unterscheidet, ob sich das Prognosemodell in Wartung befindet oder aktiv ist. Bei einer Prognoseanfrage muss die Wartungsoperation eines Prognosemodells erst beendet werden.

2.4 ZIELSTELLUNG

Die vorgestellten Wartungsoperationen für den Zustand, die Modellparameter und Metaparameter bieten eine Verbesserung der Prognose an, wie in der vorangegangenen Arbeit gezeigt werden konnte, vgl. [Keg14]. Im Rahmen mehrerer Untersuchungen wurde gezeigt, dass die Erstellung eines Prognosemodells pro Zeitreihe in einem hierarchischen System wie z. B. Produktfamilien nicht notwendig ist. Durch Ableitung von Prognosemodellen ist sowohl eine Einsparung von Modellen und damit verbundenem Wartungsaufwand als auch eine Verbesserung der Prognosegenauigkeit möglich.

Erstes Ziel dieser Arbeit ist die Untersuchung, wie Ableitungen in das existierende System integriert werden können; dazu zählen:

- die Formalisierung von *Ableitungsmodellen*, die Prognosemodelle für ihre Prognose nutzen,
- die Umsetzung von Modellerstellung und -nutzung für Ableitungsmodelle und die damit verbundene Integration von Ableitungsmodellen in den Modellindex und Modellpool,
- die Wartung der Zuteilung eines Ableitungsmodells vom Prognosemodell, dem *Ableitungsgewicht*,
- die Umsetzung einer Wartungsoperation zur automatisierten Erstellung von Ableitungsmodellen, der Wartung von *Ableitungsregeln*, und
- das Konzept zur Entfernung ersetzter Prognosemodelle im Modellindex.

Anschließend ist in dieser Arbeit der Einfluss von Prognoseanfragen auf die Systemlast zu untersuchen. Nutzer haben bei einer Prognoseanfrage unterschiedliche Anforderungen: Einige erwarten eine hohe Prognosegenauigkeit und nehmen dafür eine Verzögerung der Anfrage in Kauf. Anderen Nutzern ist eine unverzögerte Anfrage wichtig, dafür müssen sie mit einem höheren Prognosefehler rechnen. Als zweites Ziel ist daher zu untersuchen, wie nutzerdefinierte Genauigkeitsklassen zu gestalten sind, mit denen der Nutzer die anstehenden Wartungsoperationen beeinflussen kann. Vor allem die Auswirkung der Genauigkeitsklassen auf die Prognosegenauigkeit und auf die Anfrageverzögerung stehen hier im Zentrum.

Eine hohe Systemlast kann zu einer bedeutenden Anfrageverzögerung führen, da angefragte Prognosemodelle noch in Wartung sind. Als drittes Ziel ist ein Konzept für die Wartungskomponente zu entwickeln, das in Verbindung mit den Genauigkeitsanforderungen des Nutzers eine Priorisierung der zu wartenden Prognosemodelle ermöglicht. Dazu zählen:

- die Entwicklung einer Laststeuerung, die bei hoher Systemlast das Verhalten der Wartungskomponente beeinflusst, sodass Prognoseanfragen mit geringer Verzögerung bedient werden können, und
- Überlegungen zur Reduzierung der Last, die die Wartungsoperationen verursachen.

Die folgenden drei Kapitel stellen Konzepte zu den gestellten Zielen vor und bieten eine Implementierung nebst Evaluation an, die in F²DB umgesetzt wird.

3 INTEGRATION VON ABLEITUNGSMODELLEN

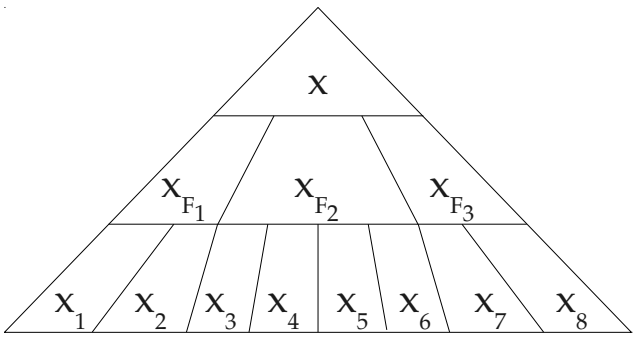
Das folgende Kapitel stellt die Integration von Ableitungsmodellen in F²DB vor. Abschnitt 3.1 präsentiert die hierarchische Prognose, die zur Konzeption von Ableitungsmodellen im eindimensionalen Fall genutzt wird. Abschnitt 3.2 stellt die multidimensionale Modellierung aus der Literatur vor, um die hierarchische Prognose in den multidimensionalen Raum zu übertragen. Abschnitt 3.3 beschreibt das multidimensionale Modell des Datenwürfels ausführlich und definiert damit Ableitungsmodelle im multidimensionalen Raum. Abschnitt 3.4 begründet Erweiterungen für die Integration von Ableitungsmodellen im Modellindex. In den Abschnitten 3.5 bzw. 3.6 wird der Lebenszyklus von Ableitungsmodellen, d. h. Modellerstellung, -nutzung bzw. Modellwartung beschrieben. Abschnitt 3.7 präsentiert Details für die Implementierung. Anhand reeller Datensätze wird in Abschnitt 3.8 die Implementierung evaluiert. Das Kapitel schließt mit einer Zusammenfassung in Abschnitt 3.9 ab.

3.1 HIERARCHISCHE PROGNOSE

Datensätze bestehen oft aus vielen Zeitreihen, die durch ein bestimmtes Merkmal eine Hierarchie bilden. So lassen sich beispielsweise Produkte aus einem Lager in Produktfamilien und höhere Aggregate hierarchisieren. Dabei kann das Prognosemodell für eine Produktfamilie auch gute Prognosen für die einzelnen Produkte der Familien liefern oder für das Aggregat aller Produkte. Die *hierarchische Prognose* ist das Erstellen einer Prognose mit den Prognosemodellen über- oder untergeordneter Zeitreihen. Der folgende Abschnitt erläutert diese Prognosetechnik.

Zeitreihen werden neben Zeit- und Messwertspalte durch Kategorienattribute definiert. Kategorienattribute können andere Kategorienattribute funktional bestimmen. Diese Eigenschaft ist wie folgt definiert, vgl. [Leh03]:

Definition 1 (Funktionale Abhängigkeit)**.** *Eine funktionale Abhängigkeit ($A \rightarrow B$) zwischen zwei Attributen A und B existiert genau dann, wenn für jedes $a \in A$ genau ein $b \in B$ existiert.*



Nachfrage nach Aggregat:
$x = \sum_{i=1}^{m} x_{F_i}$

Nachfrage nach Produktfamilie F_i:
$x_{F_i} = \sum_{j \in F_i} x_j, 1 \leq i \leq m$

Nachfrage nach Produkt:
$x_j, 1 \leq j \leq n$

Abbildung 3.1: Hierarchische Struktur einer Produktlinie nach [Fli01]

Durch Transitivität kann eine *Hierarchie* gebildet werden, d. h. eine total geordnete Menge von Kategorienattributen, die sich funktional bestimmen. Dadurch ist sichergestellt, dass Werte eines Attributs eindeutig den Wert des übergeordneten Attributs bestimmen. Im Folgenden werde angenommen, dass die funktionale Abhängigkeit auf alle Kategorienattribute zutrifft. Der Zusammenhang wird für die hierarische Prognose genutzt, vgl. [Fli01], was am Beispiel 1 erläutert wird.

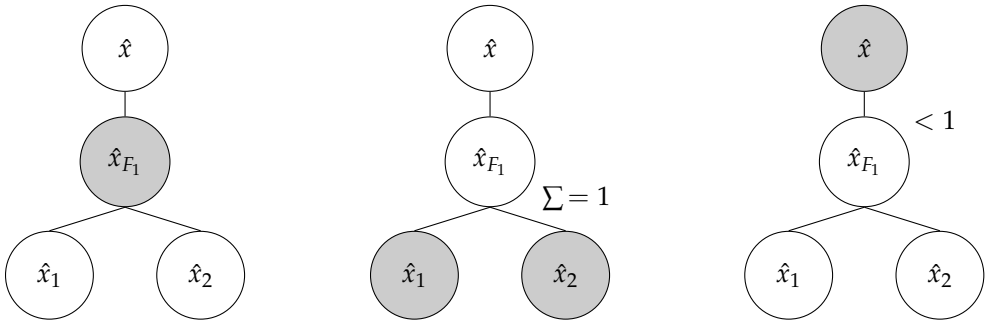
Beispiel 1. *In Abbildung 3.1 ist die Produkthierarchie einer Firma dargestellt: Produkte mit einer Produktnummer werden einer Produktfamilie mit Familiennummer zugeordnet. Alle Produktfamilien der Firma lassen sich als firmenweites Aggregat zusammenfassen. Das Kategorienattribut Produktnummer bestimmt das Kategorienattribut Familiennummer funktional. Alle Familien sind Teil des Aggregats. Jedes Produkt hat eine Nachfrage, die als Zeitreihe erfasst wird.*

Für jeden Zeitpunkt t und jedes Produkt j sei die Nachfrage $x_{j,t}$ gegeben. Wegen der funktionalen Abhängigkeit lässt sich eine Produktfamilie als Zusammenfassung ihrer Produkte ausdrücken. Für die Familie $F_1 = \{1, 2\}$ ergibt sich die Nachfrage $x_{F_1,t}$ durch die Summe aller Produktzeitreihen aus F_1. Die Nachfrage für das firmenweite Aggregat x_t ergibt sich durch die Summe der Nachfragen der m Produktfamilien $x_{F_i,t}$, $1 \leq i \leq m$.

Für die hierarchische Prognose wird dieser Zusammenhang auf Prognosemodelle übertragen: Für die Bestimmung der Prognose werden über- oder untergeordnete Prognosemodelle genutzt. Eine Produktfamilie nutzt bspw. die Prognosen ihrer Produkte oder die Prognose des Aggregats. In diesem Fall wird kein eigenes Prognosemodell, sondern ein *Ableitungsmodell* erstellt. Einem Ableitungsmodell ist eine Menge von Prognosemodellen zugeordnet, die es für die Bestimmung der Prognose nutzt. Es ist folglich eine Referenz auf andere Modelle.

Abbildung 3.2 stellt zwei Ableitungsregeln aus [Fis14] vor. Es war bisher üblich, für jede Zeitreihe ein eigenes, sog. *direktes Prognosemodell* zu erstellen. Abbildung 3.2a zeigt es für Produktfamilie F_1: Sie hat für ihre Zeitreihe ein direktes Prognosemodell (grau hinterlegt).

Eine mögliche Ableitungsregel ist die *Aggregation*, bei welcher untergeordnete Prognosemodelle für die Prognose genutzt werden. So kann die Prognose der Produktfamilie F_1 aus der Prognose aller Produkte der Familie abgeleitet werden, s. Abbildung 3.2b.



(a) Direkt **(b)** Aggregation **(c)** Disaggregation

Abbildung 3.2: Ableitungsregeln für die Prognose von Produktfamilie F_1 nach [Fis14]

Eine weitere Ableitungsregel stellt die *Disaggregation* dar: Hier ergibt sich die Prognose aus der Zuteilung von der übergeordneten Prognose. Die Zuteilung wird von einem *Disaggregationsschlüssel* bestimmt, der im Folgenden näher erläutert wird. Der Prognosewert der Produktfamilie F_1 entspricht einem Anteil des Prognosewerts vom Aggregat, s. Abbildung 3.2c.

Ableitungsmodelle stellen eine Alternative zu direkten Prognosemodellen dar. Die Motivation für hierarchische Prognose fußt auf zwei Beobachtungen: Fliedner [Fli01] führt aus, dass direkte Prognosemodelle nicht generell bessere Prognosen geben als Ableitungsmodelle. Der Prognosefehler kann bei bestimmten Zeitreihen geringer ausfallen, da die Ableitung einen stabilisierenden Einfluss auf die Prognose hat.

Darüber hinaus verursacht die Modellerstellung und -nutzung von Ableitungsmodellen weniger Kosten als direkte Prognosemodelle, vgl. Fischer et al. [FBL11]. In einem System mit sehr vielen Prognosemodellen ist dies entscheidend, denn es senkt den Wartungsaufwand und erhöht die Performance.

Aggregationsmodell

Wenn für die Prognose einer Zeitreihe eine Ableitungsregel zum Einsatz kommt, so muss ein Ableitungsmodell als Referenz auf die genutzten Prognosemodelle abgelegt werden. Für die Aggregation sei das *Aggregationsmodell* wie folgt definiert:

Definition 2 (Aggregationsmodell)**.** *Ein Aggregationsmodell ist ein Ableitungsmodell, dessen Prognose sich durch Aggregation mehrerer untergeordneter direkter Prognosemodelle oder Aggregationsmodelle ergibt.*

Sei $\hat{x}_{j,T,\tau}$ der Prognosewert für den Zeitpunkt $(T + \tau)$ einer Produktzeitreihe x_j. Der aggregierte Prognosewert $\hat{x}_{F_i,T,\tau}$ für die Produktfamilie i ergibt sich durch

$$\hat{x}_{F_i,T,\tau} = \sum_{j \in F_i} \hat{x}_{j,T,\tau} \tag{3.1}$$

Die Aggregationsstrategie ist rekursiv, vgl. [FBL11]: Untergeordnete Modelle können selbst Aggregationsmodelle sein. Die direkten Prognosemodelle und Aggregationsmodelle, die für die Ableitung genutzt werden, heißen *Quellmodelle*.

Disaggregationsmodell

Wie in Abbildung 3.2c dargestellt, kann eine Zeitreihe ein übergeordnetes Prognosemodell für die eigene Prognose nutzen. Die Referenz auf das Modell heißt *Disaggregationsmodell*:

Definition 3 (Disaggregationsmodell)**.** *Ein Disaggregationsmodell ist ein Ableitungsmodell, dessen Prognose sich durch Zuteilung von einem übergeordneten direktem Prognosemodell bestimmt. Die Zuteilung erfolgt über den Disaggregationsschlüssel.*

Sei $\hat{x}_{T,\tau}$ der Prognosewert für das Aggregat zum Zeitpunkt $(T + \tau)$. Der Prognosewert der Produktfamilie F_i ist durch

$$\hat{x}_{F_i,T,\tau} = \hat{x}_{T,\tau} \cdot P_{F_i,T} \tag{3.2}$$

gegeben, wobei $P_{F_i,T}$ der Disaggregationsschlüssel ist. Verschiedene Rechenvorschriften wurden für die Zuteilung untersucht, vgl. [GS90]. Demnach ist ein einfaches arithmetisches Mittel, das die Historie der Zeitreihen berücksichtigt, für den Disaggregationsschlüssel geeignet:

$$P_{F_i,T} = \frac{\sum_{t=1}^{T} x_{F_i,t}}{\sum_{t=1}^{T} x_t} \tag{3.3}$$

Die Summe von Messwerten einer Zeitreihe wird in der Folge als *Reihensumme* bezeichnet. Im Zusammenhang mit dem Disaggregationsmodell heißt das referenzierte direkte Prognosemodell Quellmodell.

Zusammenfassung

Die hierarchische Prognose ermöglicht zwei Ableitungsregeln für Prognosemodelle: die Aggregation und die Disaggregation. Anstelle direkter Prognosemodelle können Zeitreihen mit Referenzen auf direkte Prognosemodelle, den Ableitungsmodellen, Prognosen erstellen. Ableitungsmodelle können unter Umständen bessere Prognosen geben als direkte Prognosemodelle. Zudem lassen sich Prognosemodelle einsparen.

Die hierarchische Prognose, wie sie in [Fli01] vorgestellt wird, beschränkt sich auf eine Dimension. In der Praxis existieren jedoch Datensätze mit mehreren Dimensionen, weshalb im folgenden Abschnitt die Ableitungsregeln im multidimensionalen Raum formuliert werden.

3.2 MULTIDIMENSIONALE MODELLIERUNG

Die hierarchische Prognose berücksichtigt in der vorgestellten Form lediglich eine Dimension. Die Kategorienattribute von Zeitreihen im Date-Warehouse-System können jedoch auch mehrere Dimensionen bilden, sodass Aggregations- und Disaggregationsmodelle in jeder Dimension gefunden werden können.

Dieser Abschnitt stellt die *multidimensionale Modellierung* zur Übertragung der hierarchischen Prognose in mehrere Dimensionen vor. Unter einer *Dimension* wird, analog zur Hierarchie, eine total

geordnete Menge von Kategorienattributen verstanden, deren Relation die funktionale Abhängigkeit ist. Zusätzlich gilt, dass Dimensionen orthogonal sind, d. h. Kategorienattribute zu höchstens einer Dimension gehören. Ein Kategorienattribut bildet folglich das *Dimensionslevel* einer Dimension, was an Beispiel 2 erläutert wird.

Beispiel 2. *Der Datensatz **Tourismus** hat drei Kategorienattribute, die in Abbildung 3.3 dargestellt sind. Der Datensatz umfasst Quartalswerte zu Übernachtungen von Australiern im eigenen Land. Für die Jahre 1999 bis 2014 wurden die Messungen erstellt und nach Anlass (Freizeit, Beruf, u. a.) und nach Ziel der Reise nach Region (Sydney, Blue Mountains, Melbourne, Ballarat, u. a.) und Bundesstaat (New South Wales, Victoria, u. a.) gegliedert. Dadurch entstehen zwei Dimensionen (Geographie, Anlass). Die Top-Attribute sind spezielle Attribute, die das Aggregat einer Dimension umfassen. Die Wertebereiche der Kategorienattribute sind noch größer, werden aber für die Illustrationen auf die oben genannten Werte eingeschränkt.*

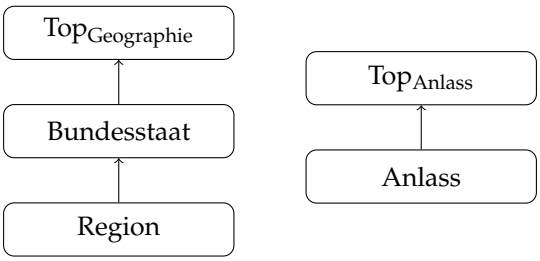


Abbildung 3.3: Dimensionen Geographie und Anlass vom Datensatz **Tourismus**

Data-Warehouse-Systeme gehen von einer multidimensionalen Datenmodellierung aus, vgl. Lehner [Leh03]. Es werden hierbei *qualifizierende Informationen* von *quantifizierenden Informationen* unterschieden. Erstere werden durch Kategorienattribute repräsentiert, die die Begriffe einer Dimension bilden und die in verschiedenen Granularitäten vorliegen, z. B. *Region* und *Bundesstaat*. Sie nehmen Werte an, z.B. $Bundesstaat =$ „Victoria“. Quantifizierbare Informationen sind die Zeitreihen selbst.

Verschiedene Ansätze werden für die multidimensionalen Modellierung vorgeschlagen. Hierzu zählen der *graphbasierte Modellierungsansatz*, s. Unterabschnitt 3.2.1, und der *multidimensionale Modellierungsansatz*, s. Unterabschnitt 3.2.2. Beide Ansätze ermöglichen die Bestimmung von Ableitungsmodellen. Es wird untersucht, welcher der Ansätze für die Integration in den Prototypen F^2DB geeignet ist.

3.2.1 Graphbasierter Modellierungsansatz

Im graphbasierten Modellierungsansatz ist ein azyklischer Objektgraph Grundlage für die Repräsentation von Kategorienattributen und den dazugehörigen Werten. Dieser Ansatz wird in [CS81] vorgestellt, um einen effizienten Zugriff auf statistische Daten, die in unterschiedlichen Kategorien klassifiziert wurden, zu ermöglichen. Insbesondere die Aggregation von statistischen Daten stand in dieser Arbeit im Fokus.

Der Objektgraph besteht aus zwei verschiedenen Knotentypen:

- Clusterknoten (C) repräsentieren Kategorienattribute. Ausgehende Kanten weisen auf die zugehörigen Werte. Zudem kann eine ausgehende Kante auf einen weiteren Clusterknoten weisen.
- Kreuzproduktknoten (X) stellen kein Kategorienattribut dar, sondern spannen auf Grundlage ihrer Kindknoten einen mehrdimensionalen Raum auf.

Abbildung 3.4 stellt den Objektgraphen für das Beispiel 2 dar. Die drei Kategorienattribute *Bundesstaat*, *Region* und *Anlass* liegen in ihren jeweiligen Dimensionen. Die zugehörigen Werte stehen an den untergeordneten Knoten oder an den Blättern. Eine Dimension wird durch eine total geordnete Menge aller Clusterknoten eines Zweigs des Kreuzproduktsknotens repräsentiert. Die Kante zwischen den Knoten verdeutlicht eine 1:N-Beziehung, und drückt die funktionale Abhängigkeit des übergeordneten vom untergeordneten Kategorienattribut aus, vgl. [Leh03]. Durch das Kreuzprodukt von Dimensionen wird ein multidimensionales Modell aufgespannt. Sie impliziert eine N:M-Beziehung der Attribute beider Dimensionen.

Die Anfrage an statistische Daten ergibt sich durch die Auswahl von Kategorienattributen und deren Werten. Wurde in einer Dimension kein Kategorienattribut gewählt, so werden alle Werte in dieser Dimension betrachtet, vgl. [CS81, S. 560]. Eine Anfrage nach Beispiel 2 ist die Zeitreihe von Übernachtungen für (*Bundesstaat* = „New South Wales“ x *Anlass* = „Freizeit“). Die dazugehörigen Werte sind unterstrichen und eine Anfrage kann an die Datenbank gerichtet werden, um das dazugehörige Prognosemodell zu suchen.

Durch den Objektgraphen sind außerdem Ableitungsmodelle ermittelbar. Sie werden durch Navigation entlang der Kanten bestimmt. Aggregationsmodelle ergeben sich aus den untergeordneten Kategorienattributen: Ein Kandidat wird durch die untergeordneten Zeitreihen (*Region* = „Sydney“ x *Anlass* = „Freizeit“) und (*Region* = „Blue Mountains“ x *Anlass* = „Freizeit“) definiert. Da das Attribut aus der Anlass-Dimension keine Kindknoten hat, ist kein Aggregationsmodell in dieser Dimension ermittelbar.

Ein Disaggregationsmodell ergibt sich aus der übergeordneteten Zeitreihe mit Bedingung

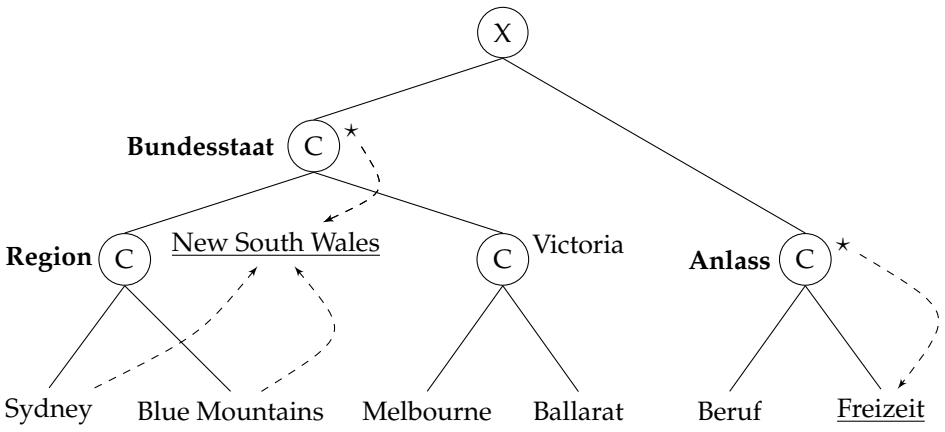


Abbildung 3.4: Graphbasiertes Schema für den Datensatz **Tourismus** nach [CS81]

($Anlass =$ „Freizeit“). Hier wird auf eine Einschränkung entlang der Geographie-Dimension verzichtet. Mit ($Bundesstaat =$ „New South Wales“) ist ein weiteres Disaggregationsmodell gegeben.

Damit ist der graphbasierte Modellierungsansatz für die Bestimmung von Ableitungsmodellen geeignet. Er ermöglicht, dass Kategorienattribute und Werte genau einmal abgelegt werden müssen. Der Modellindex, der Kategorienattribute als Spaltenknoten und Attributwerte als Prädikatsknoten abspeichert, lässt sich mit wenigen Eingriffen erweitern: Die funktionale Abhängigkeit zwischen den Prädikatsknoten muss hergestellt werden (unidirektional oder bidirektional) und die Hierarchie zwischen den Spaltenknoten (bidirektional). Das Quellmodell für die Disaggregation lässt sich durch den übergeordneten Spaltenknoten einer Dimension und dem dazugehörigen Prädikatsknoten bestimmen, der die funktionale Abhängigkeit erfüllt. Analog dazu lassen sich die Quellmodelle der Aggregation durch den untergeordneten Spaltenknoten bestimmen.

Von Nachteil an dem graphbasierten Modell ist, dass es keine Trennung zwischen Dimension, Dimensionslevel und Wert vornimmt, sondern alle drei Komponenten in einem Graph darstellt. Der Datenwürfel ermöglicht im Vergleich eine genauere Trennung und eine bessere Struktur für die Konzeption von Ableitungsmodellen.

3.2.2 Multidimensionaler Modellierungsansatz mit Datenwürfel

Der multidimensionale Raum kann als *Datenwürfel* repräsentiert werden, der aus einer Menge von orthogonalen Dimensionen $(D_1, \ldots, D_n)$ und einer Menge von *Messwerten* $(M_1, \ldots, M_m)$ besteht. Verglichen zum graphbasierten Modellierungsansatz trifft diese Repräsentation Einschränkungen der Modellierungsmächtigkeit zugunsten einer leichteren Navigation. Der graphbasierte Ansatz lässt Kreuzproduktknoten als innere Knoten zu und er erlaubt die gemeinsame Nutzung von Knoten (dargestellt durch mehrere Vaterknoten), vgl. [CS81, S.558]. Da beim Datenwürfel die Dimensionen orthogonal sind, existiert höchstens ein Kreuzproduktknoten an der Wurzel. Zudem gehören Dimensionslevels zu genau einer Dimension, sodass eine gemeinsame Nutzung von Knoten ausgeschlossen wird. Bei Abbildung 3.4 gelten diese Einschränkungen bereits.

Das unterste Level einer Dimension ist das *Primärattribut*, das alle Attribute in seiner Dimension bestimmt. Beispielsweise ist *Region* das Primärattribut der Geographie-Dimension aus Bsp. 2. Das oberste Dimensionslevel einer Dimension ist das generische Attribut *Top*, das von allen Kategorienattributen der Dimension bestimmt wird. Es umfasst das Aggregat für die gesamte Dimension. Im Bsp. 2 beschreibt $Top_{Geographie}$ alle Regionen und Bundesstaaten, vgl. [Leh03].

In der Literatur werden verschiedene Modelle für den Datenwürfel vorgestellt, vgl. Jäcksch und Lehner [JL13]. Hierzu zählt die Arbeit von Vassiliadis [Vas98], die einen Datenwürfel formalisiert und auf ihm Operationen wie die Navigation und die Auswahl eines Teilwürfels, dem *Dicing*, beschreibt.

Der Datenwürfel bietet viele Elemente, die auf die Konzeption von Ableitungsmodellen übertragen werden können:

- Die Selektionsbedingungen für eine Zeitreihe ergeben sich aus Kategorienattributen aller Dimensionen mit dazugehörenden Attributwerten, z. B. $Region =$ „Sydney“. Dabei ent-

spricht das Kategorienattribut dem Dimensionslevel und der Attributwert einem Wert für dieses Dimensionslevel.

- Dimensionslevels lassen sich über ihre funktionale Abhängigkeit hierarchisch anordnen und ergeben somit eine Dimension.

- Dimensionen stehen orthogonal aufeinander und spannen einen Datenwürfel auf.

- Ein *Basiswürfel*, d. h. eine Zelle des Datenwürfels, wird durch die Werte der Dimensionslevels aller Dimensionen bestimmt. Die quantitative Information des Basiswürfels ist eine Zeitreihe.

- Durch die funktionale Abhängigkeit der Dimensionslevels einer Dimension lassen sich über- und untergeordnete Levels finden, die zu Quellmodellen von Ableitungsmodellen führen.

- Die Modellierungsmächtigkeit vom Datenwürfel genügt zur Darstellung von Zeitreihen, weshalb die oben genannten Einschränkungen keinen Nachteil zum graphbasierten Modellierungsansatz darstellen. Zudem hat sich der Datenwürfel für die Modellierung in Data-Warehouse-Systemen etabliert.

Für das Zusammenführen von Basiswürfeln durch eine *Roll-Up-Operation* müssen die Bedingungen der *Summierbarkeit* erfüllt sein. Sie lauten Disjunktheit, Vollständigkeit und Typverträglichkeit, vgl. [LS97]. Auf die Konzeption für Ableitungsmodelle übertragen sind die drei Bedingungen erfüllbar. Die Disjunktheit ist erfüllt, wenn jeder Messwert x_t in einen Basiswürfel gehört und nicht in mehrere. Das wird aufgrund der Kategorienattribute vorausgesetzt, die jede Zeitreihe genau einem Basiswürfel zuordnet. Die Vollständigkeit ist gegeben, wenn die Zeitreihen verschiedener Basiswürfel die gleichen Zeitpunkte und die gleiche Granularität haben. Das muss durch den Datensatz gewährleistet sein, denn sonst lassen sie sich nicht aggregieren, wodurch keine Ableitungsmodelle definiert werden können. Die Typverträglichkeit ist zugesichert; diese Eigenschaft ist nur bei qualitativen Informationen mit temporalen Kategorienattributen problematisch. Da Zeitreihen nicht für temporale Kategorienattribute aggregiert werden, ist die Eigenschaft erfüllt.

Aufgrund der Trennung von Dimension, Dimensionslevel und Wert sowie der Verbreitung des Modellierungsansatzes wird der Datenwürfel für die Konzeption von Ableitungsmodellen in relationalen Datenbanken gewählt. Der folgende Abschnitt erläutert die Übertragung des Modells.

3.3 KONZEPTION VON ABLEITUNGSMODELLEN

Der folgende Abschnitt stellt die Übertragung der Datenwürfels auf die Konzeption von Ableitungsmodellen vor. Hierfür wird das Modell von Vassiliadis [Vas98] zusammengefasst, s. Unterabschnitt 3.3.1, und eine Analogie zwischen Würfeloperationen und der Suche von Ableitungsmodellen hergestellt, s. Unterabschnitt 3.3.2.

3.3.1 Modell des Datenwürfels

Vassiliadis stellt den Datenwürfel vor, in dem quantitative Informationen in Form von Basiswürfeln den qualitativen Informationen in Form von Dimensionslevels und deren Werten zugeordnet werden. Zudem ermöglicht sein Modell die Definition von Würfeloperationen, um Basiswürfel zu aggregieren. Seine Arbeit, vgl. [Vas98], wird in diesem Unterabschnitt zusammengefasst. Zudem werden Eigenschaften für die Konzeption von Ableitungsmodellen ergänzt.

Das Beispiel 2 illustriert den Datenwürfel, die Werte sind in Tabelle 3.1 gegeben. Die Dimensionen liegen im niedrigsten Dimensionslevel vor. Der Messwert ist eine Zeitreihe $x_t = \{x_1, x_2, \ldots, x_T\}$, wobei der Übersichtlichkeit halber die Zeitreihe die Länge $T = 2$ habe und nur aus ganzzahligen Werten bestehe.

Tabelle 3.1: Beispielwerte für den Datenwürfel

Geographie	Anlass	Messwert
Sydney	Freizeit	[1,3]
Sydney	Beruf	[2,8]
Blue Mountains	Freizeit	[2,4]
Blue Mountains	Beruf	[5,7]
Melbourne	Freizeit	[6,7]
Melbourne	Beruf	[6,8]
Ballarat	Freizeit	[3,4]
Ballarat	Beruf	[5,9]

Multidimensionaler Raum

Sei $\mathbf{\Omega}$ der Raum aller Dimensionen D.

Definition 4 (Dimension)**.** *Eine Dimension D hat eine Menge an Dimensionslevels $\boldsymbol{H}$, die eine Totalordnung $(\boldsymbol{H}, \rightarrow)$ bilden. Sie hat ein generisches maximales Level Top_D, sodass gilt: $\forall DL \in \boldsymbol{H} : DL \rightarrow Top_D$, vgl. [Leh03].*

Sei $\mathbf{\Psi}$ der Raum aller Dimensionslevels DL.

Definition 5 (Dimensionslevel)**.** *Ein Dimensionslevel DL ist ein Kategorienattribut mit Ausnahme von Top-Level Top_D. Für alle Dimensionslevels $DL \in \mathbf{\Psi}$ gilt, dass sie zu genau einer Dimension gehören. Es gibt keine funktionale Abhängigkeit zwischen Dimensionslevels verschiedener Dimensionen (Orthogonalität, vgl. [Leh03]).*

Im Beispiel 2 ist $\mathbf{\Omega} = \{D_{Geographie}, D_{Anlass}\}$ und $\mathbf{\Psi} = \{Region, Bundesstaat, Anlass, Top_{Geographie}, Top_{Anlass}\}$. Die Totalordnung $(DL_{Geographie}, \rightarrow)$ ist $Region \rightarrow Bundesstaat \rightarrow Top_{Geographie}$ und $(DL_{Anlass}, \rightarrow)$ besteht aus $Anlass \rightarrow Top_{Anlass}$, s. Abbildung 3.3.

Zur Vereinfachung der Operationen werden Hilfsfunktionen eingeführt:

- $levels(D)$: Die Funktion $levels(D)$ gibt die aufsteigend geordnete Liste aller Dimensionslevels von D an, z. B. für $levels(D_{Geographie}) = \{Region, Bundesstaat, Top_{Geographie}\}$.

- $dim(DL)$: Diese Funktion gibt die Dimension zurück, in der das Dimensionslevel DL liegt, z. B. $dim(Bundesstaat) = D_{Geographie}$.
- $rang(DL)$: Ermittlung des Rangs eines Dimensionslevels. Es gilt: $rang(DL) = k$, wenn $DL = levels(D)[k]$. Der kleinste Rang ist 1. Der Index k in eckigen Klammern bedeutet die Rückgabe des k-ten Elements der Liste $levels(D)$. Zum Beispiel ist $rang(Region) = 1$.

Sei $\boldsymbol{V}$ die Menge aller Werte v.

Definition 6 (Wert)**.** *Ein Wert v ist eine Belegung für ein Dimensionslevel DL. Jedem Dimensionslevel DL sind Werte zugeordnet, die aus der Domäne $dom(DL)$ stammen. Da die Top-Levels das Aggregat der Dimension beschreiben, ist ihre Domäne $dom(Top_D) = \star$. Der Stern $\star$ kennzeichnet, dass alle Belegungen zulässig sind.*

Somit entspricht der Wert eines Dimensionslevels dem Attributwert eines Kategorienattributs, mit Ausnahme von „$\star$“. Zum Beispiel gehören zur Domäne von *Region* „Sydney“, „Melbourne“ und „Ballarat“.

Ein Wert kann wegen der funktionalen Abhängigkeit höchstens einen Vorfahren, aber mehrere Nachfolger pro Dimensionslevel haben. Sei $v \in dom(DL)$, so wird wie folgt definiert:

- $vorfahre(v, DL')$: Die Funktion gibt den Vorfahren im Dimensionslevel DL' an: $vorfahre(v, DL') = w,\ w \in dom(DL'), DL \rightarrow DL'$.
- $nachfolger(v, DL')$: Die Funktion gibt die Nachfolger im Dimensionslevel DL' an: $nachfolger(v, DL') = \{w_1, w_2, \ldots, w_n\}, w_1, w_2, \ldots, w_n \in dom(DL'), DL' \rightarrow DL$.

Im Beispiel 2 gilt für $DL = Region$: $vorfahre(Sydney, Bundesstaat) =$ „New South Wales“. Für $DL = Top_{Geographie}$ gilt: $nachfolger(\star, Bundesstaat) = dom(Bundesstaat)$.

Zudem werden für das über- und untergeordnete Dimensionslevel folgende Abkürzungen eingeführt:

- *kind_level* gibt für ein Dimensionslevel DL das untergeordnete Dimensionslevel zurück: $kind\_level(DL) = levels(dim(DL))[rang(DL) - 1]$. Das unterste Dimensionslevel (Primärattribut) hat kein Kindlevel.
- *vater_level* gibt für ein Dimensionslevel DL das übergeordnete Dimensionslevel zurück: $vater\_level(DL) = levels(dim(DL))[rang(DL) + 1]$. Top-Levels haben kein Vaterlevel.

Datenwürfel

Mithilfe dieser Beschreibung qualitativer Informationen ist es möglich, einen Datenwürfel zu definieren. Die Definitionen 7 bis 9 sind der Arbeit von Vassiliadis unverändert entnommen. Als Notation führt er den Stern * zur Kennzeichnung einer Multimenge ein. Dies ist notwendig, da mehrere Messwerte im gleichen Würfel liegen können. Wenn $\boldsymbol{V}$ die Domäne eines Dimensionslevels ist, dann ist die Potenzmenge von $\boldsymbol{V}$ die Domäne einer Multimenge.

Definition 7 (Basiswürfel)**.** *Ein Basiswürfel C_b ist ein Tripel $< \underline{D_b}, \underline{DL_b}, \boldsymbol{R_b} >$:*

- $\underline{D_b} = [D_1, D_2, \ldots, D_n, M]$ *ist das Tupel aller Dimensionen des Basiswürfels ($D_i, M \in \boldsymbol{\Omega}$). M repräsentiert die Dimension des Messwerts, wie z. B. die Menge aller reellen Zahlen oder die Menge aller Zeitreihen.*
- $\underline{DL_b} = [DL_{b1}, DL_{b2}, \ldots, DL_{bn}, *ML]$ *ist das Tupel der zu $\underline{D_b}$ gehörenden Dimensionslevels ($DL_{bi}, *M \in \boldsymbol{\Psi}$). $*ML$ ist das Dimensionslevel des Messwerts. Basiswürfel existieren nur für Primärattribute, sodass gilt: $\forall DL_b \in \underline{DL_b} : rang(DL_b) = 1$.*
- $\boldsymbol{R_b}$ *ist die Menge der Tupel aus dem Würfel. Tupel sind von der Form $\underline{v} = [v_1, v_2, \ldots, v_n, *m]$. Es gilt: $\forall i (1 \leq i \leq n) : v_i \in dom(DL_{bi}) und * m \in dom(*ML)$.*

Ein Basiswürfel für das Beispiel 2 beinhaltet das Tupel [Sydney, Freizeit, [1,3]]. Es handelt sich stets um Werte der Primärattribute. Das Dimensionslevel $*ML$ ist abhängig von der Domäne des Messwerts. Eine Zeitreihe kann bspw. in verschiedenen Granularitäten vorliegen und in verschiedenen Dimensionslevels wie Tag, Monat, Jahr aufgelöst sein. Dies ist für die weitere Betrachtung jedoch nicht relevant, da angenommen wird, dass für alle Zeitreihen des Datenwürfels dieselbe unveränderliche Granularität gilt.

Definition 8 (Datenwürfel)**.** *Ein Datenwürfel C ist ein Quadrupel $< \underline{D}, \underline{DL}, C_b, \boldsymbol{R} >$:*

- $\underline{D} = [D_1, D_2, \ldots, D_n, M]$ *ist das Tupel aller Dimensionen ($D_i, M \in \boldsymbol{\Omega}$) des Datenwürfels. M repräsentiert die Dimension des Messwerts.*
- $\underline{DL} =< DL_1, DL_2, \ldots, DL_n, *ML >$ *ist das Tupel der zu $\underline{D}$ gehörenden Dimensionslevels ($DL_i, *M \in \boldsymbol{\Psi}$). $*ML$ ist das Dimensionslevel des Messwerts. Es gilt: $\forall DL_i \in \underline{DL} : DL_i \in levels(D_i)$.*
- C_b *ist der Basiswürfel, der für Berechnungen der Inhalte von C genutzt wird. Alle Dimensionen des Würfels müssen auch im Basiswürfel existieren.*
- $\boldsymbol{R}$ *ist die Menge der Tupel aus dem Würfel. Tupel sind von der Form $\underline{v} = [v_1, v_2, \ldots, v_n, *m]$. Es gilt: $\forall i (1 \leq i \leq n) : v_i \in dom(DL_i) und * m \in dom(*ML)$.*

Damit ist es möglich, die Aggregation von Zeitreihen für Dimensionslevels höheren Rangs anzugeben. Beispielsweise ist das Tupel [New South Wales, Freizeit, [3,7]] Element des Datenwürfels, dessen Dimensionslevels $D_{Geographie} = Bundesstaat$ und $D_{Anlass} = Anlass$ lauten, s. Abbildung 3.2.

Tabelle 3.2: Beispielwerte für Roll-Up-Operation entlang der Geographie-Dimension

Geographie	Anlass	Messwert
New South Wales	Freizeit	[3,7]
New South Wales	Beruf	[7,15]
Victoria	Freizeit	[9,11]
Victoria	Beruf	[11,17]

Ein Basiswürfel kann als Datenwürfel $C_b =< \underline{D_b}, \underline{DL_b}, C_b, \boldsymbol{R_b} >$ definiert werden, da er Würfel von sich selbst ist. Schließlich wird der multidimensionale Raum wie folgt beschrieben:

Definition 9 (Multidimensionaler Raum)**.** *Der multidimensionale Raum ist ein Paar $< \underline{D}, C >$: $\underline{D}$ ist eine Menge von n Dimensionen und C ein Basiswürfel, dessen Dimensionen zu $\underline{D}$ gehören.*

Der multidimensionale Raum ermöglicht die Ausführung einer Roll-Up-Operation, die Messwerte der Basiswürfel zu Messwerten höherer Dimensionslevels verdichtet. Ebenso ist es möglich, von einem höheren Dimensionslevel Teilwürfel zu ermitteln, da die Basiswürfel bekannt sind. Dies entspricht einer *Drill-Down-Operation*.

Wenn Selektionsbedingungen, d. h. Dimensionslevels mit zugehörigem Wert, angegeben sind, so kann ein bestimmter Messwert aus dem Datenwürfel ausgewählt werden:

Definition 10 (Messwert im multidimensionalen Raum)**.** *Der Messwert im multidimensionalen Raum $< \underline{D}, C >$ wird durch den Datenwürfel und durch Selektionsbedingungen bestimmt. Eine Selektionsbedingung besteht aus*

- *einem Dimensionslevel $DL \in \mathbf{\Psi}$ und*
- *einem Wert $v \in \boldsymbol{V}$ der Form „$DL = v$“.*

Liegt für eine Dimension D keine Selektionsbedingung vor, so ist in dieser Dimension jede Belegung möglich, d. h. $Top_D = \star$.

Abbildung 3.5 veranschaulicht den Datenwürfel für das Beispiel 2. Jeder Datenwürfel enthält eine Zeitreihe der Länge $T = 2$. Dargestellt sind die zwei Dimensionen mit ihren jeweiligen Dimensionslevels und Werten. Die grau hinterlegten Basiswürfel wurden durch die Selektionsbedingung ($Bundesstaat$ = „New South Wales“ x $Anlass$ = „Freizeit“) ausgewählt und ergeben einen Datenwürfel mit dem Messwert, d. h. der aggregierten Zeitreihe, [3, 7].

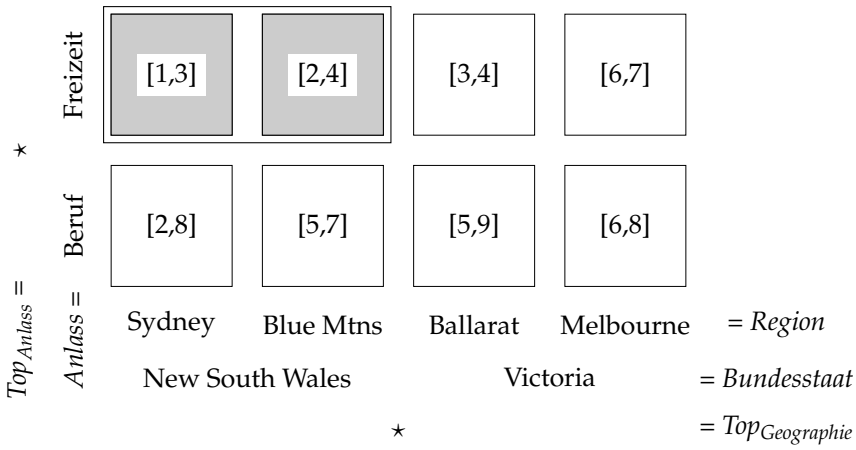


Abbildung 3.5: Messwert im multidimensionalen Raum

Vassiliadis schlägt zur Selektion des Messwerts mehrere Operationen auf dem Datenwürfel vor, die in Anhang C zusammengefasst sind. Da die Selektion im nachfolgenden Konzept von der Datenbank übernommen wird, ist die zugehörige Formalisierung nachrangig. Entscheidend ist, dass gemäß Definition 10 Messwerte im Datenwürfel auf verschiedenen Dimensionslevels ermittelt werden können.

Die hier vorgestellte Zusammenfassung weicht in einigen Punkten von der in [Vas98] vorgestellten Arbeit ab, was nachfolgend begründet wird:

- Das höchste Level jeder Dimension D ist Top_D, vgl. [Leh03]. Eine Zeitreihe hat für bestimmte Dimensionen ggf. keine Selektionsbedingung. Damit von diesen Dimensionen eine Aggregation des darunter liegenden Levels erfolgen kann, sind sie automatisch auf dem Level Top_D.
- Vassiliadis verwendet noch nicht den Begriff der funktionalen Abhängigkeit für die Ordnung der Dimensionslevels. Zudem geht er nicht auf die Orthogonalitätseigenschaft der Dimensionen ein. Dies wurde im obigen Modell ergänzt.
- Eingeschränkt wurde das Modell auf genau einen Dimensionspfad je Dimension, d. h. auf eine Totalordnung jeder Dimension. Mehrere Dimensionspfade entsprechen einer Erweiterung der Suche nach Ableitungsregeln. Für die Evaluation ist ein Ableitungsmodell je Ableitungsregel und Dimension genügend.
- Die Selektionsbedingungen lassen auch weitere Operatoren zu. Dazu gehören z. B. atomare der Form „$DL < v$“ und mehrwertige der Form „$DL \in \{v_1, v_2, \ldots, v_n\}$“. Die Wartungskomponente von F^{2}DB unterstützt diese Operatoren derzeit nicht, weshalb sie in der Folge nicht weiter betrachtet werden.

3.3.2 Übertragung auf Ableitungsmodelle

Der folgende Abschnitt definiert das Suchverfahren, mit dem Ableitungsmodelle im multidimensionalen Raum bestimmt werden. Dies ist ohne Eingriff des Nutzers möglich.

Für ein gegebenes Prognosemodell lassen sich die Zeitreihe und Kategorienattribute getrennt untersuchen, d. h. quantifizierbare getrennt von qualifizierbaren Informationen. Kategorienattribute sind qualifizierbar, weil sie lediglich eine Selektion der Zeitreihen vollziehen, die im Anschluss aggregiert werden. Die Beobachtung, dass Kategorienattribute den Dimensionslevels entsprechen, ermöglicht die Analogie mit dem Datenwürfel. Die Zeitreihen sind dabei Messwerte der Basiswürfel.

Die Dimensionen ergeben sich aus den Fremdschlüsselbeziehungen der Attribute in den Tabellen, vgl. [FBL11]. Wenn das Relationenschema unnormalisiert vorliegt und keine Fremdschlüsselbeziehungen existieren, muss das Schema der Dimensionen explizit angegeben werden. Dies ist z. B. in einer Universalrelation notwendig.

Für das Beispiel 2 zeigt Abbildung 3.6 eine Prognoseanfrage, mit der ein direktes Prognosemodell erstellt wurde. Nebenstehend sind Werte der Dimensionslevels zusammengefasst, die sich aus der Selektionsbedingung ergeben. Das Prognosemodell wurde im Modellpool abgelegt und indexiert. Es soll nun durch ein Ableitungsmodell ersetzt werden. Dafür wird zunächst die Analogie zum Datenwürfel hergestellt.

Sei $\boldsymbol{\Theta}$ die Menge aller Zeitreihen. Sei C ein Datenwürfel $C =< \underline{D}, \underline{DL}, C_b, \boldsymbol{R} >$ mit

$$\underline{D} = [D_{Geographie}, D_{Anlass}, M]$$

$$\underline{DL} = [Top_{Geographie}, Top_{Anlass}, *ML]$$

$$C_b = [\underline{D_b}, \underline{DL_b}, C_b, \boldsymbol{R_b}] \text{ mit}$$

$$\underline{D_b} = \underline{D}, \underline{DL_b} = [Region, Anlass, *ML],$$

$$\boldsymbol{R_b} = \{[v_{Region}, v_{Anlass}, *x] \mid v_{Region} \in dom(Region), v_{Anlass} \in dom(Anlass), *x \subset \boldsymbol{\Theta}\},$$

$$\boldsymbol{R} = \{[\star, \star, *x] \mid *x \subset \boldsymbol{\Theta}\}.$$

Die Position des Prognosemodells im Modellindex gibt die Auswahl eines Datenwürfels aus C vor, dessen Messwert die zum Prognosemodell gehörende Zeitreihe ist. Diese wird durch Roll-Up-Operation und anschließendes Dicing ermöglicht, wie im Anhang C erläutert ist. Im Prototypen F²DB realisiert eine Datenbankanfrage diese Operationen.

Somit ermöglicht das multidimensionale Modell von [Vas98] die Repräsentation von Zeitreihen in einem Datenwürfel. In den folgenden Unterabschnitten wird gezeigt, dass den Zeitreihen Prognosemodelle zugeordnet sind, die für die Erstellung eines Ableitungsmodells genutzt werden können.

```
1 CREATE MODEL S1
2 FOR FORECAST OF agg ON time
3 HWMODEL TRAINING_DATA (
4 SELECT zeit, SUM(messung) AS agg
5 FROM tourismus1
6 WHERE bundesstaat = 'New South Wales'
7 GROUP BY time
8 ORDER BY time);
```

Dimension	Dimensionslevel	Wert
$D_{Geographie}$	*Bundesstaat*	New South Wales
D_{Anlass}	*Anlass*	$\star$

Abbildung 3.6: Datenwürfel mit direktem Prognosemodell

Ermittlung von Aggregationsmodellen

Es sei ein direktes Prognosemodell S durch den Datenwürfel und eine Menge von Selektionsbedingungen gegeben. Es kann durch ein Aggregationsmodell ersetzt werden. Die Menge der Aggregationsmodelle werde mit $\boldsymbol{Agg}$ bezeichnet. Für alle Dimensionslevels, die unterhalb der Dimensionslevels von S liegen, kann ein Aggregationsmodell gesucht werden, wodurch die Größe von $\boldsymbol{Agg}$ bestimmt ist:

$$|\boldsymbol{Agg}| = \prod_{i=1}^{n} (rang(DL_i)) - 1 \tag{3.4}$$

Für Dimensionen mit vielen Dimensionslevels ist die Kandidatenanzahl sehr groß. Für die Konzeption der Wartungsoperation wird daher einschränkend angenommen:

1. Ein Aggregationsmodell nutzt nur Modelle, die sich auf dem Kindlevel befinden, d. h. genau ein Dimensionslevel unter dem zu ersetzenden Dimensionslevel liegen. Andere untergeordnete Dimensionslevels können durch Rekursion abgedeckt werden.
2. Es wird nicht in mehreren Dimensionen gleichzeitig das Kindlevel ermittelt, siehe Abbildung 3.7d. Diese Aggregation lässt sich auch durch Rekursion herstellen.

$\boldsymbol{Agg}$ beinhaltet somit höchstens ein Aggregationsmodell je Dimension:

$$|\boldsymbol{Agg}| = \sum_{i=1}^{n} (sgn(rang(DL_i) - 1)), \tag{3.5}$$

wobei $sgn()$ die Signumfunktion ist. Zur Verkürzung wird zunächst eine Hilfsfunktion eingeführt:

Definition 11 (Substitutionsfunktion)**.** *Es sei $\rho(\underline{DL}, i, DL')$ ist eine Substitutionsfunktion, die das Element einer Liste an einer gegebenen Position durch ein anderes Element ersetzt. Für eine Liste von Dimensionslevels $\underline{DL}$ wird das i-te Dimensionslevel DL durch DL' ersetzt:*

$$\rho(\underline{DL}, i, DL')[j] = \begin{cases} DL', & falls\ i = j \\ \underline{DL}[j], & sonst \end{cases} \tag{3.6}$$

In Anlehnung an Definition 2 (Seite 27) wird das Aggregationsmodell im multidimensionalen Raum wie folgt definiert:

Definition 12 (Aggregationsmodell, multidimensional)**.** *Sei $A_i \in \boldsymbol{Agg}$ das Aggregationsmodell, das Prognosemodelle in der Dimension D_i aggregiert. Es ersetzt das Prognosemodell S. A_i ist nicht definiert, wenn $rang(DL_i) = 1$. Ansonsten aggregiert A_i die direkten Prognosemodelle oder Aggregationsmodelle $\{S_1, \ldots, S_j, \ldots, S_m\}$. Für die Auswahl des Modells S_j gilt:*

Die Dimensionen sind die des zu ersetzenden Prognosemodells S:

$$\underline{D}' = \underline{D} \tag{3.7a}$$

Die Dimensionslevels sind die des zu ersetzenden Prognosemodells S mit Ausnahme von DL_i. Hierfür wird das Kindlevel eingesetzt:

$$\underline{DL}' = \rho(\underline{DL}, i, kind\_level(DL_i)) \tag{3.7b}$$

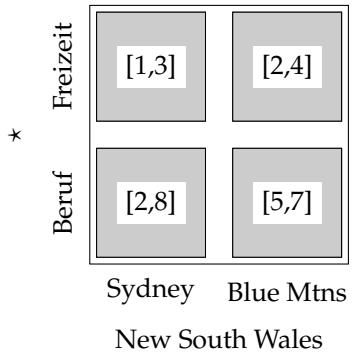
Die Selektionsbedingungen sind die des zu ersetzenden Prognosemodells S ausgenommen die Selektionsbedingung für Dimension D_i mit der Form „$DL_i = v$“. Sie wird ersetzt durch die Bedingung:

$$kind\_level(DL_i) = nachfolger(v, kind\_level(DL_i))[j] \tag{3.7c}$$

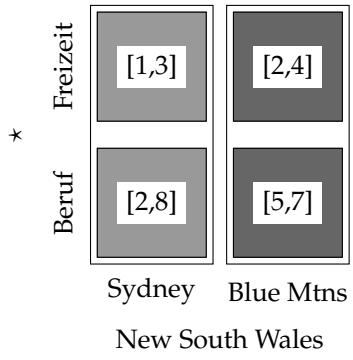
Für ein gegebenes $A_i \in \boldsymbol{Agg}$ folgt somit die Gleichung der Aggregation der Zeitreihen wie im eindimensionalen Fall, s. Gleichung 3.1 (Seite 27):

$$\hat{x}_{A_i,T,\tau} = \sum_{1 \leq j \leq m} \hat{x}_{S_j,T,\tau} \tag{3.8}$$

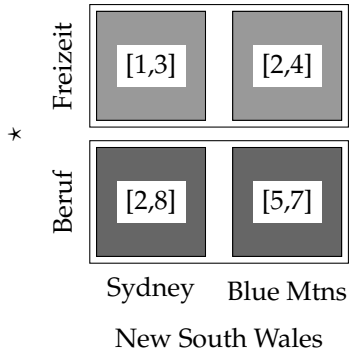
Für die Zeitreihe mit Selektionsbedingung $Bundesstaat$ = „New South Wales“ stellt Abbildung 3.7 die Aggregationsmodelle dar. Aggregationsmodell $A_{Geographie}$, s. Abbildung 3.7b, verdeutlicht die Aggregation entlang der Geographie-Dimension. Es werden die Prognosemodelle S_{Sydney} und $S_{Blue\ Mountains}$ aggregiert. Aggregationsmodell A_{Anlass}, s. Abbildung 3.7c, enspricht der Aggregation entlang der Anlass-Dimension, der die zwei Modelle $S_{Freizeit}$ und S_{Beruf} aggregiert. Beide Aggregationsmodelle können das Prognosemodell S ersetzen.



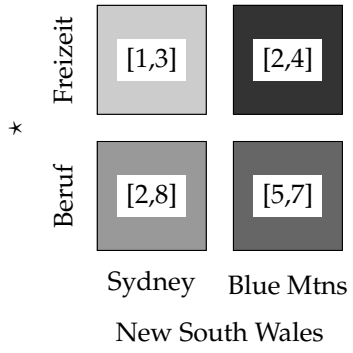
(a) Prognosemodell S: Ausgangssituation



(b) Aggregationsmodell $A_{Geographie}$: Aggregation entlang der Geographie-Dimension



(c) Aggregationsmodell A_{Anlass}: Aggregation entlang der Anlass-Dimension



(d) Aggregation über mehrere Dimensionen über Rekursion

Abbildung 3.7: Datenwürfel mit Aggregationsmodellen

Die Suche nach Quellmodellen für ein Aggregationsmodell entspricht einer Drill-Down-Operation im Datenwürfel: entlang einer gegebenen Dimension und unter Berücksichtigung der Selektionsbedingungen wird die Sicht auf ein detaillierteres Dimensionslevel heruntergebrochen, um die dortigen Zeitreihen und Prognosemodelle zu nutzen.

Ermittlung von Disaggregationskandidaten

Für das Disaggregationsmodell wird das direkte Prognosemodell des übergeordneten Dimensionslevels genutzt. Die Menge der Disaggregationsmodelle wird mit $\boldsymbol{Disagg}$ bezeichnet. Wenn alle übergeordneten Dimensionslevels ein Disaggregationsmodell bilden, ergibt sich folgende Kardinalität:

$$|\boldsymbol{Disagg}| = \prod_{i=1}^{n}(rang(Top_{D_i}) - rang(DL_i) + 1) - 1 \tag{3.9}$$

Dies führt zu einem hohen Rechenaufwand, wenn eine Dimension viele Dimensionslevels hat. Für die Konzeption wird daher einschränkend angenommen, dass für jede Dimension nur das jeweilige Vaterlevel, d. h. der unmittelbare Vorfahre, in Frage kommt. Dadurch existiert höchstens ein Disaggregationsmodell pro Dimension:

$$|\boldsymbol{Disagg}| = \sum_{i=1}^{n}(sgn(Top_{D_i} - rang(DL_i))) \tag{3.10}$$

In Anlehnung an Definition 3 (Seite 28) wird das Disaggregationsmodell im mehrdimensionalen Fall wie folgt definiert:

Definition 13 (Disaggregationsmodell, multidimensional)**.** *Sei $E_i \in \boldsymbol{Disagg}$ das Disaggregationsmodell, das in der Dimension D_i disaggregiert und das direkte Prognosemodell S ersetzt. Das Quellmodell ist das direkte Prognosemodell S_i. E_i ist nicht definiert, wenn $rang(DL_i) = Top_{D_i}$. Ansonsten gilt für die Auswahl von S_i:*

Die Dimensionen sind die des zu ersetzenden Prognosemodells S:

$$\underline{D}' = \underline{D} \tag{3.11a}$$

Die Dimensionslevels sind die des zu ersetzenden Prognosemodells S mit Ausnahme von DL_i. Hierfür wird das Vaterlevel eingesetzt:

$$\underline{DL}' = \rho(\underline{DL}, i, vater\_level(DL_i)) \tag{3.11b}$$

Die Selektionsbedingungen sind die des zu ersetzenden Prognosemodells S ausgenommen die Selektionsbedingung für Dimension D_i mit der Form „$DL_i = v$“. Sie wird ersetzt durch die Bedingung:

$$vater\_level(DL_i) = vorfahre(v, vater\_level(DL_i)) \tag{3.11c}$$

Ein Disaggregationskandidat $E_i \in \boldsymbol{Disagg}$ benötigt sowohl die Zeitreihe des zu ersetzenden Prognosemodells S als auch die Zeitreihe des Quellmodells S_i. Die Berechnung des Prognosewerts erfolgt analog zum eindimensionalen Fall, s. Gleichung 3.2 (Seite 28):

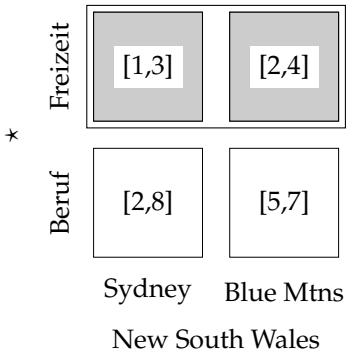
$$\hat{x}_{E_i,T,\tau} = \hat{x}_{S_i,T,\tau} \cdot P_{E_i,T} \tag{3.12}$$

$$P_{E_i,T} = \frac{\sum_{t=1}^{T} x_{S,t}}{\sum_{t=1}^{T} x_{S_i,t}} \tag{3.13}$$

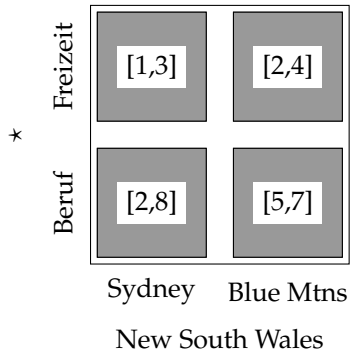
Abbildung 3.8 illustriert die Ermittlung von zwei Disaggregationsmodellen. Das zu ersetzende Prognosemodell S, s. Abbildung 3.8a, ist für die Zeitreihe $Bundesstaat$ = „New South Wales“ AND $Anlass$ = „Freizeit“ gegeben. Das Disaggregationsmodell E_{Anlass} nutzt das Quellmodell S_{Anlass} mit der Selektionsbedingung $Anlass$ = „Freizeit“, s. Abbildung 3.8b. Das Disaggregationsmodell $E_{Geographie}$ nutzt das Quellmodell $S_{Geographie}$ mit der Selektionsbedingung $Bundesstaat$ = „New South Wales“, s. Abbildung 3.8c. Die Disaggregation von einer Zeitreihe entspricht folglich einer Roll-Up-Operation im Datenwürfel: entlang einer gegebenen Dimension, unter Berücksichtigung der Selektionsbedingungen anderer Dimensionen, wird die Sicht auf ein übergeordnetes Dimensionslevel verdichtet, um die resultierende Zeitreihe und das Prognosemodell zu ermitteln.

3.4 ENTWURF VON ABLEITUNGSMODELLEN IM MODELLINDEX

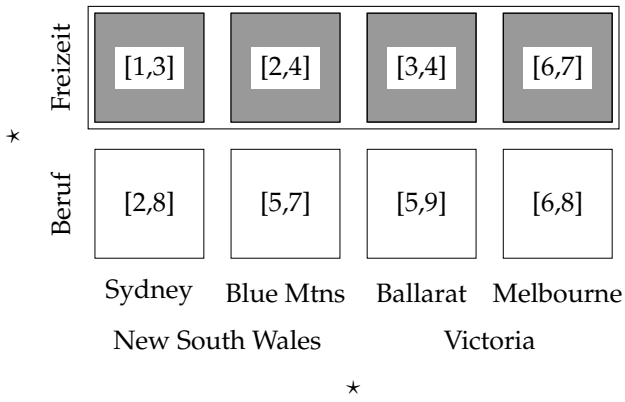
Der folgende Abschnitt stellt strukturelle Änderungen am Modellindex vor, um Ableitungsmodelle zu integrieren. Hierzu werden neue Datenstrukturen für die Ableitungsmodelle erstellt, s. Unterabschnitt 3.4.1, und die bestehende Struktur der Indexknoten für die Speicherung von Ableitungsmodellen erweitert, s. Unterabschnitt 3.4.2.



(a) Prognosemodell S: Ausgangssituation



(b) Disaggregationsmodell E_{Anlass} nutzt das Quellmodell S_{Anlass}



(c) Disaggregationsmodell $E_{Geographie}$ nutzt das Quellmodell $S_{Geographie}$

Abbildung 3.8: Datenwürfel mit Disaggregationsmodellen

3.4.1 Strukturen für Ableitungsmodelle

Die Abhängigkeit eines Ableitungsmodells von Quellmodellen ergibt sich bei der Modellerstellung und muss abgespeichert werden. In Analogie zum Modellknoten werden Ableitungsmodellknoten eingeführt, die die Ableitungsmodelle repräsentieren und die im Modellindex abgespeichert werden. Dafür wird die vorgestellte Struktur des Modellknotens aus dem Vorzustand, s. Unterabschnitt 2.3.4 (Seite 23), durch neue Klassen ersetzt. Abbildung 3.9 zeigt das dazugehörige Diagramm.

Die Entwurfsentscheidung mit Vererbung ergibt sich aus der gemeinsamen Nutzung bestimmter Attribute. Die Abhängigkeit der Ableitungsmodelle von direkten Prognosemodellen ist in einer bidirektionalen Beziehung festgehalten. Nachfolgend werden die Klassen im Detail erläutert:

- *Abstrakter Modellknoten*: Dieser Modellknoten vereinigt alle Attribute, die von direkten und Ableitungsmodellknoten benötigt werden. Sie waren Bestandteil der Klasse Modellknoten des Vorzustands. Hinzu kommt die *Tupelnummer*: Sie ist ein eindeutiger Bezeichner des letzten eingefügten Tupels, von dem der Modellknoten betroffen war. Modellknoten an einem Indexknoten sind aufgrund der asynchronen Wartung unterschiedlich weit fortgeschritten, weshalb die Beschreibung durch den Zeitstempel nicht mehr ausreicht. Die Tupelnummer

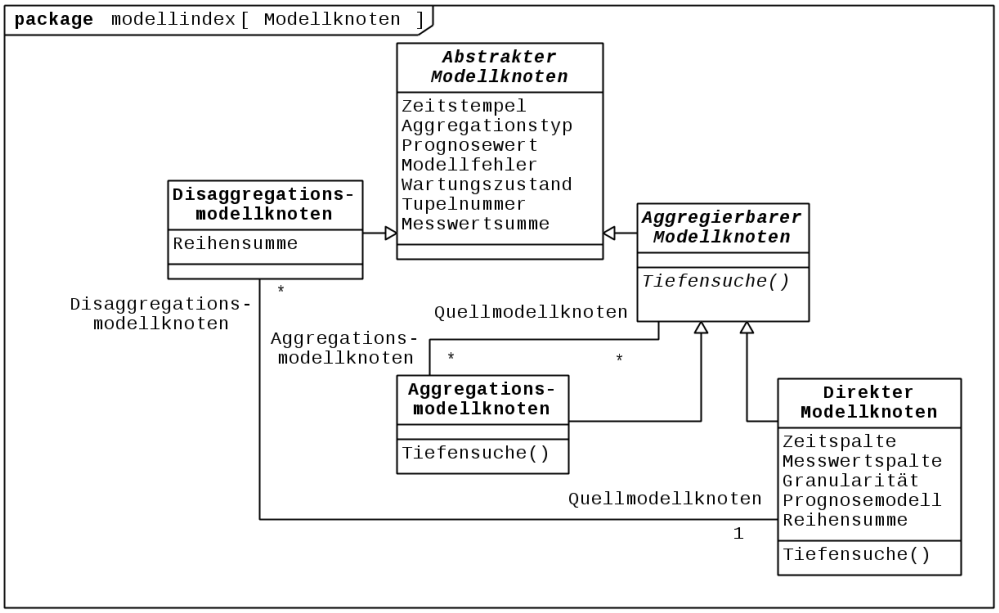


Abbildung 3.9: Klassendiagramm der Modellknoten

dient der Unterscheidung, für welche Tupel das Prognosemodell bereits zustandsgewartet wurde und für welche Tupel diese Wartung noch aussteht.

- *Aggregierbarer Modellknoten*: Aggregierbar sind alle direkten Prognosemodelle und Aggregationsmodelle, da die Aggregationsstrategie rekursiv ist. Daher wird für den Entwurf ein Kompositionsmuster gewählt. Der aggregierbare Modellknoten stellt eine Komponente dar. Genutzt wird das Muster für die Bestimmung aller direkten Prognosemodelle, die zu einem Aggregationsmodell gehören. Hierfür wird eine *Tiefensuche* durchgeführt.

- *Direkter Modellknoten*: Der Modellknoten, der ein *direktes Prognosemodell* repräsentiert. Die *Reihensumme* ist die Addition aller vergangenen Messwerte. Sie wird für die Bestimmung des Disaggregationssschlüssels benötigt. Die weiteren Attribute entstammen dem Vorzustand aus Unterabschnitt 2.3.4. Bezüglich des Kompositionsmusters stellt diese Klasse ein Blatt dar, das keine Quellmodellknoten hat.

- *Aggregationsmodellknoten*: Dieses Kompositum repräsentiert ein Aggregationsmodell. Dazu müssen die *Quellmodellknoten* bekannt sein. Sie repräsentieren die Quellmodelle, von denen das Aggregationsmodell ableitet. Das Aggregationsmodell ist seinen Quellmodellen bekannt, es ist Bestandteil der Menge *Aggregationsmodellknoten*. Quellmodelle können ihre Aggregationsmodelle bei der Wartung benachrichtigen.

- *Disaggregationsmodellknoten*: Dieser Knoten repräsentiert ein Disaggregationsmodell. Er vereint das Attribut *Reihensumme*, das für den Disaggregationsschlüssel benötigt wird, mit dem *Quellmodellknoten*, der das Quellmodell repräsentiert. Das Disaggregationsmodell ist seinem Quellmodell bekannt, es ist Bestandteil der Menge *Disaggregationsmodellknoten*. Quellmodelle können ihre Disaggregationsmodelle bei der Wartung benachrichtigen.

Die Knoten repräsentieren die direkten Prognosemodelle bzw. die Ableitungsmodelle im Modellindex. Wenn eine Unterscheidung zwischen Knoten und Prognosemodell nicht nötig ist, werden die Begriffe in der Folge synonym verwendet, um die Lesbarkeit zu erleichtern.

3.4.2 Ergänzung der Indexknoten

Ein Indexknoten verfügte bisher über eine Menge von direkten Modellknoten, die bei der Modellsuche zurückgegeben wurden. Für die Integration der Ableitungsmodelle werden die Attribute *Aggregations-* und *Disaggregationsmodellknoten* ergänzt, die auf die Knoten der Ableitungsmodelle zeigen, s. Abbildung 3.10. Ableitungsmodelle werden somit bei der Modellnutzung berücksichtigt.

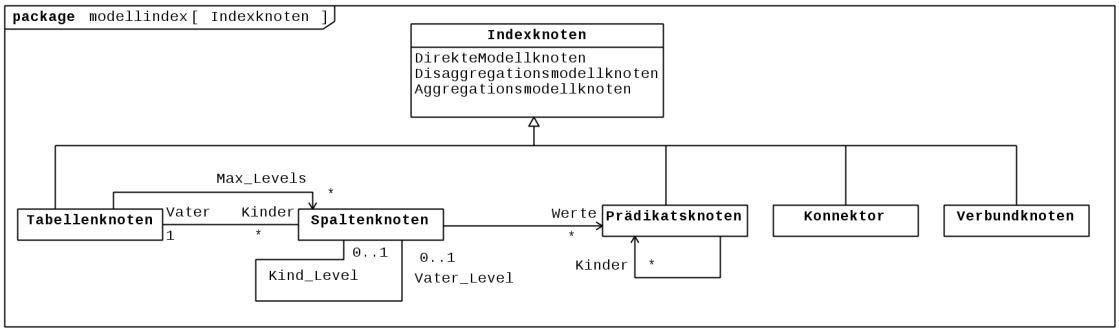


Abbildung 3.10: Klassendiagramm der Indexknoten und des Spaltenknotens

Des Weiteren werden folgende Strukturen ergänzt, um eine Roll-Up- und Drill-Down-Operation wie beim Datenwürfel zu ermöglichen:

- *Tabellenknoten*: Aus dem Vorzustand ist gegeben, dass ein Tabellenknoten alle Spalten mit Kategorienattributen kennt, die als *Kinder* bezeichnet werden. Zusätzlich muss bekannt sein, welche Dimensionen durch die Tabelle aufgespannt werden. Als Stellvertreter jeder Dimension sei das jeweils höchste Dimensionslevel ohne Top_D, dem Max_Level, ausgewählt. Hiervon lassen sich über *Kind_Level* alle untergeordneten Dimensionslevels bestimmen. Das maximale generische Level *Top* ist implizit. Das Attribut *Max_Levels* beschreibt die Menge dieser Dimesionslevels.
- *Spaltenknoten*: Ein Dimensionslevel entspricht einer Spalte in der Tabelle und wird somit durch einen Spaltenknoten repräsentiert. Die über- und untergeordneten Dimensionslevels *Vater_Level* und *Kind_Level* ergeben sich durch Assoziation zwischen zwei Spaltenknoten. Das Primärattribut hat kein Kindlevel und das *Max_Level* einer Dimension hat implizit *Top* als Vaterlevel.
- *Prädikatsknoten*: Jedem Spaltenknoten sind Prädikatsknoten zugeordnet, sie repräsentieren die *Werte* eines Dimensionslevels. Die *Kinder* eines Werts sind seine Nachfolger auf dem untergeordneten Dimensionslevel. Dieses Attribut und *Vater_Level* setzen die funktionale Abhängigkeit der Dimensionslevels um, vgl. Unterabschnitt 3.3.1 (Seite 33).

Für das Beispiel 2 wird die Kandidatensuche anhand der Abbildung 3.11 erläutert. Sie zeigt die Indexknoten für New South Wales und zwei Prognosemodell S_1 und S_2. Die Verbindungen der Objekte stellen Links dar, d. h. Ausprägungen der Assoziationen aus dem Klassendiagramm von Abbildung 3.10.

Für das Prognosemodell S_1 mit Selektionsbedingung $Bundesstaat$ = „New South Wales“ wird ein Aggregationskandidat ermittelt. Beim Durchsuchen des Modellindexes nach S_1 ergibt sich,

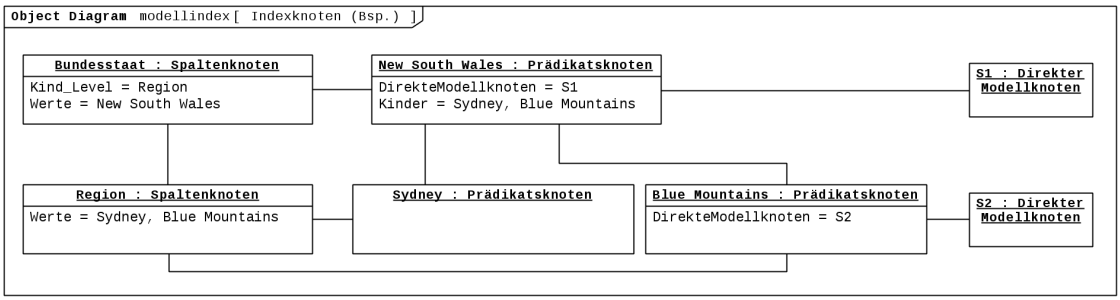


Abbildung 3.11: Objektdiagramm der Indexknoten

dass das Prognosemodell am Prädikatsknoten „New South Wales“ und dem Spaltenknoten *Bundesstaat* angehängt ist. Der Link *Kind_Level* führt zum untergeordneten Dimensionslevel und der Link *Kinder* zu den Werten, die „New South Wales“ funktional bestimmen. An den Kindern befinden sich entweder schon Quellmodelle für die Aggregation, d. h. S_2, oder sie müssen noch erstellt werden, z. B. für $Region =$ „Sydney“. Dies entspricht der Umsetzung von Definition 12 (Seite 39) und dem Aggregationsmodell $A_{Geographie}$, Abbildung 3.7b (Seite 40).

Für das Prognosemodell S_2 ist ein Disaggregationsmodell zu bestimmen. Der Spalten- und Prädikatsknoten, an dem dieses Prognosemodell angehängt ist, wird über den Modellindex ermittelt. Über den Link *Vater_Level* wird das übergeordnete Dimensionslevel bestimmt. Dessen *Werte* werden durchsucht, bis ein *Wert* gefunden wird, dessen *Kind* auf „Blue Mountains“ zeigt. Dies trifft bei $Bundesstaat =$ „New South Wales“ zu, an dem sich das Quellmodell S_1 befindet. Somit ist die Definition 13 (Seite 41) im Modellindex umgesetzt. Durch die Suche über die Spaltenknoten entfällt die bidirektionale Beziehung zwischen Prädikatsknoten.

3.5 MODELLERSTELLUNG UND -NUTZUNG

Analog zum Lebenszyklus direkter Prognosemodelle beschreiben die nachfolgenden Abschnitte die Modellerstellung -nutzung und -wartung von Ableitungsmodellen. Die Erstellung, Unterabschnitt 3.5.1, stellt die Umsetzung der manuellen und automatisierten Einfügung von Ableitungsmodellen in den Modellindex vor. Unterabschnitt 3.5.2 befasst sich mit der Nutzung von Ableitungsmodellen für eine Anfrage. Hierzu gehört erstens die Diskussion, welches Prognosemodell zum Einsatz kommt, wenn mehrere Prognosemodelle, direkt und abgeleitet, vorliegen. Zweitens gehören hierzu die Verfahren, mit denen die Prognose von Ableitungsmodellen ermittelt wird. Die Modellwartung als zentraler Bestandteil der vorliegenden Arbeit wird in Abschnitt 3.6 behandelt.

3.5.1 Modellerstellung

Ableitungsmodelle werden wie direkte Prognosemodelle in den Knoten des Modellindexes abgelegt, sodass sie bei der Modellsuche gefunden werden. Hierfür werden zwei Möglichkeiten unterschieden: die manuelle und die automatische Modellerstellung.

Bei ersterer legt der Nutzer ein Ableitungsmodell im Modellpool an. Dafür wird ein Befehl entworfen werden, der die nötigen Variablen für die eindeutige Beschreibung des Ableitungsmodells entgegennimmt. Ferner müssen alle beteiligten Modelle, von denen das Ableitungsmodell abhängig ist, im Modellpool vorhanden sein. Dies kann durch eine frühere manuelle oder automatische Modellerstellung erfolgt sein.

Die zweite Möglichkeit ist die automatische Erstellung eines Ableitungsmodells durch die Wartung der Ableitungsregeln. Alle abhängigen Modelle werden hierfür ebenso automatisch erstellt, wenn sie noch nicht im System existieren.

Unabhängig von der Ableitungsregel müssen Zeit- und Messwertspalte des Quellmodells sowie die Faktentabelle bekannt sein. Zeit- und Messwertspalte sind für Quellmodelle identisch, um zuzusichern, dass die Zeitreihe die gleichen Zeitpunkte hat und sich auf die gleichen Messwerte bezieht. Ebenso müssen die Granularitäten der Quellmodelle identisch sein, sonst wäre die Summierbarkeit der Zeitreihen (Vollständigkeits-Eigenschaft) nicht gewährleistet.

Aggregationsmodell

Gemäß Definition 12 (Seite 39) müssen Dimension, Dimensionslevels und Werte bekannt sein, die die Auswahl der Zeitreihe ermöglichen, sowie die Dimension für die Aggregation. Bei der automatischen Erstellung ist das Aggregationsmodell aus der Menge $\boldsymbol{Agg}$ gegeben.

Bei einer manuellen Erstellung gibt der Nutzer die Auswahl der Zeitreihe durch eine Anfrage an. Zudem nennt er die *Aggregationsbedingung*, eine Selektionsbedingung, für die vom Kindlevel aggregiert werden soll. Wenn z. B. der Nutzer die Aggregationsbedingung $Bundesstaat =$ „New South Wales“ angibt, so wird über alle Zeitreihen aggregiert, deren Region in „New South Wales“ liegt, z. B. $Region =$ „Sydney“.

Disaggregationsmodell

Gemäß Definition 13 (Seite 41) müssen Dimension, Dimensionslevels und Werte bekannt sein, die die Auswahl der Zeitreihe ermöglichen, sowie die Dimension für die Disaggregation. Bei der automatischen Erstellung ist das Disaggregationsmodell aus der Menge $\boldsymbol{Disagg}$ gegeben.

Bei einer manuellen Erstellung gibt der Nutzer die Auswahl der Zeitreihe durch eine Anfrage an. Zudem nennt er die *Disaggregationsbedingung*, eine Selektionsbedingung, für die vom Vaterlevel disaggregiert werden soll. Wenn die Disaggregationsbedingung $Region =$ „Blue Mountains“ lautet, so wird vom Prognosemodell mit Selektionsbedingung $Bundesstaat =$ „New South Wales“ disaggregiert.

3.5.2 Modellnutzung

Aufgabe der Modellnutzung ist es, einer Prognoseanfrage das passende Prognosemodell zuzuordnen. Im Allgemeinen ist an einem Indexknoten nur ein Prognosemodell für die Modellnut-

zung aktiv. Wenn durch die Wartung der Ableitungsregeln ein Prognosemodell durch ein Ableitungsmodell ersetzt wurde, so wird das ersetzte Prognosemodell für die Modellnutzung ignoriert.

In speziellen Fällen können auch mehrere Prognosemodelle an einem Indexknoten existieren, die die Anfrage erfüllen. Beispielsweise kann der Nutzer bei Modellerstellung mehrere Prognosemodelle, direkt und abgeleitet, am Indexknoten erstellen. Am Indexknoten kann sich auch ein Ableitungsmodell befinden und zusätzlich ein direktes Prognosemodell, das als Quellmodell für eine anderes Ableitungsmodell genutzt wird.

Daher werden die Prognosemodelle, die die Prognoseanfrage erfüllen, miteinander verglichen, um das geeignete auszuwählen. Grundlage ist der Prognosefehler, der anzeigt, wie geeignet das Prognosemodell in der Vergangenheit war. Dies dient als Kriterium für die Prognosegenauigkeit. Das Prognosemodell mit dem kleinsten Prognosefehler wird genutzt. Es ist sehr unwahrscheinlich, dass zwei Modelle den gleichen Prognosefehler haben, sodass dieses Kriterium im Allgemeinen ausreicht.

3.6 MODELLWARTUNG

Zentraler Bestandteil der Arbeit ist die Definition von Wartungsoperationen für Ableitungsmodelle. Die nötigen Schritte zur Erweiterung der Wartungskomponente werden in folgendem Abschnitt vorgestellt. Die neuen Wartungsoperationen sind die Wartung der Ableitungsgewichte und die Wartung der Ableitungsregeln. Eine Übersicht, wie die Wartungsoperationen in die Wartungskomponente integriert werden, ist in Unterabschnitt 3.6.1 gegeben. In Unterabschnitt 3.6.2 wird auf die Wartung der Ableitungsgewichte eingegangen. Die Zustandswartung wird durch die Wartungsoperation um einige Funktionen erweitert und in Unterabschnitt 3.6.3 erläutert. Unterabschnitt 3.6.4 stellt schließlich die Wartung der Ableitungsregeln vor.

3.6.1 Überblick über Wartungsoperationen

Die Wartungsoperationen sind wie in Abbildung 3.12 hierarchisiert. Die ursprünglichen Wartungsoperationen für direkte Prognosemodelle verbleiben in der Hierarchie.

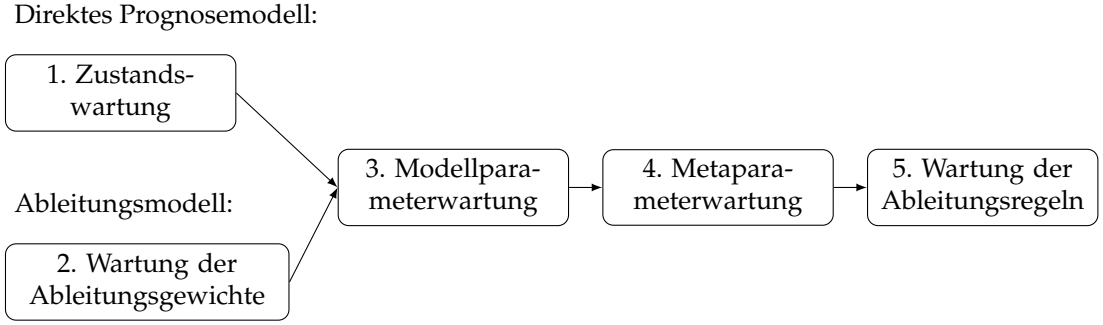


Abbildung 3.12: Übersicht über Wartungsoperationen

1. *Zustandswartung*: Diese Wartungsoperation passt den Zustand des Prognosemodells bei Fortschreiben der Zeitreihe an. Sie wurde in Abschnitt 2.1 (Seite 15) beschrieben.
2. *Wartung der Ableitungsgewichte*: Diese Wartungsoperation kann in Analogie zur Zustandswartung direkter Prognosemodelle betrachtet werden. Das Ableitungsgewicht wird aktualisiert, um die Einfügung neuer Tupel zu berücksichtigen. Sie findet daher gleichzeitig mit der Zustandswartung statt.
3. *Modellparameterwartung*: Die Wartungsoperation führt eine Modellschätzung durch. Ableitungsmodelle führen sie für ihre Quellmodelle durch.
4. *Metaparameterwartung*: Bei dieser Wartungsoperation setzt das Prognosesystem die Metaparameter des Prognosemodells neu und führt jeweils eine Modellschätzung durch. Ein Ableitungsmodell führt diese Wartungsoperation für seine Quellmodelle durch.
5. *Wartung der Ableitungsregeln*: Für Prognosemodelle werden Ableitungsregeln gesucht und geprüft. Falls eine Ableitung einen geringeren Prognosefehler hat, wird das Prognosemodell durch ein Ableitungsmodell ersetzt. Ableitungsmodelle können auch durch direkte Prognosemodelle ersetzt werden. Die Wartungsoperation ist zeitaufwändig und findet daher statt, wenn vorangegangene Operationen nicht erfolgreich waren.

Die folgenden Unterabschnitte erläutern die Einführung neuer Wartungsoperationen und die Erweiterung der Zustandswartung.

3.6.2 Wartung der Ableitungsgewichte

Ein Ableitungsgewicht beschreibt die Zuteilung des Prognosewerts eines Ableitungsmodells vom Prognosewert des zugehörigen Quellmodells. Im Zusammenhang mit dem Disaggregationsmodell ist das Ableitungsgewicht der Disaggregationsschlüssel, s. Abschnitt 3.1 (Seite 28). Für Aggregationsmodelle werden die Prognosewerte der Quellmodelle summiert, womit das Ableitungsgewicht stets 1 beträgt und daher nicht gesondert gewartet werden muss. In der Folge wird die Wartung des Ableitungsgewichts für Disaggregationsmodelle betrachtet.

Als geeigneter Disaggregationsschlüssel kommt die Division aus Messwerten der Zeitreihe des Disaggregationsmodells und des Quellmodells in Frage. Auf das Klassendiagramm in Abbildung 3.9 übertragen lautet der Disaggregationsschlüssel $P_{i,T}$ für das Disaggregationsmodell E_i mit Quellmodell S_i:

$$\begin{aligned} P_{i,T} &= \frac{\text{Reihensumme } E_{i,T}}{\text{Reihensumme } S_{i,T}} \\ &= \frac{\sum_{t=1}^{T} \text{Messwertsumme } E_{i,t}}{\sum_{t=1}^{T} \text{Messwertsumme } S_{i,t}} \end{aligned} \tag{3.14}$$

Damit der Disaggregationsschlüssel aktuell ist, muss die Messwertsumme bei Abschluss eines Zeitpunkts auf die Reihensumme aufaddiert werden. Diese Wartungsoperation ist nicht zeitaufwändig und wird daher mit der Zustandswartung durchgeführt. Somit ist keine Menge von Messwertsummen zu speichern, sondern nur eine Messwertsumme für den Gegenwartszeitpunkt. Im folgenden Abschnitt wird ein Konzept vorgestellt, mit dem die Wartung der Ableitungsgewichte in die Zustandswartung integriert wird.

3.6.3 Erweiterung der Zustandswartung

Die Zustandswartung passt den Zustand des Prognosemodells an, wenn die Zeitreihe fortgeschrieben wird, s. Abschnitt 2.3 (Seite 20). Diese Operation wird auch genutzt, um den Prognosefehler zu aktualisieren, mit dem Entscheidungen über weitere Wartungsoperationen getroffen werden. Eine Erweiterung der Zustandswartung ist nötig, um den Prognosefehler von Ableitungsmodellen zu aktualisieren. Zusätzlich wird die Wartung von Ableitungsgewichten in diese Operation integriert, um Disaggregationsschlüssel zu aktualisieren. Der Vorgang wird anhand des Aktivitätsdiagramms in Abbildung 3.13 verdeutlicht.

Der Vorgang beginnt mit der Einfügung eines neuen Tupels. Die Modellsuche gibt alle *Modelle* an, die von der Einfügung *betroffen* sind. Dazu zählen Ableitungsmodelle. Sie werden jedoch nicht unabhängig von ihren Quellmodellen aktualisiert, sondern von ihren Quellmodellen benachrichtigt. Ableitungsmodelle benötigen für die Zustandswartung den Ein-Schritt-Prognosewert des Quellmodells, weshalb das Quellmodell stets vor dem Ableitungsmodell gewartet werden muss. Die folgenden Abschnitte erläutern die Zustandswartung, bezogen auf direkte Prognosemodelle und Ableitungsmodelle.

Zustandswartung direkter Prognosemodelle

Sei ein direktes Prognosemodell ausgewählt, das sich nicht in Wartung befindet. Der Zeitstempel des Tupels entscheidet über das weitere Vorgehen:

- Ist der Zeitstempel oder die Tupelnummer veraltet, wird das Tupel ignoriert, denn es wurde durch eine andere Wartungsoperation berücksichtigt, die die Zeitreihe direkt aus der Datenbank lädt. Dies ist bei den komplexen Wartungoperationen möglich.
- Stimmt der Zeitstempel mit dem des Modellknotens überein und ist das Tupel neu, so gehört das Tupel zum aktuellen Messwert. Die *Messwertsumme* und die *Tupelnummer* werden *aktualisiert*. Zudem werden Ableitungsmodelle benachrichtigt. Die Messwertsumme betroffener, abgeleiteter Disaggregationsmodelle wird aktualisiert. Betroffen sind Disaggregationsmodelle, wenn sie in der Liste betroffener Modelle enthalten sind. Auch die Messwertsumme von Aggregationsmodellen wird aktualisiert. Wenn ein Quellmodell von einer Einfügung betroffen ist, so sind es auch alle abhängigen Aggregationsmodelle.
- Wenn der Zeitstempel neu ist, gilt der letzte Zeitpunkt als abgeschlossen. Zunächst wird der *Prognosefehler aktualisiert*, denn es liegen der Prognosewert und die abgeschlossene Messwertsumme vor. Hierbei wird auch der zeitbasierte Fehler aktualisiert. Die anschließende *inkrementelle Wartung* ist die eigentliche Zustandwartung des Prognosemodells. Beim Holt-Winters-Verfahren werden hierbei die Koeffizienten a, b und c aktualisiert. Dies ermöglicht die Bestimmung des nächsten *Ein-Schritt-Prognosewerts*. Abschließend wird die Messwertsumme auf die *Reihensumme* aufaddiert, die bei Disaggregationsmodellen benötigt wird, und die *Messwertsumme* auf den Messwert des aktuellen Tupels gesetzt.
- Wenn das Prognosemodell ohne Gruppierung definiert ist, werden Reihen- und Messwertsumme nicht benötigt: Jedes Tupel impliziert einen neuen Zeitstempel, sodass keine Messwerte in der Messwertsumme zwischengespeichert werden müssen. Zudem kann von einem Prognosemodell ohne Gruppierung kein Disaggregationsmodell abgeleitet werden,

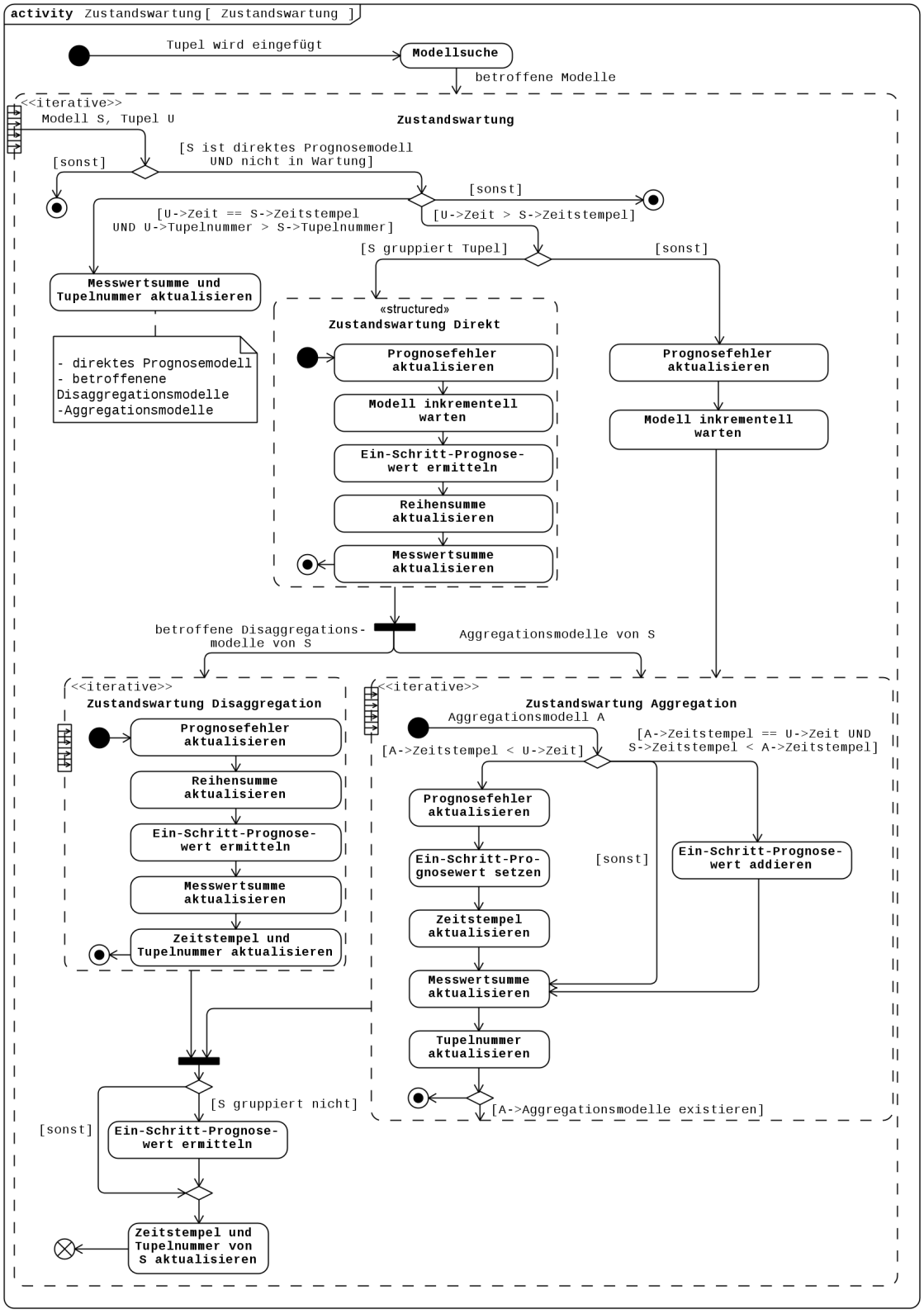


Abbildung 3.13: Aktivitätsdiagramm der Zustandswartung

weshalb die Reihensumme nicht benötigt wird. Der Übertrag des neuen Prognosewerts wird erst nach Benachrichtung abgeleiteter Aggregationsmodelle durchgeführt, weil ein Prognosemodell ohne Gruppierung den Prognosemodellen mit Gruppierung um einen Zeitpunkt voraus ist. Ein Beispiel: Sei $t = T$ der Zeitstempel des eingefügten Tupels. Ein Prognosemodell mit Gruppierung schließt dadurch den Zeitpunkt $t = T - 1$ ab und ermittelt den Prognosewert $\hat{x}_{T-1,1}$. Ein Prognosemodell ohne Gruppierung schließt dadurch den Zeitpunkt $t = T$ ab und ermittelt den Prognosewert $\hat{x}_{T,1}$. Aggregationsmodelle summieren jedoch die Prognosewerte $\sum \hat{x}_{S_i,T-1,1}$ bis zum Abschluss von $t = T$.

Anschließend erfolgt die *Zustandswartung von Disaggregations-* und *Aggregationsmodellen*.

Zustandswartung von Disaggregationsmodellen

Das eingefügte Tupel bedeutet auch für ein betroffenenes Disaggregationsmodell, dass der Zeitstempel neuer ist. Auch hier wird der *Prognosefehler aktualisiert* und die *Reihensumme aktualisiert*. Dabei wird die Messwertsumme auf die derzeitige Reihensumme addiert. Dieser Schritt gehört zur Wartung der Ableitungsgewichte: Nun ist sowohl die Reihensumme des Quell- als auch des Disaggregationsmodells aktuell. Der neue *Ein-Schritt-Prognosewert* ergibt sich durch Zuteilung vom Quellmodell, vgl. Gleichung 3.14 (Seite 48). Abschließend wird die *Messwertsumme* auf den Messwert des eingefügten Tupels gesetzt. *Zeitstempel* und *Tupelnummer* werden vom eingefügten Tupel übernommen.

Zustandswartung von Aggregationsmodellen

Ein Aggregationsmodell hat einen eigenen Zeitstempel, um zu unterscheiden, welche Quellmodelle es bereits benachrichtigt haben. Dabei treten folgende Fälle auf:

- Der Zeitstempel des eingefügten Tupels ist neuer. Folglich wird der *Prognosefehler aktualisiert*, denn der vergangene Zeitpunkt ist abgeschlossen. Anschließend wird der neue *Prognosewert* auf den Prognosewert des Quellmodells *gesetzt*, von dem es benachrichtigt wurde. Der *Zeitstempel* wird vom eingefügten Tupel übernommen.

- Stimmt der Zeitstempel mit dem des Tupels überein und hat das benachrichtigende Quellmodell noch den vergangenen Zeitstempel, so wurde das Aggregationsmodell bereits über den Zeitstempel benachrichtigt. Es kennt aber nicht den Prognosewert des benachrichtigenden Quellmodells, sodass nur der *Prognosewert hinzuaddiert* wird, vgl. Gleichung 3.8 (Seite 39).

Abschließend wird die *Messwertsumme aktualisiert*: War der Zeitstempel neuer, wird sie auf den neuen Messwert gesetzt. War er der gegenwärtige, so wird der Messwert hinzuaddiert. Die *Tupelnummer* wird vom eingefügten Tupel übernommen. Der rekursive Fall wird umgesetzt, indem das Aggregationsmodell weitere Aggregationsmodelle benachrichtigt, die von ihm ableiten.

3.6.4 Wartung der Ableitungsregeln

Die Wartung der Ableitungsregeln ermöglicht das automatische Ersetzen direkter Prognosemodelle durch Ableitungsmodelle und umgekehrt. Zwei Ziele werden verfolgt: Erstens kann eine Ableitung gefunden werden, die eine Verbesserung des Prognosegenauigkeit erzielt; zweitens kann die Anzahl zu wartender Prognosemodelle reduziert werden, da durch die Ableitung Prognosemodelle mehrmals genutzt werden. Die Wartungsoperation gliedert sich in vier Komponenten, s. Abbildung 3.14. Die folgenden Absätze erläutern das Verfahren der Wartungsoperation für die Ersetzung eines direkten Prognosemodells durch ein Ableitungsmodell. Der umgekehrte Fall, die Ersetzung eines Ableitungsmodells durch ein direktes Prognosemodell oder ein anderes Ableitungsmodell, ist mit dem gleichen Verfahren möglich.

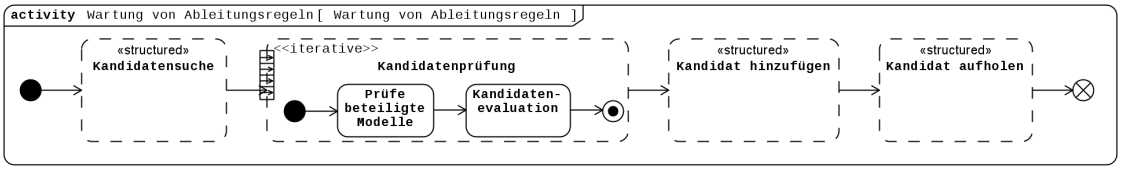


Abbildung 3.14: Wartung der Ableitungsregeln: Komponenten

Ausgehend vom Indexknoten des zu ersetzenden Prognosemodells ermittelt die *Kandidatensuche* Quellmodelle, die für eine Ableitung in Frage kommen, s. Unterabschnitt 3.3.2 (Seite 37). Ein *Kandidat* repräsentiert eine mögliche Ableitung, ohne dass dafür bereits die Modelle existieren müssen.

Die anschließende *Kandidatenprüfung* untersucht für jeden Kandidaten, ob sich an betroffenen Indexknoten Quellmodelle befinden, die genutzt werden können. Sie werden wiederverwendet, indem eine temporäre Kopie erstellt wird. Existiert ein Quellmodell noch nicht, so wird es temporär erstellt. Für die Untersuchungen wird angenommen, dass alle Zeitreihen die gleichen Zeitpunkte und die gleiche Granularität aufweisen. Der Kandidat ist nach dieser Phase ein temporäres Ableitungsmodell, das temporäre Quellmodelle nutzt und nicht im Modellindex abgelegt ist.

Anschließend wird der Kandidat auf dem gleichen Zeitraum wie das zu ersetzende Prognosemodell *evaluiert*. Beim abschließenden Vergleich der Prognosefehler aller Kandidaten wird der Kandidat mit dem geringsten Fehler für die Ersetzung ausgewählt, er ist der *minimale Kandidat*. Die anderen Kandidaten verfallen. Wenn der minimale Kandidat keine Verbesserung aufweist, endet die Wartungsoperation ohne Ersetzung.

Die Komponente *Kandidat hinzufügen* fügt den minimalen Kandidaten in den Modellindex ein. Die Modelle, die bisher temporär waren, werden in den Modellpool übertragen und indexiert.

Die abschließende Komponente *Kandidat aufholen* aktualisiert den Modellzustand des minimalen Kandidaten. Da es sich um eine asynchrone Wartungsoperation handelt, müssen Tupel, die nach der Kandidatenevaluation eingefügt wurden, den Modellen bekannt sein. Der Puffer für die Einfügetupel hält sie nicht zurück, da noch nicht bekannt war, dass ein betroffenes Prognosemodell existiert.

3.7 IMPLEMENTIERUNG

Der folgende Abschnitt erläutert die Umsetzung vorgestellter Konzepte im Prototypen F²DB. Dabei werden Entscheidungen hinsichtlich Änderungen an der Wartungskomponente (Unterabschnitt 3.7.1) und der Bestimmung der funktionalen Abhängigkeit (Unterabschnitt 3.7.2) erläutert.

3.7.1 Änderungen der Wartungskomponente

Die Wartungskomponente teilt sich in mehrere Aktivitäten auf, die in Abbildung 2.2 (Seite 21) zusammengefasst sind. Im Vorzustand lief die Komponente in einem Hintergrundprozess, den PostgreSQL seit Version 9.3 ermöglicht [Pos12]. Dabei wurden Arbeiter als Threads realisiert, die die Wartungsoperationen (Modellparameter-, Metaparameterwartung) verarbeiten. Obwohl PostgreSQL kein Multithreading unterstützt [Lan12], konnten diese Operationen mit Threads realisiert werden. Für die vorliegende Umsetzung werden Arbeiter als Prozesse umgesetzt. Sie sind zwar schwergewichtiger als Threads, können jedoch Funktionen von PostgreSQL für die Allokation (`palloc`, `MemoryContext`) nutzen, die bei der Implementierung unverzichtbar waren.

Die Tupelnummer identifiziert Tupel eindeutig. Bei Wartungsoperationen werden Zeitreihen aus der Datenbank geladen, die Tupelnummer belegt dabei das letzte eingefügte Tupel, um eine doppelte Einfügung zu verhindern. PostgreSQL wird durch `default_with_oids = on` so konfiguriert, dass jedem Tupel ein eindeutiger Identifikator zugeordnet wird [Pos12]. Es ermöglicht die eindeutige Identifikation von bis zu 4 Milliarden Tupel, was für die Evaluation des Prototyps hinreichend ist.

3.7.2 Bestimmung der Dimensionen

Um die Wartung der Ableitungsregeln zu nutzen, müssen die funktionalen Abhängigkeiten der Kategorienattribute bekannt sein. In Anlehnung an die `HIERARCHY`-Klausel von Oracle [Ora12] wird für PostgreSQL die Anweisung `CREATE HIERARCHY` implementiert. Sie bestimmt die funktionale Abhängigkeit in jeder Dimension einer Universalrelation. Für das Beispiel 2 lautet die Anweisung wie in Abbildung 3.15. Es wird nach Erstellung eines Prognosemodells aufgerufen. Auf Basis vorhandener Tupel setzt es automatisch die Assoziationen *Max_Level*, *Kind_Level*, *Vater_Level* bzw. *Kinder* der Spalten- bzw. Prädikatsknoten im Modellindex, siehe Abbildung 3.10 (Seite 44). Es ist keine informative Zusicherung, der Nutzer ist für die Einhaltung der funktionalen Abhängigkeiten selbst verantwortlich. Wie bei der Konzeption ausgeführt, beschränkt sich der Prototyp auf einen Dimensionspfad pro Dimension.

```
1 CREATE HIERARCHY 'Geographie' (Region CHILD OF Bundesstaat),
2 'Anlass' (Anlass) ON (SELECT * FROM Tourismus)
```

Abbildung 3.15: Anweisung zur Erstellung funktionaler Abhängigkeiten

3.8 EVALUATION

Die vorgestellte Konzeption von Ableitungsmodellen wurde im Prototypen F²DB implementiert. Der folgende Abschnitt untersucht anhand reeller Datensätze den Einsatz von Ableitungsmodellen für die Verbesserung der Prognosegenauigkeit und für die Reduzierung der Anzahl genutzter Prognosemodelle. Unterabschnitt 3.8.1 stellt eine Untersuchung an einer einzelnen Zeitreihe vor. Die Unterabschnitte 3.8.2 und 3.8.3 evaluieren das Verhalten an einem Datensatz mehrerer Zeitreihen.

Für die Untersuchungen gelten die in Tabelle 3.3 angegebenen Systemvariablen.

Tabelle 3.3: Systemvariablen für F²DB

Systemvariable	Wert	Einheit	Anmerkung
`naptime`	`10`	`ms`	Länge der Inaktivität der Wartungsplanung
`max_tuples`	$2{\cdot}10^7$	–	Länge des Einfügepuffer je Tabelle
`max_workers`	`60`	–	Anzahl Arbeiter-Prozesse
`optim_term_maxtime`	`5`	`s`	Maximale Zeit für Modellschätzung

3.8.1 Untersuchung an einzelner Zeitreihe

In folgendem Unterabschnitt wird eine einzelne Zeitreihe untersucht. Ziel ist der Nachweis, dass der Einsatz von Wartungsoperationen, insbesondere der Wartung der Ableitungsregeln, eine Verbesserung der Prognosegenauigkeit ermöglicht.

Der im Beispiel 2 eingeführte Datensatz **Tourismus** zählt 85 Regionen Australiens und 5 Anlässe der Reise (Freizeit, Freunde besuchen, Beruf, Anderes, Anlass der Reise nicht erfragt) auf. Die Datenanalyse der Plots ergab, dass die zwei letztgenannten Anlässe (Anderes, Anlass der Reise nicht erfragt) für die Prognose ungeeignet sind und ausgeschlossen werden. Des Weiteren werden Regionen ausgeschlossen, die den Transit beschreiben, also Übernachtungen zur Durchreise. Diese Zeitreihen haben oft Messwerte $x = 0$ und lassen keine Trend- oder Saisonkomponente erkennen.

Die verbleibenden 76 Regionen sind 8 Bundesstaaten zugeordnet. Wenn alle möglichen Kombinationen von Dimensionslevels berücksichtigt werden, ergeben sich insgesamt 340 Zeitreihen. Die Länge der Zeitreihe beträgt 16 Jahre (September 1998 bis Juni 2014) mit je einem Messwert pro Quartal, d. h. $16 \cdot 4 = 64$ Messwerten pro Zeitreihe.

Versuchsaufbau Für alle 340 Zeitreihen des Datensatzes **Tourismus** werden Prognosemodelle erstellt. Der Trainingszeitraum beträgt 4 Jahre. In den folgenden 12 Jahren werden die Zeitreihen fortgeschrieben, weshalb die Prognosemodelle evtl. gewartet werden müssen. Die Systemvariablen und Schwellwerte sind in den Tabellen 3.3 und 3.4 angegeben. Die Schwellwerte wurden empirisch ermittelt und ermöglichen eine optimale Wartung durch das System. Der Schwellwert `error_time_elapsed_min` bedeutet, dass frühestens nach der gegebenen Anzahl Tupel komplexe Wartungsoperationen (Modellparameter-, Metaparameterwartung, Wartung der Ableitungsregeln) durchgeführt werden. Ohne diesen Schwellwert würden Aureißer zu einer zu

häufigen Wartung führen. Der zeitbasierte Schwellwert ist nicht definiert, da eine Obergrenze keine Änderung der Prognosegenauigkeit ergibt, dafür ist die Zeitreihe zu kurz. Der fehlerbasierte Schwellwert setzt sich stets durch.

Tabelle 3.4: Schwellwerte für den Datensatz **Tourismus**

Systemvariable	Wert	Einheit	Anmerkung
`error_smape_max`	24	%	fehlerbasierter Schwellwert
`error_time_elapsed_min`	4	–	1 Jahr
`error_time_elapsed_max`	–	–	kein zeitbasierter Schwellwert

Nacheinander werden die Wartungsoperationen eingeschaltet, um eine Vergleichbarkeit der Ergebnisse zu ermöglichen. Zunächst wird nur die Zustandswartung erlaubt. Anschließend kommt Modellparameterwartung hinzu, danach die Metaparameterwartung. Sie sind Wartungsoperationen aus [Keg14], die für diese Untersuchung wiederholt wurden. Abschließend wird zusätzlich die Wartung von Ableitungsregeln erlaubt. Hiermit wird automatisch die Wartung von Ableitungsgewichten berücksichtigt, sodass alle Wartungsoperationen aus Abbildung 3.12 (Seite 47) einsetzbar sind.

Versuchsdurchführung Nach Modellerstellung werden die 48 Tupel der verbleibenden 12 Jahre eingetragen und führen zur Wartung der Prognosemodelle. Arbeiter nehmen Wartungsoperationen entgegen und führen sie aus. Abschließend liegt für jeden Zeitpunkt der Messwert x_t und der dazugehörige Ein-Schritt-Prognosewert $\hat{x}_{t-1,1}$ vor.

Ergebnis Repräsentativ sei das Prognosemodell S_3 von Abbildung 3.16 ausgewählt. Der Boxplot in Abbildung 3.17 zeigt den Prognosefehler jedes einzelnen Prognosewerts als SAPE, dem ungemittelten SMAPE. Der Mittelwert aller 48 Prognosewerte ist als SMAPE angegeben (rotes Kreuz).

```
1 CREATE MODEL S3
2 FOR FORECAST OF messwert ON zeit
3 ALGORITHM HWMODEL TRAINING_DATA (
4   SELECT zeit, messwert FROM Tourismus
5   WHERE bundesstaat = 'Northern Territory'
6   AND anlass = 'Beruf'
7   ORDER BY zeit
8 ) STORAGE modellindex;
```

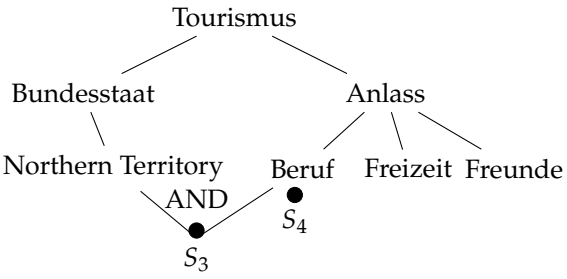


Abbildung 3.16: Anfrage und Modellindex für untersuchtes Prognosemodell

Für dieses Prognosemodell fällt der SMAPE bei Einsatz von Wartungsoperationen von 26 %, wenn nur die Zustandswartung („Zustand“) durchgeführt wird, auf 19 % bei Einsatz eines Disaggregationsmodells, das vom Quellmodell S_4, siehe Abbildung 3.16, ableitet. Dadurch nimmt der Fehler bei Disaggregation („Disagg.“) auch im Vergleich zur Modellparameter- („Param.“) bzw. Metaparameterwartung („Metap.“) ab. Bei jenen liegt der Fehler bei 22 % bzw. 21 %. Ein Aggregationsmodell kann für S_3 nicht ermittelt werden, weshalb der Prognosefehler („Aggreg.“) dem Fehler der Metaparameterwartung entspricht.

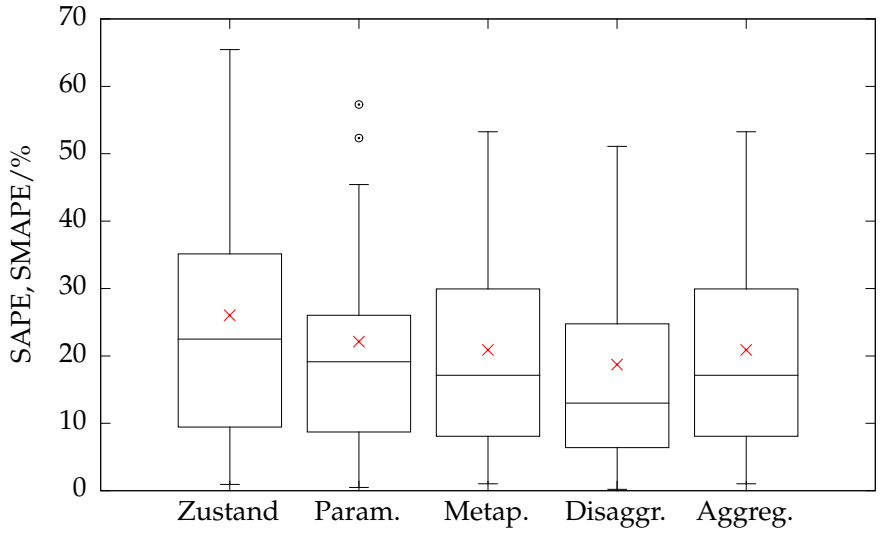


Abbildung 3.17: Prognosefehler für einzelne Zeitreihe

Die Box zeigt außerdem, dass der Median bzw. das 25 %-Quartil fallen: von 23 % bzw. 10 % bei Zustandswartung auf 13 % bzw. 7 % bei der Nutzung der Disaggregation. Das 75 %-Quartil fällt ebenso: von 35 % auf 24 %. Es erhöht sich jedoch bei Zulassung der Metaparameterwartung auf 29 %. Der Grund hierfür ist, dass die Wartungsoperation zu Beginn des Fortschreibens sehr ungünstige Metaparameter setzt, die nach 4 Zeitpunkten jedoch korrigiert werden.

Interpretation Diese Untersuchung zeigt, dass der Einsatz von Wartungsoperationen eine Verbesserung der Prognosegenauigkeit ermöglicht. Die automatische Ersetzung durch ein Disaggregationsmodell zeigt den Vorteil, dass das Prognosemodell S_3 ignoriert werden kann (wenn nicht andere Modelle davon ableiten). Die Ersetzung durch ein Aggregationsmodell konnte noch nicht nachgewiesen werden, dies wird u. a. im folgenden Unterabschnitt untersucht.

3.8.2 Untersuchung am Datensatz Tourismus

Der folgende Unterabschnitt untersucht den Einsatz von Wartungsoperationen am gesamten Datensatz **Tourismus**. Einerseits ist dies durch die Frage motiviert, ob insgesamt eine Reduzierung des Prognosefehlers nachweisbar ist, andererseits, inwieweit die Anzahl zu wartender Prognosemodelle durch den Einsatz von Ableitungsmodellen reduziert werden kann.

Versuchsaufbau und -durchführung

Für alle 340 Zeitreihen des Datensatzes werden Prognosemodelle, wie in Unterabschnitt 3.8.1 beschrieben, erstellt. Die Systemvariablen und Schwellwerte sind wie in den Tabellen 3.3 und 3.4 aufgeführt. Für den Vergleich werden Wartungsoperationen nacheinander eingeschaltet. Tupel werden zeitlich geordnet eingefügt. Ein Arbeiterprozess führt Wartungsoperationen aus. Als Resultat liegen die Mess- und Prognosewerte aller Zeitreihen vor. Zusätzlich ist bekannt, welches Prognosemodell oder welche Prognosemodelle den Prognosewert zu einem Zeitpunkt liefern.

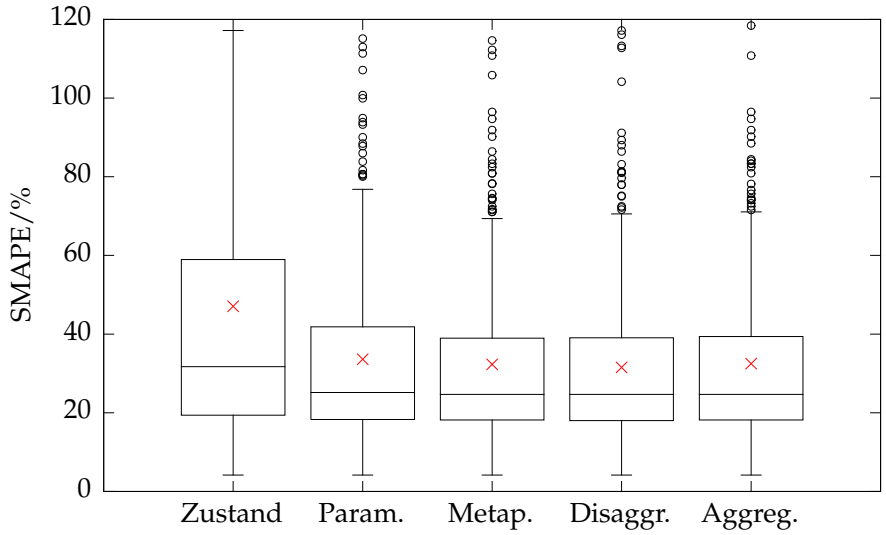


Abbildung 3.18: Prognosefehler für den Datensatz **Tourismus**

Ergebnisse zum Prognosefehler

Abbildung 3.18 zeigt den Boxplot für 5 Fälle, die nach der höchsten zugelassenen Wartungsoperation benannt sind. Die im Boxplot dargestellten Lagemaße charakterisieren den Prognosefehler aller 340 Zeitreihen, das Kreuz repräsentiert das arithmetische Mittel (Mittelwert) aller Prognosefehler. Die ersten drei Fälle stellen die Boxen unter Verwendung der in [Keg14] untersuchten Wartungsoperationen vor. Zu erkennen ist, dass der Median kleiner wird, d. h. der Prognosefehler bei Einschalten zusätzlicher Wartungsoperationen insgesamt geringer wird. Zudem verkleinert sich die Box und sowohl das 25 %- als auch das 75 %-Quartil liegen näher bei 0 %.

Der Einsatz von Disaggregationsmodellen kann insgesamt nur eine geringe Verbesserung im Vergleich zur Metaparameterwartung erzielen: Das 25 %-Quartil verbessert sich von 17,9 % auf 17,7 %. Das 75 %-Quartil verschlechtert sich geringfügig von 38,1 % auf 38,3 %. Der Median ist unverändert bei 24,5 %. Der Mittelwert verbessert sich von 31,8 % auf 31,0 %.

Beim Einsatz von Aggregationsmodellen bleiben Median und 25 %-Quartil gleich, das 75 %-Quartil wird geringfügig größer im Vergleich zur Metaparameterwartung (38,7 %). Der Mittelwert verschlechtert sich ebenfalls geringfügig (31,9 %).

Interpretation Diese Untersuchung zeigt, dass der Einsatz von Ableitungsmodellen eine geringfügige Reduzierung des Prognosefehlers ermöglicht. Die Disaggregation führt bei ausgewählten Zeitreihen zu einer Verbesserung, bei anderen verursacht sie jedoch einen höheren Prognosefehler. Das heißt, dass sich ein Disaggregationsmodell bei der Wartung der Ableitungsregeln durchsetzen kann, aber in der Folge keine bessere Prognose als das (ignorierte) direkte Prognosemodell erzielen muss.

Das größere 75 %-Quartil und der Mittelwert bei der Aggregation deuten darauf hin, dass diese Ableitungsregel für den Datensatz ungeeignet ist. Dies zeigen auch die folgenden Ergebnisse.

Ergebnisse zur Anzahl genutzter Prognosemodelle

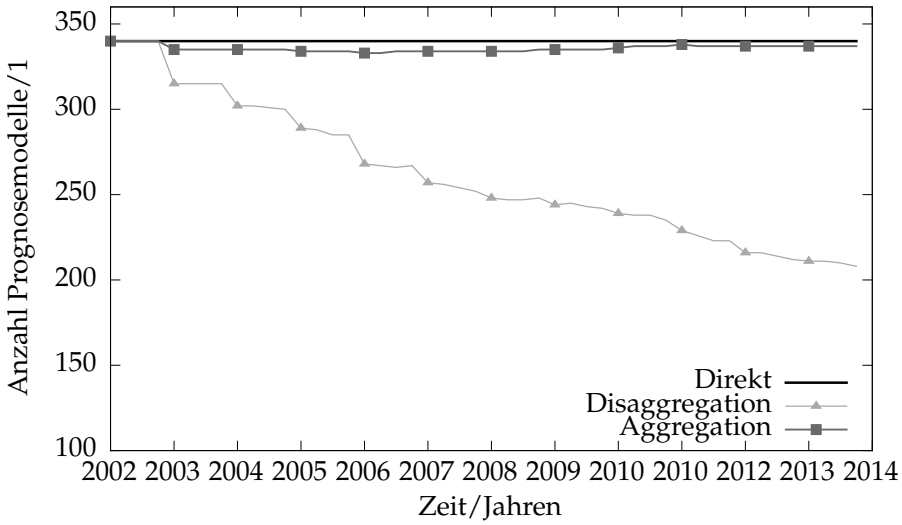


Abbildung 3.19: Anzahl genutzter Prognosemodelle für Datensatz **Tourismus**

Durch Disaggregations- und Aggregationsmodelle ergibt sich eine gemeinsame Nutzung von Prognosemodellen, wie in Unterabschnitt 3.8.1 für einen Fall illustriert wird: S_3 wird ignoriert und die Prognose der Zeitreihe künftig von S_4 bestimmt. Für den gesamten Datensatz zeigt Abbildung 3.19 die Anzahl aktiver direkter Prognosemodelle für jeden Zeitpunkt. Hierbei ist erkennbar, dass die Wartungsoperationen (ausgenommen Zustandswartung) erst nach 4 Tupeln einsetzen, dies legt der Schwellwert `error_time_elapsed_min` fest.

Die Einfügung von Disaggregationsmodellen ermöglicht die Reduzierung der genutzten direkten Prognosemodelle: bei vielen Wartungsoperationen werden direkte Prognosemodelle durch Disaggregationsmodelle ersetzt, sodass schließlich 208 der 340 direkte Prognosemodelle für die Prognose verbleiben. Das gewählte Vaterlevel war hierbei unterschiedlich: Oft ist es Top_{Anlass}, d. h. eine Roll-Up-Operation entlang der Anlass-Dimension, aber auch *Bundesstaat* und $Top_{Geographie}$ sind möglich.

Die Wartung der Ableitungsregeln mit Aggregationsmodellen ist weniger effektiv: Hier werden bis zu 6 direkte Prognosemodelle ignoriert. In der Folge werden sie jedoch wieder aktiv, da Aggregationsmodelle keine bessere Prognose lieferten. Dies zeigt sich an dem gestiegenen Prognosefehler, siehe Abbildung 3.18. Schließlich sind noch 337 direkte Prognosemodelle aktiv, 3 werden ignoriert. Es sei erwähnt, dass für 339 direkte Prognosemodelle die Möglichkeit zur Disaggregation bestand und für 112 direkte Prognosemodelle die Möglichkeit zur Aggregation. Folglich kann die Aggregation nicht so häufig genutzt werden.

Interpretation Die Disaggration erweist sich für diesen Datensatz als effektive Ableitungsregel, denn sie ermöglicht, dass 132 direkte Prognosemodelle ignoriert werden können und nicht mehr gewartet werden müssen. Dabei bleibt der Prognosefehler in der Größenordnung wie bei der Metaparameterwartung, bei der jedoch alle 340 direkte Prognosemodelle gewartet werden müssen.

3.8.3 Untersuchung am Datensatz Wind

Die folgende Untersuchung prüft den Einsatz von Wartungsoperationen am Datensatz **Wind**, s. Beispiel 3. Neben der Überprüfung der Prognosegenauigkeit stellt die Untersuchung aufgrund ihres Umfangs einen Lasttest dar, bei dem die Skalierbarkeit des Prognosesystems ausgenutzt wird. Für die Durchführung kommt ein System mit 64 Prozessoren à 2,13 GHz und 128 GiB Arbeitsspeicher zum Einsatz.

Beispiel 3. *Der Datensatz **Wind** besteht aus 1326 Zeitreihen mit simulierten Leistungsprofilen von Windkraftanlagen im Osten der USA. Die Messwerte (in Watt) wurden im Zeitraum Januar 2004 bis Dezember 2006 erhoben mit einer Granularität von 10 Minuten. Durch Aggregation der Zeitreihen nach Bundesstaaten können die Zeitreihen aggregiert werden. Somit ergeben sich 1361 Prognosemodelle: 1326 für jede Windkraftanlage, 24 für die Bundesstaaten und 1 Modell für das gesamte Aggregat.*

Die Betrachtung der Zeitreihen als Plots zeigen ein sehr chaotisches Verhalten, das den Schluss nahelegt, dass eine Trend- und Saisonkomponente auf der gesamten Länge einer Zeitreihe kaum nachweisbar ist. Dies wird auch in den Messergebnissen zu erkennen sein.

Versuchsaufbau und -durchführung Für die 1361 Zeitreihen werden direkte Prognosemodelle erstellt, der Trainingszeitraum beträgt zwei Jahre. Im folgenden Jahr werden unter Einsatz von Wartungsoperationen fortgeschrieben, sodass für ein Jahr die Messwerte x_t und die dazugehörigen Prognosewerte $\hat{x}_{t-1,1}$ vorliegen.

Die Schwellwerte sind in Tabelle 3.5 angegeben. Der fehlerbasierte Schwellwert wurde empirisch ermittelt. Die zeitbasierten Schwellwerte sind dadurch motiviert, dass jedes Prognosemodell mindestens einmal gewartet werden muss, aber frühestens nach einem Monat. Da die Wartungsoperationen sehr zeitintensiv sind, dient das einer Beschränkung, um das Experiment in vertretbarer Zeit durchzuführen.

Tabelle 3.5: Schwellwerte für den Datensatz Wind

Schwellwert	Wert	Einheit	Anmerkung
`error_smape_max`	14	%	-
`error_time_elapsed_min`	4320	–	Früheste Wartung nach einem Monat
`error_time_elapsed_max`	26352	–	Späteste Wartung nach einem halben Jahr

Ergebnis Abbildung 3.20 zeigt den Boxplot für 4 Wartungsoperationen. Zu erkennen ist, dass der Prognosefehler insbesondere bei Einsatz der Metaparameterwartung abnimmt. Die Modellparameterwartung ermöglichte nur eine sehr geringe Reduzierung. Die Wartung der Ableitungsregeln fand hier keinen Einsatz, weshalb die Disaggregation das gleiche Ergebnis wie die Metaparameterwartung zeigt. Auch die Aggregation (nicht dargestellt) fand keinen Einsatz, wie in der Interpretation erläutert wird.

Interpretation Auch hier ermöglicht der Einsatz von Wartungsoperationen eine Reduzierung des Prognosefehlers. Aufgrund der Natur der Zeitreihen ist die Anwendung des Holt-Winters-Verfahrens als Prognosemethode nicht geeignet, da die Erkennung von Trend- und Saisonkomponente versagt. Die Verbesserung der Prognose durch Metaparameterwartung ist effektiv, da

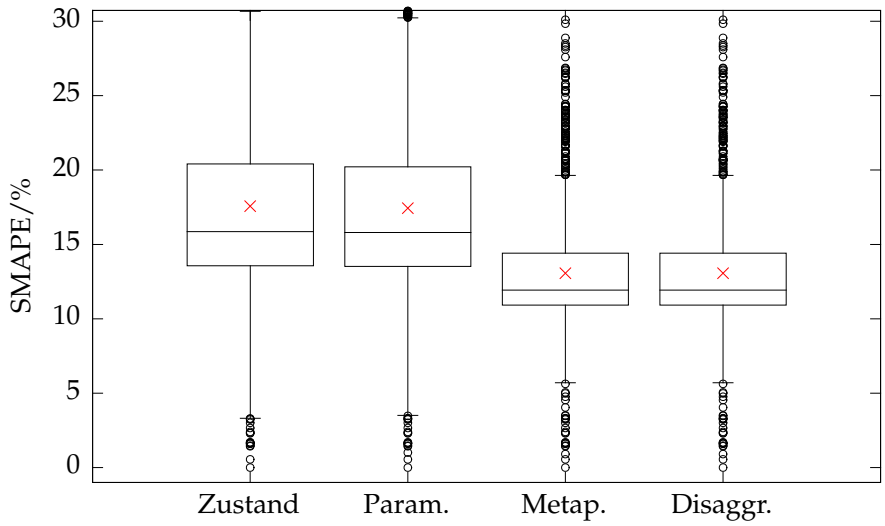


Abbildung 3.20: Prognosefehler für den Datensatz **Wind**

die Modelle vom Prognosemodell nach Holt (Exponentielles Glätten zweiter Ordnung, im System voreingestellt) auf das Exponentielle Glätten erster Ordnung wechselt. Überdies ist der Glättungsparameter α sehr hoch, sodass nahezu der vergangene Messwert als nächster Prognosewert angenommen wird.

Aufgrund der Granularität von zehn Minuten ist der Prognosefehler, der sich aus Ein-Schritt-Prognosen zusammensetzt, sehr gering. Spätestens bei einer Prognose für einen längeren Zeitraum zeigt sich, dass das Holt-Winters-Verfahren für diesen Datensatz keine gute Prognose ermöglicht. Die Wartung der Ableitungsregeln konnte sich nicht durchsetzen, da das Exponentielle Glätten erster Ordnung stets überlegen war.

3.9 ZUSAMMENFASSUNG

Das Kapitel untersucht die Integration von Ableitungsmodellen in das Prognosesystem F²DB. Ableitungsmodelle können mithilfe der Analogie zum Datenwürfel im multidimensionalen Raum definiert werden. Der Ansatz wird zur Erweiterung des Modellindexes zur automatischen Bestimmung von Ableitungsregeln genutzt.

Die Wartung der Ableitungsregeln wird in die Wartungskomponente integriert, sodass Ableitungsmodelle direkte Prognosemodelle automatisch ersetzen können. Die Wartung der Ableitungsgewichte wird in Analogie zur Zustandswartung betrachtet: Sie ermöglicht die regelmäßige Anpassung von Ableitungsmodellen bei Fortschreibung der Zeitreihe. Da sie nicht zeitaufwändig ist, findet sie mit der Zustandswartung statt.

Die Evaluation zeigt, dass direkte Prognosemodelle durch Ableitungsmodelle ersetzt werden können und die Prognosegenauigkeit erhalten bleibt bzw. geringfügig besser wird. Zudem kann die Anzahl direkter Prognosemodelle reduziert werden; dies spart Wartungsoperationen ein.

4 NUTZERDEFINIERTE GENAUIGKEITSKLASSEN

Das folgende Kapitel stellt nutzerdefinierte Genauigkeitsklassen vor, mit denen das Prognosesystem besser auf Anforderungen des Nutzers an Prognoseanfragen eingeht. Bisher wurden Wartungsoperationen vom System angefordert, wenn Schwellwerte überschritten wurden. Genauigkeitsklassen ermöglichen dem Nutzer nun, Wartungsoperationen anfragespezifisch anzufordern. Dadurch kann er beeinflussen, mit welcher Genauigkeit eine Prognose vorzuliegen hat und wieviel Verzögerung der Anfrage akzeptiert wird.

Abschnitt 4.1 gibt einen Überblick über die wichtigsten Nutzeranforderungen an eine Prognose. Abschnitt 4.2 nennt Möglichkeiten aus der Literatur, um die Prognosegenauigkeit zu messen. Dies schließt mit einer Diskussion ab, welches Maß für die Genauigkeit ausgewählt wird. Abschnitt 4.3 stellt das Konzept der nutzerdefinierten Genauigkeitsklasse vor. Abschnitt 4.4 geht auf die Implementierung im Prototypen F²DB ein. In Abschnitt 4.5 wird der Prototyp anhand reeller Daten evaluiert und bewertet. Abschnitt 4.6 schließt das Kapitel mit einer Zusammenfassung ab.

4.1 ÜBERBLICK ÜBER NUTZERANFORDERUNGEN

Die Literatur zählt zahlreiche Anforderungen auf, die Nutzer an ein Prognosesystem stellen. In den von Yokuma und Armstrong [YA95] untersuchten Studien stellt sich die Prognosegenauigkeit als wichtigste Anforderung an Prognosesysteme heraus. Als zweite wichtige Forderung benennen die Autoren die Anfrageverzögerung. Das vorliegende Kapitel konzentriert sich auf diese zwei Forderungen.

Die *Prognosegenauigkeit* ist die Anforderung an ein Modell, eine genaue Prognose der Zeitreihe zu ermöglichen. Im folgenden wird die Prognosegenauigkeit immer durch das Ziel, den Prognosefehler zu minimieren, behandelt. Die *Anfrageverzögerung* ist die Anforderung an das System, die Rechenzeit für die Erstellung der Prognose zu minimieren. Diese Ziele müssen je nach Nutzeranforderung abgewogen werden, da sie sich widersprechen: Verlangt ein Nutzer nach einer hohen

Prognosegenauigkeit, verzögert sich die Rückgabe des Prognosemodells wegen durchzuführender Wartungsoperationen. Erwartet der Nutzer hingegen eine geringe Verzögerung, so wird das Prognosemodell zeitnah zurückgegeben und Wartungsoperationen ignoriert. Beide Anforderungen müssen messbar sein, sodass sie vom Nutzer definiert werden können. Die Verzögerung ergibt sich durch die Anfragezeit des Nutzers. Für die Prognosegenauigkeit wird unterschiedlich gemessen; der folgende Abschnitt stellt zwei Möglichkeiten vor.

4.2 MESSUNG DER PROGNOSEGENAUIGKEIT

Die Prognosegenauigkeit wird durch den Prognosefehler gemessen, vgl. [Kü12]. Üblich ist die Technik der Punktprognose, die den Prognosefehler über die Abweichung von Mess- und Prognosewert ermittelt. Sie wurde bereits in der Modellschätzung und -evaluation eingesetzt, s. Abschnitt 2.1 (Seite 15). Eine weitere Evaluationstechnik ist die Intervallprognose $[\hat{x}_{t,u}, \hat{x}_{t,o}]$. Sie drückt den Prognosefehler zu einer vorgegebenen Überdeckungswahrscheinlichkeit $1 - \alpha$ mit $P(x_t \in [\hat{x}_{t,u}, \hat{x}_{t,o}]) = 1 - \alpha$ aus. Beide Techniken werden in den nachfolgenden Unterabschnitten erläutert.

Punktprognose

Durch das Fortschreiben der Zeitreihe ergibt sich eine Sequenz an Paaren $(x_T, \hat{x}_{T-1,1})$, d. h. ein Messwert und der zugehörige Ein-Schritt-Prognosewert. Durch Mittelwertbildung mit einem *Fehlermaß* ergibt sich eine skalare Größe, die den Prognosefehler der Punktprognose ausdrückt, vgl. [Kü12]. Ein Mittelwert hat eine höhere Aussagekraft als ein einzelner Prognosefehler, muss aber bezüglich seines Einsatzes passend gewählt werden. Tabelle 4.1 stellt ausgewählte Fehlermaße vor. Sie werden in zwei Gruppen eingeteilt: in absolute und relative Fehlermaße.

Die *absoluten Fehlermaße*, zu denen der mittlere absolute Fehler (MAE) und der mittlere quadratische Fehler (MSE) gehören, setzen den Fehler nicht ins Verhältnis zum Messwert. Dadurch sind sie nur in Bezug auf eine Zeitreihe einsetzbar. Sie sind symmetrisch, d. h. sie beachten negative und positive Abweichung mit gleichem Gewicht. MSE ist überproportional: Während die Verdoppelung der Abweichung $e_t = x_t - \hat{x}_t$ zu einer Verdoppelung des absoluten Fehlers $|x_t - \hat{x}_t|$ führt, verursacht sie eine Vervierfachung des quadratischen Fehlers $(x_t - \hat{x}_t)^2$. Dadurch werden Ausreißer stärker gewichtet.

Die *relativen Fehlermaße* setzen die Abweichung ins Verhältnis zum Messwert. Dadurch sind sie auf absoluten Skalen darstellbar und ermöglichen eine Vergleichbarkeit von Fehlern verschiedener Zeitreihen. Zu ihnen gehören der mittlere absolute prozentuale Fehler (MAPE) bzw. der symmetrische mittlere absolute prozentuale Fehler (SMAPE).

Das Fehlermaß MAPE ist symmetrisch, besitzt mit dem Wert 0 % eine untere Grenze, jedoch keine obere Grenze. Von Nachteil ist außerdem, dass bei einem Messwert $x = 0$ der MAPE nicht definiert ist.

SMAPE ermöglicht den Ausdruck der mittleren prozentualen Abweichung der Prognosewerte von den Messwerten im begrenzten Intervall [0 %, 200 %]. Daher ist er für den Einsatz als zeit-

Tabelle 4.1: Ausgewählte Fehlermaße nach [Kü12]

Maß	Operation	Absolutes (A) bzw. relatives (R) Maß	Proportionale (P) oder überproportionale (Ü) Fehlergewichtung
Mittlerer absoluter Fehler	$\text{MAE} = \frac{1}{J}\sum_{j=1}^{J}\lvert x_j - \hat{x}_j\rvert$	A	P
Mittlerer quadratischer Fehler	$\text{MSE} = \frac{1}{J}\sum_{j=1}^{J}(x_j - \hat{x}_j)^2$	A	Ü
Mittlerer absoluter prozentualer Fehler	$\text{MAPE} = 100 \cdot \frac{1}{J}\sum_{j=1}^{J}\lvert\frac{x_j-\hat{x}_j}{x_j}\rvert$	R	P
Symmetrischer mittlerer absoluter prozentualer Fehler	$\text{SMAPE} = 100 \cdot \frac{1}{J}\sum_{j=1}^{J}\frac{2\cdot\lvert x_j-\hat{x}_j\rvert}{\lvert x_j\rvert+\lvert\hat{x}_j\rvert}$	R	P

reihenunabhängiges Fehlermaß geeignet. Von Nachteil ist, dass positive und negative Prognosefehler unterschiedlich bewertet werden, vgl. [GL99]. Je größer der Prognosefehler ist, desto weniger symmetrisch ist der SMAPE. Um negative Mess- und Prognosewerte zuzulassen, wird die Formulierung des SMAPE von [CY04] verwendet, die um den Faktor 100 für die prozentuale Darstellung ergänzt wird.

Die Fehlermaße MAE, MSE, MAPE und SMAPE haben verschiedene Vor- und Nachteile: MAE und MSE sind als absolute Fehlermaße zwar sehr genau, aber spezifisch für eine Zeitreihe. Eine Genauigkeitsklasse sollte unabhängig von Zeitreihen formuliert werden, weshalb diese Maße nicht in Frage kommen. MAPE als ein relatives Fehlermaß ist nach oben unbeschränkt. Dadurch ist eine geeignete Definition eines Schwellwerts nicht möglich. Zudem ist er für einen Messwert $x = 0$ nicht definiert.

Aufgrund der Unabhängigkeit des SMAPE von einer Zeitreihe und dem beschränkten Intervall wird dieses Fehlermaß für den Ausdruck der Punktprognose gewählt. Zudem ist es für beliebige Mess- und Prognosewerte definiert (für $x = \hat{x} = 0$ ist $SMAPE = 0\ \%$). Ein Nutzer kann anschaulich den fehlerbasierten Schwellwert als SMAPE angeben, sodass Prognosemodelle mit einem Fehler über dem Schwellwert zu warten sind, bevor sie für eine Anfrage genutzt werden.

Illustriert wird die Punktprognose an der Zeitreihe **Beschäftigung** aus [YC90], s. Abbildung 4.1. Die Zeitreihe besteht aus monatlichen Messwerten von 1970 bis 1979 mit Trend- und Saisonkomponente. Auf dem Trainingszeitraum 1970/1971 wird ein Prognosemodell erstellt und anschließend unter Nutzung von Wartungsoperationen bis 1980 fortgeschrieben. Im März 1978 findet die letzte Modellparameter- oder Metaparameterwartung statt und das Prognosemodell wird neu geschätzt. Die Prognose zwischen März 1978 und Dezember 1979 sind Ein-Schritt-Prognosewerte, vgl. Abschnitt 2.1 (Seite 15). Deren Prognosefehler werden als SMAPE zusammengefasst. Bei einer Prognoseanfrage zum Zeitpunkt Januar 1980 kann der Nutzer den Fehler akzeptieren oder ablehnen: Letzteres führt zu einer erneuten Modellschätzung, die die Messwerte von März 1978 bis Dezember 1979 berücksichtigt.

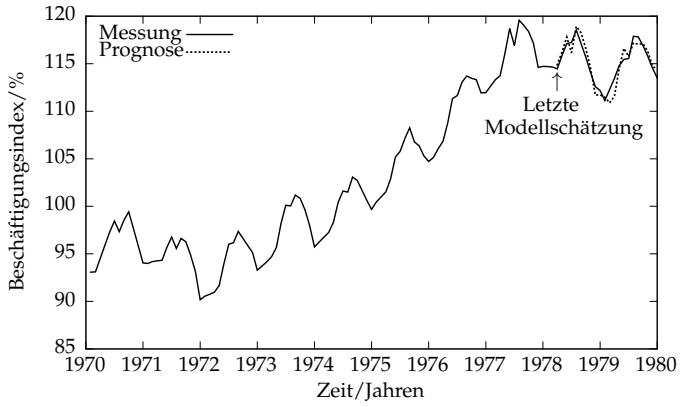


Abbildung 4.1: Punktprognose der Zeitreihe **Beschäftigung**

Intervallprognose

Durch Intervallprognose ergibt sich die obere und untere Grenze eines *Prognoseintervalls* $[\hat{x}_{t,u}, \hat{x}_{t,o}]$. Mit einer Wahrscheinlichkeit von $1 - \phi$ befindet sich der Messwert innerhalb dieses Intervalls, wobei ϕ vom Nutzer anzugeben ist.

Für das Holt-Winters-Verfahren erarbeiteten Chatfield und Yar Formeln für das Prognoseintervall bei additiver und multiplikativer Saisonkomponente [YC90, CY91]. Für die Ein-Schritt-Prognose lautet es in beiden Fällen:

$$\hat{x}_{T,1} \pm Z_{\phi/2} \sqrt{Var(e_T(1))}, \tag{4.1}$$

wobei $Z_{\phi/2}$ für das $\frac{\phi}{2}$-Quantil der Normalverteilung steht. Bei einer Überdeckungswahrscheinlichkeit $\phi = 0,05$ lautet $Z_{\phi/2} = Z_{0,025} = -1,96$. Die Varianz des Fehlers $Var(e_T(1)) = \sigma_e^2$ ist nicht bekannt, hierfür wird der Maximum-Likelihood-Schätzer eingesetzt, vgl. [Hyn08]:

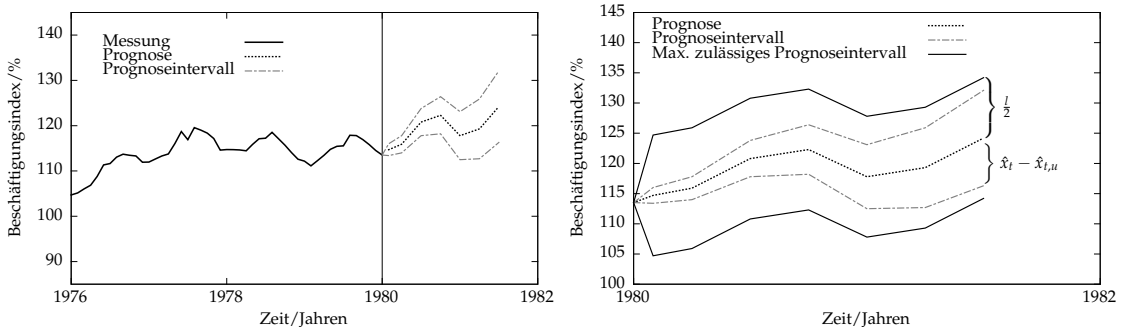
$$s^2 = \frac{1}{T} \sum_{t=1}^{T} (x_t - \hat{x}_{t-1,1})^2 \tag{4.2}$$

Abbildung 4.2 illustriert die Intervallprognose an der Zeitreihe **Beschäftigung**. Die Zeitreihe besteht aus monatlichen Messwerten von 1970 bis 1979. Abbildung 4.2a zeigt die Zeitreihe ab 1976. Für die Jahre 1980 und 1981 wird eine Prognose mit einem 95 %-Prognoseintervall erstellt, dessen Ober- und Untergrenze eingezeichnet sind.

Auswahl eines Maßes für die Prognosegenauigkeit

Die zwei vorgestellten Evaluationstechniken, Punktprognose und Intervallprognose, repräsentieren die Prognosegenauigkeit auf unterschiedliche Art. In diesem Abschnitt wird diskutiert, welches Maß für die Definition von Genauigkeitsklassen zum Einsatz kommt. Für die Punktprognose wurde der SMAPE zum Ausdruck der Prognosegenauigkeit gewählt.

Die Intervallprognose kann ebenso für die Formulierung der Prognosegenauigkeit genutzt werden. Der Nutzer muss hierfür zwei Parameter angeben: die Überdeckungswahrscheinlichkeit ϕ



(a) Prognoseintervall **(b)** Maximal zulässiges Prognoseintervall $l = 20$

Abbildung 4.2: Intervallprognose der Zeitreihe **Beschäftigung** mit $\phi = 0.05$

und ein maximal zulässiges Prognoseintervall $[d_u, d_o]$, das als Abstand $l = d_o - d_u$ formuliert wird. Abbildung 4.2b verdeutlicht dies an der Zeitreihe **Beschäftigung**. Die Prognose im Zeitraum 1980/1981 ist umgeben vom Prognoseintervall. Das maximal zulässige Prognoseintervall wird vom Nutzer durch den Abstand l angegeben und fordert eine Beschränkung des Prognoseintervalls.

Wenn das Prognoseintervall die Eigenschaft erfüllt, im gesamten Horizont H kleiner als das maximal zulässige Intervall zu sein, d. h.

$$\forall t \in H : \frac{l}{2} \geq \hat{x}_t - \hat{x}_{t,u}$$

so wird das Prognosemodell akzeptiert. Erfüllt es diese Eigenschaft nicht, werden Wartungsoperationen angestoßen. In der Abbildung 4.2b ist die Eigenschaft erfüllt. Durch diesen Ansatz kann Intervallprognose für die Formulierung genutzt werden. Problematisch an ihm ist, ähnlich wie bei MAE und MSE, dass der Abstand l des maximal zulässigen Prognoseintervalls ein absolutes Maß und somit von der Zeitreihe abhängig ist. Außerdem wurde nicht für alle Prognosemethoden und Metaparameter ein Prognoseintervall beschrieben. Somit kann dieses Verfahren nicht allgemein eingesetzt werden.

In der Folge wird daher die Punktprognose als Genauigkeitsmaß eingesetzt.

4.3 ENTWURF DER GENAUIGKEITSKLASSEN

Der folgende Abschnitt untersucht, wie eine nutzerdefinierte Genauigkeitsklasse für die zwei Anforderungen der Prognosegenauigkeit und der Anfrageverzögerung zu gestalten ist und wie sie bei einer Prognoseanfrage eingesetzt wird.

Das existierende System sieht eine Modellevaluation beim Fortschreiben der Zeitreihe vor. Beim Überschreiten des fehler- oder zeitbasierten Schwellwerts werden Wartungsoperationen angefordert, die eine Reduzierung des Prognosefehlers für die vergangenen Messwerte ermöglichen. Die Schwellwerte sind vom System festgelegt, sodass Wartungsoperationen von Prognosemodellen ohne Prognoseanfrage angefordert werden.

Es wird eine *nutzerdefinierte Genauigkeitsklasse* vorgeschlagen, die für ein angefragtes Prognosemo-

dell die Schwellwerte des Systems durch Schwellwerte des Nutzers ersetzt. So kann der Nutzer entscheiden, ob das Prognosemodell seinen Anforderungen entspricht. Hat das Prognosemodell in der Vergangenheit einen zu großen Fehler, wird es für die Prognoseanfrage erneut gewartet. Dem Nutzer stehen drei Genauigkeitsklassen zur Verfügung:

- *Keine Wartung*: Der fehler- und zeitbasierte Schwellwert des Prognosemodells werden ignoriert, um die Prognoseanfrage nahezu unverzögert zurückzugeben. Abgesehen von der Zustandswartung werden ausstehende Wartungsoperationen ignoriert. Die Verzögerung durch Wartung ist niedrig, der Prognosefehler ist hoch. Wenn der Nutzer eine Anfrage keiner Genauigkeitsklasse zuordnet, fällt sie automatisch in diese Klasse.

- *Beste Wartung*: Für das Prognosemodell werden alle Wartungsoperationen durchgeführt, um eine Verbesserung der Prognosegenauigkeit zu erzielen. Die Verzögerung durch Wartung ist hoch, der Prognosefehler ist niedrig.

- *Nutzerdefinierte Wartung*: Diese Klasse bietet dem Nutzer eine Abstufung der Anforderungen. Er legt den fehler- und zeitbasierten Schwellwert fest, sodass eine Wartungsoperation nur erfolgt, wenn einer von beiden überschritten wurde. Ebenso legt der Nutzer die Auswahl der Wartungsoperationen und deren Parameter fest. Dadurch kann er beeinflussen, welche Wartungsoperationen geeignet sind. Wurden die Metaparameter bspw. durch einen Experten festgelegt, so kann das System diese Wartungsoperation überspringen. Ebenso gilt, dass die Wartung der Ableitungsregeln nicht durchgeführt werden muss, wenn der Nutzer dies nicht wünscht. Eine Zustandswartung wird stets durchgeführt, um neu eingefügte Messwerte zu berücksichtigen. Damit liegt diese Genauigkeitsklasse zwischen den Fällen „Keine Wartung“und „Beste Wartung“.

Die Entwicklung von Prognosefehler und Anfrageverzögerung ist keine lineare, da bspw. die Metaparameterwartung um einen Faktor länger dauert als die Parameterwartung. Dieser Faktor spiegelt wider, dass die Modellschätzung bei der Metaparameterwartung bis zu 14-mal durchgeführt wird. Dies sind die möglichen Kombinationen von Metaparametern, die unterstützt werden, vgl. [Keg14]. Ebenso kann die Hinzunahme einer Wartungsoperation keine lineare Verbesserung der Prognosegenauigkeit zusichern.

4.4 IMPLEMENTIERUNG

Nutzerdefinierte Genauigkeitsklassen mit fehler- und zeitbasiertem Schwellwert werden in F²DB implementiert. Abbildung 4.3 zeigt das Flussdiagramm für eine Prognoseanfrage mit Genauigkeitsklasse, das eine Erweiterung zum Vorzustand (Abbildung 2.2, Seite 21) darstellt. Der Daten- und Kontrollfluss wird in den folgenden Absätzen erläutert.

Im Rahmen der *Modellnutzung* stellt der *Nutzer* eine *Prognoseanfrage*. Über den *Modellindex* wird der *Modellpool* nach einem *Prognosemodell*, das zur Prognoseanfrage passt, durchsucht. Wenn es gefunden wurde, muss zugesichert werden, dass die Zustandswartung abgeschlossen ist. Wäre dies nicht der Fall, gäbe das Prognosemodell veraltete Prognosen aus. Daher werden im Puffer *Anfragen* alle Prognoseanfragen gesammelt und so der *Modellwartung* übergeben.

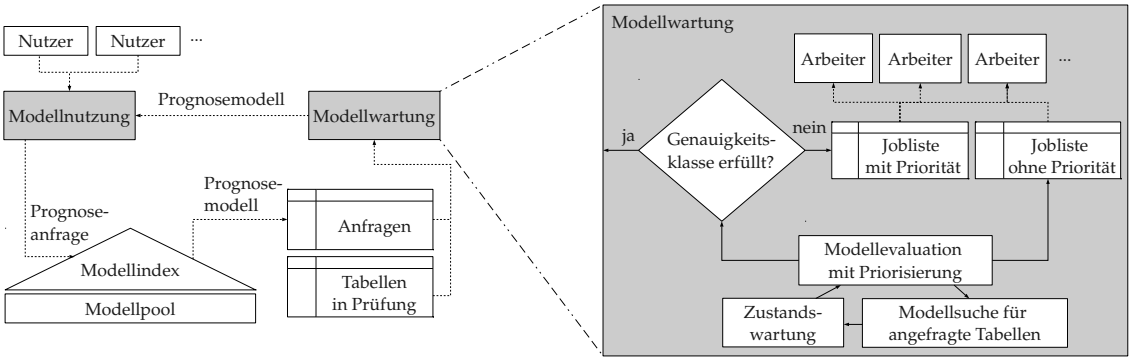


Abbildung 4.3: Erweiterung des Prototyps F²DB für die nutzerdefinierte Genauigkeitsklasse

Die Wartungskomponente setzt unmittelbar nach Eintreffen einer Anfrage ein. Anstatt alle *Tabellen in Prüfung* zu bearbeiten, werden nur Tabellen fortgeschrieben, für die eine Anfrage vorliegt. Daher findet die *Modellsuche*, d. h. die Zuordnung von Tupeln zu Prognosemodellen, nur für angefragte Tabellen statt.

Ferner wird die *Zustandswartung* für alle Prognosemodelle dieser Tabelle durchgeführt. Hier ist eine Beschränkung auf angefragte Prognosemodelle nicht sinnvoll: Die Modellsuche müsste während einer Anfrage Tupel mehrmals untersuchen, obwohl ein betroffenes (nicht angefragtes) Prognosemodell bereits bekannt ist. Das hätte eine Verzögerung der Modellwartung zur Folge.

Die anschließende *Modellevaluation mit Priorisierung* prüft unter Berücksichtigung der Genauigkeitsklasse, ob Wartungsoperationen durchzuführen sind. Diese Aktivität ist iterativ: Sie prüft nacheinander alle Prognosemodelle auf ihre Fehler und fügt zu wartende Prognosemodelle in die Jobliste ein. Um angefragte Prognosemodelle schneller zu verarbeiten, wird eine Priorisierung eingerichtet: Zunächst prüft die Modellevaluation alle Prognosemodelle mit Anfrage (mit Priorität), anschließend alle Prognosemodelle ohne Anfrage (ohne Priorität). Entscheidend ist hierbei, ob die Prognosegenauigkeit bei den vergangenen Ein-Schritt-Prognosewerten eingehalten wurde. Ist die Genauigkeitsklasse erfüllt, erfolgt die Rückgabe des Prognosemodells. Ist dies nicht der Fall, wird ein Job für die Wartungsoperation erstellt und in die *Jobliste* eingefügt.

Auch Jobs werden unterteilt in Jobs mit Anfrage (mit Priorität) und Jobs ohne Anfrage (ohne Priorität). Jobs mit Anfrage werden in die *Jobliste mit Priorität* eingefügt. Arbeiter bedienen bevorzugt Jobs mit Priorität, sodass die Bedienung schneller erfolgt. Da die Jobs in der gleichen Reihenfolge bedient werden, wie sie eingefügt wurden (First In - First Out), erfolgt eine Zweiteilung: Dies vermeidet das Durchsuchen der Jobliste nach Priorität.

Arbeiter bedienen die Joblisten und führen die Jobs aus. Sollte die Genauigkeitsklasse trotz Wartung nicht erfolgreich sein, d. h. konnte der fehlerbasierte Schwellwert nicht unterschritten werden, so wird der Nutzer darüber informiert.

Nach Rückgabe des Prognosemodells erfolgt die Ermittlung der Prognose, die dem Nutzer ausgegeben wird. Damit endet die Prognoseanfrage mit Genauigkeitsklasse.

Abbildung 4.4 zeigt eine Anfrage mit nutzerdefinierter Genauigkeitsklasse am Beispiel von **Tourismus**. Das Schlüsselwort `MAINTENANCE` ergänzt die Prognoseanfrage, wobei zwischen `BEST`, `OFF` und `CUSTOM` gewählt werden kann. Bei letzterem gibt der Nutzer die Anforderungen mit.

Hierbei setzt der Parameter MAX_ERROR den fehlerbasierten Schwellwert als SMAPE, im Beispiel 24 %. Der Parameter MAX_TIME setzt den zeitbasierten Schwellwert, sodass im Beispiel nach spätestens 16 Tupeln eine Wartung erfolgen muss. Mit OPERATIONS werden zulässige Wartungsoperationen angegeben, in diesem Fall sind Modellparameter- und Metaparameterwartung erlaubt.

```
1 SELECT zeit, messwert FROM Tourismus
2 WHERE region = 'Goulburn' AND anlass = 'Freunde'
3 GROUP BY zeit ORDER BY zeit NUMBER 8 STORAGE modelindex
4 MAINTENANCE CUSTOM (
5         MAX_ERROR 24, MAX_TIME 16
6 ) OPERATIONS (
7         MODELPARAMETER, METAPARAMETER
8 );
```

Abbildung 4.4: Prognoseanfrage mit Genauigkeitsklasse für den Datensatz **Tourismus**

Mit Bezug auf den zeit- und fehlerbasierten Schwellwert sind die Genauigkeitsklassen anwendbar bei Prognosemodellen, die nicht vor kurzem neu geschätzt wurden, da nur für sie ist die Durchführung von Wartungsoperationen sinnvoll ist. Für Prognosemodelle, deren Komponenten vor weniger als error_time_elapsed_min Messwerten neu geschätzt oder gesetzt wurden, wird keine erneute Wartung durchgeführt.

4.5 EVALUATION

Der folgende Abschnitt evaluiert das Konzept der nutzerdefinierten Genauigkeitsklasse, mit der ein Nutzer auf die Prognosegenauigkeit und die Verzögerung seiner Prognoseanfrage eingehen kann. Unterabschnitt 4.5.1 evaluiert zunächst die Optimierungen an der Wartungskomponente, um Prognoseanfragen priorisiert zu bedienen. Unterabschnitt 4.5.2 vergleicht die Genauigkeitsklassen hinsichtlich des Prognosefehlers und der Anfrageverzögerung. Die Zeit ist für die nachfolgende Evaluation in Echtzeit angegeben. Sie wird mit der Funktion gettimeofday im Prototyp bestimmt.

4.5.1 Optimierungen zur Minimierung der Anfrageverzögerung

Die Implementierung sieht zwei Optimierungen der Wartungskomponente vor, um die Prognoseverzögerung zu reduzieren: Zum einen werden nur Tabellen untersucht, für die eine Prognoseanfrage vorliegt, zum anderen werden angefragte Prognosemodelle priorisiert bedient. Die Auswirkungen der beiden Optimierungen sind Gegenstand nachfolgender Untersuchung. Die Versuche werden auf einem System mit 64 Prozessoren à 2,13 GHz und 128 GiB durchgeführt.

Beschränkung auf Tabellen mit Prognoseanfragen

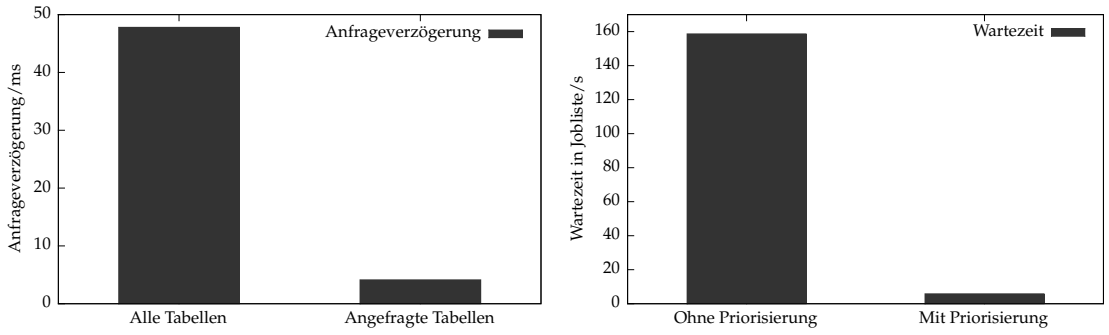
Nachfolgend wird die Optimierung untersucht, mit der die Modellsuche bei Prognoseanfragen nur angefragte Tabellen auf Änderungen prüft. Dies ist dadurch motiviert, dass in einem Mehrbenutzersystem ein Nutzer Anfragen an das System richtet, während ein anderer Nutzer Zeitreihen fortschreibt, die von den Anfragen unberührt sind. In diesem Fall sollen die Prognoseanfragen nicht verzögert werden.

Versuchsaufbau und -durchführung Ein Nutzer A stellt nacheinander Prognoseanfragen an die Zeitreihen vom Datensatz **Wind**. Alle 1361 Prognosemodelle werden gleichverteilt angefragt. Die Prognosemodelle wurden auf zwei Jahren (2004/2005) trainiert und werden nicht fortgeschrieben. Als Genauigkeitsklasse wird `MAINTENANCE OFF` gewählt, wodurch die geringste Anfrageverzögerung erwartet wird.

Ein Nutzer B führt einen zweiten Datensatz mit Prognosemodellen. Dieser Datensatz wird mit Messwerten fortgeschrieben. Es werden keine Anfragen an den Datensatz gerichtet. Für das Experiment wird ein Duplikat vom **Wind**-Datensatz eingesetzt, die Prognosemodelle werden auf den Jahren 2004/2005 trainiert und für das Jahr 2006 fortgeschrieben. Damit ist die Anzahl eingefügter Tupel ca. 70 Millionen, wodurch eine Verzögerung der Modellsuche, der Zustandswartung und der Modellevaluation zu erwarten ist, wenn die o. g. Optimierung nicht eingesetzt wird. Es werden für das Experiment keine komplexen Wartungsoperationen durchgeführt, die Angabe von Schwellwerten entfällt.

Ergebnis Abbildung 4.5 zeigt die Ergebnisse der Optimierungen. Abbildung 4.5a vergleicht die Wartungskomponente ohne bzw. mit Berücksichtigung angefragter Tabellen („Alle Tabellen“ bzw. „Angefragte Tabellen“). In dieser Versuchsanordnung kann die Anfrageverzögerung durch die Optimierung von 47,8 ms auf 4,1 ms reduziert werden. So werden ohne Optimierung im Mittel 21 Prognoseanfragen pro Sekunde bedient, während es mit Optimierung im Mittel 244 Prognoseanfragen pro Sekunde sind. Die Zustandswartung der nicht angefragten Tabelle von Nutzer B wird zu einem Zeitpunkt durchgeführt, zu dem eine Anfrage für die Tabelle vorliegt oder keine Anfrage für alle Tabellen vorliegen.

Interpretation Die Optimierung zeigt, dass die Anfrageverzögerung durch Optimierung der Wartungskomponente minimiert werden kann. Der Fokus liegt dabei auf der Bearbeitung angefragter Tabellen, wie sie der Nutzer A stellt. Für nicht angefragte Tabellen (Nutzer B) entsteht kein Mehraufwand: sie werden bearbeitet, wenn keine Anfragen am System anliegen. Wenn Nutzer B hingegen eine Prognoseanfrage an seine Tabelle stellt, entsteht durch das Fortschreiben der Zeitreihen und der nachzuholenden Zustandswartung eine Verzögerung.



(a) Zustandswartung angefragter Tabellen **(b)** Priorisierung von Jobs

Abbildung 4.5: Minimierung der Anfrageverzögerung

Priorisierung von Jobs mit Prognoseanfrage

Die folgende Optimierung betrifft weniger die Genauigkeitsklasse `MAINTENANCE OFF`, sondern alle Genauigkeitsklassen, die Jobs an Arbeiter weitergeben. Nach der Modellevaluation werden die Jobs in die Jobliste mit Priorität eingefügt. Das folgende Szenario untersucht, ob die Wartezeit in der Jobliste durch die Priorisierung reduziert werden kann.

Versuchsaufbau und -durchführung Es werden alle 1361 Prognosemodelle des Datensatzes **Wind** erstellt und auf den Jahren 2004/2005 trainiert. Der Nutzer fragt die Prognosemodelle gleichverteilt an. Dabei wird die Genauigkeitsklasse `MAINTENANCE CUSTOM (MAX_ERROR 10, MAX_TIME 26352) OPERATIONS (MODELPARAMETER)` gefordert, d. h. es wird bei Überschreiten des Fehlers von 10 % oder spätestens nach einem halben Jahr eine Modellparameterwartung durchgeführt.

Parallel zu den Prognoseanfragen des Nutzers werden die Zeitreihen für das Jahr 2006 fortgeschrieben, wobei das System Wartungsoperationen für nicht angefragte Prognosemodelle anfordert. Hierbei sind alle Wartungsoperationen zugelassen. Die Schwellwerte sind in Tabelle 3.5 (Seite 59) angegeben. Der Schwellwert `error_time_elapsed_min` gilt auch für die Prognoseanfragen.

Ergebnis Abbildung 4.5b zeigt die Auswirkungen der Priorisierung auf die Wartezeit. Die Wartezeit ist die mittlere Aufenthaltszeit eines Jobs (für ein angefragtes Prognosemodell) in der Jobliste. Der Fall „Ohne Priorisierung“ trifft keine Unterscheidung zwischen angefragten und nicht angefragten Prognosemodellen, sodass deren Jobs in einer Jobliste eingefügt und gleichermaßen nach dem Prinzip First In - First Out bedient werden. Beim Fall „Mit Priorisierung“ werden Jobs mit Prognoseanfrage in eine eigene Jobliste eingetragen, die bevorzugt durch die Arbeiter bedient wird. Diese Priorisierung wirkt sich auf die Wartezeit aus: sie kann von 158,6 s auf 5,8 s reduziert werden.

Interpretation Durch die Einführung der Priorisierung von Jobs wird die Prognoseverzögerung reduziert, da die Wartezeit in der Jobliste einen entscheidenden Anteil an der gesamten Prognoseverzögerung ausmacht. Dass die Wartezeit ohne Priorisierung im Minutenbereich liegt, ist am Versuchsaufbau zu erklären: Die Metaparameterwartung ist für nicht angefragte Prognosemodelle zugelassen. Diese Wartungsoperation prüft eine Dekomposition der Zeitreihe in Trend- und Saisonkomponente, die aufgrund der Länge von ca. 150.000 Tupeln pro Zeitreihe sehr zeitaufwändig ist.

4.5.2 Prognosefehler und Anfrageverzögerung im Vergleich

Das nachfolgende Experiment untersucht den Zusammenhang zwischen Prognosefehler und Anfrageverzögerung bei verschiedenen Genauigkeitsklassen. Hierbei werden neben „Keine Wartung“ und „Beste Wartung“ auch nutzerdefinierte Klassen definiert, die eine Abstufung ermöglichen.

Versuchsaufbau und -durchführung

Der Datensatz **Tourismus** wird wie in Unterabschnitt 3.8.2 (Seite 56) vorbereitet. Für einen Trainingszeitraum von vier Jahren werden 340 direkte Prognosemodelle erstellt; hierin sind die Prognosemodelle für aggregierte Zeitreihen enthalten. In den zehn folgenden Jahren werden die Zeitreihen fortgeschrieben, wobei alle Wartungsoperationen erlaubt sind. Für die Wartung der Ableitungsregeln wird die Disaggregation eingesetzt, die eine Reduzierung der genutzten direkten Prognosemodelle ermöglicht. Die Schwellwerte für den Datensatz, s. Tabelle 3.4 (Seite 55), sind unverändert.

Schließlich liegen bis Juni 2012 gewartete Prognosemodelle vor. Die Wartung erfolgte aufgrund der Schwellwerte, die das System spezifizierte. Für die letzten zwei Jahre, September 2012 bis Juni 2014, werden alle Prognosemodelle mit nutzerdefinierten Genauigkeitsklassen angefragt. Der Vergleich mit den Messwerten ergibt den Prognosefehler als SMAPE. Dabei werden die Prognosefehler aller Anfragen zusammengefasst. Es handelt sich nicht um Ein-Schritt-Prognosewerte, sondern um Prognosewerte im Horizont von 2 Jahren. Dies soll untersuchen, ob über einen längeren Zeitraum eine Verbesserung der Prognosegenauigkeit möglich ist.

Die zur Wartung benötigte Zeit, die *Prognoseverzögerung*, ist die vergangene Rechenzeit des Nutzerprozesses vom Aufruf der Wartungskomponente, d. h. der Einfügung der Anfrage in den Puffer, bis zur Rückgabe des gewarteten Prognosemodells. Die Rechenzeit zur Aufbereitung und Rückgabe der Ergebnistupel durch die Datenbank wird nicht berücksichtigt, da diese Verzögerung unabhängig von der Genauigkeitsklasse ist.

Fehlerbetrachtung

Während eines Durchlaufs werden alle Prognosemodelle genau einmal angefragt, wobei der Prognosefehler und die Prognoseverzögerung erfasst werden. Bei mehreren Durchläufen sind diese Werte nicht exakt gleich, was wie folgt begründet wird:

- Das Prognosemodell hat nach den vier Jahren Training und zehn Jahren Fortschreiben mit Wartungsoperationen Komponenten mit schwankenden Werten (Metaparameter, Modellparameter, Zustand). Einerseits hängt dies vom verwendeten Optimierer ab und dessen Abbruchkriterien, andererseits von der Länge der Zeitreihe, auf der die Modellschätzung stattfindet. Da die Zeitreihe asynchron fortgeschrieben wird, ist diese Länge und damit die Modellschätzung unterschiedlich.
- Die Prognoseverzögerung wird in Echtzeit gemessen. Wann die Wartungsprozesse ausgeführt werden, hängt vom Betriebssystem ab, sodass sich leichte Schwankungen ergeben. Zudem gibt es einige Prognosemodelle, die bei mehreren Durchläufen verschiedene Wartungsoperationen angefordert haben, wodurch die Verzögerung stark schwankt (zwischen 100 µs und 10 ms). Daher wird der Mittelwert der Zeiten für alle Prognoseanfragen betrachtet, der stabiler ist.

Die Schwankungen sind als zufällige Fehler zu betrachten, für die angenommen wird, dass sie normalverteilt sind. Um aussagekräftige Messwerte sicher zu stellen, werden $n = 10$ Durchläufe

gemittelt und der prozentuale Fehler ermittelt. Bei OFF, C1, C2 werden dafür werden $n = 20$ Durchläufe benötigt. Sei y die Messung, z. B. der Prognosefehler oder die mittlere Prognoseverzögerung. Sei y_i die Messung des i-ten Durchlaufs, so ergibt sich das arithmetische Mittel $\bar{y}$ der Messungen durch

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \tag{4.3}$$

Für die Bestimmung der *Unsicherheit* dieses Mittelwerts wird aus [Hä67] zitiert. Jede einzelne Messung y_i hat den mittleren quadratischen Fehler s, der sich ergibt aus

$$s = \pm \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{x})^2} \tag{4.4}$$

Dieser wird auch als empirische Standardabweichung bezeichnet. Interessant ist jedoch der mittlere quadratische Fehler des Mittelwerts $m_{\bar{y}}$, der die Unsicherheit aller Durchläufe angibt:

$$m_{\bar{y}} = \pm \frac{1}{\sqrt{n}} \cdot s \tag{4.5}$$

Bspw. lautet der SMAPE 38,64 % ± 0,08 % bei der Genauigkeitsklasse „Keine Wartung“. Der *prozentuale Fehler* ist schließlich das Verhältnis von Unsicherheit und Mittelwert $\frac{m_{\bar{y}}}{\bar{y}}$.

Für den Prognosefehler SMAPE lag der prozentuale Fehler für alle nutzerdefinierten Genauigkeitsklassen bei höchstens 0,2 %, für die Prognoseverzögerung lag er bei höchstens 6,1 %. Damit ist die Unsicherheit der Messung hinreichend klein.

Ergebnis

Es werden acht nutzerdefinierte Genauigkeitsklassen ausgewählt, siehe Tabelle 4.2.

Tabelle 4.2: Genauigkeitsklassen für die Evaluation am Datensatz **Tourismus**

Genauigkeitsklasse	Höchste Wartungsoperation	Fehlerbasierter Schwellwert
OFF (Keine Wartung)	Zustand	-
C1	Modellparameter	24 %
C2	Metaparameter	24 %
C3	Ableitung	24 %
C4	Modellparameter	12 %
C5	Metaparameter	12 %
C6	Ableitung	12 %
BEST (Beste Wartung)	Ableitung	-

Abbildung 4.6 zeigt den Prognosefehler und die Prognoseverzögerung für den vorgestellten Datensatz. Auf der horizontalen Achse sind die acht ausgewählten Genauigkeitsklassen abgetragen. Der Prognosefehler ist der SMAPE aller Prognosen. Die Prognoseverzögerung zeigt die mittlere Verzögerung durch Wartungsoperationen pro Prognosemodell.

Zu erkennen ist, dass der Prognosefehler bei steigender Nutzeranforderung fällt. Dabei hat der fehlerbasierte Schwellwert von 24 % keinen nachweisbaren Einfluss, jedoch der Schwellwert von

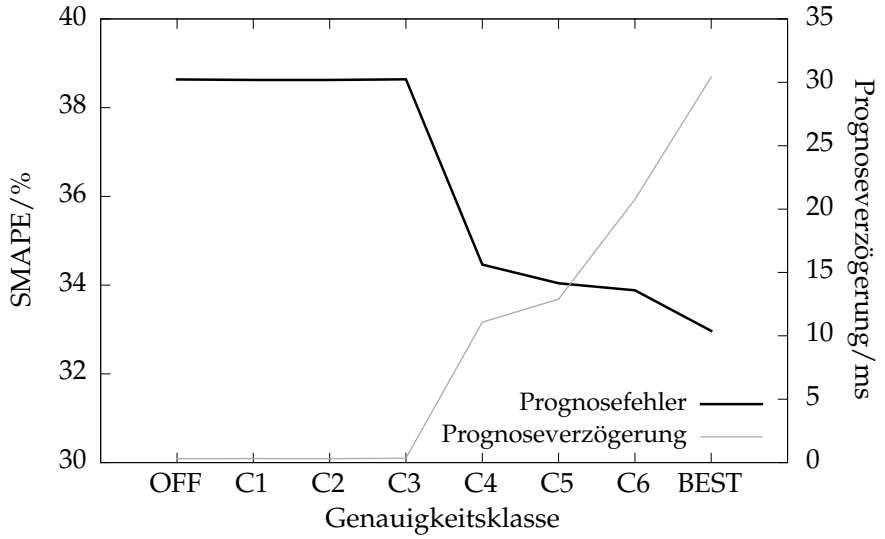


Abbildung 4.6: Evaluation der nutzerdefinierten Genauigkeitsklassen

12 %. Das Hinzufügen einer Wartungsoperation führt insgesamt zur Reduzierung des Prognosefehlers.

Interpretation

Zunächst wird der Prognosefehler betrachtet. Bei einem fehlerbasierten Schwellwert von 24 % (C1, C2, C3) kann im Vergleich zu OFF keine Reduzierung nachgewiesen werden. Dies hängt damit zusammen, dass der Schwellwert gleich dem systembasierten Schwellwert ist: alle Prognosemodelle, die einen größeren Fehler haben, wurden bereits durch automatische Wartungsoperationen geändert. Da dies auch bei OFF zutrifft, sind die Prognosefehler gleich.

Bei einem fehlerbasierten Schwellwert von 12 % (C4, C5, C6) werden zusätzlich auch Prognosemodelle gewartet, deren Fehler unterhalb des Schwellwerts vom System lagen. Dies führt zu einem Abstieg der Kurve. C6 zeigt außerdem, dass die Wartung der Ableitungsregeln kaum eine zusätzliche Reduzierung des Prognosefehlers ermöglicht. Dies wurde bereits in Abschnitt 3.8.2 (Seite 56) untersucht. Bei der Klasse BEST („Beste Wartung“) ist der Abstieg wieder stärker, da hier auch Prognosemodelle unterhalb des Schwellwerts von 12 % gewartet werden.

Die Prognoseverzögerung verhält sich genau entgegen gesetzt. Da während der Klassen OFF, C1, C2, und C3 kaum Wartungsoperationen durch die Anfrage veranlasst werden, fällt die Verzögerung mit rund 0,3 ms pro Prognosemodell bei OFF, C1, C2, C3 gering aus. Der niedrigere Schwellwert führt zu einer höheren Verzögerung: Sie liegt zwischen 11,1 ms bei C4 und 30,4 ms bei BEST.

Der Prognosefehler liegt mit rund 36 % weit über den Schwellwerten vom Nutzer. Das heißt, dass die Schwellwerte zu einer Durchführung von Wartungoperationen und in der Folge zu einer Verbesserung der Prognose führen. Aber der SMAPE der Prognose muss nicht zwingend unterhalb des fehlerbasierten Schwellwerts liegen. Insbesondere bei diesem Datensatz, in dem sehr viele Zeitreihen nur eine schlechte Prognose ermöglichen, fällt der SMAPE im Mittel höher aus.

4.6 ZUSAMMENFASSUNG

Dieses Kapitel führt nutzerdefinierte Genauigkeitsklassen ein. Sie integrieren zwei wichtigen Anforderungen an Prognosesysteme: Genauigkeit und geringe Verzögerung der Prognoseanfrage. Diese Anforderungen sind anfragespezifisch und können vom Nutzer angepasst werden.

Die Charakterisierung von Prognoseanfragen durch diese Klassen ist relevant für die nachfolgende Untersuchung der lastgesteuerten Wartung. Bei unterschiedlicher Systemlast soll die Wartungskomponente verschieden reagieren, um Prognoseanfragen mit geringer Verzögerung zu beantworten. Die Genauigkeitsklassen geben der Komponente Informationen über den Wartungsaufwand.

5 LASTGESTEUERTE WARTUNGSKOMPONENTE

Prognoseanfragen wurden bisher für eine mittlere Systemlast beschrieben, d. h. es standen stets Betriebsmittel zur Verfügung, um die Anfrage zu bedienen. Zudem ermöglicht die Priorisierung, dass das Warten auf einen Arbeiter verringert wird. Bei hoher Systemlast stehen evtl. keine Arbeiter zur Verfügung, die die Bedienung vornehmen. Das folgende Kapitel stellt daher eine laststeuerte Wartungskomponente für den Umgang mit hoher Systemlast vor.

Abschnitt 5.1 erläutert zwei mögliche Repräsentationen der Systemlast. Abschnitt 5.2 stellt den Entwurf der Laststeuerung vor. Abschnitt 5.3 evaluiert die Laststeuerung bei unterschiedlicher Anfragerate. Abschnitt 5.4 fasst die Ergebnisse des Kapitels zusammen.

5.1 REPRÄSENTATION DER SYSTEMLAST

Die Systemlast setzt sich aus verschiedenen Komponenten zusammen. Während einer Wartungsoperation zeichnet sich vor allem eine erhöhte Prozessorauslastung ab. Die Modellschätzung ist sehr zeitaufwändig sowie die Zerlegung der Zeitreihe in Trend- und Saisonkomponente. Bei einer parallelen Ausführung mehrerer Jobs kann dies zu einer hohen Systemlast führen. In der Folge wird die Systemlast als Prozessorauslastung dargestellt. Weitere Betriebsmittel, die sich auf die Systemlast auswirken (Speicherbandbreite, Festplattenzugriff) bleiben zunächst unberücksichtigt, ihr Einfluss ist nachrangig.

Für das Betriebssystem Linux werden zwei mögliche Repräsentationen vorgestellt: die momentane Prozessorauslastung mit `/proc/stat`, s. Unterabschnitt 5.1.1, und die mittlere Systemlast mit `/proc/loadavg`, s. Unterabschnitt 5.1.2. Für das Betriebssystem Windows gibt es eine Formulierung ähnlich zu `/proc/stat`. Da die Implementierung und Evaluation unter Linux stattfindet, wird hierfür auf die Literatur verwiesen, z. B. [Wil06, S. 264].

5.1.1 Prozessorauslastung mit /proc/stat

Unter Linux dient das `/proc`-Dateisystem als Schnittstelle zum Kernel und zur Ablage von Leistungsindikatoren. Die Dateien `/proc/stat` und `/proc/uptime` kommen für die Messung der Prozessorauslastung zum Einsatz. Sie werden dynamisch erzeugt und nicht auf Festplatte abgelegt.

Die erste Datei, `/proc/stat`, gibt die Anzahl aktiver (*total_time*) und inaktiver (*idle*) Takte eines Prozessors aus. Als Takt wird hierfür der Systemzeitmesser f eingesetzt, der in Linux üblicherweise 100 Hz beträgt, vgl. [Pro09]. Die Datei `/proc/uptime` gibt die vergangene Zeit seit Systemstart in Sekunden (*uptime*) aus. Mit diesen Werten lässt sich die *momentane Prozessorauslastung* in einem Zeitintervall $[uptime_1, uptime_2]$ ausdrücken:

$$cpu = 100 \cdot \frac{(total\_time_2 - total\_time_1)/f}{uptime_2 - uptime_1} \tag{5.1}$$

Unter momentaner Auslastung wird die Auslastung in einem vergangenen, kurzen Zeitintervall verstanden. Je länger das Zeitintervall ist, desto eher stellt es eine mittlere Prozessorauslastung dar. Je kürzer es ist, desto höher sind die Schwankungen. Bei mehreren Prozessoren wird die Auslastung gemittelt. Somit ist eine Darstellung der momentanen Prozessorauslastung auf einer Skala von [0 %, 100 %] möglich. Die Wartungskomponente kann diese Information nutzen und darauf reagieren, wie im Abschnitt 5.3 untersucht wird.

Ein Programm, das dieses Verfahren nutzt, heißt *ps*. Die Auslastung der Prozessoren für jeden Prozess *pid* kann durch den Befehl `ps -p pid -o %cpu` abgefragt werden. Intern berechnet das Programm:

$$cpu_{pid} = 100 \cdot \frac{total\_time_{pid}/f}{uptime - start\_time_{pid}/f} \tag{5.2}$$

Dabei bedeuet $start\_time_{pid}$ die vergangene Zeit des Prozesses *pid* seit Prozessstart in Takten. Diese Formel ergibt somit die mittlere Prozessorauslastung durch den Prozess *pid*. Entsprechend der Gleichung 5.1 lässt sich diese Berechnung auch auf andere Intervalle übertragen. Somit ist es möglich, bspw. die momentane Prozessorauslastung durch die Wartungskomponente zu ermitteln.

5.1.2 Mittlere Systemlast mit /proc/loadavg

Der Kernel führt Statistiken über die Aktivität der Prozessoren. Die *mittlere Systemlast* der letzten Minute, der letzten fünf und der letzten 15 Minuten werden dem Nutzer durch die Datei `/proc/loadavg` zur Verfügung gestellt. Diese drei Werte stellen keine prozentuale Auslastung, sondern Kennzahlen dar.

Ein Wert 0 bedeutet, dass kein Prozess im vergangenen Zeitraum aktiv war, während ein Wert 1 bedeutet, dass ein Prozessor zu 100 % mit einem oder mehreren Prozessen ausgelastet war. Hierin eingeschlossen sind aktive Prozesse (`TASK_RUNNING`) und nicht unterbrechbare Prozesse (`TASK_UNINTERRUPTIBLE`): Erstere werden entweder vom Prozessor bedient oder warten auf Bedienung, zweitere sind schlafende Prozesse, die auf eine Bedingung warten und dabei nicht

von Signalen unterbrochen werden dürfen. Nicht unterbrechbare Prozesse sind normalerweise sehr selten und führen nicht zu einer hohen Last, vgl. [BC06]. Die mittlere Systemlast stellt folglich nicht die Prozessorauslastung dar, sondern die mittlere Anzahl der Prozesse, die bearbeitet werden oder auf Bearbeitung warten.

Wenn die mittlere Systemlast der Anzahl an Prozessoren entspricht oder diese übersteigt, kann von einer hohen Auslastung des Systems ausgegegangen werden. Was hohe Systemlast bedeutet, muss hingegen experimentell ermittelt werden, vgl. [GA07].

Die Messung der mittleren Systemlast ist von Vorteil, da sie auch wartende Prozesse berücksichtigt. Bei der Auslastung der Prozessoren mit *ps* können 100 % erreicht werden. Dies zeigt aber nicht, wieviele Prozesse noch auf Bedienung warten. Weiterhin stellen die Zahlen gleitende Mittelwerte dar, was bei der momentanen Prozessorauslastung nachträglich implementiert werden muss. Von Nachteil ist, dass die Zeiträume der Messung bereits fixiert sind und der Nutzer keinen Zeitraum kürzer als eine Minute messen kann. Der verbreitete Einsatz der mittleren Systemlast in Linux-Anwendungen (`uptime`, `top`, `who`) zeigt jedoch, dass diese drei Kennzahlen ein verbreitetes Maß für die Angabe der Systemlast sind.

5.2 ENTWURF DER LASTSTEUERUNG

Der folgende Abschnitt stellt den Entwurf einer Laststeuerung vor, um Prognoseanfragen bei hoher Systemlast zu begünstigen. Abbildung 5.1 zeigt die Position der *Laststeuerung* in der Wartungskomponente: Sie wird während der *Wartungsplanung* durchgeführt, zusammen mit der *Modellsuche*, der *Zustandswartung* und der *Modellevaluation*. Die *Wartungsoperationen* werden weiterhin von *Arbeitern* ausgeführt, die Jobs aus den *Joblisten* entgegennehmen. Neu ist die Einrichtung von *exklusiven Arbeitern*, die lediglich Jobs mit hoher Priorität, d. h. angefragte Prognosemodelle, bedienen. Der Status *Exklusiv* wird ihnen durch die Laststeuerung verliehen, wenn eine hohe Systemlast anliegt, und entzogen, wenn die Systemlast zurückgeht. Auch weiterhin ändert sich das Verhalten der Wartungskomponente zugunsten von Prognoseanfragen, s. Abbildung 4.3 (Seite 67). Die Laststeuerung ist auch aktiv, wenn keine Anfrage anliegt, weshalb hier der allgemeine Fall dargestellt ist.

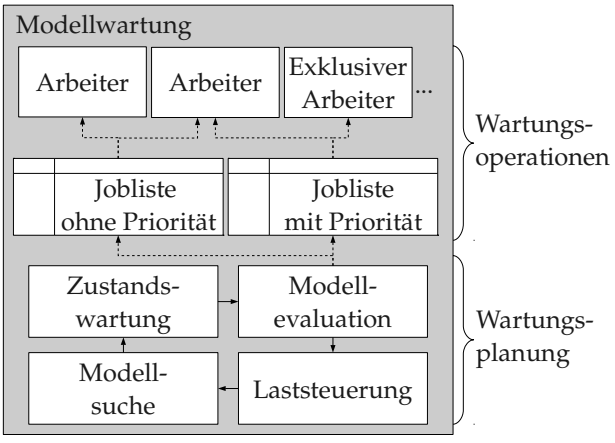


Abbildung 5.1: Erweiterung der Wartungskomponente um eine Laststeuerung

Experimentelle Untersuchungen haben gezeigt, dass die Systemlast der Datenbank weniger bei den Nutzerprozessen liegt, die eine Prognoseanfrage stellen. Vielmehr liegt sie bei der Wartungskomponente, die abschließende Wartungsoperationen durchführen muss, bevor die Prognoseanfrage bedient werden kann.

Die Priorisierung angefragter Prognosemodelle, s. Unterabschnitt 4.4 (Seite 66) ermöglicht eine reduzierte Anfrageverzögerung. Bei einer mittleren Systemlast sind stets Arbeiter frei, die die Wartungsoperationen angefragter Prognosemodelle annehmen können. Bei einer hohen Systemlast sind alle Arbeiter besetzt, sodass die Priorisierung versagt: Die Prognoseanfrage kann erst bedient werden, wenn ein Arbeiter wieder frei wird. Dies motiviert zu einer Steuerung, die exklusive Arbeiter bereithält. Sie können Jobs unverzögert von angefragten Prognosemodellen bedienen.

Die Steuerung wird durch einen *Dreipunktregler* realisiert, wie er z. B. in der Elektrotechnik Anwendung findet. Abbildung 5.2 stellt sein Schaltverhalten schematisch dar. Die x-Achse stellt die Abweichung der *Regelgröße* z vom Sollwert dar, was als Abweichung e angeben wird. Auf der y-Achse ist die *Stellgröße* y, die eine Steuerung der Regelgröße ermöglicht.

Unterschreitet die Regelgröße den Sollwert z_u, wird die Stellgröße auf $y = -Y_h$ gesetzt. Überschreitet sie den Sollwert z_o, wird der Stellgröße auf $Y = +Y_h$ gesetzt. Im dazwischenliegenden *Nullbereich* ist keine Steuerung notwendig, vgl. [MSF09].

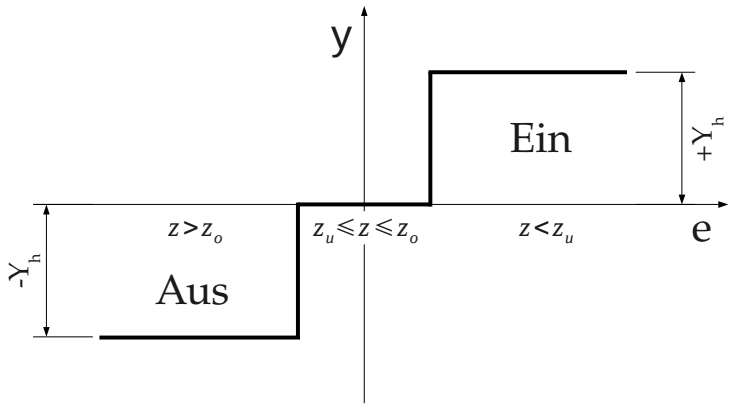


Abbildung 5.2: Dreipunktregler nach [MSF09]

Dieses Konzept wird wie folgt auf die Laststeuerung übertragen: Als Regelgröße wird die Systemlast eingesetzt, z. B. die Prozessorauslastung oder die mittlere Systemlast. Die Sollwerte, zwischen denen sich der Nullbereich befindet, müssen hierfür empirisch bestimmt werden. Bei Überschreiten des Sollwerts z_o werden Arbeiter exklusiv gesetzt, sodass sie ausschließlich für die Bedienung von Jobs angefragter Prognosemodelle zur Verfügung stehen („Aus“). Dies geschieht so lange, bis die Systemlast wieder unterhalb des Sollwerts liegt. Eine Verzögerung der Stellgröße sorgt dafür, dass die Reaktion der letzten Steuerung abgewartet wird und nicht alle Arbeiter nacheinander exklusiv werden. Liegt die Systemlast unterhalb des Sollwerts z_u, so wechseln Arbeiter wieder in den normalen Zustand zurück („Ein“).

5.3 EVALUATION

Der folgende Abschnitt stellt die Evaluation der Laststeuerung vor. In Unterabschnitt 5.3.1 wird die Auswahl der Maßes für die Systemlast untersucht. Unterabschnitt 5.3.2 zeigt die Auswirkungen der Laststeuerung auf die Systemlast. Unterabschnitt 5.3.3 evaluiert, in welchem Maße die Prognoseverzögerung einer Anfrage durch die Laststeuerung reduziert werden kann.

Da der Versuchsaufbau und die -durchführung für alle drei Unterabschnitte identisch ist, wird zunächst auf die gemeinsamen Eigenschaften eingegangen. Die Ergebnisse werden anschließend einzeln vorgestellt. Die Versuche werden auf einem System mit 64 Prozessoren à 2,13 GHz und 128 GiB durchgeführt.

Versuchsaufbau Die Evaluation wird am Datensatz *Wind* durchgeführt, der durch seinen Umfang zu einer Auslastung des Prognosesystems führt. Für alle 1361 Zeitreihen werden direkte Prognosemodelle erstellt, die auf dem Zeitraum 2004/2005 trainiert werden. Tabelle 3.3 (Seite 54) zeigt die Systemvariablen. Es sind alle Wartungsoperationen zugelassen.

Die Datenbank wird durch die Parameter `shared_buffers` und `work_mem` derart konfiguriert, dass die Zeitreihen im Hauptspeicher gepuffert werden und der Festplattenzugriff minimiert wird. Somit ist eine Skalierbarkeit der Wartungskomponente möglich und die 60 Arbeiter können parallel Jobs ausführen. Die Prozessoren erreichen dabei eine Auslastung von 100 %.

Für die Laststeuerung werden die Werte aus Tabelle 5.1 eingesetzt. Sie stellen eine Potenzialanalyse dar. Mit der Motivation, die Systemlast bei 90 % zu begrenzen wird der obere Sollwert gesetzt. Der untere Sollwert von 75 % trägt dafür Sorge, dass alle Arbeiter nach einer hohen Systemlast in den normalen Zustand zurückwechseln. Die Werte der mittleren Systemlast gelten entsprechend auf 64 Prozessoren bezogen. Die Verzögerung der Stellgröße sorgt dafür, dass zunächst eine Minute gewartet wird, bis eine neue Regelung erfolgt. Dies ist für die mittlere Systemlast wichtig, deren Messung sich auf die vergangene Minute bezieht. Zur Vergleichbarkeit beträgt auch das Zeitintervall für die Messung der momentanen Prozessorauslastung eine Minute, entsprechend auch die Verzögerung. Eine genauere Abstimmung der Werte auf das Prognosesystem bedarf einer weiteren Untersuchung.

Tabelle 5.1: Systemvariablen für die Evaluation der Laststeuerung

Variable	Wert	Einheit	Anmerkung
`cpuusage_max`	`90`	`%`	Oberer Sollwert (Prozessorauslastung)
`cpuusage_min`	`75`	`%`	Unterer Sollwert (Prozessorauslastung)
`cpuusage_delay`	`60`	`s`	Verzögerung der Stellgröße (Prozessorauslastg.)
`loadaverage_max`	`57.6`	–	Oberer Sollwert (mittl. Systemlast)
`loadaverage_min`	`48`	–	Unterer Sollwert (mittl. Systemlast)
`loadaverage_delay`	`60`	`s`	Verzögerung der Stellgröße (mittl. Systemlast)

Versuchsdurchführung Nach Erstellung der Prognosemodelle werden alle Zeitreihen für Januar und Februar 2006 fortgeschrieben. Die Einfügung der Tupel einer Stunde, d. h. $(6 \cdot 1326)$ Tupel, benötigt fünf Sekunden. Sie werden in einem Schub eingefügt, es entspricht einer Einfügegeschwindigkeit von etwa 1591 Tupeln pro Sekunde. Eine höhere Geschwindigkeit führt zu einer Überfüllung des Einfügepuffers und verhindert eine genaue Messung. Wegen dieser Begrenzung

wird die Evaluation lediglich auf zwei Monaten des Datensatzes durchgeführt, so kann ein Experiment in etwa drei Stunden realisiert werden. Die zuvor verwendeten Schwellwerte für den Datensatz **Wind** werden dafür angepasst, s. Tabelle 5.2.

Tabelle 5.2: Schwellwerte für den Datensatz Wind (Laststeuerung)

Schwellwert	Wert	Einheit	Anmerkung
error_smape_max	14	%	-
error_time_elapsed_min	1008	–	Früheste Wartung nach einer Woche
error_time_elapsed_max	4320	–	Späteste Wartung nach einem Monat

Das System fordert aufgrund der Schwellwerte Wartungsoperationen für die Prognosemodelle an. Parallel dazu werden Prognoseanfragen gestellt, die wie in Kapitel 4 beschrieben priorisiert werden. Nachdem eine Prognoseanfrage beantwortet wurde, wird unmittelbar eine neue Prognoseanfrage gestellt. Alle Prognosemodelle werden gleichverteilt angefragt, die Genauigkeitsklasse ist für alle Prognoseanfragen gleich. Beispielhaft ist eine Anfrage in Abbildung 5.3 dargestellt.

```
1 SELECT zeit, messwert FROM Wind
2 WHERE windanlage = 1
3 GROUP BY zeit ORDER BY zeit NUMBER 3 STORAGE modelindex
4 MAINTENANCE CUSTOM (
5         MAX_ERROR 10, MAX_TIME 4320
6 ) OPERATIONS (
7         MODELPARAMETER
8 );
```

Abbildung 5.3: Prognoseanfrage mit Genauigkeitsklasse für den Datensatz **Wind**

5.3.1 Prozessorauslastung und mittlere Systemlast im Vergleich

Zunächst wird untersucht, welches Maß für die Repräsentation der Systemlast geeignet ist. Die vorgestellten Maße, momentane Prozessorauslastung und mittlere Systemlast, werden in oben beschriebenem Versuchsaufbau miteinander verglichen. Während der Einfügung stellt genau ein Nutzer Prognoseanfragen an das System.

Ergebnis Abbildung 5.4 zeigt den Einsatz der beiden Maße im Vergleich. Die x-Achse stellt die vergangene Echtzeit seit Beginn der Tupeleinfügungen dar. Die erste y-Achse stellt die mittlere Systemlast und die Anzahl Arbeiter dar, beide sind einheitenlos. Die zweite y-Achse zeigt die Prozessorauslastung in Prozent. Die beiden Geraden zeigen die Sollwerte der Maße: der untere liegt wie angegeben bei $z_u = 48$ (mittlere Systemlast) bzw. $z_u = 75$ % (Prozessorauslastung), der obere Sollwert bei $z_o = 57,6$ bzw. $z_o = 90$ %.

Zunächst wird die Systemlast im Vergleich untersucht. Beide Maße verzeichnen einen starken Anstieg, sobald Wartungsoperationen veranlasst werden. Dadurch übersteigen sie den oberen Sollwert. Die Laststeuerung sorgt für eine Reduzierung der Systemlast, sodass sich die Maße unterhalb des oberen Sollwerts einschwingen. Nach dem Ende der Wartungsoperationen fällt die Last wieder ab. Für beide Maße werden vier Arbeiter exklusiv gesetzt. Dabei setzt die Reaktion der Prozessorauslastung eher ein als die mittlere Systemlast. Nachdem die Last zurückgeht, wechseln die Arbeiter zurück in den normalen Zustand.

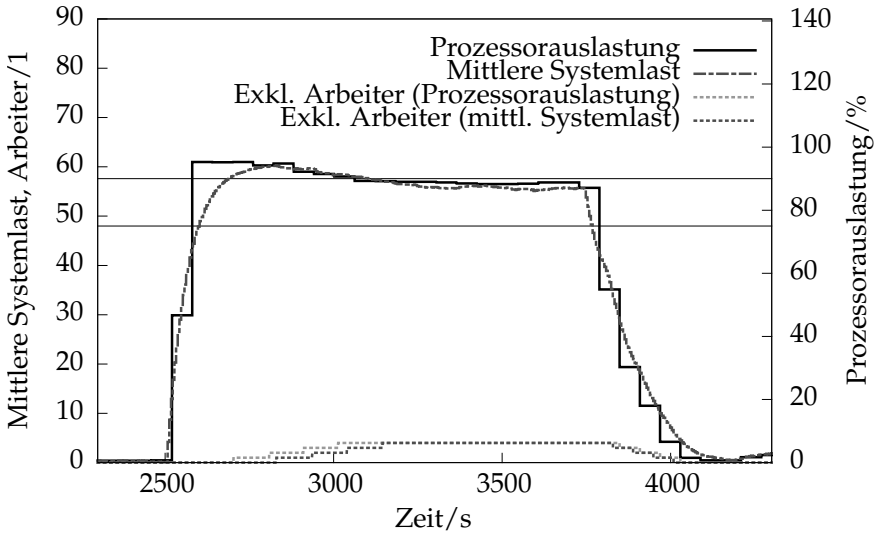


Abbildung 5.4: Vergleich von Prozessorauslastung und mittlerer Systemlast

Interpretation Nach Überschreiten des oberen Sollwerts bedienen Arbeiter exklusiv Jobs für Prognoseanfragen. Der Aufwand für die Modellparameterwartung, die von den Prognoseanfragen gefordert wird, ist verglichen zur Metaparameterwartung und Wartung der Ableitungsregeln gering. Dadurch sinkt die Systemlast.

Die beiden Maße sind in ihrem Verhalten sehr ähnlich. Da für die Prozessorauslastung nur im Abstand von einer Minute gemessen wird, fallen manche Reaktionen schnell aus, z. B. als der obere Sollwert überschritten wird. Andere sind verglichen zur mittleren Systemlast verzögert. Dies wirkt sich auch auf das Setzen der exklusiven Arbeiter aus. Von Vorteil an der Prozessorauslastung ist, dass die Zeitintervalle für die Messung frei wählbar sind. Somit ist auch die Reaktion auf Laständerungen anpassbar.

Die mittlere Systemlast ist ein gleitender Mittelwert, der laufend angepasst wird, jedoch ein festes Zeitintervall von einer Minute hat. Von Vorteil ist, dass sein Einsatz in Programmen weit verbreitet ist. In der Folge wird daher die mittlere Systemlast für die Evaluation der Laststeuerung genutzt.

5.3.2 Auswirkungen auf die Systemlast

Die vorgestellte Versuchsdurchführung ermöglicht die Evaluation der Laststeuerung. Ziel ist es, die Systemlast zu begrenzen, um exklusive Arbeiter für eine unverzögerte Prognoseanfrage bereitzustellen. Als Maß dient die mittlere Systemlast mit den Werten aus Tabelle 5.1. Während der Einfügung stellt genau ein Nutzer Prognoseanfragen mit der in Abbildung 5.3 gegebenen Genauigkeitsklasse.

Ergebnis Abbildung 5.5 zeigt die Auswertung der Systemlast ohne Laststeuerung, s. Abbildung 5.5a, und mit Laststeuerung, s. Abbildung 5.5b. Die x-Achse stellt die vergangene Zeit in Sekunden dar, die y-Achse beschreibt sowohl die mittlere Systemlast als auch die Anzahl Arbeiter, beide

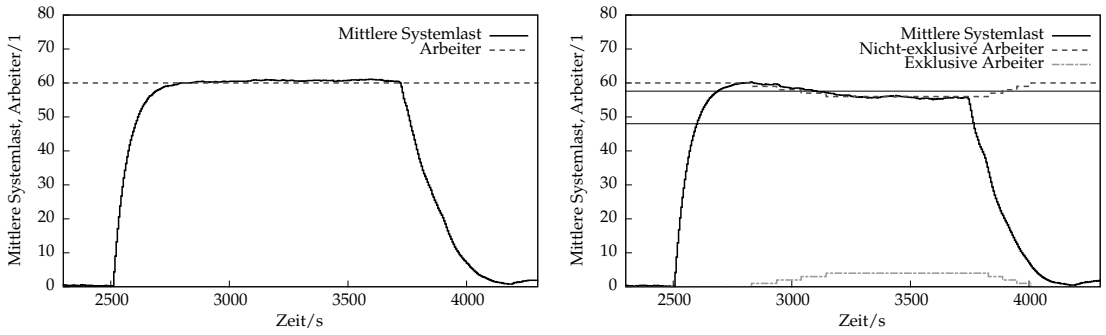
sind einheitenlos. Abbildung 5.5b zeigt zusätzlich die Sollwerte der Laststerung.

In beiden Fällen ist zunächst eine geringe Systemlast messbar. Nach 2500 Sekunden steigt sie schnell an. Ohne Laststeuerung liegt sie 1200 Sekunden lang zwischen 60 und 61, mit Laststeuerung fällt sie nach einer Einschwingphase in den Nullbereich und bleibt bei 55 bis 56. Anschließend fällt die Systemlast in beiden Fällen wieder herab und bleibt bei etwa 1. Ohne Laststeuerung ist die Anzahl Arbeiter konstant bei 60, wie vom System vorgegeben. Mit Laststeuerung werden 4 exklusive Arbeiter bereitgestellt, folglich sind 56 Arbeiter nicht-exklusiv.

Interpretation Aufgrund des Fortschreibens der Zeitreihen werden vom System komplexe Wartungsoperationen angefordert. Wegen des Schwellwerts `error_time_elapsed_min` beginnen sie erst nach einer geforderten Anzahl Einfügetupel. Die Systemlast liegt bei 60, offensichtlich verursachen die Arbeiter die höchste Last.

Mit Laststeuerung führt das Überschreiten des oberen Sollwerts zur Beschränkung von Arbeitern, die nun exklusiv Jobs für angefragte Prognosemodelle bedienen. Dadurch reduziert sich die Systemlast und verharrt nach einer Einschwingphase im Nullbereich zwischen den Sollwerten. Wie von dem Nutzer gefordert führen die exklusiven Arbeiter die Modellparameterwartung durch. Die mittlere Systemlast ist nach der Bearbeitung der Jobs leicht höher, sie liegt etwa bei 1. Grund hierfür ist, dass kontinuierlich Anfragen eine Modellparameterwartung benötigen, die von einem Arbeiter ausgeführt wird.

Diese Evaluation zeigt, dass die Laststeuerung zur Reduzierung der Systemlast führen kann, die die Wartungskomponente verursacht. Zudem stellt sie exklusive Arbeiter bereit, die eine reduzierte Anfrageverzögerung ermöglichen. Dies wird im nächsten Unterabschnitt untersucht.



(a) Systemlast ohne Laststeuerung **(b)** Systemlast mit Laststeuerung

Abbildung 5.5: Vergleich der Systemlast ohne und mit Laststeuerung

5.3.3 Auswirkungen auf Prognoseverzögerung

Die nachfolgende Untersuchung vergleicht die Prognoseverzögerung mit und ohne Einsatz der Laststeuerung. Dabei werden drei verschiedene Anfrageraten miteinander verglichen, um zu zeigen, dass die Laststeuerung unabhängig von der Nutzeranzahl eine Reduzierung der Prognoseverzögerung ermöglicht.

Für Prognoseanfragen können drei verschiedene Fälle eintreten:

1. Wenn das zugehörige Prognosemodell die Schwellwerte der Prognoseanfrage erfüllt, so passiert die Prognoseanfrage nahezu unverzögert das Prognosesystem, weil keine Wartungsoperationen anfallen. Dieser Zusammenhang wurde bei den nutzerdefinierten Genauigkeitsklassen evaluiert, s. Abschnitt 4.5 (Seite 68).
2. Wenn das zugehörige Prognosemodell in Wartung ist, muss die Prognoseanfrage auf die Beendigung der Wartungsoperationen warten, wobei zugesichert wird, dass das Prognosesystem die geforderte Genauigkeitsklasse berücksichtigt.
3. Wenn das zugehörige Prognosemodell aktiv ist, aber die Genauigkeitsklasse der Prognoseanfrage nicht erfüllt, so werden die geforderten Wartungsoperationen durchgeführt.

Für die vorliegende Untersuchung ist der Fall 1 und der Fall 2 nachrangig: Ersterer wurde bereits untersucht und die Laststeuerung hat hierauf keinen Einfluss. Der Fall 2 muss gesondert untersucht werden, da nicht bekannt ist, in welcher Wartungsoperation sich das Prognosemodell befindet und welche Wartungsoperation noch ausstehen. Daher ist eine Zusammenfassung mit Fall 3 nicht sinnvoll. Für den Fall 3 setzt sich die Prognoseverzögerung stets aus drei Komponenten zusammen: die Wartezeit für die Wartungsplanung, die Wartezeit des Jobs in der Jobliste und die Bedienung des Jobs durch einen Arbeiter. Die Zeit für die Rückgabe des Prognosemodells wird zur Wartezeit für die Wartungsplanung gezählt.

Für sechs verschiedene Szenarien, die in Tabelle 5.3 genannt werden, stellen Nutzer Prognoseanfragen. Dabei wird die Anzahl paralleler Anfragen variiert und die Aktivität der Laststeuerung.

Tabelle 5.3: Szenarien für die Evaluation der Laststeuerung

Kürzel	Anzahl paralleler Anfragen	Aktivität der Laststeuerung
1 Ohne	1	nicht aktiv
1 Mit	1	aktiv
10 Ohne	10	nicht aktiv
10 Mit	10	aktiv
20 Ohne	20	nicht aktiv
20 Mit	20	aktiv

Ergebnis Abbildung 5.6 stellt die dazugehörigen Ergebnisse vor. Sie zeigt die mittlere Prognoseverzögerung, die aus ihren drei Komponenten besteht. Die Wartezeit für die Wartungsplanung liegt zwischen 0,34 s und 0,78 s und wird nicht von der Laststeuerung beeinflusst. Die Wartezeit in der Jobliste verkürzt sich je nach Szenario: von 2,85 s auf 1,10 s (1 Anfrage), von 3,48 s auf 1,96 s (10 parallele Anfragen) und von 6,29 s auf 3,98 s (20 parallele Anfragen). Die Verzögerung durch die Bedienung, zwischen 0,57 s und 0,81 s, verändert sich nur leicht und wird nicht von der Laststeuerung beeinflusst.

Interpretation Der Einsatz der Laststeuerung zeigt sich an der Bevorzugung von Jobs durch exklusive Arbeiter: Diese stehen wegen der hohen Systemlast zur Verfügung und ermöglichen somit eine zügige Bedienung. Die Wartungsplanung verursacht stets eine geringe Verzögerung der Wartungskomponente und skaliert nicht. Sie wird nicht von der Laststeuerung beeinflusst, sondern von der Anzahl neu eingetroffener Tupel sowie der Menge an Prognosemodellen, die sich noch in Wartung befinden. Die Arbeiter, die die Jobs bedienen, benötigen im Beispiel eine

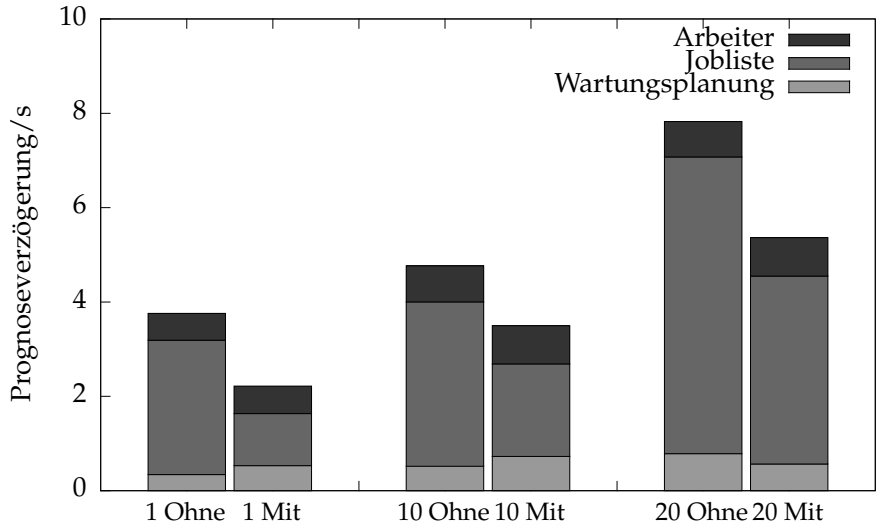


Abbildung 5.6: Prognoseverzögerung bei Einsatz der Laststeuerung

ähnlich hohe Rechenzeit, da es sich um die gleiche Wartungsoperation handelt. Die Bedienung ist schnell, da die Zeitreihen bereits gepuffert werden. Die Wartezeit in der Jobliste bei 10 und 20 parallelen Anfragen ist im Mittel höher: Wenn die Laststeuerung noch nicht aktiv ist, d. h. noch keine exklusiven Arbeiter bereit stehen, schwankt sie und ist abhängig davon, wann der nächste Arbeiter für eine Bedienung zur Verfügung steht. Dies betrifft vor allem die Szenarien mit mehreren parallelen Anfragen. Um die Ausreißer zu minimieren, müssen die Systemvariablen der Laststeuerung noch weiter angepasst werden.

5.4 ZUSAMMENFASSUNG

Dieses Kapitel stellt die Erweiterung der Wartungskomponente um eine Laststeuerung vor. Die Systemlast wird dabei als mittlere Systemlast angegeben, da es ein verbreitetes Maß für die Beschreibung ist.

Experimentell wurde gezeigt, dass eine hohe Systemlast von der Wartungskomponente des Prognosesystems ausgeht und weniger von den Prognoseanfragen. Daher ist die Laststeuerung mit der Motivation entworfen, die Aktivität der Wartungskomponente zu reduzieren. Sie lässt auch andere Prozesse als Verursacher der hohen Systemlast zu, denn sie zieht für die Messung der mittleren Systemlast alle Prozesse in Betracht.

Die Laststeuerung ist ein Dreipunktregler. Überschreitet die Systemlast den oberen Sollwert, stehen Arbeiter exklusiv für Jobs mit angefragten Prognosemodellen zur Verfügung. Unterhalb des unteren Sollwerts wechseln alle Arbeiter in den normalen Zustand. Es wurde gezeigt, dass die Koppelung von Systemlast und Wartungskomponente sinnvoll ist, da sie die Bedienung von Prognoseanfragen beschleunigt.

Die Prognoseverzögerung setzt sich aus mehreren Wartezeiten zusammen. Es wurde gezeigt, dass die Wartezeit in der Jobliste durch exklusive Arbeiter reduziert wird, andere Wartezeiten bleiben unberührt.

6 ZUSAMMENFASSUNG UND AUSBLICK

Die vorliegende Arbeit untersucht mehrere Erweiterungen des Prognosesystems Flash-Forward Query Framework. Im Fokus stehen dabei Verbesserungen der Wartungskomponente, die dem Nutzer eine genauere und weniger verzögerte Prognose ermöglichen. Das folgende Kapitel fasst die Ergebnisse der Arbeit zusammen und gibt einen Ausblick auf künftige Erweiterungen.

ZUSAMMENFASSUNG

Zentrales Ziel der Arbeit ist die Integration von Ableitungsmodellen für die Prognose. Mit einer knappen Erweiterung des Modellindexes lässt sich die Datenstruktur zur automatischen Bestimmung von Ableitungsregeln nutzen. Die Modellerstellung, -nutzung und -wartung der Ableitungsmodelle werden analog zum Lebenszyklus direkter Prognosemodelle gestaltet, wobei sich zwei neue Wartungsoperationen, die Wartung der Ableitungsgewichte und die Wartung der Ableitungsregeln, ergeben. Zentrales Resultat ist, dass durch ihre Nutzung die Anzahl der zu wartenden direkten Prognosemodelle reduziert werden kann und dabei die Prognosegenauigkeit erhalten bzw. geringfügig verbessert wird. Insbesondere die Disaggregationsstrategie konnte in der Evaluation an reellen Daten ihren Einsatz nachweisen.

Das zweite Ziel ist die Definition von nutzerdefinierte Genauigkeitsklassen. Für eine Prognoseanfrage kann ein Nutzer zwischen einer hohen Prognosegenauigkeit und einer geringen Prognoseverzögerung wählen. Dabei ermöglichen Optimierungen der Wartungskomponente eine bevorzugte Bedienung von angefragten Prognosemodellen. Die Evaluation zeigt, dass die zwei Nutzeranforderungen abgewogen werden müssen: Die Forderung nach hoher Prognosegenauigkeit führt zu einer Prognoseverzögerung führt. Andererseits muss ein Nutzer für eine unverzögerte Prognoseanfrage eine geringere Prognosegenauigkeit in Kauf nehmen.

Schließlich schlägt die Arbeit die Optimierung der Wartungskomponente unter Berücksichtigung der Systemlast vor. Hierfür wird eine Laststeuerung entworfen, die abhängig von der Auslastung

der Prozessoren eine Bereitstellung von exklusiven Arbeitern ermöglicht, um auch bei hoher Systemlast Prognoseanfragen mit geringer Verzögerung zu bedienen. Die Anzahl an Arbeitern ermöglicht wie in [Keg14] auch weiterhin die Skalierbarkeit der Komponente.

AUSBLICK

Optimierung der Kandidatenevaluation Während der Kandidatenevaluation müssen Ableitungsmodelle auf ihre Eignung für die Prognose geprüft werden, indem sie ihren Prognosefehler mit dem Prognosefehler anderer Kandidaten vergleichen. Hierfür werden temporär alle Quellmodelle für den Kandidaten erstellt, was zu einer hohen Rechenzeit der Wartungsoperation führt. Dabei können die Prognosen der Quellmodelle anders ermittelt werden, bspw. durch das Puffern vergangener Prognosewerte oder durch Backcasting. Letzteres ist für das Holt-Winters-Verfahren möglich, vgl. [Sch12b], was jedoch nicht für andere Prognosemethoden gelten muss.

Austausch der Prognosemethode Mit den vorgestellten Wartungsoperation sind drei von vier Komponenten eines Prognosemodells wartbar. Als weitere Operation kann der Austausch der Prognosemethode in Betracht gezogen werden, falls eine Zeitreihe durch eine andere Methode, z. B. die Box-Jenkins-Methode, besser charakterisiert wird.

Formulierung der Systemlast Für die vorgeschlagene Laststeuerung wurde der Einfluss von Festplattenzugriffen nicht berücksichtigt, um das zur Verfügung stehende System vollständig auszulasten. Dies ist möglich durch die Pufferung der Datenbank im Hauptspeicher. Wenn Daten von Festplatte gelesen werden müssen, werden die Wartungsoperationen durch die Zugriffszeit auf die Zeitreihen beschränkt. Folglich ist die Skalierbarkeit nicht mehr gewährleistet. Daher muss für eine künftige Untersuchung auch die Zugriffszeit auf die Festplatte in die Formulierung der Systemlast einbezogen werden.

Einschränkung von Wartungsoperationen Die Wartungskomponente verursacht durch die komplexen Wartungsoperationen die höchste Prognoseverzögerung. Der Entwurf der Laststeuerung kann sich auch auf die Einschränkung von Wartungsoperationen erstrecken. Bspw. kann die Modellschätzung auf einen aktuellen Ausschnitt der Historie an Stelle des gesamten Zeitraums beschränkt werden, vgl. [RZ06]. Von Nachteil ist hingegen, dass das Risiko ungenauer Modellparameter steigt. Für die Metaparameterwartung bzw. die Wartung der Ableitungsregeln lassen sich Statistiken über die Wahl einer Konfiguration führen, d. h. die Wahl von Metaparametern bzw. Ableitungsstrategien. Wenn die Häufigkeit einer Konfiguration zu gering ist, kann bei hoher Systemlast auf diese Prüfung verzichtet werden. Jedoch müssen zunächst Kenntnisse existieren, inwieweit von der Nutzung einer Konfiguration für ein Prognosemodell S_1 auf die bevorzugte Nutzung dieser Konfiguration für ein Prognosemodell S_2 geschlossen werden darf.

Prognosemodell für Anfragerate erstellen Die Anfragerate wurde für die Evaluation als fest angenommen. Schaffner und Januschowski [SJ13] belegen jedoch empirisch, dass sie im Tagesverlauf Schwankungen unterliegt, und stellen ein Modell vor, das die Anfragerate in der Realität widerspiegelt. Ein solches Modell ließe sich auch für das Prognosesystem erstellen, um die künftige Anfragerate zu berechnen. Benötigte Arbeiter könnten reserviert werden, bevor Prognoseanfragen eintreffen. Zudem belegt dies die Einsatzfähigkeit des Prognosesystems Flash-Forward Database System.

LITERATURVERZEICHNIS

[Arm01] ARMSTRONG, Jon S. (Hrsg.): *Principles of forecasting*. Boston : Kluwer Academic, 2001

[BB13] BACHEM, Achim ; BUCHAL, Christoph: Energiewende – quo vadis? In: *Physik-Journal* (2013), Dezember

[BC06] BOVET, Daniel P. ; CESATI, Marco: *Understanding the Linux kernel*. Sebastopol, CA : O'Reilly, 2006

[BJ70] BOX, George E. P. ; JENKINS, Gwilym M.: *Time series analysis forecasting and control*. San Francisco : Holden-Day, 1970

[CGS12] CHIANG, Roger H. L. ; GOES, Paulo ; STOHR, Edward A.: Business Intelligence and Analytics Education, and Program Development: A Unique Opportunity for the Information Systems Discipline. In: *ACM Trans. Manage. Inf. Syst.* 3 (2012), Nr. 3, S. 1–13

[Cha78] CHATFIELD, C.: The Holt-Winters Forecasting Procedure. In: *Journal of the Royal Statistical Society* 27 (1978), Nr. 3, S. 264–279

[CS81] CHAN, Paul ; SHOSHANI, Arie: SUBJECT: A Directory Driven System for Organizing and Accessing Large Statistical Databases. In: *Proceedings of the Seventh International Conference on Very Large Data Bases* Bd. 7, VLDB Endowment, 1981, S. 553–563

[CY91] CHATFIELD, Chris ; YAR, Mohammed: Prediction intervals for multiplicative Holt-Winters. In: *International Journal of Forecasting* 7 (1991), Nr. 1, S. 31–37

[CY04] CHEN, Zhuo ; YANG, Yuhong: *Assessing Forecast Accuracy Measures*. `http://www.researchgate.net/publication/228774888_Assessing_forecast_accuracy_measures`, 2004. – Abgerufen: 09.02.2015

[FBL11] FISCHER, Ulrike ; BÖHM, Matthias ; LEHNER, Wolfgang: Offline Design Tuning for Hierarchies of Forecast Models. In: *Proceedings der 14. GI-Fachtagung für Datenbanksysteme in Business, Technology und Web*, Gesellschaft für Informatik e. V., 2011, S. 167–186

[Fis14] FISCHER, Ulrike: *Forecasting in database systems*. Dresden, Technische Universität Dresden, Dissertation, 2014

[Fli01] FLIEDNER, Gene: Hierarchical forecasting: issues and use guidelines. In: *Industrial Management & Data Systems* 101 (2001), Nr. 1, S. 5–12

[FRBL10] FISCHER, Ulrike ; ROSENTHAL, Frank ; BÖHM, Matthias ; LEHNER, Wolfgang: Indexing forecast models for matching and maintenance. In: *Fourteenth International Database Engineering and Applications Symposium (IDEAS 2010)*, ACM, 2010, S. 26–31

[FRL12] FISCHER, Ulrike ; ROSENTHAL, Frank ; LEHNER, Wolfgang: F2DB: The Flash-Forward Database System. In: *IEEE 28th International Conference on Data Engineering (ICDE 2012)*, IEEE Computer Society, 2012, S. 1245–1248

[GA07] GANTEN, Peter H. ; ALEX, Wulf: *Debian GNU/Linux*. Berlin, Heidelberg : Springer, 2007

[GL99] GOODWIN, Paul ; LAWTON, Richard: On the asymmetry of the symmetric MAPE. In: *International Journal of Forecasting* 15 (1999), Nr. 4, S. 405–408

[GS90] GROSS, Charles W. ; SOHL, Jeffrey E.: Disaggregation methods to expedite product line forecasting. In: *Journal of Forecasting* 9 (1990), Nr. 3, S. 233–254

[Hol57] HOLT, Charles C.: Forecasting Trends and Seasonal by Exponentially Weighted Averages. In: *Office of Naval Research Memorandum* 52 (1957)

[Hyn08] HYNDMAN, Rob J.: *Forecasting with exponential smoothing*. Berlin, Heidelberg : Springer, 2008

[Hyn14] HYNDMAN, Rob J.: *R Forecast Package*. `http://cran.r-project.org/web/packages/forecast/index.html`, 2014. – Abgerufen: 21.02.2014

[Hä67] HÄNSEL, Horst: *Grundzüge der Fehlerrechnung*. Berlin : Dt. Verl. der Wissensch., 1967

[JL13] JAECKSCH, Bernhard ; LEHNER, Wolfgang: The Planning OLAP Model – A Multidimensional Model with Planning Support. In: *Transactions on Large-Scale Data- and Knowledge-Centered Systems* Bd. 7790. Berlin Heidelberg : Springer, 2013 (Lecture Notes in Computer Science 8), S. 32–52

[Keg14] KEGEL, Lars: *Asynchrone Wartung von Prognosemodellen*. Dresden, Technische Universität Dresden, Belegarbeit, 2014

[Kü12] KÜSTERS, Ulrich: Evaluation, Kombination und Auswahl betriebswirtschaftlicher Prognoseverfahren. In: MERTENS, Peter (Hrsg.) ; RÄSSLER, Susanne (Hrsg.): *Prognoserechnung*. Berlin : Physica-Verlag, 2012

[Lan12] LANE, Tom: *Multithreading von PostgreSQL*. `http://www.postgresql.org/message-id/6870.1327422359@sss.pgh.pa.us`, 2012. – Abgerufen: 15.01.2015

[Leh03] LEHNER, Wolfgang: *Datenbanktechnologie für Data-Warehouse-Systeme: Konzepte und Methoden*. Heidelberg : dpunkt-Verlag, 2003

[LS97] LENZ, Hans-Joachim ; SHOSHANI, Arie: Summarizability in OLAP and Statistical Data Bases. In: *Proceedings of the Ninth International Conference on Scientific and Statistical Database Management*. Washington : IEEE Computer Society, 1997, S. 132–143

[MAT14] MATLAB: *Create ARIMA or ARIMAX time series model*. `http://www.mathworks.de/de/help/econ/arimaclass.html`, 2014. – Abgerufen: 25.02.2014

[MR05] MERTENS, Peter (Hrsg.) ; RÄSSLER, Susanne (Hrsg.): *Prognoserechnung*. Berlin : Physica-Verlag, 2005

[MSF09] MANN, Heinz ; SCHIFFELGEN, Horst ; FRORIEP, Rainer: *Einführung in die Regelungstechnik*. München : Hanser, 2009

[NM65] NELDER, J. A. ; MEAD, R.: A Simplex Method for Function Minimization. In: *The Computer Journal* 7 (1965), Nr. 4, S. 308–313

[Ora12] ORACLE: *SQL Reference 10.2*. Redwood Shores : `http://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_5006.htm`, 2012. – Abgerufen: 15.01.2015

[Peg69] PEGELS, C. C.: Exponential Smoothing: Some New Variations. In: *Management Science* 15 (1969), Nr. 5, S. 311–315

[Pos12] POSTGRESQL: *PostgreSQL 9.3.6 Documentation*, 2012

[Pro09] PROCPS: *The proc file system utilities*. `http://procps.sourceforge.net`, 2009. – Abgerufen: 16.02.2015

[RZ06] RAUDYS, Šarunas ; ZLIOBAITE, Indre: The Multi-Agent System for Prediction of Financial Time Series. In: *Artificial Intelligence and Soft Computing (ICAISC 2006)* Bd. 4029. Berlin Heidelberg : Springer, 2006, S. 653–662

[Sch12a] SCHRÖDER, Michael: Einführung in die kurzfristige Zeitreihenprognose und Vergleich der einzelnen Verfahren. In: MERTENS, Peter (Hrsg.) ; RÄSSLER, Susanne (Hrsg.): *Prognoserechnung*. Berlin : Physica-Verlag, 2012

[Sch12b] SCHUHR, Roland: Einführung in die Prognose saisonaler Zeitreihen mithilfe exponentieller Glättungstechniken. In: MERTENS, Peter (Hrsg.) ; RÄSSLER, Susanne (Hrsg.): *Prognoserechnung*. Berlin : Physica-Verlag, 2012

[SJ13] SCHAFFNER, Jan ; JANUSCHOWSKI, Tim: Realistic tenant traces for enterprise DBaaS. In: *Workshops Proceedings of the 29th IEEE International Conference on Data Engineering (ICDE 2013)*, IEEE Computer Society, 2013, S. 29–35

[Vas98] VASSILIADIS, Panos: Modeling Multidimensional Databases, Cubes and Cube Operations. In: *Proceedings of the 10th International Conference on Scientific and Statistical Database Management (SSDBM)*, IEEE Computer Society, 1998, S. 53–62

[Wil06] WILSON, Ed: *Windows-Scripting mit WMI*. Unterschleißheim : Microsoft Press, 2006

[Win60] WINTERS, Peter R.: Forecasting Sales by Exponentially Weighted Moving Averages. In: *Management Science* 6 (1960), Nr. 3, S. 324–342

[YA95] YOKUMA, J. T. ; ARMSTRONG, J. S.: Beyond accuracy: Comparison of criteria used to select forecasting methods. In: *International Journal of Forecasting* 11 (1995), Nr. 4, S. 591–597

[YC90] YAR, Mohammed ; CHATFIELD, Chris: Prediction intervals for the Holt-Winters forecasting procedure. In: *International Journal of Forecasting* 6 (1990), Nr. 1, S. 127–137

[ZLE07] ZHOU, Jingren ; LARSON, Per-Åke ; ELMONGUI, Hicham G.: Lazy Maintenance of Materialized Views. In: *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB 2007)*, ACM, 2007, S. 231–242

A VERWENDETE DATENSÄTZE

Für die Illustration und Evaluation der vorliegenden Arbeit wurden drei Datensätze verwendet, die nachfolgend erläutert werden. Die Quelldokumente befinden sich auf der beiliegenden DVD-ROM.

Beschäftigung Die Arbeiten [Cha78, YC90] nutzen für die Evaluation den kanadischen Beschäftigungsindex in der Produktion. Die Messwerte liegen monatlich von 1970 bis einschließlich 1979 vor. Die Zeitreihe dient nicht der Evaluation, sondern ausschließlich der Illustration.

Tourismus Der Datensatz umfasst Quartalswerte zu Übernachtungen von Australiern im eigenen Land. Für die Jahre 1999 bis 2014 wurden die Messungen erhoben und nach *Anlass* (Freizeit, Beruf, u. a.) und nach Ziel der Reise nach *Region* (Sydney, Blue Mountains, Melbourne, Ballarat, u. a.) und *Bundesstaat* (New South Wales, Victoria, u. a.) gegliedert. Die Zeitreihen existieren für 85 Regionen und sind in 5 Anlässe untergliedert. Die Regionen sind 10 Bundesstaaten zugeordnet. Die australische Tourismusbehörde (`http://tra.gov.au/`) hat diese Daten per E-Mail zur Verfügung gestellt.

Wind Der Datensatz besteht aus 1326 Zeitreihen mit simulierten Leistungsprofilen von Windkraftanlagen im Osten der USA. Die Messwerte (in Watt) wurden im Zeitraum Januar 2004 bis Dezember 2006 erstellt mit einer Granularität von 10 Minuten. Durch Aggregation lassen sich zudem die Zeitreihen für die US-Bundesstaaten ermitteln, in denen sich die Windkraftanlagen befinden. Der Datensatz ist über die Internetseite vom National Renewable Energy Laboratory (NREL) erhältlich, siehe `http://www.nrel.gov/electricity/transmission/eastern_wind_dataset.html`.

B LISTE DER SYMBOLE

a, b, c	Koeffizienten des Holt-Winters-Verfahrens
d	Grenzen des maximal zulässigen Prognoseintervalls
e	Abweichung
l	Abstand des maximal zulässigen Prognoseintervalls
n, m	Kardinalität einer Menge
q	Saisonkomponente einer Zeitreihe
r	Trendkomponente einer Zeitreihe
s	Empirische Standardabweichung
s^2	Maximum-Likelihood-Schätzer
u	Restkomponente einer Zeitreihe
v, w	Wert eines Dimensionslevels
x	Messwert einer Zeitreihe
$\hat{x}$	Prognosewert einer Zeitreihe
y	Messung, Regelgröße
z	Stellgröße
A	Aggregationsmodell
$\boldsymbol{Agg}$	Menge der Aggregationsmodelle
C	Datenwürfel
D	Dimension
$\boldsymbol{Disagg}$	Menge der Disaggregationsmodelle
DL	Dimensionslevel
E	Disaggregationsmodell
H	Horizont
$\boldsymbol{H}$	Menge der Dimensionslevels einer Dimension
J	Anzahl fortgeschriebener Zeitpunkte (zeitbasierter Fehler)
L	Saisonlänge
M	Dimension des Messwerts
ML	Dimensionslevel des Messwerts
P	Disaggregationsschlüssel
$\boldsymbol{R}$	Menge der Tupel im Datenwürfel
S	Prognosemodell

T	Gegenwartszeitpunkt
Top_D	Maximales generisches Dimensionslevel von D
U	Tupel
R	Werte im Datenwürfel
$\boldsymbol{V}$	Domäne der Werte der Dimensionslevels
Z	Quantil der Normalverteilung

GRIECHISCHE SYMBOLE

α, β, γ	Glättungsparameter des Holt-Winters-Verfahrens
ϕ	Überdeckungswahrscheinlichkeit der Intervallprognose
ρ	Substitution
σ	Standardabweichung
$\boldsymbol{\Theta}$	Menge aller Zeitreihen
$\boldsymbol{\Psi}$	Raum aller Dimensionslevels
$\boldsymbol{\Omega}$	Raum aller Dimensionen

INDIZES

i, j	Position in einer Liste
k	Rang in Dimension
t	Zeitpunkt
τ	Prognosezeitpunkt

SONDERZEICHEN

$\star$	Beliebiger Wert für ein Top-Level
$*$	Kennzeichnung einer Multimenge

C WÜRFELOPERATIONEN NACH VASSILIADIS

Für die Realisierung der Roll-Up-Operation und des Dicing im Datenwürfel führt Vassiliadis fünf Operationen ein, die nachfolgend zusammengefasst werden, vgl. [Vas98]. Anhand des Beispiels 2, dem Datensatz **Tourismus**, werden die Resultate der Operationen illustriert, siehe Abbildung C.1 (Seite 98). Zunächst werden folgende Funktionen zur Verkürzung eingeführt. Es seien der Datenwürfel $C =< \underline{D}, \underline{DL}, C_b, \boldsymbol{R} >$ und der Wert $\underline{v} = [v_1, v_2, \ldots, v_n, *m]$ gegeben:

- $messwert\_dimension(C) = M$
- $messwert\_level(C) = *ML$
- $dimensionen(\underline{v}) = [v_1, v_2, \ldots, v_n]$
- $messwert(\underline{v}) = *m$
- $dimensionen(\underline{v})[i] = v_i$, falls $v \in \underline{R}$
- $dimensionen(\underline{v})(d) = v_i$, falls $d \in \underline{D}$ und $d = \underline{D}(i)$ und $\underline{v} \in \underline{R}$

Es sei $C =< \underline{D}, \underline{DL}, C_b, \boldsymbol{R} >$ der Datenwürfel im Ausgangszustand. Das Resultat einer Operation sei $C' =< \underline{D}', \underline{DL}', C_b', \boldsymbol{R}' >$.

LEVEL ERHÖHEN

Die Operation erhöht in der Teilmenge von Dimensionen auf die geforderten Dimensionslevels. Die Operation wird als $C' = LC(C, \underline{d}, \underline{dl'})$ dargestellt, dabei sind $\underline{d}$ die Teilmenge der betroffenen Dimensionen und $\underline{dl'}$ die geforderten Dimensionslevels.

Im Datenwürfel werden dadurch die Vater- und Kindlevel bestimmt, für die später eine Aggregation durchgeführt wird. Beispielhaft zeigt Abbildung C.1b die Erhöhung in der Geographie- bzw. Anlass-Dimension auf die Dimensionslevels *Bundesstaat* bzw. *Anlass*.

Definition 14. *Sei $\underline{d} \subset \underline{D}$ eine Menge von Dimensionen und $\underline{dl'} \subset \underline{DL}$ eine Menge von Dimensionslevels, die zu D gehören. $\underline{dl}$ seien die derzeitigen Dimensionslevels, die zu $\underline{d}$ gehören und durch $\underline{dl}$ ersetzt werden. $C' = L(C, \underline{d}, \underline{dl'})$ ist definiert als:*

$$\underline{D'} = \underline{D}, \underline{DL'} = \underline{DL} - \underline{dl} \cup \underline{dl'}\ , {C_b}' = C_b$$

$$\begin{aligned}\boldsymbol{R'} = \{\underline{v} | \exists \underline{w} \in \boldsymbol{R} : \forall D_i \notin \underline{d} : dimensionen(\underline{v})(D_i) = dimensionen(\underline{w})(D_i) \\ \wedge \forall D_i \in \underline{d}, dl_j \in \underline{dl'}, dl_j \in levels(D_i) : vorfahre(dimensionen(\underline{w})(D_i), dl_j) \\ \wedge messwert(\underline{v}) = messwert(\underline{w}), wenn\ M \notin \underline{d}\}\end{aligned}$$

Es wird vorausgesetzt, dass Dimensionslevels aus $\underline{dl}$ den gleichen oder einen kleineren Rang wie die Dimensionslevels aus $\underline{dl'}$ haben.

PACKEN

Die Operation *Packen*, $C' = P(C)$, fasst Messwerte, deren Dimensionslevels den gleichen Wert haben, in einer Menge zusammen. Dies stellt eine Vorstufe zur anschließenden Aggregation der Messwerte dar. Für diese Operation ist es notwendig, dass die Domäne des Messwerts Multimengen unterstützt.

Auf Ableitungskandidaten übertragen bedeutet diese Operation, Zeitreihen zu gruppieren. Ein Aggregationsmodell muss bspw. die Messwerte seiner untergeordneten Zeitreihen packen, um sie anschließend zu aggregieren. Abbildung C.1c verdeutlicht dies.

Definition 15. *Die Operation Packen, $C' = P(C)$, fasst Messwerte, deren Dimensionswerte gleich sind, in einer Menge zusammen.*

- $\underline{D'} = \underline{D}, \underline{L'} = \underline{L}, {C_b}' = C_b$
- $\boldsymbol{R'} = \{\underline{v} | \exists \underline{w} \in \boldsymbol{R} : \forall i (1 \leq i \leq n) : dimensionen(\underline{v})(D_i) = dimensionen(\underline{w})(D_i) \wedge messwert(\underline{v}) = \{messwert(\underline{w'}) | \exists \underline{w'} \in \boldsymbol{R} : dimensionen(\underline{w}) = dimensionen(\underline{w'})\}\}$

AGGREGATION

Um die Menge von Messwerten zu einem Wert aggregieren, wird die Operation $C' = F(C, f)$ eingeführt, wobei f die Aggregationsfunktion darstellt.

Für die Funktion *agg*, die die Summation von Messwerten einer Zeitreihen realisiert, ist das Resultat in Abbildung C.1d dargestellt. Ein Aggregationsmodell addiert die Messwerte seiner untergeordneten Zeitreihen, um aus ihnen seinen eigenen Messwert zu bestimmen. Dies gilt analog für die Bestimmung des Prognosewerts.

Definition 16. *Es sei f eine Funktion aus {sum, avg, count, agg, ...}. Die Operation „Aggregation" ist definiert als $C' = F(C, f)$ mit*

- $\underline{D'} = \underline{D}, \underline{L'} = \underline{L}, {C_b}' = C_b$
- $\boldsymbol{R'} = \{\underline{v} | \exists \underline{w} \in \boldsymbol{R} : dimensionen(\underline{v}) = dimensionen(\underline{w}) \wedge messwert(\underline{v}) = f(messwert(\underline{w}))$

NAVIGATION

Die drei genannten Operationen werden unter der Operation *Navigation* zusammengefasst. Entlang einer Dimensionen ist dadurch ein Verdichten der Daten auf einem höheren Level möglich. Sie lautet $C' = Nav(C, d, dl, f)$, wobei d die Dimension und dl das resultierende Dimensionslevel darstellt. Dies entspricht der Roll-Up-Operation.

Für einen gegebenen Datenwürfel und eine gegebene Dimension wird das Dimensionslevel geändert und gleichzeitig werden die Messwerte für das neue Dimensionslevel zusammengefasst. Abbildung C.1e verdeutlicht diese Operation, in dem der Basiswürfel nach Bundesstaaten gruppiert wird.

Definition 17. *Sei d die Dimension, über die navigiert werden soll, dl sei das resultierende Dimensionslevel und f die anzuwendende Aggregationsfunktion. Es gelte $d = \underline{D}[i]$. Die Navigation $C' = Nav(C, d, dl, f)$ ist definiert als die zusammengefasste Anwendung der Operationen Level erhöhen, Packen und Aggregazion: $C' = F(P(LC(C_b, \rho(\underline{D}, i, d), \rho(\underline{DL}, i, dl))), f)$.*

Die Navigation über mehrere Dimensionen kann nacheinander ausgeführt werden, um entlang mehrerer Dimensionen zu aggregieren.

Trotzdem $LC()$ nur eine Levelerhöung vorsieht, ist für die Navigation eine Levelreduzierung möglich, da sie stets vom Basiswürfel C_b ausgeht.

DICING

Mit der Operation *Dicing* wird ein Datenwürfel ausgewählt. Eine Selektionsbedingung gibt die Beschränkung des Datenwürfels entlang einer Dimension an.

Definition 18. *Sei D_i die Dimension, in der Dicing durchgeführt wird, sei eine Selektionsbedingung von der Form „$DL \;\; op \;\; v'$ “gegeben, wobei op für einen Operator wie $=$ oder $\in$ steht. Die Selektionsbedingung werde als Formel $\sigma(v')$ dargestellt. Dicing wird definiert als $C' = Dice(C, D_i, \sigma(v'))$ mit*

$\underline{D'} = \underline{D}, \underline{DL'} = \underline{DL},$

$C_b' = < {\underline{D_b}}', {\underline{L_B}}', {C_b}', {\boldsymbol{R_B}}' >$ *mit*

$\underline{D_b'} = C_b.\underline{D_b}, \underline{DL_b'} = C_b.\underline{DL_b},$

$\boldsymbol{R_b'} = \{\underline{v} \in C_b.\boldsymbol{R_b} | \underline{v}[i] \;\; op \;\; w, w \in nachfolger(v', levels(d)[1]\}$

$\boldsymbol{R'} = \{\underline{v} \in \boldsymbol{R} | \underline{v}[i] \;\; op \;\; v'\}$

Dicing kann über mehrere Dimensionen nacheinander ausgeführt werden. Beispielhaft die Operatione über zwei Dimensionen in Abbildung C.1f dargestellt. Die Selektionsbedingungen lauten: $\{\{D_1, \sigma_1(v_1)\}, \{D_2, \sigma_2(v_2)\}\} = \{\{D_{Geographie}, D_{Geographie} \in \{\text{„New South Wales“}, \text{„Victoria“}\}\}, \{D_{Anlass}, D_{Anlass} = \text{„Freizeit“}\}$.

Geographie	Anlass	Messwert
Sydney	Freizeit	[1,3]
Sydney	Beruf	[2,8]
Blue Mountains	Freizeit	[2,4]
Blue Mountains	Beruf	[5,7]
Melbourne	Freizeit	[6,7]
Melbourne	Beruf	[6,8]
Ballarat	Freizeit	[3,4]
Ballarat	Beruf	[5,9]

(a) Basiswürfel $C = (D, DL, C_b, R)$ mit $D = \{D_{Geographie}, D_{Anlass}\}$ und $DL = \{Region, Anlass\}$

Geographie	Anlass	Messwert
New South Wales	$\star$	[1,3]
New South Wales	$\star$	[2,8]
New South Wales	$\star$	[2,4]
New South Wales	$\star$	[5,7]
Victoria	$\star$	[6,7]
Victoria	$\star$	[6,8]
Victoria	$\star$	[3,4]
Victoria	$\star$	[5,9]

(b) Level erhöhen: $C_1 = L(C, \{D_{Geographie}, D_{Anlass}\}, \{Bundesstaat, \star\})$

Geographie	Anlass	Messwert
New South Wales	$\star$	[1,3], [2,8], [2,4], [5,7]
Victoria	$\star$	[6,7], [6,8], [3,4], [5,9]

(c) Packen von Werten: $C_2 = P(C_1)$

Geographie	Anlass	Messwert
New South Wales	Freizeit	[3,7]
New South Wales	Beruf	[7,15]
Victoria	Freizeit	[9,11]
Victoria	Beruf	[11,17]

(e) Navigation entlang Dimension $D_{Geographie}$: $C_4 = Nav(C, D_{Geographie}, Bundesstaat, agg)$

Geographie	Anlass	Messwert
New South Wales	$\star$	[10,22]
Victoria	$\star$	[20,28]

(d) Aggregation der Zeitreihen: $C_3 = F(C_4, agg)$

Geographie	Anlass	Messwert
Sydney	Freizeit	[1,3]
Blue Mountains	Freizeit	[2,4]
Melbourne	Freizeit	[6,7]
Ballarat	Freizeit	[3,4]

(f) *Dicing* auf zwei Dimensionen nacheinander ausgeführt: $C_5 = Dice(C, \boldsymbol{d}, \boldsymbol{\sigma}(\boldsymbol{v}))$

Abbildung C.1: Resultate der Würfeloperationen für den Datensatz **Tourismus**

D INHALTSVERZEICHNIS DVD-ROM

Die vorliegende DVD-ROM enthält alle Dokumente, die während der Diplomarbeit entstanden sind oder genutzt wurden. Ausgenommen sind Literaturverweise.

1. Schriftliche Arbeit
2. Prototyp Flash-Forward Database System
3. Evaluation: Quelltexte für die Evaluation und für die Erstellung der Diagramme
   (a) Derived: Integration von Ableitungsmodellen
   (b) Accuracy: Nutzerdefinierte Genauigkeitsklassen
   (c) LoadManager: Lastgesteuerte Wartungskomponente
4. Datensätze
   (a) Tourimus: Quelldokument und Anschreiben
   (b) Wind: Formatierte Quelldokumente und Link
   (c) Beschäftigung: Formatiertes Quelldokument