Workload Classification and Forecasting

Nikolas Roman Herbst, Nikolaus Huber, Samuel Kounev, Erich Amrehn (IBM R&D)

KoSSE-Symposium on Application Performance Management

November 29th, 2012
Motivation

Forecast approaches of the time series analysis:

1) Which tools are in the toolkit? properties - requirements - strength?
2) How can we characterize possible scenarios?
3) How do we select and apply a tool in a certain scenario?
4) Direct Feedback: ... Did we select the most appropriate tool and was it beneficial?

Idea: Intelligent and dynamic use of different tools out of the toolkit

Goal: Providing information on most likely future developments

„Knowing about a problem before feeling it“
Time Series Analysis

Foundations ▶ Approach ▶ Architecture ▶ Evaluation ▶ Related Work ▶ Summary

[BFAST]

Nikolas Herbst – Workload Classification and Forecasting

Software Design and Quality Group
Institute for Program Structures and Data Organization
# Forecasting Strategies

## Basic Methods

<table>
<thead>
<tr>
<th>Naïve, Moving Averages, Random Walk</th>
</tr>
</thead>
</table>

## Trend Interpolation

<table>
<thead>
<tr>
<th>Simple Exponential Smoothing (SES)</th>
<th>[Hynd08]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cubic Smoothing Splines</td>
<td>[Hynd02]</td>
</tr>
<tr>
<td>Croston’s method for intermittent time series</td>
<td>[Shen05]</td>
</tr>
<tr>
<td>Autoregressive Moving Averages (ARMA11)</td>
<td>[Box08]</td>
</tr>
</tbody>
</table>

## Estimation and Modelling of Seasonal Pattern

<table>
<thead>
<tr>
<th>Extended Exponential Smoothing (ETS)</th>
<th>[Hynd08, Hyn08]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARIMA framework with automatic model selection</td>
<td>[Box08, Hynd08]</td>
</tr>
<tr>
<td>tBATS for complex seasonal patterns</td>
<td>[Live11]</td>
</tr>
</tbody>
</table>
Forecast Accuracy Metric

Mean absolute scaled error (MASE) [Hynd06]

\[
e_t = \text{forecastValue}_t - \text{observedValue}_t
\]

\[
b_n = \frac{1}{n} \sum_{i=1}^{n} |\text{observedValue}_i - \text{observedValue}_{i-1}|
\]

\[
mase(0; n) = \text{mean}_{t=\{1;n\}}\left(\frac{|e_t|}{b_n}\right)
\]
Workload Intensity Characterization

High level data analysis to gain information on:

- Noise level & occurrences of unpredictable bursts
- Influence of trends and seasonal patterns

**Workload Intensity Trace:**
Time Series of Request Arrival Rates

- **Length**
- **Positivity**
- **Frequency**
- **Gradient**
- **Monotonicity**
- **Variance**
- **Quartile Dispersion**
- **Burstiness**

Foundations

Approach

Architecture

Evaluation

Related Work

Summary
Classification Mechanism

(II) Evaluate accuracy

(II) Select strategies

Forecast Strategy Overhead Groups

Overhead Limitation
Forecast Frequency

Forecast Objectives
Confidence Level

Workload Intensity Trace
Forecast Objectives
Accuracy Feedback

INITIAL
Naïve, Smoothing, Random Walk

FAST
Trend Interpolation

COMPLEX
Decomposition & Seasonal Patterns

Overhead Limitation
Forecast Frequency

Forecast Objectives
Confidence Level

 Foundations  ➤  Approach  ➤  Architecture  ➤  Evaluation  ➤  Related Work  ➤  Summary

Nikolas Herbst – Workload Classification and Forecasting
Decision Tree for Classification

Initial

- MASE
  - length of time series
    - Naive + Mean Forecaster
      - short, medium, long
        - overhead > 2
          - yes
            - positivity of time series
              - > x % zero
                - yes
                  - Croston’s Method
                    - > x % zero
                      - yes
                        - Moving Average (3)
                          - yes
                            - Simple Exponential Smoothing - ARIMA 101
                              - e, f thresholds
                                - yes
                                  - Cubic Spline Interpolation
                                    - MASE
                                      - MASE
                                        - ETS + tBATS
                                          - overhead > 3
                                            - yes
                                              - tBATS + ARIMA
                                                - MASE
                                                  - time series characteristic
                                                    - forecasting or filter strategy
                                                      - empirical decision parameter
                                                        - fast

Complex
Data, Timing, Parameters

classification: (I) select strategies

forecast execution & result output

forecast horizon/ frequency (must not be equal)

classification frequency (# forecast executions)

classification: (II) evaluate selection

forecast execution & result output

Data input stream:
- Time series of request arrival rates [0; maxSize] most recent values
time unit, delta time & start time

Result output stream:
- Time series of forecast mean values, confidence intervals & MASE metric

new classification: (I) select strategies

forecast execution & result output

new forecast horizon
Experiment: Example for Forecast Accuracy Improvement

- Real-world workload intensity trace (IBM CICS transactions on System z)
- Comparison of **Workload Classification & Forecasting (WCF)** approach to **Extended Exponential Smoothing (ETS)** and **Naive** forecast

![Graph showing workload classification and forecast comparison](image-url)
Experiment

Cumulative Percentage Error Distribution
Comparison of WCF to Naive and ETS strategy
CICS transactions (5 days, 48 frequency, 240 forecast values)
Case Study: Example for Using Forecast Results

- **Scenario**: Additional server instances at certain thresholds, 3 weeks
- Real-world workload intensity trace (*Wikipedia DE* page requests per hour)

![Diagram showing workload intensity and forecast results.](image)
Case Study

Resource provisioning:
(I) Without forecasting (solely reactive):
   Resource provisioning actions triggered by 76 SLA violations
(II) Interpreting WCF forecast results (add. proactive):
   Reduction to 34 or less SLA violations

→ No significant change in resource usage observed (server instances per hour)
Related Work

Workload & Performance Monitoring  
e.g. [vHoo12]

Time Series Analysis  
[Box08, Hynd08, Shum11]

Forecasting for Proactive Resource Provisioning  
[Grun12, Hedw10, Gmac08, Mena04, Bens07]

Self-Adaptive Resource Provisioning  
e.g. [Hube11, Koun10]

→ Focus on forecasting of performance metrics  
→ Focus on single tools of the toolkit or other toolkits  
not workload intensity  
no dynamic selection
Summary & Outlook

Survey on Forecast Approaches

Implementation of the WCF-System
provides continuous forecast results at run-time

Construction of a Workload Classification Scheme

Experiments and Case Study:
Evaluation based on real-world workload intensity traces

Forecast Accuracy Improvement:
> 37 % compared to ETS as an established approach

Proactive Resource Provisioning enabled:
> Up to 75 % less SLA violations than reactive

Future Work:
> System Integration with Kieker
> Filters: Objective Selection, Splitting
> Use for Anomaly Detection [Biel12]
Literature


[Hynd02] R. J. Hyndman, M. L. King, I. Pitrun, and B. Billah, Local linear forecasts using cubic smoothing splines, Monash University, Department of Econometrics and Business Statistics, 2002
Literature


## Backup: Forecast Objectives

<table>
<thead>
<tr>
<th>parameter name</th>
<th>parameter space</th>
<th>proposed setting</th>
<th>explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>forecast period</td>
<td>[1:max_int]</td>
<td>[1; frequency]</td>
<td>This objective defines how often a forecast is executed in times of new time series points. For a value of 1 a forecast is requested every new time series point and can be dynamically increased by period factors in the classification setting to reach the configured maximum horizon. This value should be equal or smaller than the <em>start horizon</em> objective (if continuous or even overlapping forecasts are needed).</td>
</tr>
<tr>
<td>highest</td>
<td>[1;4]</td>
<td>[2;4]</td>
<td>This objective defines the highest overhead group from which the forecast strategies will be chosen. A value of 2 may be sufficient if the time series data have strong trend components that are not overlaid by seasonal patterns, as the strength of class 2 strategies is the trend extrapolation. For time series with seasonal patterns, a setting of 3 for a maximum forecast computation time of 30 seconds and 4 for forecast computation times below 1 minute is recommended.</td>
</tr>
<tr>
<td>overhead group</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>confidence level</td>
<td>[0;100)</td>
<td>may be given by a forecast interpreter</td>
<td>The confidence level $\alpha$ of the returned forecast confidence intervals is defined by this objective.</td>
</tr>
<tr>
<td>start horizon</td>
<td>[1:max_int]</td>
<td>[1; 1/8x frequency]</td>
<td>The <em>start horizon</em> defines the number of time series points to be forecasted at the beginning and can be dynamically increased by period factors in the classification setting up to the <em>maximum horizon</em> setting. This value should be equal or higher than the <em>forecast period</em> objective (if continuous or even overlapping forecasts are needed).</td>
</tr>
<tr>
<td>maximum</td>
<td>[1:max_int]</td>
<td>frequency</td>
<td>The value of <em>maximum horizon</em> setting defines the maximum number of time series points to be forecasted. A recommendation for this setting is the value of the <em>frequency</em> setting of the time series, as a higher horizon setting may lead to broad confidence intervals.</td>
</tr>
<tr>
<td>horizon</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Backup: Forecast Strategy Overhead Groups

<table>
<thead>
<tr>
<th>overhead group</th>
<th>strategies</th>
<th>application</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 – nearly none</td>
<td>naive, arithmetic mean</td>
<td>These two strategies are only applied if less than initial size threshold values are in the time series. The arithmetic mean strategy can have an forecast accuracy below 1 and therefore be better than a solely reactive approach using implicitly the naive strategy. This is only true in cases of nearly constant base level of the arrivals rates. These strategies should be executed as frequently as possible every new time series point.</td>
</tr>
<tr>
<td>2 - low</td>
<td>cubic spline interpolation, ARIMA 101, simple exponential smoothing, Croston’s method for intermittend demands</td>
<td>The strengths of these strategies are the low computational efforts below 100ms and their ability to extrapolate the trend component. They differ in sensitivity to noise level or seasonal components. These strategies need to be executed in a high frequency with small horizons.</td>
</tr>
<tr>
<td>3 - medium</td>
<td>extended exponential smoothing, tBATS</td>
<td>The computational effort for both strategies is below 30 sec for a maximum of 200 time series points. They differ in the capabilities of modeling seasonal components.</td>
</tr>
<tr>
<td>4 - high</td>
<td>ARIMA, tBATS</td>
<td>The computational effort for the ARIMA approach can reach up to 60 sec for a maximum of 200 time series points and may achieve smaller confidence intervals than the tBATS approach.</td>
</tr>
</tbody>
</table>
Backup: Sequence Diagram

Nikolas Herbst – Workload Classification and Forecasting
Backup: Class Diagram - Management

Nikolas Herbst – Workload Classification and Forecasting
Backup: Class Diagram - Classification

```
<<enumeration>>
ClassificationStrategyEnum
+Initial
+Fast
+Complex

<<interface>>
IClassification
+classify()

<<creates>>
InitialClassificationStrategy
+classify()

<<creates>>
FastClassificationStrategy
+classify()

<<creates>>
ComplexClassificationStrategy
+classify()

<<call>>
ClassificationUtilities
+calcForecastQuality(): double
+calcIndex(): double
+calcDeviation(): double
+calcVarianceCoefficient(): double
+countNumberOfZeroValues(): int
+calcMaximum(): double
+calcArithMean(): double
+calcMedian(): double
+calcLengthOfBiggestMonotonicSection(): int
+calcRelativeMonotonicity(): double
+calcRelativeGradient(): double
+calcQuartileDispersionCoefficient(): double
+calcQuartiles(): double
+calcBurstinessIndex(): double
+applyMovingAverage(): ITimeSeries
+smoothTSbyCombination(): ITimeSeries
+getLastXofTS(): ITimeSeries
+getValueGreaterThan(): ITimeSeries
+getValueLessThan(): ITimeSeries

<<call>>
ClassificationSetting
+classificationStrategy: ClassificationStrategyEnum
+class_period: int
+recentFcStrategy1: ForecastStrategyEnum
+recentFcStrategy2: ForecastStrategyEnum
+getForecaster1(): IForecaster
+getForecaster2(): IForecaster
```

Nikolas Herbst – Workload Classification and Forecasting
Backup: Class Diagram – Forecasting

```
ForecastObjectives
  - start_horizon : int
  - max_horizon : int
  - overhead : int
  - fc_period : int
  - confidenceLevel : int

ForecastStrategy
  - getConfidenceLevel() : int
  - getDuration() : long
  - getForecast() : ForecastStrategy

AbstractForecaster
  - confidenceLevel : int
  - historyTimeSeries
  - forecast() : IForecastResult

AbstractRForecaster
  - forecast() : IForecastResult

RServerBridge
  - host : string
  - port : int
  - getInstance() : RServerBridge
  + assign()
  + evaluate()
  + uniqueVarName()
  + toTimeSeries()

Mean
  - Naive

CubicSpline

ARIMA101
  - ETS
  - ARIMA

ForecastResult
```

```
interface IForecaster
  + forecast() : IForecastResult
  + getForecastResult() : int
  + getToOriginal() : ITimeSeries

interface IForecastResult
  + getConfidenceLevel() : int
  + getDuration() : long
  + getForecastStrategy() : ForecastStrategy
  + getMeanAbsoluteScaledError() : double
  + getForecast() : ITimeSeries
  + getLower() : ITimeSeries
  + getUpper() : ITimeSeries
  + getOriginal() : ITimeSeries
```
Backup: Experiment WCF4 vs. tBATS, ARIMA

WCF limited to choose from tBATS and ARIMA

→ Significant accuracy improvement by combination and dynamic strategy selection