Project Report
Design of Forcasting Engine
Course Project in System Lab (IT 680)

Kaushal Mittal
KReSIT, IIT Bombay

Abhishek Seth
KReSIT, IIT Bombay

Amar Agrawal
KReSIT, IIT Bombay

Apr 07, 2005

Abstract

Most of the research work in forecasting domain concentrates on obtaining forecasts using multiple forecasting models rather than trusting a single model. In this article, we propose a design of a generic forecasting engine that can use a number of forecasting models provided by differnet forecasting systems and analyse the results for various research goals. Our experiments using this engine has revealed that a single forecasting model may not give the best predictions throughout the time series. A time series can be partitioned and different forecast models can be fitted for each of this segments to get the best forecast.

Definition

- Expert - It is a parameterized forecasting model that takes a series as input and performs forecasts on it. For example: ARIMA(1,0,1) is a forecasting model with parameters $p = 1, d = 0, q = 1$.

- Backends - Forecasting applications like SAS, MATLAB, user applications consisting of various expert packages.

- Error measures - The criteria used to evaluate forecasts. For example: MAPE (Mean Absolute Percentage Error) RMSE (Root Mean Squared Error)

1 Introduction

A time series can be defined as a sequence of observation taken sequentially in time[?]. The interval of observation is usually fixed. These observation can be monthly sales in a super market, hourly electricity load at power generators, and many more. What makes the time series interesting to analyze is the dependency of adjacent observations. This dependency of adjacent observations can be trivial or it can be complicated. Many real world time series has such complicated dependency, hence require sophisticated mathematics to explain them. Time Series analysis is concered with fitting of mathematical models such as stochastic and dynamic models, to explain dependencies in observations.

The analysis of time series gives us the power of forecasting. Forecasting is concerned with the prediction of the future values of a given time series. Forecasting has many real world applications such as predicting of future sales, which will aid the businessmen in their planning.

Trivially one could think of using a forecasting expert that results in minimum error over a given time series. However, opinions from multiple experts can increase the confidence in the forecast values. A single expert cannot perform optimally over the entire series. Knowledge about which expert performs best at some segment of a time series can be used to improve the forecasting accuracy. We have designed an application that can be used to obtain this knowledge.

In section 2, we discuss the motivation behind the application. Section 3, we describe the requirements design and implementation aspects of the system. Section 5 describes the future extensions possible in this direction. Finally we conclude in section 6.

2 Motivation

A number of systems like SAS, MATLAB etc. are available that can be used for forecasting. Each of these systems provides implementations for various kind of forecasting models.
For a given series, using a combination of a number of experts over these systems can certainly improve the confidence in the forecast values. We need a system, that allows to know the top-k experts for a given series, irrespective of the system that provides it. One of the ways for combining the results could be getting the votes from these top-k experts for a series.
Another interesting enhancement possible is to get the best experts for different parts of the series. This will help us to know for what characteristics of a series, which expert will give better results. So the application should get forecast from different experts supported by various systems and find the best experts at each instance of time. With the above aim, we have developed a prototype/design/implementation that combines these software systems in some useful way so as to predict the best forecast. This application is useful for various research goals in the forecasting domain, and practioners dealing with forecasting of time series.

3 System Specifications

In this section we describe the requirements , design and implementation aspects of our application.

3.1 Problem Definition

- Develop a forecast engine that interacts with multiple forecasting back-ends such as SAS, MATLAB and other forecasting applications,to generate useful forecasts for a given time series.
- Provide support for analysing the forecast results from an ensemble of experts supported by different systems.

3.2 Requirement Specification

- Support Multiple heterogenous forecasting backends for forecasting.
- Provide the support for analysis of results from an ensemble of forecasting experts to obtain -
    1. the top 'k' best experts for the entire series using an error criteria (eg. MAPE, RMSE, etc.) and use them to generate forecasts.
    2. the top 'k' best experts for each observation point in the series. This information can be used to identify if some experts perform better on some segment of the series which could help in selecting the best expert to use for each future forecast.
- Support adaptive forecasting i.e. generate forecasts on the series considering only the past values upto that point and not look into future values.
  For example: If the series contains say 100 observation points then, use the first 30 points to predict the 31^{st}, then include the 31^{st} point(actual value) in the model to predict the 32^{nd} next point and so on. This avoids the bias that occurs when we use all the points to build a model and then use it to calculate prediction errors on the series points.
- The application must facilitate the easy integration of new forecasting backends in the future. A developer must be able to do this with minimal efforts and no changes to the existing software.

3.3 Design and Implementation

Different forecasting backend systems differ in the format of input and outputs. They have their own scripting language. Few of them do not support adaptive forecasting directly. To provide integration support for such a variety of forecasting systems, following design decisions were considered.

- The backend system must have command line support.
- They must take the input time series in a flat file and should be able to generate the output forecast in a file.

- By default the system generate the reports containing the top-k models for the whole series and the top-k models for each observation. It uses the experts to forecast the value for each time instance and then calculate the error using the actual values for that time instance. Based on the error measure, it rank the experts.

- The template file is provided for defining the syntax and semantics of the scripts corresponding to the forecasting backend system. The template is an XML file with the simple programming features like parameter substitutions, looping. The system can generate the scripts for forecasting systems automatically using this file and the input parameters.

The architecture of this application has following components:

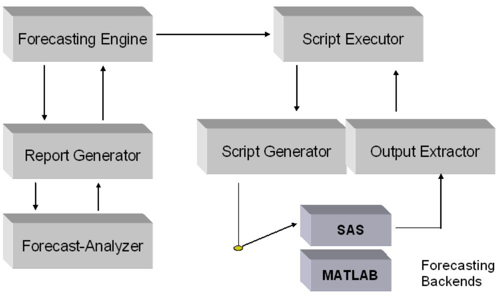

Figure 1: Architecture of Forecast Engine

- Forecasting Engine: This component takes in the input parameters and generate forecasts for a given time series. It act as a controller and coordinates among other components.

- Script Generator: This component generates forecasting backend specific scripts for the experts specified in the list of experts. This template file and the input parameters are used to generate the required script. The use of a template file makes the Script generator generic.

- ScriptExecuter: It interacts with the forecasting backends by executing the script files generated, using the appropriate command specified in the **backend-definition** file. Since the command can be different for different forecasting backend systems, a new ScriptExecutor is required for different forecasting system.

- Output Extractor: This component extracts the results from the output file of forecasting backend. Since the forecasting backend systems have different output formats, a separate output extractor is required for each forecasting system.

- Report Generator: This component is responsible for presenting the statistics of various forecasting experts as a report in text and HTML format.

- Forecast Analyzer: This component is responsible for analyzing the forecasts obtained from various experts. The analysis consist of finding out the top-k expert on the overall time series, and finding out the top k experts for each observation along with various error measures.

The application can be used in following ways:

- Use it to generate the forecast using a expert.

- Use it to generate the forecast with the best-k expert.

- use it to find the best-k experts at each time instance, for analysis purpose.

- User can use analyse the results from experts in his own way by using the API support.

- User can plug-in the new forecasting system .

Addition of a new Forecast system

- Append a node in the backend-definition XML file. This contains information about the paths of scripts generated, output/log files, command to be executed and extensions of the script files.

- Write a general script to be executed on the new forecasting backend. Write a template XML file corresponding to this script.

- Write an expert definition file. This file contains the values of the parameters to be substituted in the template file for generating scripts.
- Make an entry of this new expert in the list of experts.
- If the output format is different from default format, write an output extractor and a customized engine. [1].

4 Experimental Results

In this section, we provide a summary of the results obtained on the seven different time series provided to us [2]. The series provides the monthly data. Table 1 gives the experts that appear in the results. This list is not exhaustive. The experiments were conducted using 73 different experts.

Table 2 gives the best model over the entire series and for the first three predictions i.e. for the forecast of observation 31, 32 and 33. The results shows that the best model found based on MAPE error criteria, for the overall series are different from the best models for individual observations. This, suggest for using different forecasting models for different part of the series.

Table 3 gives the MAPE over the entire series and APE error for the first three predictions i.e. for the forecast of observation 31, 32 and 33. The results shows that, accuracy would have increased tremendously, if the best model is used for prediction at each and every point.

Abbrevations

Abbreviation	Full Form
E1	arima (0,2,2)(0,1,1)s
E2	arima (0,1,2)(0,1,1)s
E3	arima (2,0,0)(1,0,0)s
E4	arima (0,2,2)(0,1,1)s NOINT
E5	arima (0,1,2)(0,1,1)s NOINT
E6	arima (2,0,0)(1,0,0)s NOINT
E7	log arima (0,1,1)s
E8	log arima (0,2,2)(0,1,1)s NOINT
E9	log arima (1,1,1)
E10	log arima (2,0,0)(1,0,0)s
E11	log arima (2,0,0)(1,0,0)s NOINT
E12	log arima (2,1,0)(0,1,1)s NOINT
E13	Linear Trend AR4
E14	Linear Exponential
E15	log Linear Exponential
E16	log Winters Additive

Table 1: List of Experts

Series	BestModel	Obs31	Obs32	Obs33
1	E7	E7	E1	E16
2	E4	E4	E9	E4
3	E7	E5	E10	E6
4	E12	E2	E9	E13
5	E7	E14	E7	E7
6	E1	E3	E13	E8
7	E7	E15	E15	E11

Table 2: Best Models

Series	Best	Obs31	Obs32	Obs33	Optimal
1	3.21	1.26	0.36	1.68	0.45
2	8.62	0.35	2.49	0.57	2.32
3	7.72	2.35	0.19	1.15	1.76
4	7.87	1.66	3.25	0.60	1.55
5	5.34	0.40	0.78	0.02	1.10
6	9.51	0.19	0.04	0.01	2.08
7	14.72	0.58	0.12	3.95	2.27

Table 3: MAPE Errors

5 Future Enhancement

1. **For Developer** The application currently uses a single execution thread on a single machine. A distributed parallel implementation might be supported inorder to reduce the time for forecasts for a single series. This requires the scheduling of experts on multiple machines and combining their results.
2. **For User** The implementation provides an extensive set of experts for SAS which can be used directly by the user. The forecasting system can be made more accurate/robust by incorporating additional forecasting experts into it. The application facilitate such extension through use of templates.

[1] Refer User Manual for details
[2] Source: ABS

6 Conclusion

We have designed a generic forecasting engine that can analyze the results obtained from number of experts from different backend forecasting systems, and built a prototype for the same. The experimental results shows that if we somehow develop a way to find the best forecasting models for different segments of the series, the overall accuracy of the forecast values can improve significantly.

7 References

Java Online References SAS online help ETS Package Language Reference Matlab User Guide Neural Network toolbox manual Scripting Language