Design Document for the Final Project

Wenyan Gong, Zongxi Li, Cong Ma, Qingcan Wang, Zhuoran Yang, Hao Zhang

1 Overview

Time series analysis comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data. It's widely used in statistics, signal processing, pattern recognition, mathematical finance, weather forecasting, earthquake prediction, control engineering, and largely in any domain of applied science and engineering which involves temporal measurements. In this project, we will play a game with time series in finance. It has gained its popularity in Wall Street recently, since it is fundamental to the most promising quantitative investment strategies.

We develop a system that can predict future prices of stocks, with different time series models. Given input of a stock price series, our system will first fit some powerful and popular time series models, such as autoregressive (AR) model, moving average (MA) model and other derived models. This procedure will give you the estimation of parameters in these models. The most important task in estimation is the optimization procedure. Users can select one of optimization methods in the system based on their preference. The optimization method includes but not limited to gradient descent and Newton's method.

After model fitting and estimation, our system provides a fast way to do statistical inference. Users can do different kinds of statistical test as well as confidence interval. Moreover, with fitted model, future price prediction is made and it's compared with real price data. Moreover, we provide different methods to assess the prediction accuracy, which will be visualized afterwards.

More interestingly, if users input multiple stock prices, we can divide these stocks into different groups, which is called clustering. Among each group, stocks share similarity. Different clusters will also be visualized.

In the following, we will present the basic organization and components.

1.1 Model fitting

For a given time series, there are several potential classes of models that could be used to describe its variation. Among them, the most frequent used are the autoregressive(AR) model, integrated(I) model and the moving-average(MA) model. The combination of these classes lead to the autoregressive moving-average(ARMA) model, the autoregressive integrated moving-average(ARIMA) model and the autoregressive fractionally integrated moving-average(ARFIMA) model. The user could choose the model class he/she wish to fit based on his understanding of the input time series.

1.2 Optimization

After the model type being fixed, there would be certain loss function regarding the selected model. The key of model fitting is to decide the parameters, which are derived by minimizing the loss

function. Therefore, certain optimization techniques should be applied during the fitting. In our software, the user could choose the optimization tool between gradient descent, Newton method and stochastic gradient descent according to the scale of the problem.

1.3 Inference

After model fitting, in order to provide the user with more overall information about the model, our software also do inference work regarding the parameters. The software would construct confidence interval of given level for each parameter, conduct test to decide the non-zero parameters and calculate the corresponding P-value. This could help to identify the pattern of the model and help the user develop deeper understanding of the given time series.

1.4 Clustering

When a large number of time series are given, some of them may share similar patterns since they might be commonly affected by several intrinsic factors. With clustering techniques, the software would be able to identify the similar time series, i.e. stocks with similar variation, and divide them into groups. This would help the user gain a better knowledge of the stocks and their patterns.

1.5 Visualization

The software is able to predict the future price for each stock based on the fitted model. Hence, in the data visualization part, it would plot the estimated future price and an prediction interval along with the previous price that is already known. Together, some trading strategy would be made based on the prediction. The software could provide the user with the selling or buying point of certain stocks.

2 User Interface

As stated in previous sections, our code library integrates sampling, estimation, clustering, and statistical inference for time series models. To support these functionalities, we build interfaces that decompose the whole project into well-organized parts that enables integration.

In specific, our project consists of five classes: “Model”, “Simulation”, “Optimization”, “Inference”, and “ Clustering”. The pivotal idea is to decouple the times series models and the methods that work on models. By doing so, we can not only obtain a clearer view of the whole picture, but, more importantly, enables users to apply the statistical methods to more datasets, thus enhance usability. A detailed description of these class are as follows.

The “Model” class consists of a variety of time series models. Each model is identified by its parameters and the major goal of data analysis is to learn these parameters from the data. For each model, we first define a function that formulates the model. Then we define a loss function and the goal of learning the model is reduce to minimizing the loss function. Since we use first- and second-order optimization methods, we also compute its gradient and Hessian.

The “Simulation” class consists tools to sample data from the time series models. Although working on real-world datasets is more interesting, simulated data enables us to access the performance of the statistical procedures. The sampling function treats a model as input and use

Bayesian sampling method such as Markov Chain Monte Carlo and Gibbs sampling to generate simulated data.

The "Optimization" class consists of optimization tools. In general, an optimization procedure minimizes the loss function using gradient and/or Hessian information of the loss function. In this class, we will realize various popular optimization algorithms that enjoy great empirical success in deep learning. Some examples are gradient descent, stochastic gradient descent, momentum method, Adgrad, conjugate gradient, Newton method.

The "Inference" class consists of statistical procedures for time series models. We construct confidence intervals and hypothesis tests for the model parameters. After estimating the model, it is not clear whether estimation is accurate. Statistical inference enables us to access the uncertainty of our estimation procedure. Out inference functions will take the model and data as input, and outputs a confidence interval or test statistic. When using the simulated data, statistical inference enables us to quantitively understand the performance of estimation.

Finally, the "Clustering" class consists methods to cluster data into groups. For finance data, this task is of great importance. For example, stocks in different sectors behave differently. It is meaningful to cluster these stocks to reveal more fine-grained information of the market. Some of the procedures consists of K-means, Gaussian mixture model, hierarchical clusterin, and DB-SCAN algorithm. Note that clustering algorithms work for general datasets. Thus this class may potentially have broader application.

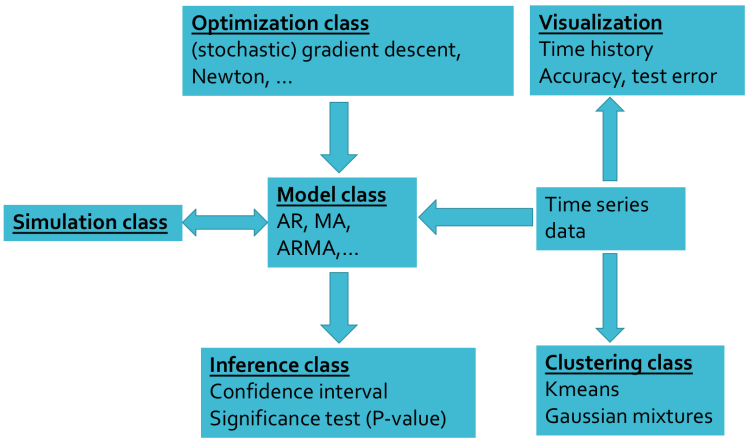

Figure 1: The relationship between the five classes of our project

3 Schedule

3.1 Design Document and Presentation

We will produce the design document which is a fairly detailed description of the project, and it will be submitted by 11/30/2016.

We will produce the design presentation slides, and make an oral presentation in class on 12/01/2016, which will give an overview of the project, and discuss specifics of the design.

3.2 Prototype

In this release, we will have at least one implementation for each step.

Zongxi and Wenyan will develop the implementation of AR and MA in the "Model" class, and take some sample data for testing. Qingcan will develop the implementation of gradient descent method in the "Optimization" class. Zhuoran will develop the implementation of correlation testing in "Inference" class. Cong will develop the interface for "Model", "Optimization" and "Inference" class. Hao will develop one method in "Clustering" class.

3.3 Alpha Version

In this release, we will complete most methods in each class.

Zongxi will continue working on combination methods in the "Model" class. Wenyan will develop the "Simulation" class. Qingcan will finish the "Optimization" class. Zhuoran will implement confidence interval and hypothesis testing in the "Inference" class. Hao will finish the "Clustering" class. Cong will provide interface for "Simulation" and "Clustering" class.

3.4 Final Version

In this relase, we will impelement the "Visualization" class, and compelet all the code, then get more data for testing. We will also finish the software manual and project report.