Apply Machine Learning to Performance Trend Analysis

Araya Eamrurksiri

Markov switching model is implemented for analyzing the thesis' problem - we are interested to discover change points and regime shift in time series when the time instant is unknown. This model is one of the most well-known non linear time series models. Markov switching model is first introduced by Hamilton [1] and is extensively implemented in economics and finance field. The model takes the presence of shifting regime in time series into account and models multiple structures that can explain these characteristics in different states at different time. The shift between states or regimes comes from the switching mechanism which is assumed to follow an unobserved Markov chain. Thus, the model is able to capture more complex dynamic patterns and identify the switch in states when change-point is most likely to occur. In speech recognition, such processes are described as hidden Markov model. Each software package in the system is viewed as a time point in time series and the performance of each software package is treated as an observed value. In this study, the observed value is not completely independent of each other i.e., the performance of the current software package depends on the performance from the prior version of the software package. Therefore, additional dependencies, with the first order autoregression, is taken into consideration when modeling the Markov switching model and becomes Markov switching autoregressive model.

The first step when doing an analysis is to perform data preprocessing and transformation. In this study, each software product consists of several software packages. Also, numerous test cases are executed in one software package. A test case which has a minimum value of CPU utilization is, therefore, selected to represent the performance of a specific software package. Moreover, some fields in the dataset store multiple values separated by a tab character. These tab-separated values are split to columns in order to use for later analysis. Lastly, incomplete test cases are removed from the dataset.

The MSwM [1] package in CRAN developed by Josep A. Sanchez-Espigares is used for performing an univariate autoregressive Markov switching model for linear and generalized model. The package implemented expectation-maximization (EM) algorithm to estimate the Markov switching model. Source code and functions in this package have been studied and reviewed in detail. The purpose of this task is to get a general idea of how the package works and also to un-

[1] https://cran.r-project.org/web/packages/MSwM/index.html

derstand algorithms and concepts behind the package. Even though most of the coding have been done, code is further implemented in order to properly deal with the problem and the given dataset. Some modifications have been made in the function to handle errors and warnings produced when fitting the model. For instance, when setting variance to have a non-switching effect, the function generates a warning. It proved that there is a minor mistake in the code. Furthermore, the package uses a Hessian (the matrix of second order partial derivatives with respect to parameters) for numerical optimization. In some cases, Hessian matrix will not be invertible as the matrix is singularity. Consequently, the function can not compute standard error of estimated parameters. This non-invertible Hessian is solved by using generalized inverse (or pseudoinverse) procedure [2]. Apparently, the package does not work well with categorical predictor variables. Hence, a further implementation in the code for handling with categorical variables is done. Another error in the package occurs when any initial coefficients from the model are NAs. A function will first randomly divide data into different subsets and separately fit the model to get initial coefficients in each state. Sometime a variable in a subset will have the same value in all observations. Then, NA coefficient is generated because of singularity. The solution is to reshuffle data and fit the model until there is no NA as coefficient. For a categorical variable, it is rather difficult for every subset to contain all levels of variable. As a result, the model generates an NA coefficient for that particular variable. It is worth noting that the function computes conditional means for each state by multiplying data with coefficient matrix. Therefore, if NAs exist in the coefficient matrix, these conditional means will also become NAs. This issue is resolved by first removing any variables which have NAs coefficient from the coefficient matrix and later on performing matrix multiplication.

Results from fitting Markov switching autoregressive model in the package are described here. The function returns estimated parameters in each state. For each observation, the function provides states assignment and probability assignment in each state. The package includes a function to plot periods where the observation is in the specific state and also a function to create several plots for the residual analysis. It shows a plot of residuals against fitted values, a Normal Q-Q plot, and ACF/PACF of residuals.

References

[1] James D Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the Econometric Society*, pages 357–384, 1989.

[2] Jeff Gill and Gary King. What to do when your hessian is not invertible: Alternatives to model respecification in nonlinear estimation. *Sociological methods & research*, 33(1):54–87, 2004.