# Betasmartz Research – Confidential - Business Cycle Classification Algorithm

## Objective

The intention of this document is to introduce and outline all the steps that are involved in determination of the business cycle classification algorithm.

This algorithm provides the probability forecasting model for a particular labeling in time.

## Data Needed

Two types of data are needed, one will be the labeling that wants to be identified for the algorithm and the other will be the potential explanatory data for the label variable.

## Modelling framework definition

There are numerous classification algorithms that could provide a classification regarding the business cycle identification. However, not all the methodologies are robust and most of them are prone to overfitting. In this case the algorithm that was used is called a Bayesian Modeling Average. This technique is very robust and is not sensitive to overfitting so it almost assures from its building blocks that the model reflects the actual state of the population distribution.

### Brief introduction to BMA

The Bayesian model averaging is a technique created to adress model uncertainty in selecting a particular model structure (it is not only dependent on the regression problem).

Let me explain further with a little example:

*There is a researcher that wants to assess the size of a covariate effect on survival time with a view of designing future interventions and additionally he would like to predict the survival time for different patients. So he conducts a data-driven search to select covariates for a specific proportional hazards regression model, M\*, that will provide the framework for inference. So he checks if the model M\* fits to the data and then it proceeds to estimate the models significance, test the covariates size effects and finally make predictions. This is a standard statistical procedure but sadly it is not entirely satisfactory. Suppose there exists an alternative way to fit the proportional hazard model, M\*\*, that also provides a good fit but leads to significantly different estimated effect sizes, different standard errors and different predictions? Sadly there is no standard procedure on how to tackle this situation, proceed with the model M\* will be too risky and could lead to ambiguity. There is where the model selection procedure comes as a solution of this problem and Bayesian model averaging is a way around this problem.*

The BMA adress this problem with the following equation:

This has a practical interpretation of:

***“The probability of that one effects is effectively that size given actual data is equal to the probability of the effect given a particular model and the actual data averaged by the probability that the model is correct given the actual data”***

This will forcedly lead to a use of Bayesian transformation for averaging terms that leads to:

Where the probability depending of the model is calculated as:

This represents the integrated likelihood of model represents the vector parameters in the case of a regression will be: .

* Represents the prior density of under the model .
* is the actual likelihood.
* is the prior probability that the is the true model.

Theoretically this is a very appealing solution to the model uncertainty; sadly the actual implementation of this methodology has several difficulties as:

1. The number of models to be considered in (1) could be enormous, so the exhaustive summation could be infeasible.
2. The integrals that need to be computed are very difficult to solve numerically speaking, gladly a good Markov Chain Montecarlo engine will overcome most of the difficulties but there are still technical challenges to this.
3. The prior distributions for the models and parameters are very challenging and there is not work in there yet.
4. Choosing what class of models to average is a modeling task where there is not a final word on it yet.

### BMA implementation framework for Betasmartz

In the Betasmartz framework the BMA is used for a multi-class classifier that will provide the forecasting information on what economic cycle is yet to come and its future probability.

In the Beta framework the model structure for estimating the BMA will be a generalized linear model.

The typical structure of a generalized linear model is:

Where the link function corresponds to the probit function, so the estimation equation will be:

In the Betasmartz system the implementation is based on the standardized approach that the BMA package could provide. There are caveats to this package:

* It uses a model reduction technique that is very sharp, so there might be a very broad discrediting of the candidate models by the “Leaps Algorithm”, so some modifications are done in order to use the Occam methodologies for model discarding.
* There are ways to improve the Markov Chain Montecarlo engine; the engine is not accessible from the package so there will be limitations on the convergence of the methods.

### Step by Step implementation

The idea of this section is to outline the algorithm that is being followed to generate the probabilities for each business cycle.

### Step 1 – Data collection and initial cleansing

The algorithm filters out the data that is not complete in order to ensure that the algorithm will not have convergence problems due to missing variables.

In this data collection step there are three types of data considered to be entered:

* Level data: This data correspond to time series that are of level type, this mean that the time series have only positive values and will normally be interpreted as a real value in the economy for a good or a price(i.e. stock price index)
* Percentage data: This data correspond to time series that are of percentage level type, this mean that the variable is naturally presented as a percentage but also represents level information. (i.e. inflation, spread)
* Percentage change data: This data represent the percentage level change, this means that it represents the change in other time series and it make more sense to express the variable like a change that like a level (Normally this type of variables are economically reported as changes).
* Label variable: This variable is the one that has all the labels to be explained by the model.

### Step 2 – Data transformation

In order for the data to be statistically more informative there exists certain transformation that could be done to the same, there is one peculiarity, the transformation depends on the data so all the transformations could not be done to all the data.

* Level data: Since the domain of these variables is restricted to the positive real numbers that are greater than 1, a Box-Cox power transform of logarithm could be applied without complications. After that it could be differentiated to obtain some nicely behaved logarithmic returns.
* Percentage data: The data of this group has is restricted to the interval: this restricts the transformation that could be performed in these variables. As shown in Meucci “Risk and Asset Allocation”, the market invariant for this type of variable is the absolute difference of the same.
* Percentage change data: This data will be the equivalent of the above transformed, so it should be used in as it comes.

### Step 3 – Data cleansing and pre-processing for classification

The data now is ready for training the BMA classification model. Before training the model, a proper “forecasting adjustment” needs to be done. As the intention of the model is to forecast the probability of the next X-months periods belonging to a particular business cycle stage an adjustment of the true data should be made.

The label data should be lagged X-periods in order for the actual covariate data match the lagged one, this will mean that the classifier will be trained with a label that is X-months in the future. In this manner the trained models will be prepared to naturally forecast the label variable.

This process generates a decaying in the data (X-periods less of data), consequently a new data cleansing should be made deleting the incomplete periods.

### Step 4 – Decoding the Label data into Dummy

It is important to take into consideration that the categorical data should be coded as a set of dummy variables; this is made in the algorithm before the training period.

The dummy encoding in the algorithm won’t be having a base case and each dummy has a direct interpretational meaning.

### Step 5 – Training the BMA Probit models in a one-vs-all classification