Thesis Plan: Research question and methodology

Ricardo Semião e Castro

1 Research Question

Summarized question:

> What is the relationship between regimes' characteristics and forecasting performance, across different regime-switching models and DGPs?

Sub-questions:

1. Does regimes' characteristics matters for performance? [Assumedly yes].
2. How does regimes' characteristics matters?
   - In general, which characteristics matter, which don't?
   - Does the answer for the above change across different contexts? Specially, across DGPs and models?
   - What is important about each characteristic: the absolute value, the difference with the other regimes, the difference with the DGP?
   - Are both within-regime performance and the global performance of the model affected?
3. What practical use can be made of this information?
   - How does this discussion contribute to differentiating each RS model?
     - Is there evidence for a universal approximator? That is, is there a model that performs well in all cases?
   - Are there practical recommendations for metrics that econometricians should calculate when creating RS models?

Claims to defend:

- About the literature:
  - There is demand for more studies on RS model performance. [Very generic, we could go straight to the next one].
  - Little attention has been given to the relation between the characteristics of the regimes and the forecasting performance of the model.
  - More could be learned about in which context each RS model performs better.
- About why regime identification is important:
  - It could provide researchers with tools to better understand the expected performance of their models, by analyzing the characteristics of the estimated regimes.
  - It offers further information on the differences between RS models, and in which types of regime structures each performs better.
  - [Essentially, if my exercise yields useful practical recommendations, it is important; otherwise, not as much (and that is a result in itself).]

1.1 Possible Exercises

There are several possible exercises. I will focus on those whose results depend only on information available to the econometrician. Among these, the idea below is the most promising.

For each metric, calculate some measure of its dispersion across all regimes of an (estimated) series, and study its relationship with the model's performance. Two possible extensions are:

- Analyze the dispersions in terms of the true dispersion of the DGP, yielding information on which characteristics' dispersion is most important to match.
- Analyze the dispersions in terms of the number of estimated regimes, yielding information on when increasing the number of regimes is useful for each 'level' of regime dispersion.

There are several other exercises, but as will become clear in Section 2.2, they all share the same foundation, differing mainly in how the calculated metrics are aggregated and compared.

All exercises start from a matrix with each row being a regime - rows indexed by (p, m, s, r) (DGP, model, simulation, and regime) - and columns a metric, be it regime-conditional characteristics or model performance metrics. Then, some exercises might aggregate the rows by (p, m, s) only. It is from these matrices that analyses are performed.

2 Methodology

First, in the section below, I define the objects of interest. Then, the general algorithm to address the exercises is derived.

2.1 Generating Processes, Models, and Metrics

Let $y_t \in \mathbb{R}$ denote the series of interest, and $r_t \in \mathbb{N}$ denote the (categorical) regime, at time $t \in 1 : T$[1], $T \in \mathbb{N}$.

A data generating process (DPG) can be written in terms of a pair regime generating process (RGP) and a series generating process (SGP). These are functions with parameters Θ_r and Θ_s, respectively, such that:

$$\begin{array}{lllllll} r_t & = rgp(& t, & y_{1:(t-1)}, & r_{t-1} & ; \Theta_r &) \\ y_t & = sgp(& t, & y_{1:(t-1)}, & r_t & ; \Theta_s &) \end{array} \tag{1}$$

Note that the DGP can then be written as:

$$\begin{array}{lllll} y_t & = sgp(& t, & y_{1:(t-1)}, & rgp(t, y_{1:(t-1)}, r_{t-1}; \Theta_r) \; ; \Theta_s \quad) \\ y_t & = dgp(& t, & y_{1:(t-1)}, & r_{t-1} \qquad\qquad\qquad\qquad\quad ; (\Theta_r, \Theta_s) \quad) \end{array} \tag{2}$$

The random error lies within the functional form of the DGP. But, for our purposes, it is more useful to write the DGP as function that receives a set of random error vectors - possibly one for each regime - and returns the series and the regimes:

$$(y_{1:T},\ r_{1:T}) = dgp(\varepsilon_{1:T}\ ;\ (\Theta_r, \Theta_s)) \tag{3}$$

[1]Let $a : b := \{a, a + 1, \dots, b\}$ for $a \leq b \in \mathbb{Z}$, and $y_{a:b} := y_t : t \in a : b$.

Also, consider a model *mod* as a function with parameters Θ_s that generates the fitted values and h-step ahead predictions of the series and regimes:

$$(\hat{y}_{1:T},\ \hat{r}_{1:T}) = mod(y_{1:(T-h)},\ r_{1:(T-h)}\ ;\ \Theta_s) \tag{4}$$

Notably, the number of regimes is included within the parameter set of a DGP/model. From now on, I will refer to a DGP/model as a combination of functional forms and parameter sets. Let P denote the set of DGPs, and M the set of models.

One of the first challenges is to define comprehensive sets of RGPs, SGPs, and models to be studied. This will be addressed in the future, after the set of exercises to be conducted is defined.

Finally, there will be defined a set of regime-conditional metrics, i.e.: functions that take a contiguous subset of the serie, associated with an 'instance' of a regime, and return a real number. Let M denote the set of metrics. Note that one could consider more general metrics that consider the whole pair of.

2.2 General Algorithm

To answer the research question, there are five main steps:

1. Generate random errors given the DGP.
2. For each DGP and simulation, generate $(y_{1:T}, r_{1:T})$.
3. For each model, DGP, and simulation, obtain $(\hat{y}_{1:T},\ \hat{r}_{1:T})$.
4. For each model, DGP, and simulation, compute each metric.
5. Aggregate the metrics and compute the comparisons/statistics of interest.

The order and nesting used to organize the steps can be changed to optimize performance (e.g., via parallelization) and/or modularity. Below, I describe the chosen algorithm. Its implementation is done in the R language.

2.2.1 Generating Random Errors

Let $s \in 1:S$, $S \in \mathbb{N}$ be the simulation index. For each DGP, we need to create S sets of random error vectors, each of size $1:T$. For most DGPs, each set will be a single vector, but others define a different error generating process for each regime. The nesting order does not matter, and the errors were generated for each pair (dgp, s), in parallel, using TRNG. Let E denote the set of all errors.

For each $p \in 1:|P|$ and $s \in 1:S$, let $\mathrm{E}_{p,s}$ denote the set of errors generated for the p-th DGP and the s-th simulation. Similar definitions will be used for similar collections throughout this document.

2.2.2 Generating Series

Again, for each DGP, we need to generate S series. The nesting order does not matter, and the errors were generated for each DGP and then for each error, as this order reduced parallel process communication overhead.

Let Y and R denote the sets of generated series and regime series. For each p and s, the elements $Y_{p,s}$ and $R_{p,s}$ are computed given $\mathrm{E}_{p,s}$:

Initialize Y and R

for $p = 1$ **to** $|P|$ **do**
    Spawn a new parallel task
    $dgp \leftarrow P_p$
    for $s = 1$ **to** S **do**
        $Y_{p,s},\ R_{p,s} \leftarrow dgp(\mathrm{E}_{p,s})$
    end for
end for

2.2.3 Estimating Models

Now, for each simulation, we estimate each model, generating the sets $\hat{Y}$ and $\hat{R}$. The nesting order is the same as above, for consistency, but with an additional inner loop for the models.

Initialize $\hat{Y}$ and $\hat{R}$

for $p = 1$ **to** $|P|$ **do**
    Spawn a new parallel task
    for $m = 1$ **to** $|M|$ **do**
        $mod \leftarrow M_m$
        for $s = 1$ **to** S **do**
            $\hat{Y}_{p,m,s},\ \hat{R}_{p,m,s} \leftarrow mod(Y_{p,s},\ R_{p,s})$
        end for
    end for
end for

2.2.4 Computing Metrics

In general, most considered exercises will involve series-wide metrics, as is the case for the dispersion metrics described in Section 1.1. Performance metrics also follow this logic. Thus, for each (p, m, s), we compute a vector of metrics, which can then be aggregated by rows into a matrix for various analyses. Let X denote such a matrix[2].

Initialize X

for $p = 1$ **to** $|P|$ **do**
    Spawn a new parallel task
    for $m = 1$ **to** $|M|$ **do**
        for $s = 1$ **to** S **do**
            for $\mu = 1$ **to** M **do**
                metric $\leftarrow D_d$
                $X[(p, m, s),\ d] \leftarrow \text{metric}(\hat{Y}_{p,m,s}, \hat{R}_{p,m,s}, Y_{p,m,s}, R_{p,m,s})$
            end for
        end for
    end for
end for

[2]Whose row and column dimensions are indexed, in order, by the operator $[(p, m, s),\ d]$.

2.2.5 Comparisons and Results

This step of the algorithm will vary the most depending on the chosen exercise. I will formalize it once the set of exercises is defined.