Combinação de previsões

J. Renato Leripio

11 de julho de 2019

Em posts anteriores apresentei algumas metodologias capazes de melhorar previsões. Em particular, falei um pouco sobre **bagging** — uma técnica que estima um modelo específico sobre variações da série original e, em seguida, computa a média/mediana destas previsões (ver aqui) — e sobre **rectify** — uma abordagem que considera eventuais informações contidas nos erros de previsão (ver aqui). Em todos os casos considerei apenas um único modelo para realizar as previsões. Porém, com frequência temos à disposição mais de um modelo para a mesma variável. Neste caso, o que fazer?

Uma estratégia muito comum consiste em combinar previsões de diversos modelos. Isso não é novidade e vem sendo explorado desde o paper seminal de Bates e Granger em 1969, *"The combination of forecasts"*, com resultados bastante promissores. Entretanto, a estratégia parece ter definido um novo padrão no campo de previsões uma vez que **12 dos 17 modelos mais acurados na competição M4 foram combinações**. Isto se deve, em grande medida, ao menor risco de repousar exclusivamente em um modelo mal especificado ou com baixa capacidade de adaptação a novos eventos.

Existem diversas estratégias para combinar previsões. As abordagens mais comuns utilizam alguma medida como média simples/mediana ou fazem uso de alguma combinação linear das previsões, conforme a expressão abaixo:

$$y_t^{FC} = \alpha_1 y_{1,t}^{FC} + \alpha_2 y_{2,t}^{FC} + ... + \alpha_k y_{k,t}^{FC} = \sum_i^k \alpha_i y_{i,t}^{FC}$$

em que $y_{i,t}^{FC}$ é a previsão do modelo i para o período t.

Os pesos, α_i, podem ser definidos de diversas formas. Em geral, considera-se alguma medida do erro de previsão de cada modelo, dando menor peso ao modelo que historicamente errou mais; ou então obtém-se os pesos através da minimização de alguma função perda (RMSFE, MSFE, etc). Métodos mais sofisticados permitem, por exemplo, mudanças no valor dos parâmetros ao longo do tempo e até mesmo a utilização de algoritmos de Machine Learning para aprender o valor destes parâmetros.

Neste post, vou considerar quatro modelos univariados: **ETS**, **CES** (complex exponential smoothing), **ARIMA** e **DOTM** (dynamic optimised theta). A primeira abordagem para combinação será computar a mediana das projeções individuais. A escolha conjunta destes modelos e da mediana para combinação não é arbitrária, mas segue a proposta de Petropoulos e Svetunkov (2019, IJF), a qual, embora simples, obteve excelentes resultados. A segunda abordagem para combinação considera os mesmos modelos e pesos α_i que minimizam o RMSFE (root mean squared forecast error), de acordo com:

$$min_{\alpha_i} \frac{\sum_{t=1}^T (y_t - \sum_i^k \alpha_i y_{i,t}^{FC})^2}{n}$$

Os leitores mais familiarizados vão notar que este problema pode ser reduzido à uma regressão linear entre o y observado em t e as projeções de cada um dos modelos para o mesmo t. Em especial, ao elevar ao quadrado os resíduos da regressão, calcular a média e extrair a raiz, obteremos o RMSFE. Entretanto, para deixar o tratamento mais geral, vou considerar o problema de otimização acima. Adicionalmente, para que os coeficientes α_i sejam não-negativos e somem um, vou aplicar uma transformação sobre eles utilizando a função **softmax**. O objetivo é deixar mais intuitiva a noção de pesos. Portanto, os coeficientes α_i padronizados serão dados por:

$$\bar{\alpha_i} = \frac{e^{\alpha_i}}{\sum_i^k e^{\alpha_i}}$$

Por fim, vamos comparar os resultados dos modelos individuais com aqueles obtidos através das combinações. Antes de começarmos, é preciso chamar atenção para três pontos. Em primeiro lugar, como quase tudo em forecasting, as evidências que apontam vantagem de previsões combinadas sobre as individuais são obtidas ao aplicar o método sobre um grande conjunto de séries. Ou seja, a superioridade dos métodos de combinação vale na média e não necessariamente para todos os casos particulares.

Em segundo lugar, a combinação de modelos pressupõe que os modelos gerem previsões não-viesadas. Caso contrário, o viés de um dos modelos acaba contaminando a previsão combinada. Por esta razão, incluir uma constante no problema de otimização pode melhorar o resultado, uma vez que captura algum eventual viés.

Por último, é preciso ter cuidado ao avaliar o poder preditivo dos modelos. No caso da combinação linear, como precisamos gerar previsões para calcular o valor dos pesos α_i, vamos separar uma parte da amostra para validação. Para ficar mais claro, faremos o seguinte:

1. A amostra de treino será utilizada para computar as projeções de cada método;
2. O peso de cada método será computado tendo como referência o poder preditivo sobre a amostra de validação;
3. Os pesos obtidos na etapa 2 serão utilizados para combinar as projeções obtidas na amostra de treino ampliada (treino+validação).
4. Estas projeções da etapa 3 serão comparadas com os valores da amostra de teste.

Entendido o exercício, abra o R e acompanhe!

Passo 1: carregar os pacotes necessários, importar os dados e definir as amostras. Para esta aplicação, vamos utilizar a série do núcleo do IPCA EX-3, calculado pelo BCB (SGS 27839).

```r
# 1. Carregar bibliotecas

library(tidyverse)
library(rbcb) # Para instalar: devtools::install_github("wilsonfreitas/rbcb")
library(forecast)
library(smooth)
library(forecTheta)
library(knitr)

# 2. Importar dados dados

dados <- rbcb::get_series(list("ipca_ex3" = 27839),
                          start_date = "2006-07-01",
                          end_date = "2019-05-01")

dados_ts <- ts(dados$ipca_ex3, start = c(2006,7), freq = 12)

# 3. Separar as amostras (cerca de 55% para treino, 30% para validação e 15% para teste)

dados_treino <- window(dados_ts, end = c(2013,7))

dados_valida <- window(dados_ts, start = c(2013,8), end = c(2017,5))

dados_teste <- window(dados_ts, start = c(2017,6))
```

Passo 2: realizar as projeções individuais para cada modelo com horizonte igual ao período de validação. Estes dados serão utilizados para estimar os pesos.

```
modelo_i <- list(

"ets" = function(x,h) forecast(ets(x, lambda = "auto"), h = h),

"ces" = function(x,h) forecast(smooth::auto.ces(x), h = h),

"arima" = function(x,h) forecast(auto.arima(x), h = h),

"dotm" = function(x,h) forecTheta::dotm(x, h = h)

)

fc_i <- purrr::invoke_map(.f = modelo_i,
                          .x = list(dados_treino),
                          h = length(dados_valida))

fc_i_mean <- purrr::map_dfc(.x = fc_i, .f = function(x) x[["mean"]]) %>%

             dplyr::mutate(y_valida = dados_valida)
```

Passo 3: computar os parâmetros que minimizam a RMSFE e normalizá-los. Aqui, como eram apenas 4 modelos eu abri o somatório para ficar mais claro. Para o caso de um conjunto grande de modelos, o ideal é substituir por uma operação matricial.

```
msfe_comb <- function(x){

alpha_ets <- x[1]
alpha_ces <- x[2]
alpha_arima <- x[3]
alpha_dotm <- x[4]

((fc_i_mean$y_valida - alpha_ets*fc_i_mean$ets - alpha_ces*fc_i_mean$ces -
    alpha_arima*fc_i_mean$arima - alpha_dotm*fc_i_mean$dotm)^2) %>%

  mean() %>%

  sqrt()

}

pesos <- optim(c(1,1,1,1), msfe_comb)

pesos_norm <- round(exp(pesos$par)/sum(exp(pesos$par)), 3)
```

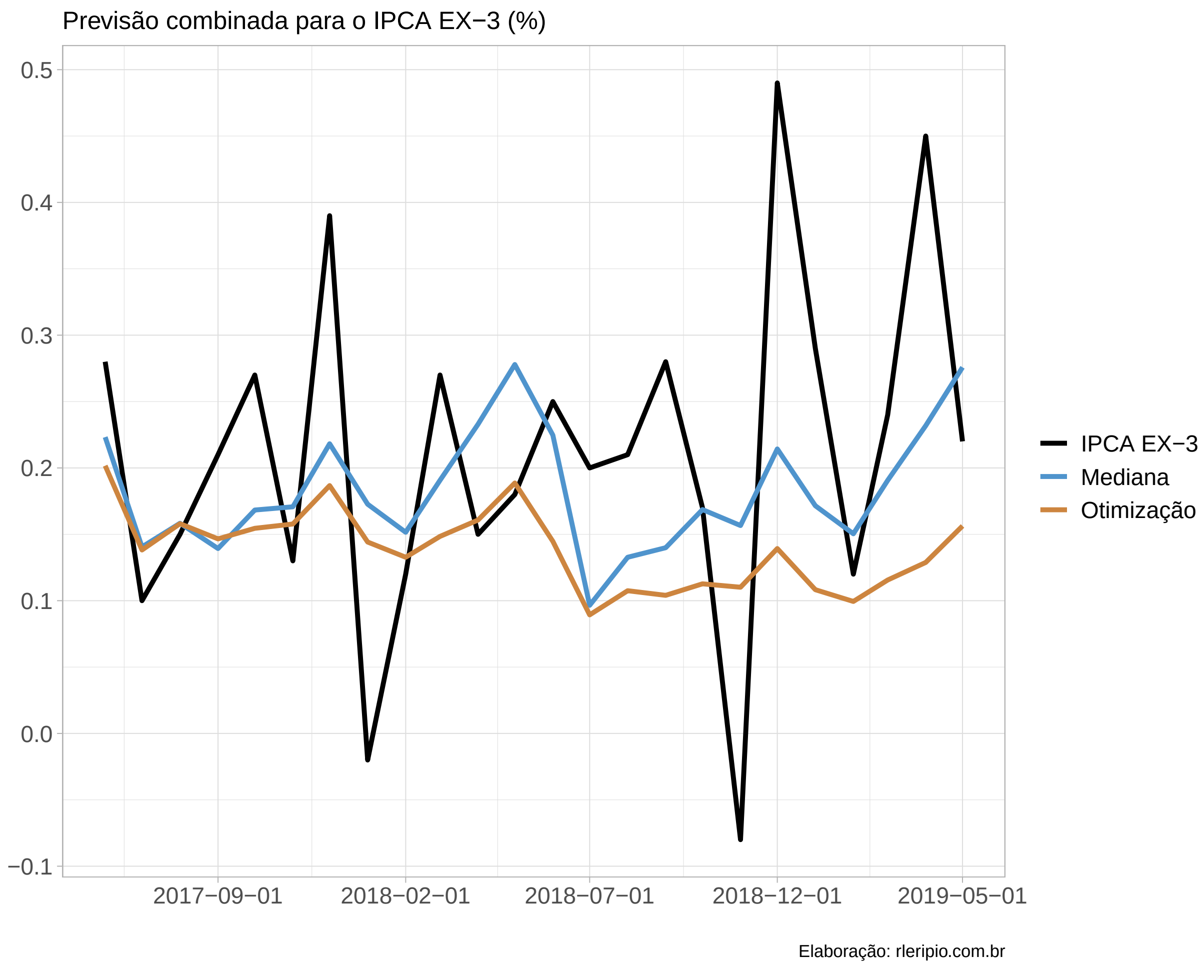
Passo 4: realizar as projeções combinadas utilizando os parâmetros estimados e a mediana. Em seguida, comparar com as realizações da amostra de teste.

```
dados_treino_amplo <- window(dados_ts, end = c(2017,5))

fc_i_amplo <- purrr::invoke_map(.f = modelo_i,
                                .x = list(dados_treino_amplo),
                                h = length(dados_teste))

fc_i_mean_amplo <- purrr::map_dfc(.x = fc_i_amplo,
                                  .f = function(x) x[["mean"]]) %>%

dplyr::mutate("y_teste" = dados_teste) %>%

dplyr::rowwise() %>%

dplyr::mutate("Mediana" = median(c(ets,ces,arima,dotm))) %>%

dplyr::ungroup() %>%

dplyr::mutate("Otimização" = pesos_norm[1]*ets + pesos_norm[2]*ces +
                             pesos_norm[3]*arima + pesos_norm[4]*dotm)

fc_i_acc <- fc_i_mean_amplo %>%

  dplyr::summarise_at(vars(-y_teste),
                      funs(forecast::accuracy(., y_teste)[, "RMSE"]))
```

As medidas de acurácia são exibidas na tabela abaixo. A estratégia de combinação através da mediana apresentou o melhor resultado, superando ligeiramente os modelos CES e ETS. A combinação através de otimização, por sua vez, não foi capaz de bater todos os modelos individuais. Mais especificamente, o bom desempenho do modelo arima no período de validação fez com que este recebesse um peso mais elevado. Entretanto, essa vantagem não se materializou no período de teste. Isto reforça a necessidade de reavaliar modelos e estratégias de tempos em tempos, sobretudo quando ocorrem mudanças estruturais na série de interesse – como foi o caso do IPCA EX-3. Por outro lado, também reforça a capacidade de estratégias que utilizam medidas de tendência menos sensíveis a extremos – como a mediana – em responder melhor a ambientes mais incertos.

Modelo	RMSFE
Mediana	0.121
ces	0.123
ets	0.123
dotm	0.131
Otimização	0.143
arima	0.184

O gráfico abaixo apresenta as observações para o IPCA EX-3 da amostra de teste e as previsões pontuais geradas pelos dois métodos de combinação. Vale ressaltar que uma análise mais rigorosa levaria em conta também a performance para cada horizonte. Também fica claro ao observar o gráfico que os picos e vales mais pronunciados podem ter um papel relevante sobre a magnitude da medida RMSFE. Uma boa prática seria considerar medidas alternativas, sobretudo aquelas mais robustas a este tipo de situação. Pretendo abordar isso em algum momento.



Por fim, cabe notar que intervalo de confiança nesses casos não é trivial, uma vez que é preciso obter uma expressão para a variância da combinação das previsões, o que requer computar as covariâncias entre os erros dos modelos. Uma solução conservadora é utilizar o intervalo mais amplo dos modelos individuais, porém não me agrada muito. Talvez possamos voltar nesse ponto em uma próxima oportunidade.

Sugestão: Para os interessados em aplicar metodologias de combinação de previsões, existem alguns pacotes disponíveis para R. Dois deles (opera e ForecastHybrid) são tratados neste post do Rob. Hyndman: https://robjhyndman.com/hyndsight/forecast-combinations/

Os códigos dos exercícios encontram-se disponíveis no repositório do blog no github.

Ficou alguma dúvida ou tem sugestões? Entre em contato!

Siga nossa página RLeripio – Economia e Data Science no Facebook e fique sabendo de todas as nossas publicações!

Aviso legal: Todo o conteúdo desta página é de responsabilidade pessoal do autor e não expressa a visão da instituição a qual o autor tem vínculo profissional.