Using Ensemble of Decision Trees to Forecast Travel Time

José P. González-Brenes

Guido Matías Cortés

What to Model?

Goal

- Predict travel time at time *t* on route *s* using a set of explanatory variables

- We apply our estimation procedure separately for each route *s*, and for each prediction time horizon (i.e. 15 min ahead, 30 min ahead, etc)
  - We call each of these a "model"
  - We will have 610 models (61 routes x 10 prediction time horizons)

Variables we considered relevant to predict travel time:

- What time is it? (Hour + (Minute/60))
- What is the date? (Month + (Day/31))
- What day of the week is it?

Additional variables we considered relevant to predict travel time:

- What is the recent trend for the travel time on this route?
  - Growth rate over the two most recent periods: $y_{t-1} - y_{t-2}$
  - Growth rate between the average travel time in the two most recent periods and the average travel time in the two periods immediately preceding them: $(y_{t-1} + y_{t-2})/2 - (y_{t-3} + y_{t-4})/2$

- What is the most recent observation for the travel time on neighboring routes?

Intuition for the additional variables:

- Example:
  We would like to predict travel time at time *t* for route A:
  - If the travel time on route A has been growing steadily over the past few minutes, it's likely that it will keep growing at time t.
  - If the most recent data for route B show longer travel times, and route B is next to route A, it's likely that, as traffic moves along to route A, travel times on route A will increase.

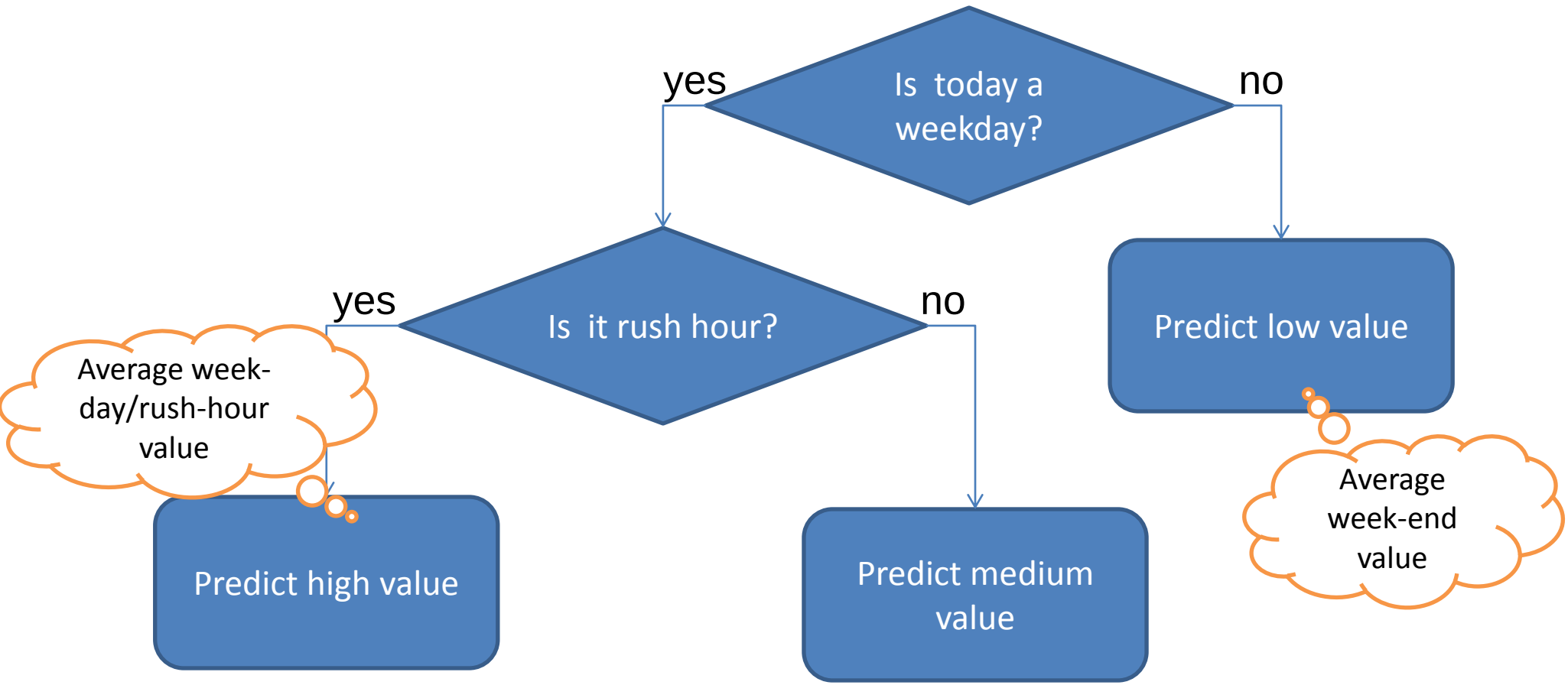
- Note: We eliminate observations corresponding to holidays from our sample.

Brief Introduction to Ensemble of Decision Trees

AKA Random Forests™

Introduction

- A decision tree:

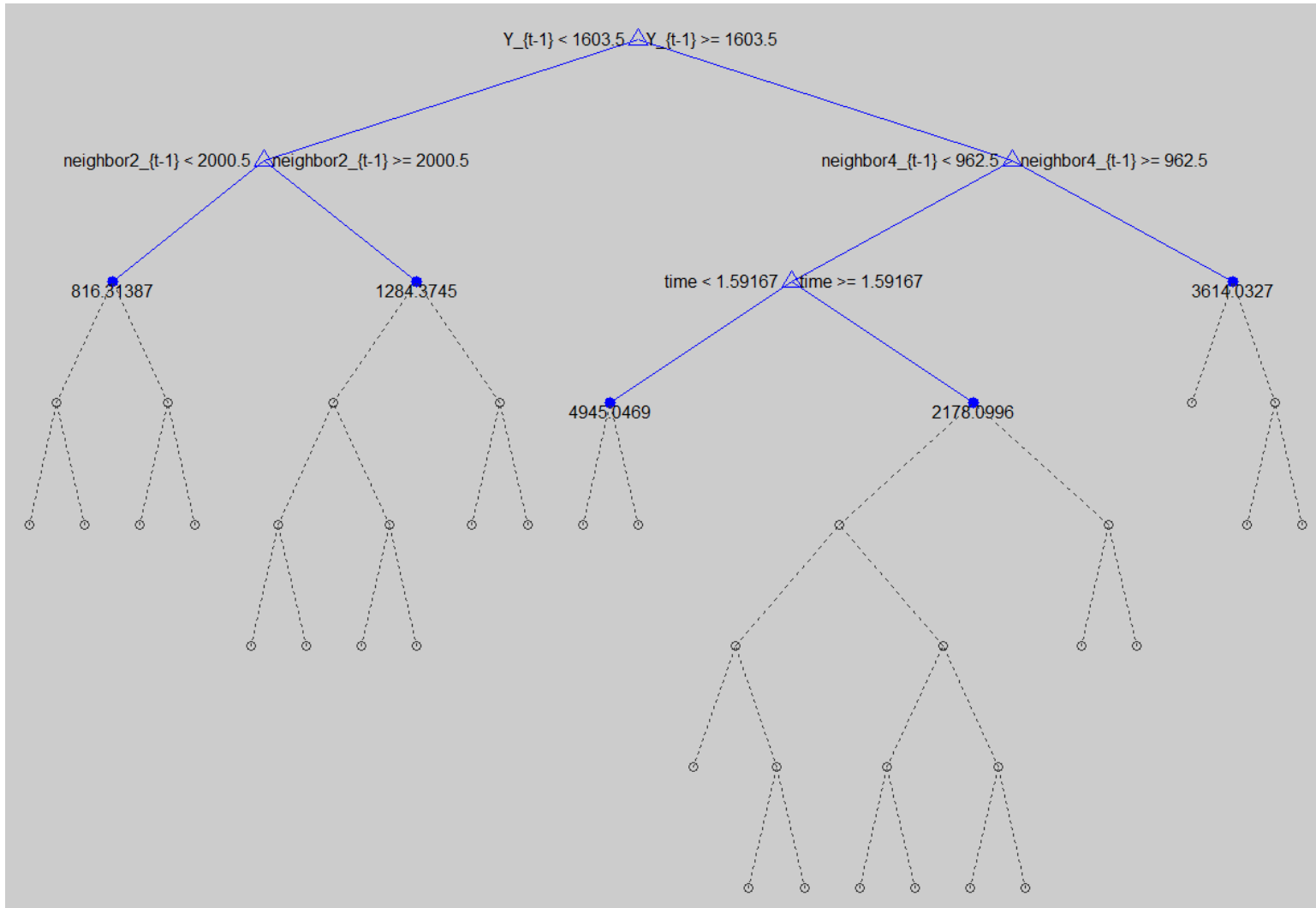


Growing Decision Trees

- What questions to ask? Let the data talk!

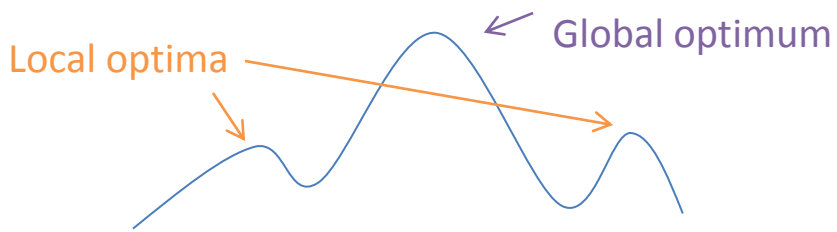
- Matlab offers the functionality to grow decision trees
  - At each step, create a branch using the “explanatory variable+threshold” combination that improves the forecast the most
  - Example:
    - is time < 5 pm? is time < 6pm? is time < 7pm? etc …
    - is month < 2? is month < 3? etc …
  - Stop when the bin only <= 5 observations

A decision tree grown by Matlab



From Decision Tree ➔ Random Forest

- Forest = Multiple decision trees
  - The output of every decision tree in the “forest” is averaged
- What’s random in a Random Forest?
  - Random subset of the explanatory variables
  - Random subset of the training data
- Why random?
  - Avoids modeling noise
  - Decision trees are greedy: Using the best split at every point might overlook better solutions in the long-term (stuck at local optimum)



Why random forests are great:

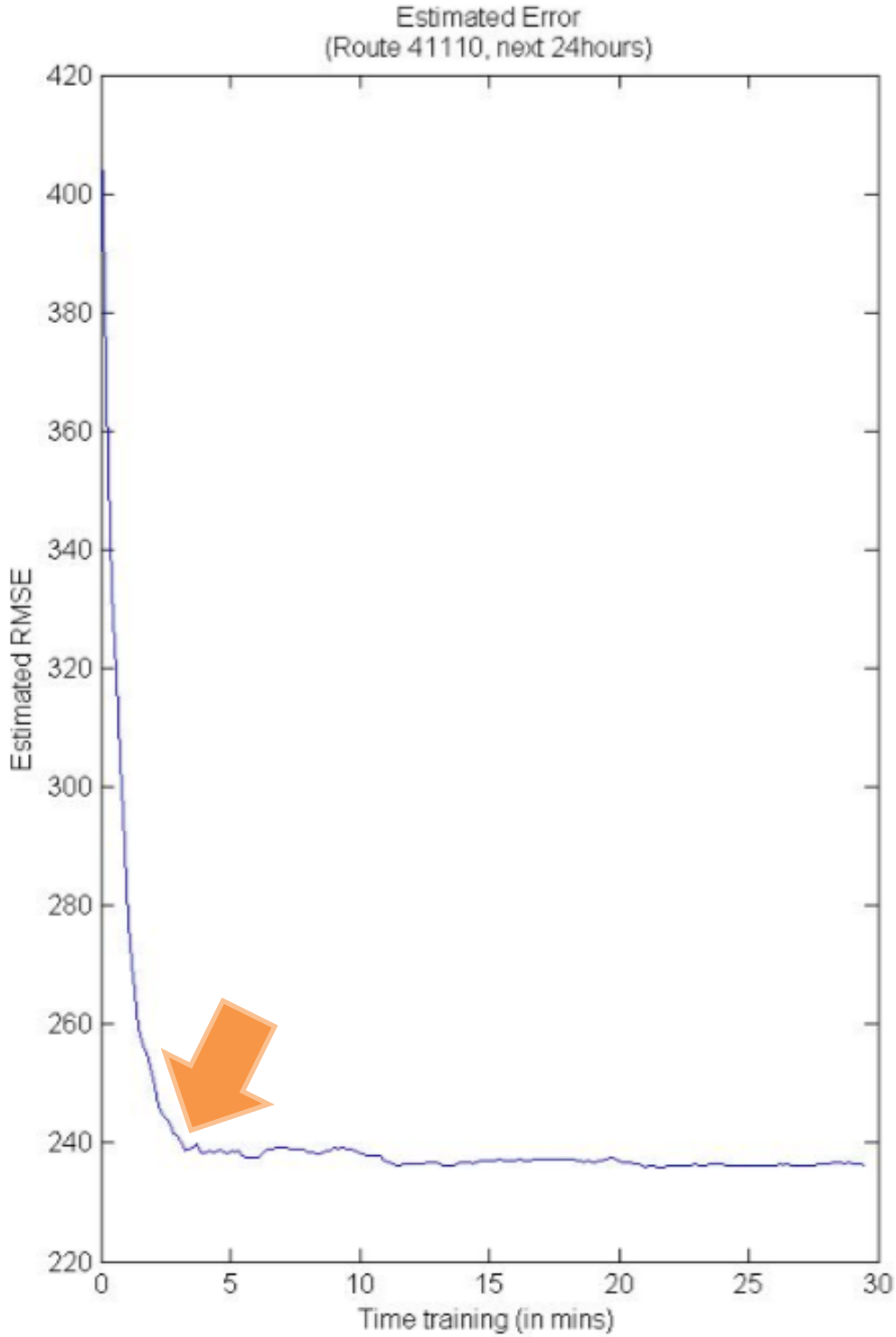
- Random forest is non-parametric and non-linear:
  - Does not impose a specific relationship between our explanatory variables and our predictions.

- Linear regression would impose a specific relationship between the explanatory variables and the predicted travel time.
  - Random forest is much more flexible. We do not need to make any special assumptions, or arbitrary decisions on how to cut the data. These decisions are all made by the random forest procedure.

- Linear regression would estimate effects of each explanatory variable separately.
  - Random forest incorporates **interactions** between all the explanatory variables. It makes predictions for specific combinations of e.g. day of week, month and time.

Making new predictions is fast!

- We get the impression that there's a concern that our approach is "slow"
- Our approach is decoupled:
  - Making new predictions – Really fast!
    - 610 predictions in <4 seconds!
  - Training the model (growing the forests) – Can take some time
    - Between 20-60 mins per model


Estimated Error
(Route 41110, next 24hours)
Estimated RMSE
420
400
380
360
340
320
300
280
260
240
220
0
5
10
15
20
25
30
Time training (in mins)

Our Code

90 lines of code in a few slides

Data Preparation

- Arguments of Matlab function `forecast_segment` are:
  - segment_number: 1..61 specifying what route (41160, 41155, etc)
  - time_range: 1..10 specifying prediction horizon (15-min ahead, 30-min ahead, etc.)
- Lines 1-29 read the “training data”, and set-up some variables
  - We only used RTAData.txt
  - If replicable results are necessary, seed the random number generator

Choices that need to be made

- Number of trees:
  - Increasing the number of trees increases accuracy, but also increases computation time
- Number of neighbors:
  - Including recent travel times on neighboring routes might improve accuracy, but including too many neighbors might decrease it

Two methods

Our final algorithm combines two methods. Both methods include time, date, and day of week as explanatory variables

Method A:

- 200 trees
- Additional explanatory variables:
  - Most recent observation for 2 neighboring routes

Method B:

- 300 trees
- Additional explanatory variables:
  - Most recent observation for 4 neighboring routes
  - Recent trend in travel time (growth rates, as explained earlier)

Secret recipe: Use both methods!

- We tried Method A and Method B, and discovered that for some route segments, Method A performed better, while for others, Method B performed better.
- Our final algorithm uses:
  - Method A for route segments 40105-41160
  - Method B for route segments 40010-40100

Encoding Explanatory Variables

- Lines 31-61 encode the explanatory variables into two matrices:
    - Xtrain
    - Xtest
- The output variables are encoded as:
    - Ytrain

Training the model (aka: Growing a Forest)

- Lines 70-79 grows the forest:

```
model =  TreeBagger(TREES, Xtrain, Ytrain,
'method', 'regression','oobpred','on');
```

Making new predictions

```
yhat = predict(model, Xtest);
```

Possible improvements

- Further exploring combinations of trees and neighbors, and their performance on different route segments → might improve the secret recipe further!
- Explore using historical data, and data on failing circuits
- Use weather information as an additional explanatory variable
- Generate predictions for holidays