Gradient Boosting Trees to Predict Store Sales

Maksim Korolev, Kurt Ruegg
Stanford University

Goal

Predict store sales for Rossman from August 1, 2015 to the September 17, 2015 with lowest root mean squared percent error (RMSPE).

Data

- 1115 German Stores
- Various features including Promo, State and School Holiday, and Competition
- Sales from 2013

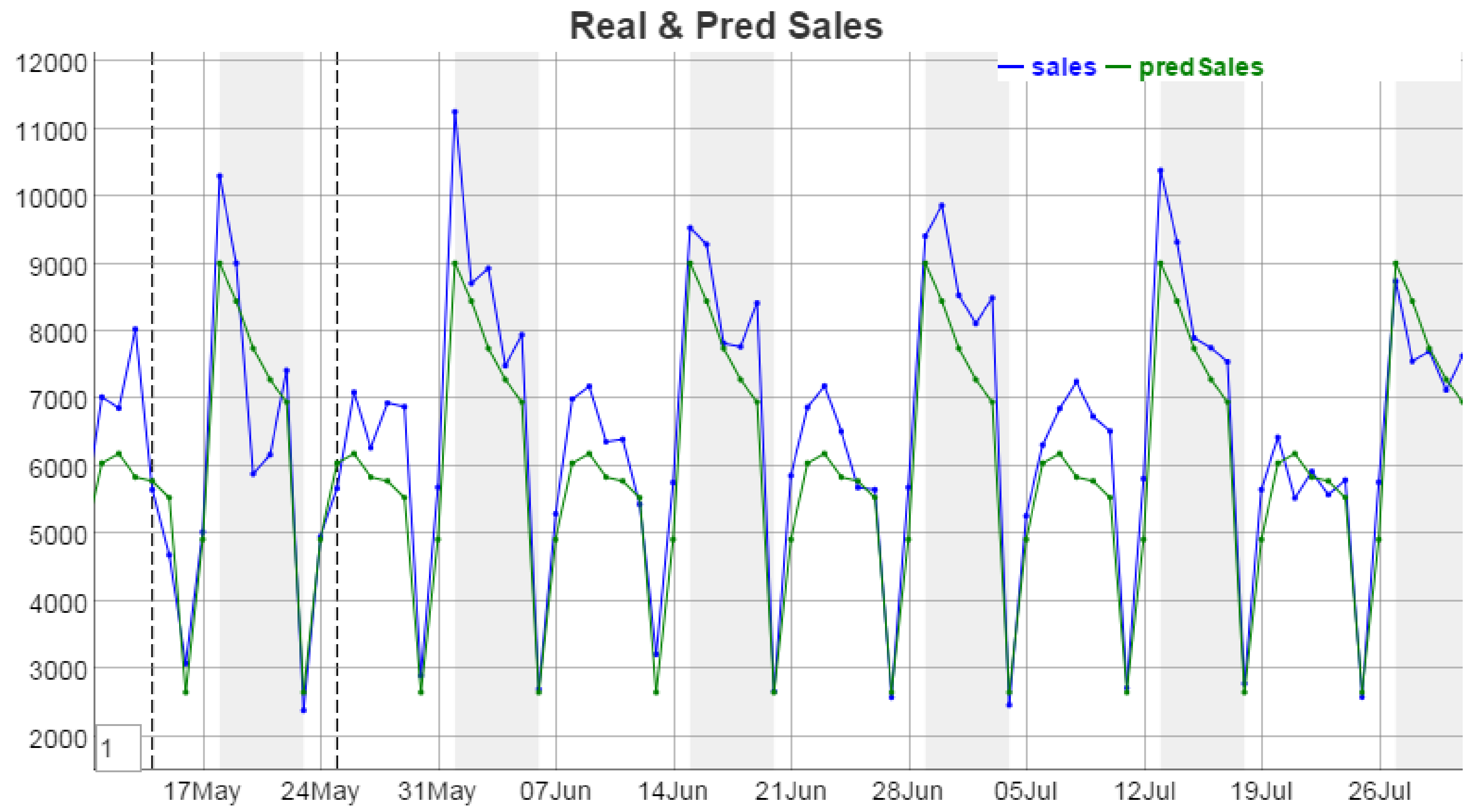


Figure 1. Example of Sales. Highlighted portions are portions where promo occurs.

Initial Models

- OLS Model with features picked by hand.
- Mean Guess model (predictions shown in Figure 1, equation in Figure 2).

$$h(x) = \frac{\sum_{i=1}^{n} y^{(i)} \prod_{j \in A} 1\{x_j^{(i)} = x_j\}}{\sum_{i=1}^{n} \prod_{j \in A} 1\{x_j^{(i)} = x_j\}}$$

Figure 2. Our mean guess model takes the mean of the sales in 2013, 2014, and 2015 for the day of the year and store of interest.

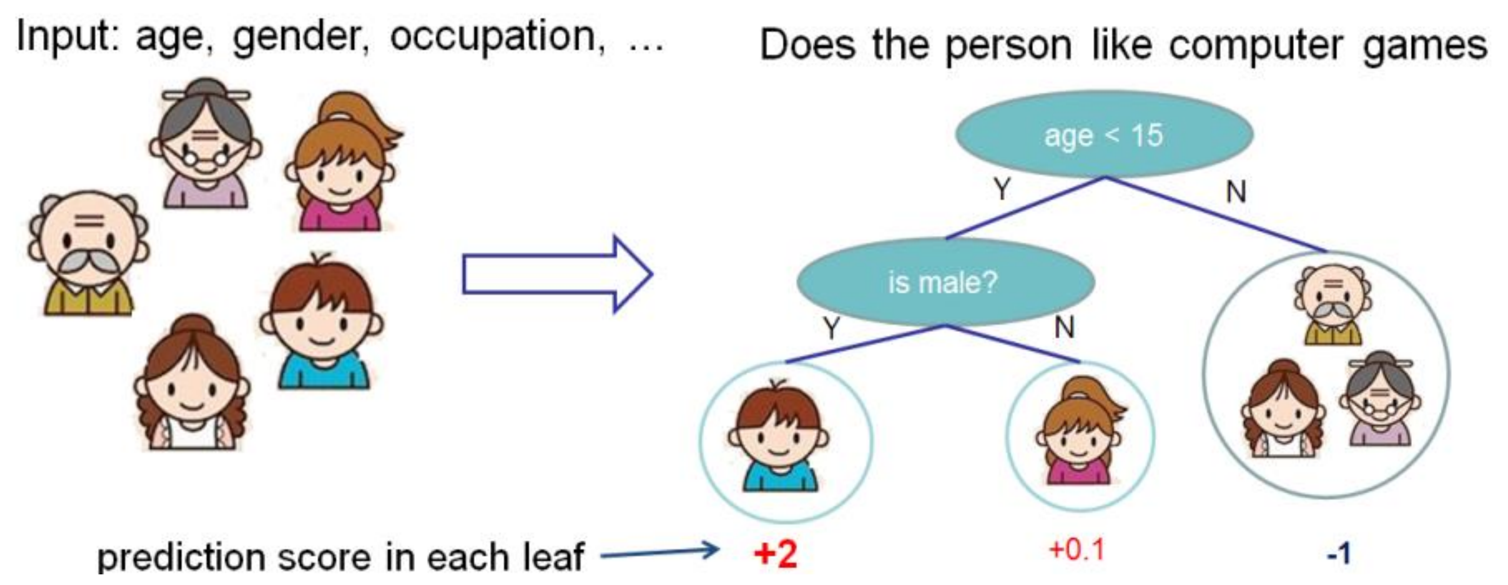
Base Model	RMSPE
OLS	0.2586
MG	0.1915

Figure 3. RMSPE for each baseline model.

Using Gradient Boosting Trees

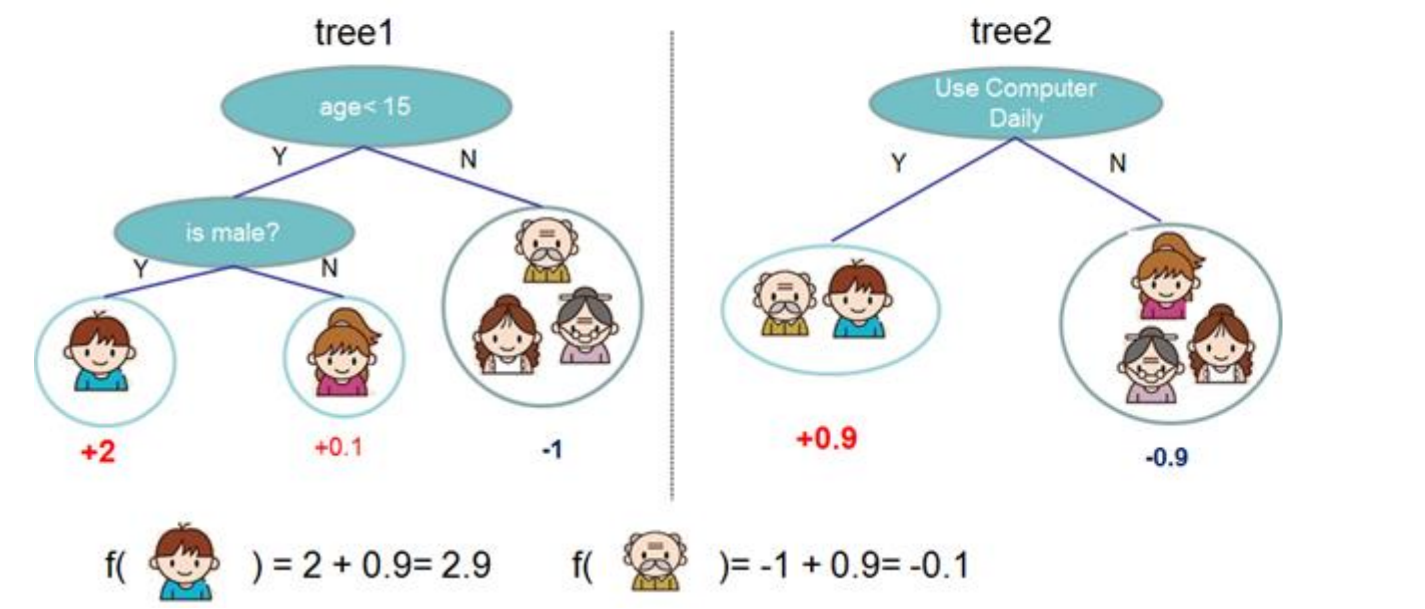
Mean Guess Model is similar to a decision tree! How does a decision tree work?

A decision tree is a tree where each node splits the input space of the node among its children.


Input: age, gender, occupation, ...
Does the person like computer games
age < 15
Y
N
is male?
Y
N
prediction score in each leaf
+2
+0.1
-1

In our simple mean guess decision tree, we performed a multiway split along 1115 stores and then did a 365 way split on each day of the year to produce our result. We then took the mean of the members of each leaf to make our predictions.

A single decision tree is usually not strong enough for prediction, so we use multiple trees.



We choose to use gradient boosting trees. This decision tree algorithm has been shown to perform the best once optimized. Specifically, this method is an example of **boosting**, which combines a number of **weak learners** into a **strong learner**. This is done by simply summing all of our weak learners.

Each of our weak learners is created by selecting the tree that minimizes our objective function. Our objective function is composed of two terms: the **loss function** and the **regularization function**. The loss function works to decrease the bias of our tree by seeking to improve the training error, while the regularization function combats high variance by punishing overfitting.

Initial Results of GBM

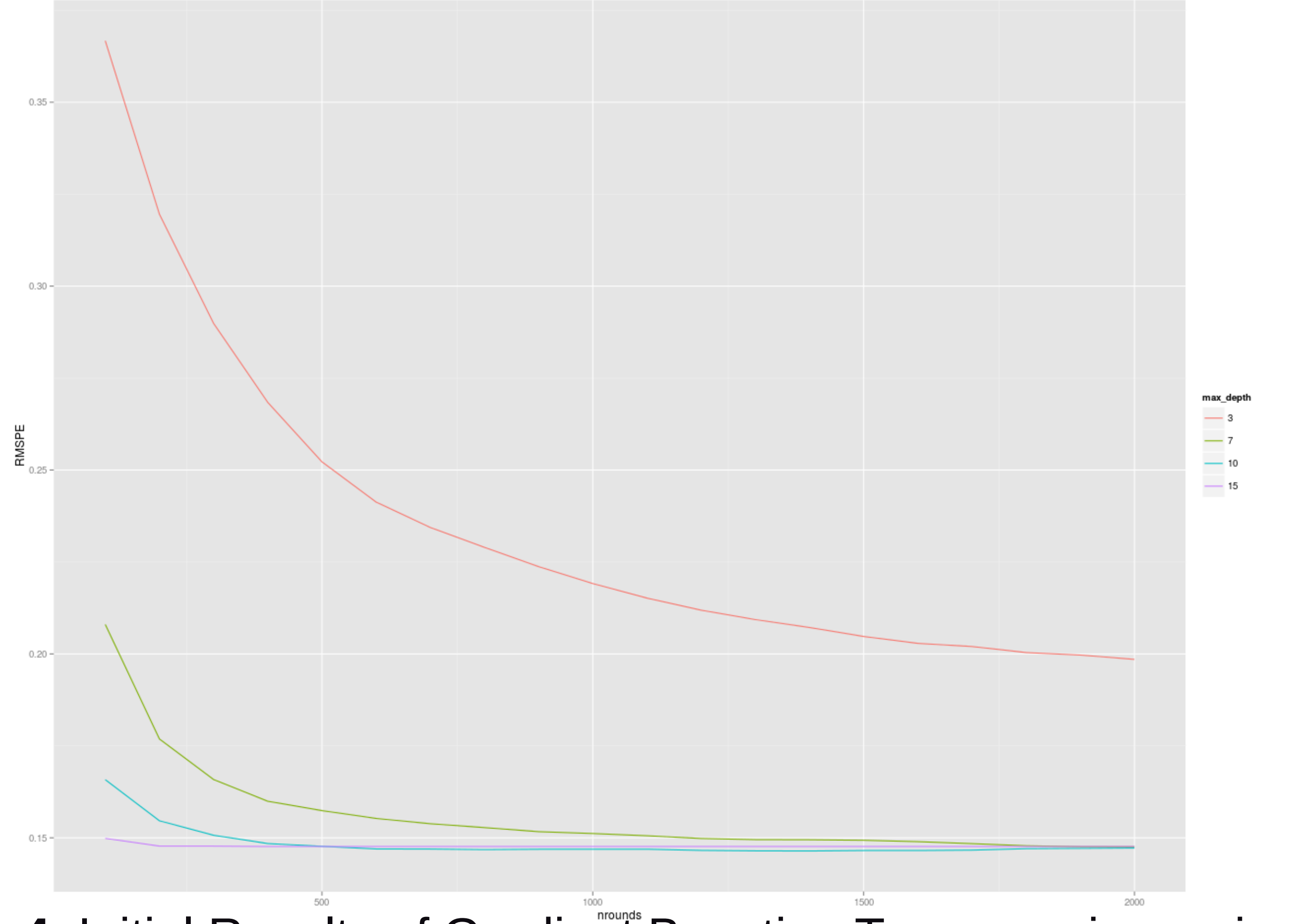


Figure 4. Initial Results of Gradient Boosting Trees varying a single parameter..

But GBM has many parameters (8 total!). How to optimize these? Each run of GBM is costly. How to maximize these parameters in the least amount of time?

We use **Bayesian Optimization**.

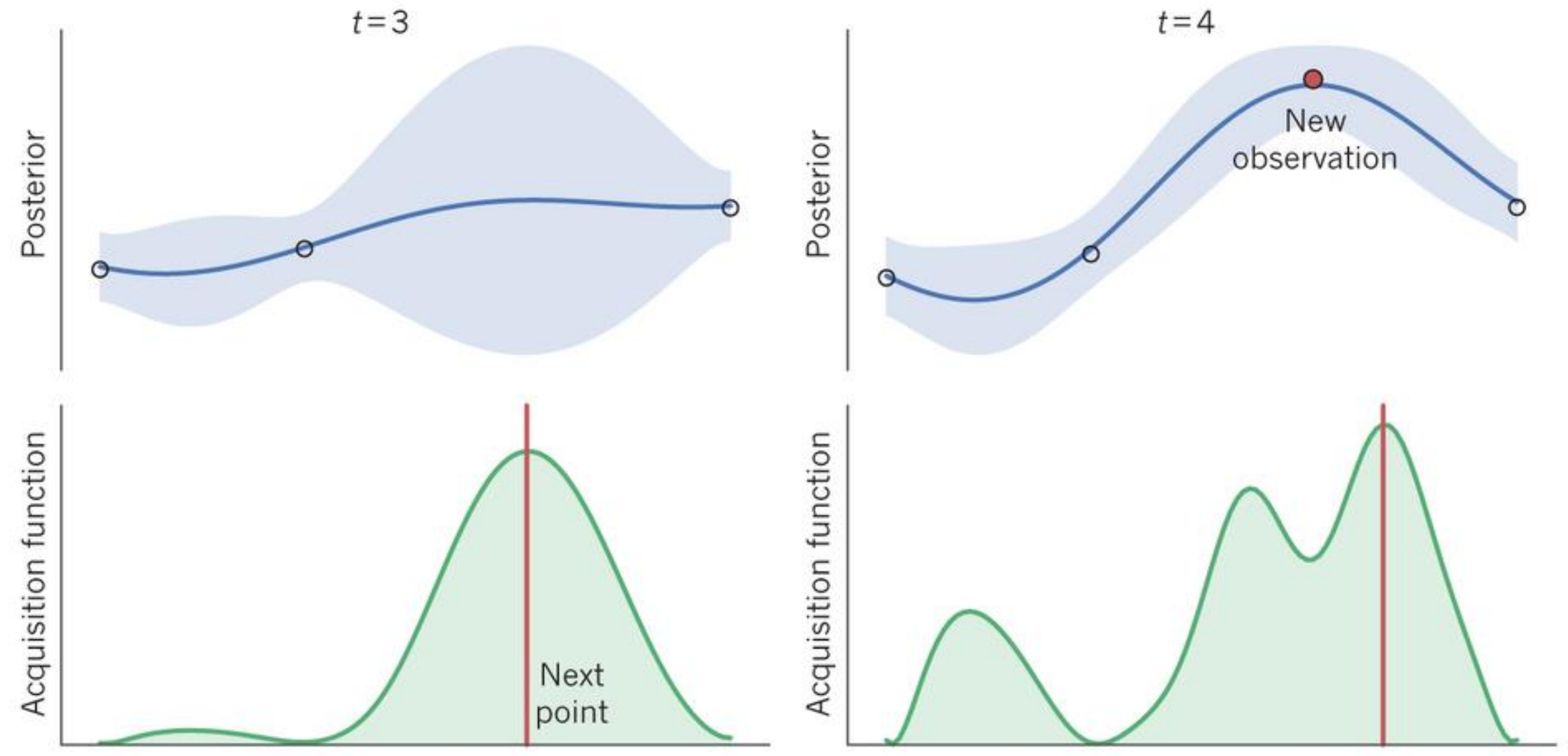


Figure 4. Bayesian Optimization works iteratively by fitting a Gaussian process to observed data from a black box function, then decides the next point to sample based on an acquisition function.

Final result is **0.1125 RMSPE**!