Gradient Boosted Trees

Motivation

- fast and scalable
- successfully used in kaggle competitions
- great xgboost package (R, Python, Julia, Scala)
- LightGBM - new competitor developed by Microsoft
- regression, classification, ranking

Example - predicting person's age

PersonID	Age	LikesGardening	PlaysVideoGames	LikesHats
1	13	FALSE	TRUE	TRUE
2	14	FALSE	TRUE	FALSE
3	15	FALSE	TRUE	FALSE
4	25	TRUE	TRUE	TRUE
5	35	FALSE	TRUE	TRUE
6	49	TRUE	FALSE	FALSE
7	68	TRUE	TRUE	TRUE
8	71	TRUE	FALSE	FALSE
9	73	TRUE	FALSE	TRUE

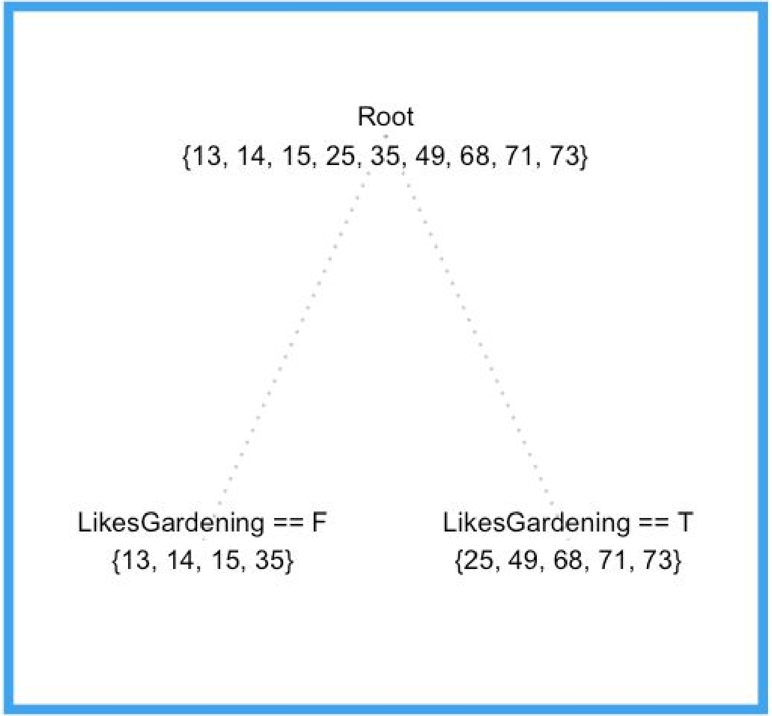
Intuition

- The people who like gardening are probably older
- The people who play video games are probably younger
- LikesHats is probably just random noise

Feature	FALSE	TRUE
LikesGardening	{13, 14, 15, 35}	{25, 49, 68, 71, 73}
PlaysVideoGames	{49, 71, 73}	{13, 14, 15, 25, 35, 68}
LikesHats	{14, 15, 49, 71}	{13, 25, 35, 68, 73}

Decision Tree

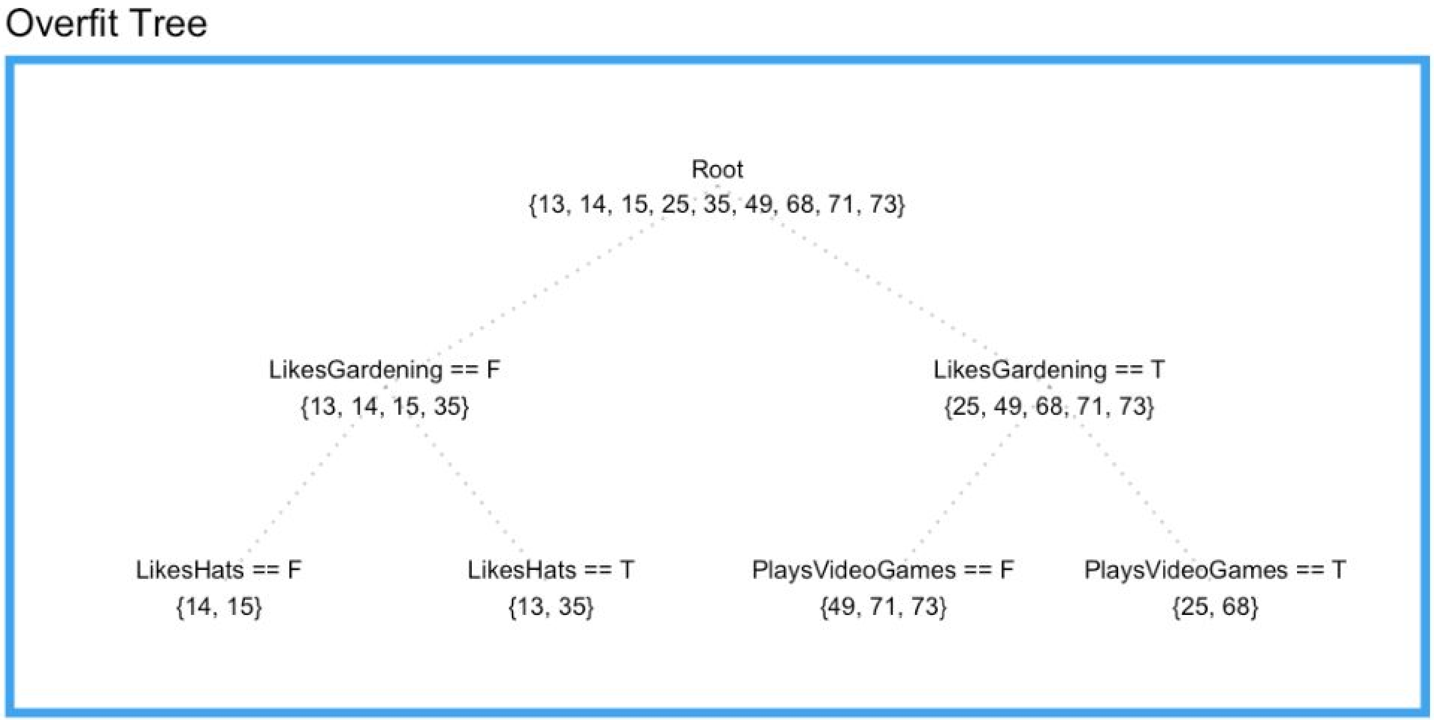
Tree 1



Residual Error

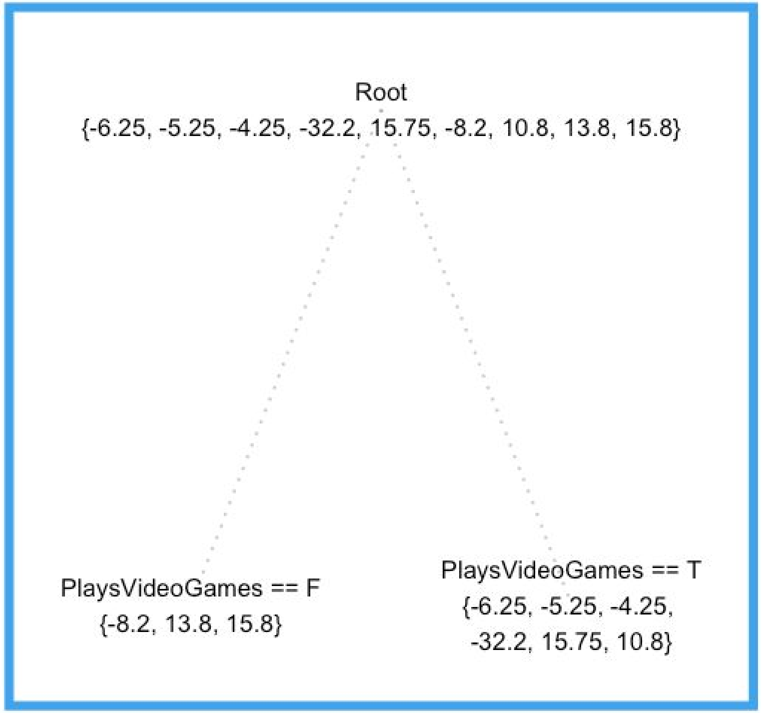
PersonID	Age	Tree1 Prediction	Tree1 Residual
1	13	19.25	-6.25
2	14	19.25	-5.25
3	15	19.25	-4.25
4	25	57.2	-32.2
5	35	19.25	15.75
6	49	57.2	-8.2
7	68	57.2	10.8
8	71	57.2	13.8
9	73	57.2	15.8

Overfitted Tree



Fitting Residuals

Tree2



Residual Error of the Combined Model

PersonID	Age	Tree1 Prediction	Tree1 Residual	Tree2 Prediction	Combined Prediction	Final Residual
1	13	19.25	-6.25	-3.567	15.68	2.683
2	14	19.25	-5.25	-3.567	15.68	1.683
3	15	19.25	-4.25	-3.567	15.68	0.6833
4	25	57.2	-32.2	-3.567	53.63	28.63
5	35	19.25	15.75	-3.567	15.68	-19.32
6	49	57.2	-8.2	7.133	64.33	15.33
7	68	57.2	10.8	-3.567	53.63	-14.37
8	71	57.2	13.8	7.133	64.33	-6.667
9	73	57.2	15.8	7.133	64.33	-8.667

Tree1 SSE	Combined SSE
1994	1765

Boosting Scheme

1. Fit a model to the data: $F_1(x) = y$
2. Fit a model to the residuals: $h_1(x) = y - F_1(x)$
3. Create a new model: $F_2(x) = F_1(x) + h_1(x)$

- Generally: $F(x) = F_1(x) \rightarrow F_2(x) = F_1(x) + h_1(x) \ldots \rightarrow F_M(x) = F_{M-1}(x) + h_{M-1}(x)$

Inicialization and Terminal Condition

- We can initialize the model with a single prediction value minimizing MSE

$$F_0(x) = \arg\min_{\gamma} \textstyle\sum_{i=1}^{n} L(y_i, \gamma) = \arg\min_{\gamma} \textstyle\sum_{i=1}^{n} (\gamma - y_i)^2 = \displaystyle\frac{1}{n}\sum_{i=1}^{n} y_i.$$

- The number of weak learners can be determined by cross-validation

Gradient Descent

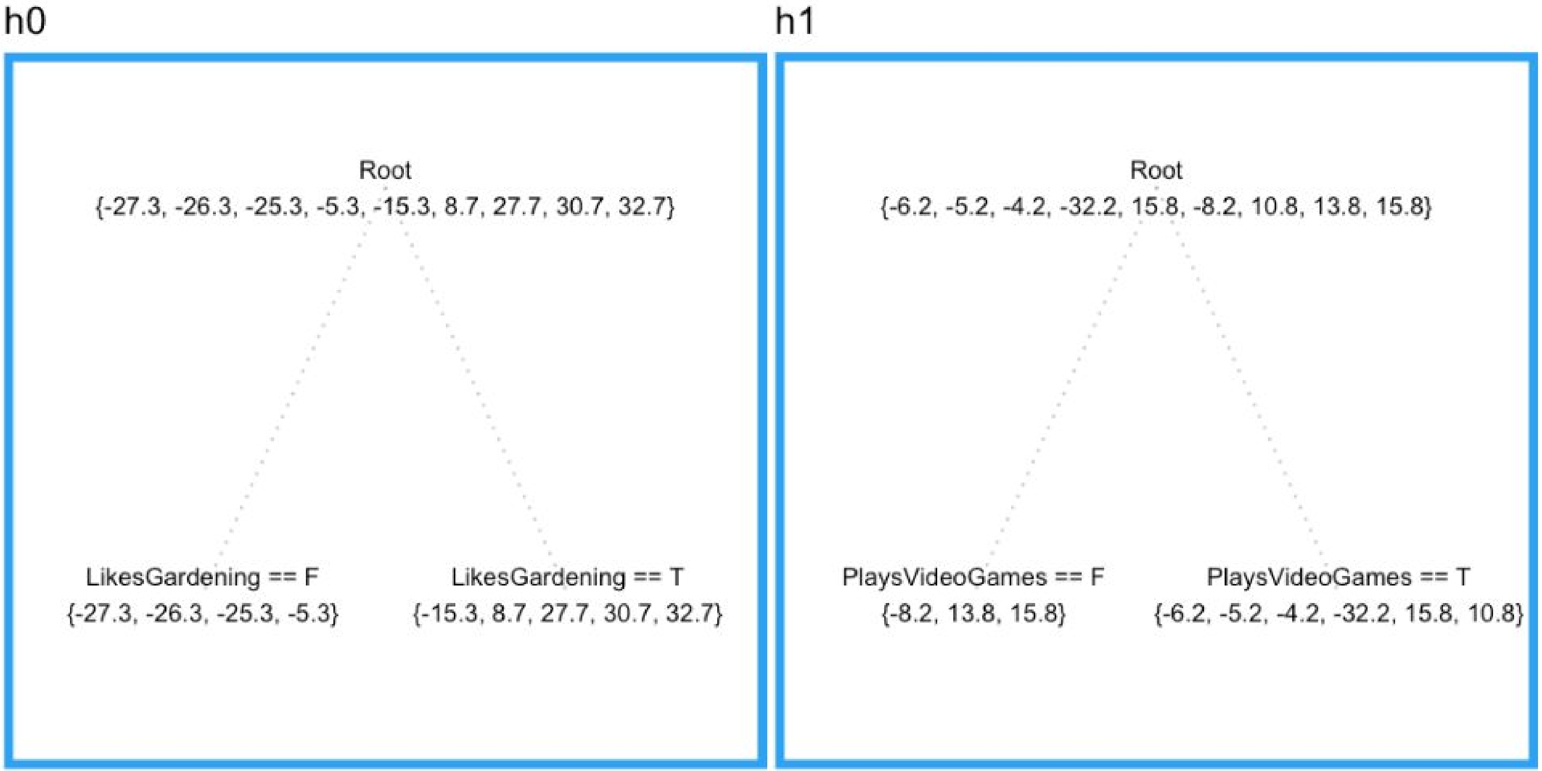
- Instead of fitting the actual residuals, **we can fit the gradient of the loss func.**

1. Initialize the model with a constant value
2. For m = 1 to M:
3. Compute pseudo residuals
4. Compute step magnitude multiplier G
5. $F_M(x) = F_{M-1}(x) + G * h_{M-1}(x)$

Squared Error I

Age	F0	PseudoResidual0	h0	gamma0	F1	PseudoResidual1	h1	gamma1	F2
13	40.33	-27.33	-21.08	1	19.25	-6.25	-3.567	1	15.68
14	40.33	-26.33	-21.08	1	19.25	-5.25	-3.567	1	15.68
15	40.33	-25.33	-21.08	1	19.25	-4.25	-3.567	1	15.68
25	40.33	-15.33	16.87	1	57.2	-32.2	-3.567	1	53.63
35	40.33	-5.333	-21.08	1	19.25	15.75	-3.567	1	15.68
49	40.33	8.667	16.87	1	57.2	-8.2	7.133	1	64.33
68	40.33	27.67	16.87	1	57.2	10.8	-3.567	1	53.63
71	40.33	30.67	16.87	1	57.2	13.8	7.133	1	64.33
73	40.33	32.67	16.87	1	57.2	15.8	7.133	1	64.33

Squared Error II

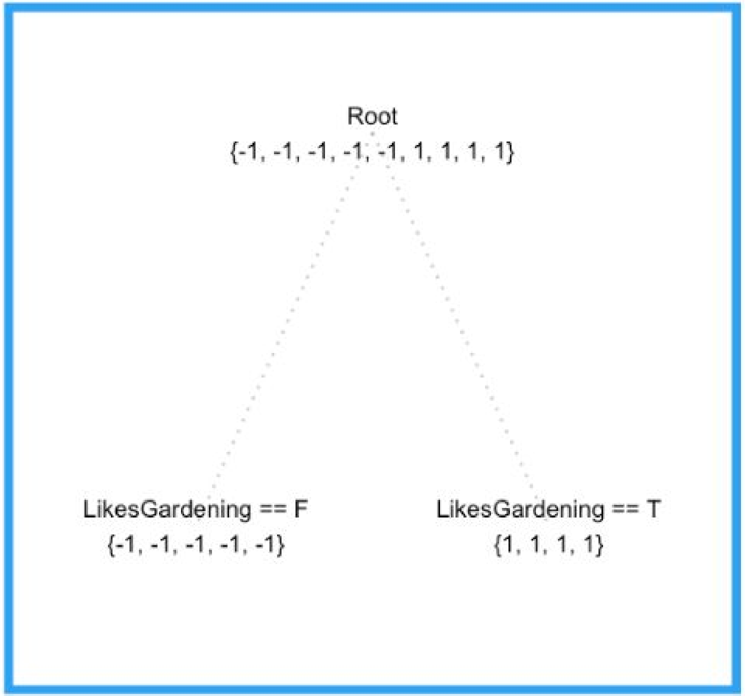


Absolute Error I

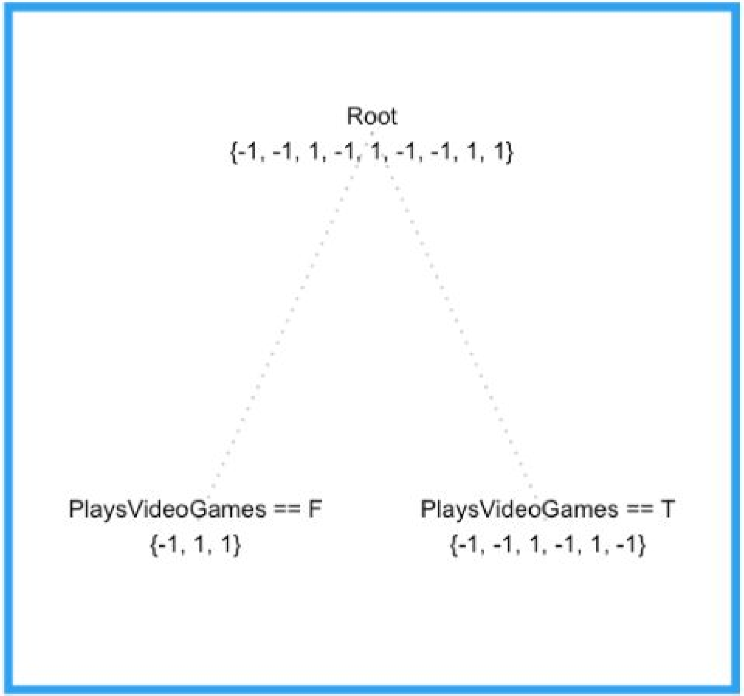
Age	F0	PseudoResidual0	h0	gamma0	F1	PseudoResidual1	h1	gamma1	F2
13	35	-1	-1	20.5	14.5	-1	-0.3333	0.75	14.25
14	35	-1	-1	20.5	14.5	-1	-0.3333	0.75	14.25
15	35	-1	-1	20.5	14.5	1	-0.3333	0.75	14.25
25	35	-1	0.6	55	68	-1	-0.3333	0.75	67.75
35	35	-1	-1	20.5	14.5	1	-0.3333	0.75	14.25
49	35	1	0.6	55	68	-1	0.3333	9	71
68	35	1	0.6	55	68	-1	-0.3333	0.75	67.75
71	35	1	0.6	55	68	1	0.3333	9	71
73	35	1	0.6	55	68	1	0.3333	9	71

Absolute Error II

h0



h1



Other Improvements

- Learning Rate
  - step magnitude is multiplied by number from (0, 1)
  - slower convergence ➔ samples closer to their targets grouped into larger leaves
  - form of regularization
- Row & column sampling
  - done before each boosting iteration
  - leads to different splits

http://blog.kaggle.com/2017/01/23/a-kaggle-master-explains-gradient-boosting