COMP9417 - Machine Learning
Homework 2

Question 1 a

Find the gradient of this function with respect to $\hat{y}$.

$$\frac{\partial}{\partial \hat{y}}\ell(y,\hat{y}) = -(y-\hat{y})$$

When $j \neq i$, there are no $f(x_i)$ terms in $\ell(y_j, f(x_j))$, so $\dfrac{\partial}{\partial f(x_i)}\ell(y_j, f(x_j)) = 0$. The derivative is only non-zero for the term where $j = i$ Hence, Substitute $j = i$ and $\hat{y} = f(x_i)$ in $r_{t,i}$,

$$\begin{aligned} r_{t,i} &= -\frac{\partial \ell(y_i, f(x_i))}{\partial f(x_i)}\,|_{f(x_i)=f_{t-1}(x_i)} \\ &= -(-(y_i - f_{t-1}(x_i))) \\ &= y_i - f_{t-1}(x_i) \end{aligned}$$

According to the $\ell(y,\hat{y}) = \dfrac{1}{2}(y-\hat{y})^2$ and $y = r_{t,i}$,

$$h_t = \arg\min_{f\in\mathcal{F}} \sum_{i=1}^{n} \ell(r_{t,i}, f(x_i)) = \arg\min_{f\in\mathcal{F}} \sum_{i=1}^{n} \frac{1}{2}(r_{t,i} - f(x_i))^2$$

Question 1 b

According to the topic, set

$$L(\alpha) = \sum_{i=1}^{n} \ell(y_i, f_{t-1}(x_i) + \alpha h_t(x_i))$$

As $\ell(y,\hat{y}) = \frac{1}{2}(y-\hat{y})^2$

$$L(\alpha) = \sum_{i=1}^{n} \ell(y_i, f_{t-1}(x_i) + \alpha h_t(x_i)) = \frac{1}{2}\sum_{i=1}^{n} \left(y_i - f_{t-1}(x_i) - \alpha h_t(x_i)\right)^2$$

Since the minimum case needs to be obtained, find the case where the derivative of the loss function at α is equal to 0.

$$\begin{aligned} \frac{d}{d\alpha}L(\alpha) = -\sum_{i=1}^{n} \left(y_i - f_{t-1}(x_i) - \alpha h_t(x_i)\right) h_t(x_i) = 0 \\ \sum_{i=1}^{n} \left(y_i - f_{t-1}(x_i))\right) h_t(x_i) = \sum_{i=1}^{n} \alpha h_t^2(x_i) \\ \frac{\sum_{i=1}^{n} \left(y_i - f_{t-1}(x_i))\right) h_t(x_i)}{\sum_{i=1}^{n} h_t^2(x_i)} = \alpha \end{aligned}$$

Hence, the step-size expression according to the adaptive approach (SS2) can derive

$$\alpha_t = \frac{\sum_{i=1}^{n} \left(y_i - f_{t-1}(x_i))\right) h_t(x_i)}{\sum_{i=1}^{n} h_t^2(x_i)}$$

Question 1 c

Q: Comment on your results, what happens as the number of base learners is increased?

When fewer base learners are used, the complexity of the model is relatively low and may not be able to fully fit the complex patterns in the data. As the number of base learners increases, the complexity of the model increases accordingly, so it is able to fit the data better.

Q: Comment on the differences between your fixed and adaptive step-size implementations.

Figure 1 shows the case of adaptive step size, which has more ups and downs and the results are more in line with the real results.

Figure 2 is the case of fixed step size, which does not need to consider potential changes in the data distribution.

Comparing with both, we can find that for the interval of base learner number 5 50, the adaptive model changes less after 25; the fixed step model causes a larger convergence change for each number change, but it still does not fit the real results as well as the adaptive step model.

Q: How does your model perform on the different x-ranges of the data?

I have tried increasing and decreasing the range of X. The results seem to be similar to the above results.

Question 1 d

Figure 3 is the case of adaptive step size with depth 2 decision trees, which has more ups and downs and is susceptible to noise.

Figure 4 shows the case of a fixed step size with depth 2 decision trees, which does not need to consider potential changes in the data distribution. It has less model ups and downs and is less susceptible to noise. The choice of step size affects the convergence result.

Comparing the two, we can find that when there are fewer base learners, the adaptive step size fits the real results better than the fixed step size; however, as the number of base learners increases, the adaptive step size is more and more susceptible to noise, which may lead to overfitting of the model. In contrast, the fixed step size fits the formal results better.

Question 1 e

Find the gradient of this function with respect to $\hat{y}$. The derivative is only non-zero for the term where $j = i$ Hence, Substitute $j = i$ and $\ell(y, \hat{y}) = \log(1 + e^{-y\hat{y}}$,

$$\begin{aligned} r_{t,i} &= -\frac{\partial \ell(y_i, f(x_i))}{\partial f(x_i)} |_{f(x_i)=f_{t-1}(x_i)} \\ &= -\frac{\partial \log(1 + e^{-y_i f(x_i)})}{\partial f(x_i)} \\ &= -\frac{\partial \ln(1 + e^{-y_i f(x_i)})}{\partial f(x_i)} \\ &= \frac{y_i e^{-y_i f(x_i)}}{1 + e^{-y_i f(x_i)}} \\ &= \frac{y_i}{1 + e^{y_i f(x_i)}} \end{aligned}$$

As for the optimization problem in step (GC3),

$$h_t = \arg\min_{h\in\mathcal{F}} \sum_{i=1}^{n} (r_{t,i} - h(x_i))^2 = \arg\min_{h\in\mathcal{F}} \sum_{i=1}^{n} \left(\frac{y_i}{1 + e^{y_i f(x_i)}} - h(x_i)\right)^2$$

Question 1 f

According to the topic, set

$$L(\alpha) = \sum_{i=1}^{n} \ell\left(y_i, f_{t-1}(x_i) + \alpha h_t(x_i)\right)$$

Substituting $r_{t,i} = \dfrac{y_i}{1 + e^{y_i f(x_i)}}$ into loss function

$$L(\alpha) = \sum_{i=1}^{n} \log\left(1 + e^{-y_i(f_{t-1}(x_i)+\alpha h_t(x_i))}\right)$$

Assuming that $p = -y_i\left(f_{t-1}(x_i) + \alpha h_t(x_i)\right)$, try to find the derivative of $\log(1+e^p))$

$$\frac{\partial}{\partial p}\log(1 + e^p)) = \frac{e^p}{1 + e^p}$$

Since e^x is always greater than 0 no matter what value x takes, the result of this inverse must be greater than 0. Let's consider the derivative of $-y_i\left(f_{t-1}(x_i) + \alpha h_t(x_i)\right)$

$$\frac{\partial}{\partial \alpha} - y_i\left(f_{t-1}(x_i) + \alpha h_t(x_i)\right) = -h_t(x_i)y_i$$

$h_t(x_i)$ is the prediction of the base learner (base learner) for the ith sample point x_i, its prediction can only be the value of its leaf node, and will not be equal to zero.
y_i is the true label of the sample point x_i, and in our setup, y_i can only be -1 or 1 and will not be equal to zero.

Therefore, $-h_t(x_i)y_i \neq 0$, we cannot find a critical point for this alpha, so there is no direct closed-form solution.

Question 1 g

I think we can initialize a range of α_t within which a step size is selected to generate a series of candidate α_t values, compute their loss function values, find the α_t value corresponding to the smallest loss function value, and use it as the best step size.

The additional computational cost is to generate candidate α_t values and compute the corresponding loss function values.

Question 2 a

Create an array to record the results.

For all each element i in the test set, let them be compared with each hypothesis h_j in the set $\mathcal{H}$ as follows:

1. First, create a new hypothesis $h' = h_j$.
2. Set $h'(x_i) = 0$ for all terms $k \neq i$.
3. Then compare rMSE(h') with c_0. If rMSE$(h') < c_0$, put h_j in the ith in array. After that stop the comparison go to $i + 1$.

The time of this algorithm requires comparing all elements of the test set with all hypotheses in $\mathcal{H}$. Since it is two nested loops, the test size n is linearly increasing when the number of hypothesis sets is constant.

The best case is that all elements of rMSE$(h') < c_0$ for $h' = h_1$, with a time complexity of $O(n)$. The worst case is that $\mathcal{H}$ needs to be traversed, with a time complexity of $O(nT)$. Hence, the time complexity of this algorithm is $O(nT)$.

It need at most nT queries for rMSE.

Question 2 b

According to the definition of the dot product of linear algebra,

$$y^\top h(X) = \sum_{i=1}^{n} y_i h(x_i)$$

For two arbitrary two variables a, b, their product can again be obtained from the difference between the square of their sum and the square of their difference

$$(a+b)^2 - (a-b)^2 = (a^2 + 2ab + b^2) - (a^2 - 2ab + b^2) = 4ab$$

Since rMSE is the square root of the mean square error, we obtain a similar form of sum and difference as above by changing the positive and negative of $h(x_i)$. Suppose $m = h(x_i), n = -h(x_i)$,

$$\text{rMSE}(m)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - h(x_i))^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i^2 - 2y_i h(x_i) + h(x_i)^2)$$
$$\text{rMSE}(n)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i + h(x_i))^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i^2 + 2y_i h(x_i) + h(x_i)^2)$$
$$\sum_{i=1}^{n} y_i h(x_i) = -\frac{n}{4}\left(\text{rMSE}(m)^2 - \text{rMSE}(n)^2)\right)$$

Hence, $y^\top h(X)$ can be converted to

$$y^\top h(X) = -\frac{n}{4}\left(\text{rMSE}(h_1)^2 - \text{rMSE}(h_2)^2)\right)$$

Therefore, a minimum of 2 queries is required.

Question 2 c

The interval of $f(x) = x^2(x > 0)$ is monotonically increasing, according to the topic, since rMSE is a positive number, we can convert the problem to find the minimum value of (rMSE)^2.

$$\min_{\alpha_1,\dots,\alpha_k} (\text{rMSE})^2(\sum_{k=1}^{K}\alpha_k h_k) = \min_{\alpha_1,\dots,\alpha_k} \frac{1}{n}\sum_{i=1}^{n}(y_i - \sum_{k=1}^{K}\alpha_k h_k(x_i))^2$$

This is a quadratic function of the α's and can be minimized by setting its derivative with respect to each $p \in \{1,\dots,K\}$ to zero:

$$\frac{\partial(\text{rMSE})^2}{\partial\alpha_p} = \frac{-2}{n}\sum_{i=1}^{n}(y_i - \sum_{k=1}^{K}\alpha_k h_k(x_i))h_p(x_i) = 0 \quad p \in \{1,\dots,K\}$$
$$\sum_{i=1}^{n} y_i h_p(x_i) - \sum_{k=1}^{K}\alpha_k \sum_{i=1}^{n} h_k(x_i)h_p(x_i) = 0 \quad p \in \{1,\dots,K\}$$

$\sum_{k=1}^{K}\alpha_k \sum_{i=1}^{n} h_k(x_i)h_p(x_i)$ do not need to make rMSE. $\sum_{i=1}^{n} y_i h_p(x_i)$ can be counted in the same way as in the previous question, each time makes 2 queries, p takes values in the range 1 to K, so $2K$ of queries is required.

Question 3 a

We can see that for this part $\frac{g'(x^{(k)})}{g''(x^{(k)})}$, it is the first-order derivative $g'(x^{(k)})$ divided by the second-order derivative $g''(x^{(k)})$. In the multidimensional metric, the first-order derivative is replaced by the gradient function at position $\nabla f(x^{(k)})$, and the second-order derivative is replaced by the Hessian matrix which is a matrix containing the second derivatives of the function $\nabla f(x^{(k)})$.

From conversions that only allow numbers, to conversions of vector and matrices.

Question 3 b

Solution in Figure 5

Question 3 c

x^0 = [-1.2 1.]
x^1 = [-1.1752809 1.38067416]
x^2 = [0.76311487 -3.17503385]
x^3 = [0.76342968 0.58282478]
x^4 = [0.99999531 0.94402732]
x^5 = [0.9999957 0.99999139]
x^6 = [1. 1.]

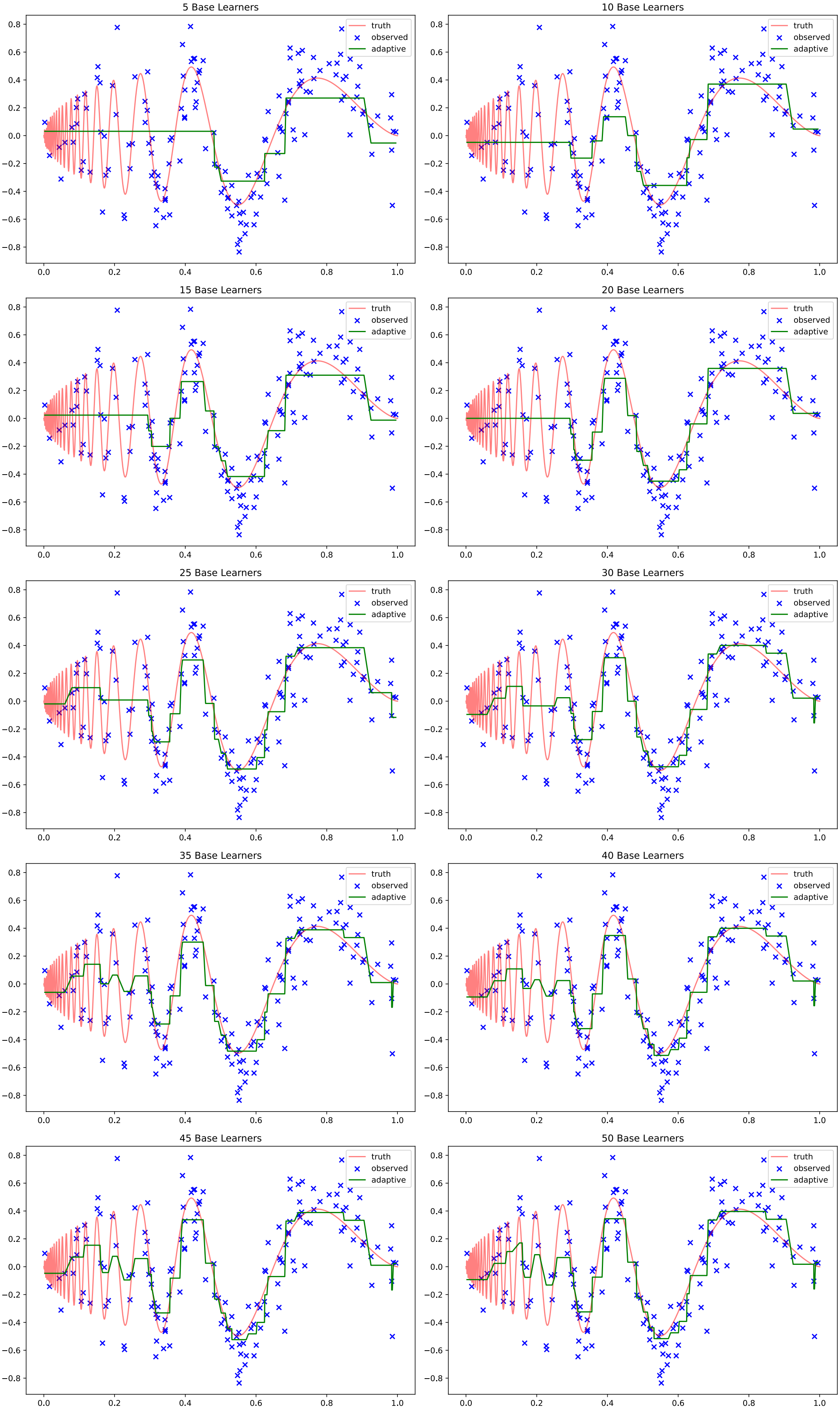


Figure 1: Adaptive Step Size with Depth 1 Decision Trees

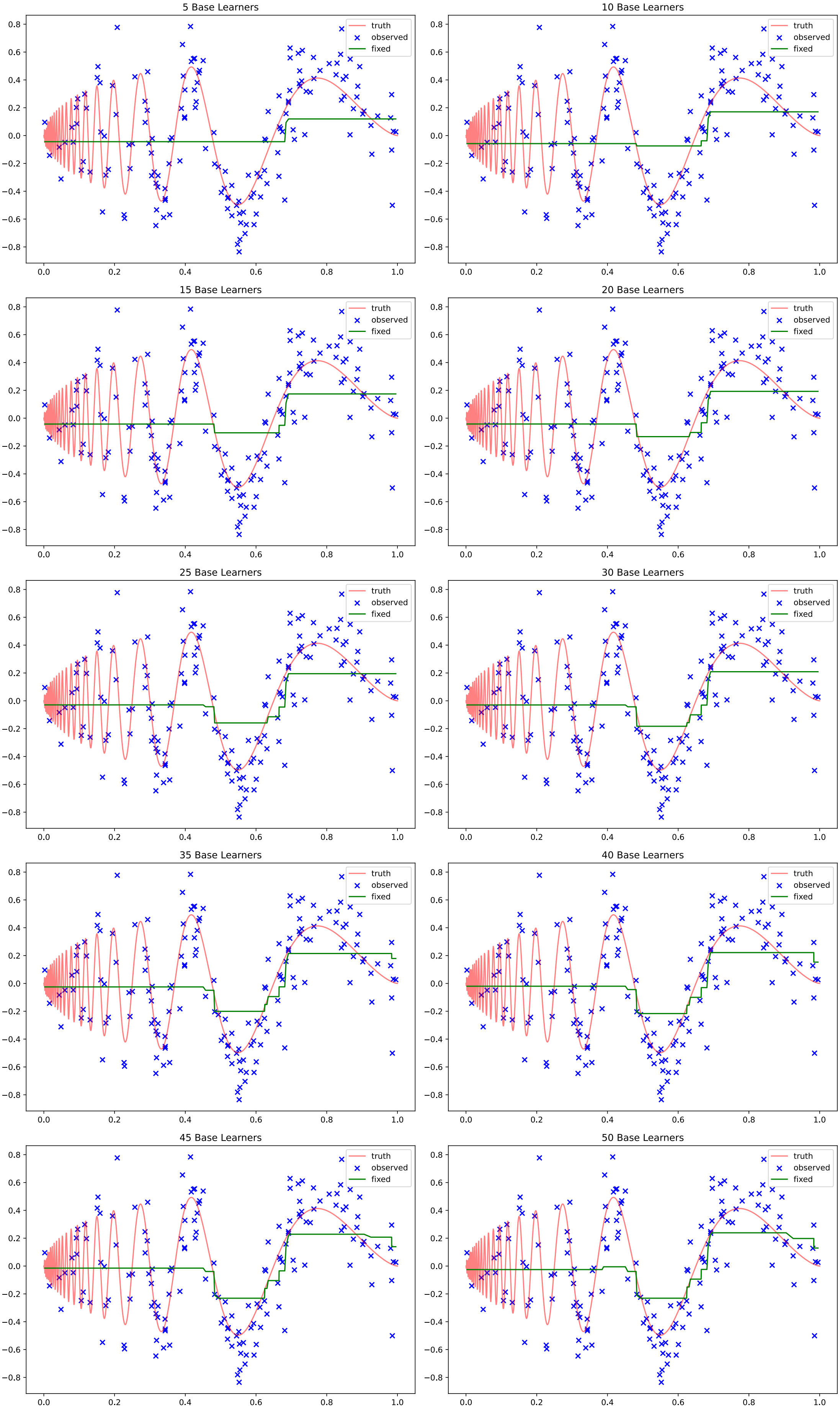
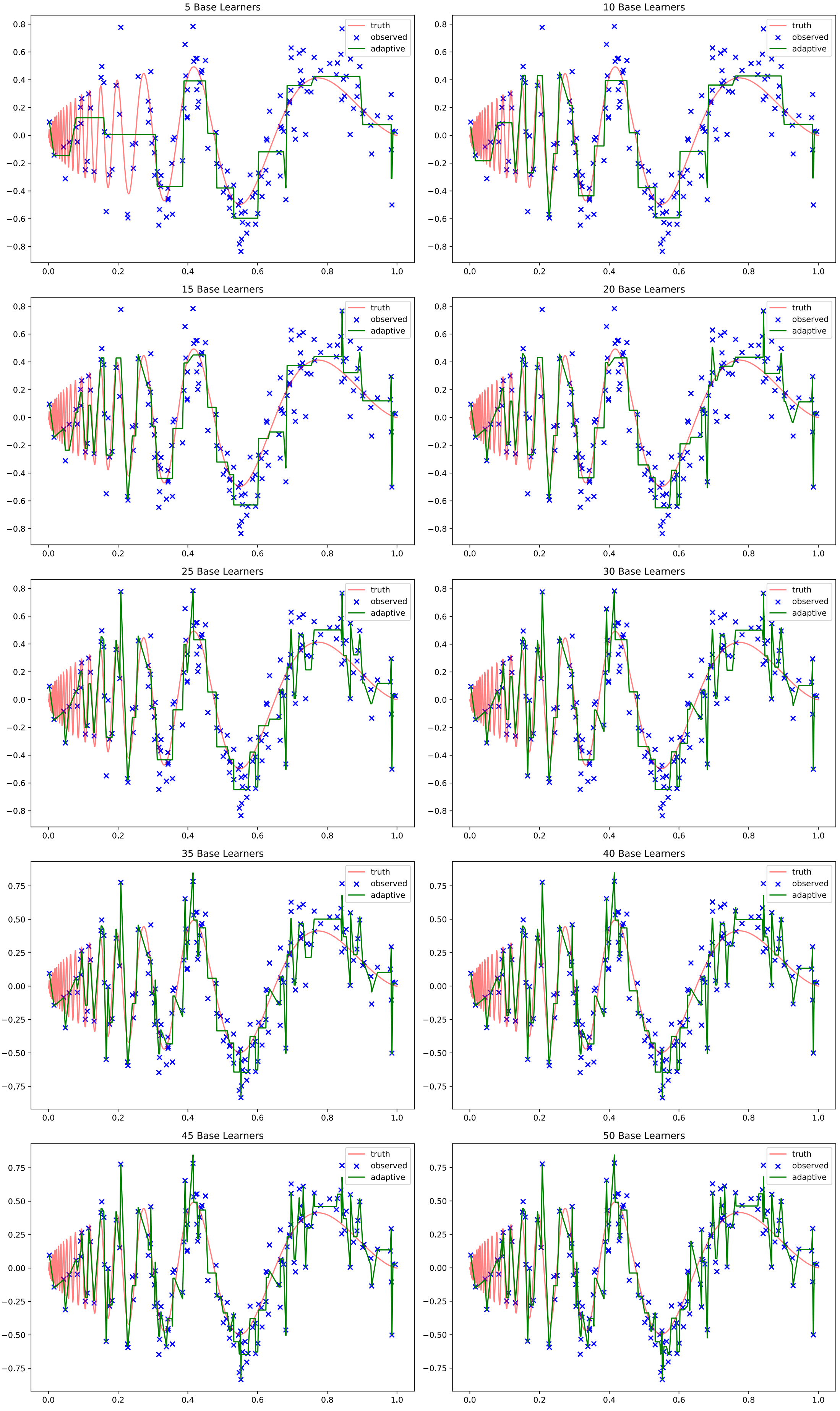


Figure 2: $\alpha = 0.1$ Fixed Step Size with Depth 1 Decision Trees



5 Base Learners

10 Base Learners

15 Base Learners

20 Base Learners

25 Base Learners

30 Base Learners

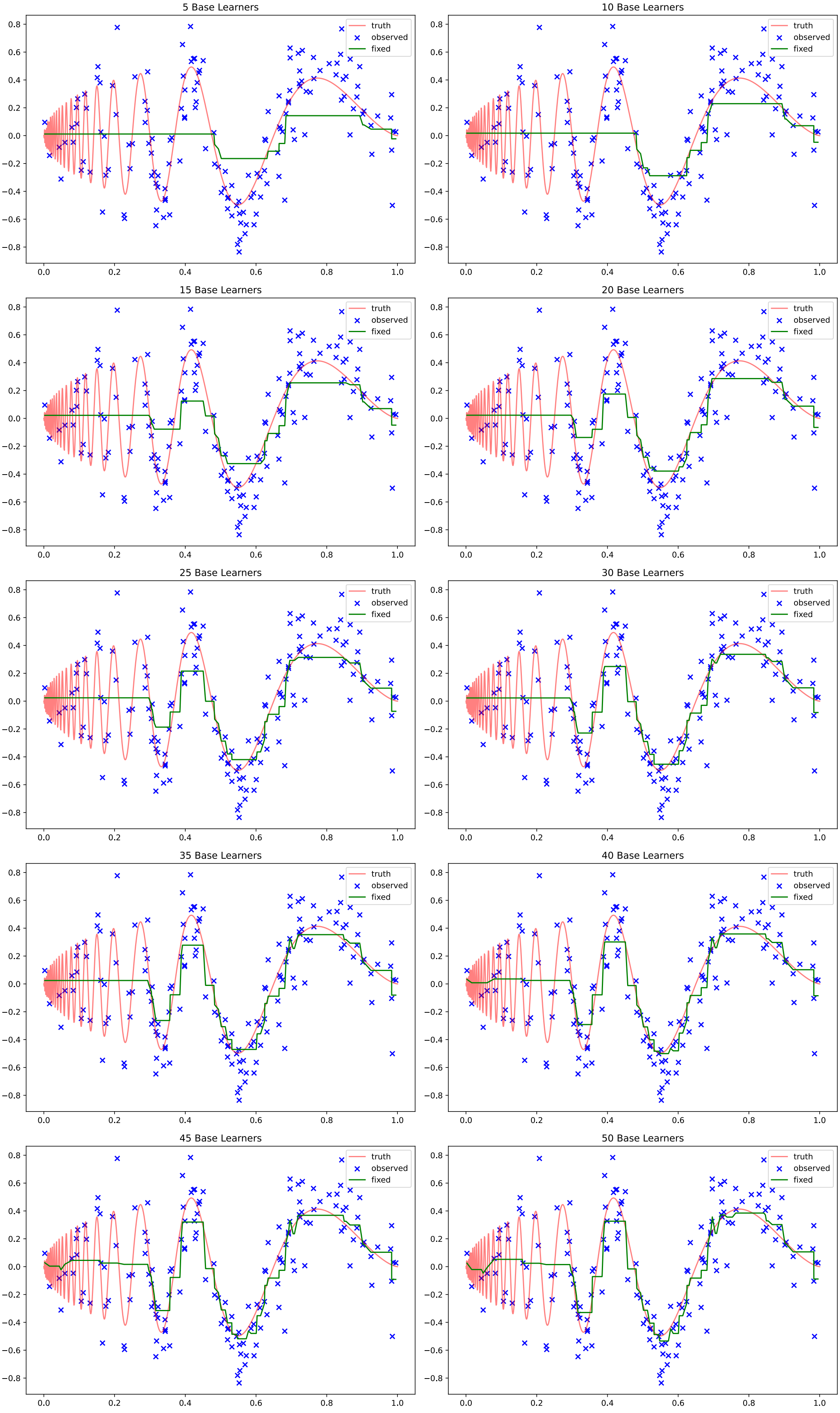
35 Base Learners

40 Base Learners

45 Base Learners

50 Base Learners

Figure 3: Adaptive Step Size with Depth 2 Decision Trees



5 Base Learners

10 Base Learners

15 Base Learners

20 Base Learners

25 Base Learners

30 Base Learners

35 Base Learners

40 Base Learners

45 Base Learners

50 Base Learners

Figure 4: $\alpha = 0.1$ Fixed Step Size with Depth 2 Decision Trees

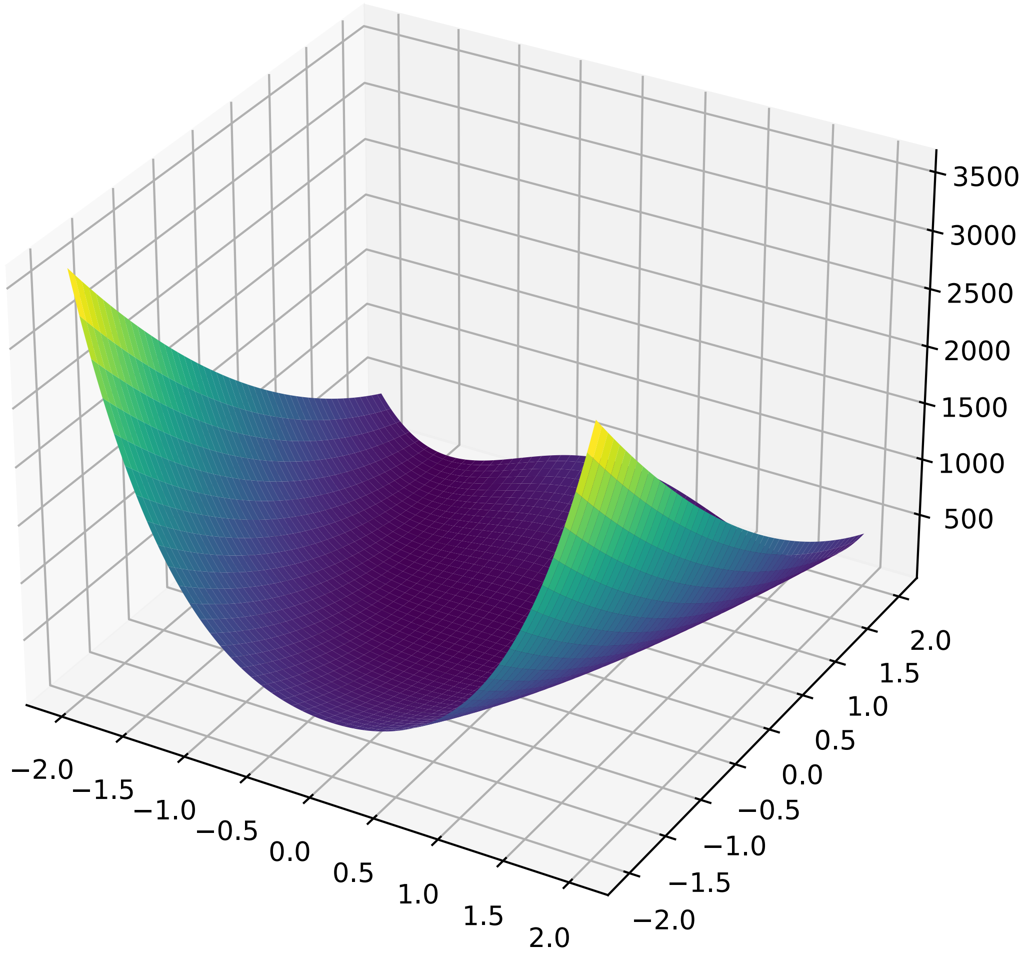


Figure 5: $f(x, y) = 100(y - x^2)^2 + (1 - x)^2$