Homework 6: Multiclass, Trees and Gradient Boosting

Lin Pengyun

August 10, 2021

1 Reformulation of Multiclass Hinge Loss

1.1 Multiclasss setting review

There is no question in this part.

1.2 Two version of multiclass hinge loss

1. We have:

$$\max_{y\in\mathcal{Y}} f(y) = \max_{y\in\mathcal{Y}-\{y_i\}}\{\max[f(y_i), f(y)]\}$$

If $\Delta(y,y)=0$, then $\Delta(y_i,y_i)+h(x_i,y_i)-h(x_i,y_i)=0$, we have:

$$\begin{aligned}\max_{y\in\mathcal{Y}}[\Delta(y_i,y)+h(x_i,y)-h(x_i,y_i)] &= \max_{y\in\mathcal{Y}-\{y_i\}}\{\max[0,\Delta(y_i,y)+h(x_i,y)-h(x_i,y_i)]\} \\ m_{i,y}(h) &= h(x_i,y_i)-h(x_i,y)\end{aligned}$$

So we have:

$$\max_{y\in\mathcal{Y}}[\Delta(y_i,y)+h(x_i,y)-h(x_i,y_i) = \max_{y\in\mathcal{Y}-\{y_i\}}\{\max[0,\Delta(y_i,y)-m_{i,y}(h)]\}$$

2. If $\forall y\in\mathcal{Y}, m_{i,y}(h)\geq\Delta(y_i,y)$, we have:

$$l_1(h,(x_i,y_i)) = l_2(h,(x_i,y_i)) = 0$$

The classification creteria is $f(x_i)=\arg\max_{y\in\mathcal{Y}} h(x_i,y)$, $m_{i,y}(h)=h(x_i,y_i)-h(x_i,y)\geq 0$ and the equality holds only when $y=y_i$.
Hence, $f(x_i)=y_i$ in the conditions decribed in this question.

2 SGD for Multiclass Linear SVM

1. If for a set of functions $\{f_1(x), ..., f_n(x)\}$, $f_i(x)$ is convex function, then:

   $$\max_i f_i[\alpha x+(1-\alpha)y] \leq \max_i \alpha f_i(x)+(1-\alpha)f_i(y) \leq \alpha \max_i f_i(x)+(1-\alpha)\max_i f_i(y)$$

   So $\max_i f_i(x)$ is also convex.

   $$J(w) = \lambda w^T w + \frac{1}{n}\sum_{i=1}^{n}\max_{y\in\mathcal{Y}}[\Delta(y_i, y)+ < w, \Psi(x_i, y) - \Psi(x_i, y_i) >]$$

   The norm and inner product of w are both convex, for the additivity of convex function and max invariance shown before, $J(w)$ is also convex

2. The subgradient of $\lambda w^T w$ w.r.t. w is $2\lambda w$.
   Denote $\hat{y_i}$ as $\arg\max_{y\in\mathcal{Y}}[\Delta(y_i, y)+ < w, \Psi(x_i, y) - \Psi(x_i, y_i) >]$, one of the subgradients of each max term is:

   $$\Psi(x_i, \hat{y_i}) - \Psi(x_i, y_i)$$

   So one subgradient of $J(w)$ could be:

   $$2\lambda w + \frac{1}{n}\sum_{i=1}^{n}\Psi(x_i, \hat{y_i}) - \Psi(x_i, y_i)$$

3. According to 2, the subgradient on (x_i, y_i) could be:

   $$2\lambda w + \Psi(x_i, \hat{y_i}) - \Psi(x_i, y_i)$$

4. The minibatch subgradient based on $(x_i, y_i), ..., (x_{i+m}, y_{i+m})$ is:

   $$2\lambda w + \frac{1}{m}\sum_{k=i}^{i+m}\Psi(x_i, \hat{y_i}) - \Psi(x_i, y_i)$$

3 Hinge Loss is a Special Case of Generalized Hinge Loss

When the problem reduces to binary classification where the output space $\mathcal{Y} = \{-1, 1\}$, the loss function on (x_i, y_i) is:

$$\max_{y\in\{-1,1\}}[\Delta(y_i, y) + h(x_i, y) - h(x_i, y_i)]$$

Denote the score function of binary SVM is $g(x)$, let:

$$\Delta(-1,1) = \Delta(1,-1) = 1, \Delta(1,1) = \Delta(-1,-1) = 0$$
$$h(x_i, y_i) = \frac{y_i}{2} g(x_i)$$

$y \in \{-1, 1\} = \{-y_i, y_i\}$, the loss function will be:

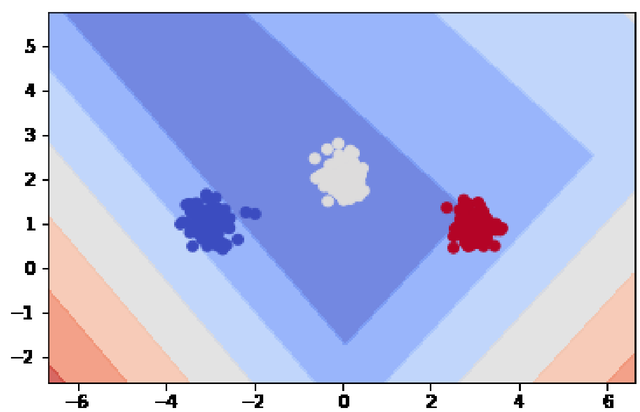
$$\max[0, 1 - y_i g(x)],$$

which is as same as the that of binary SVM.

4 Multiclass classification–Implementation

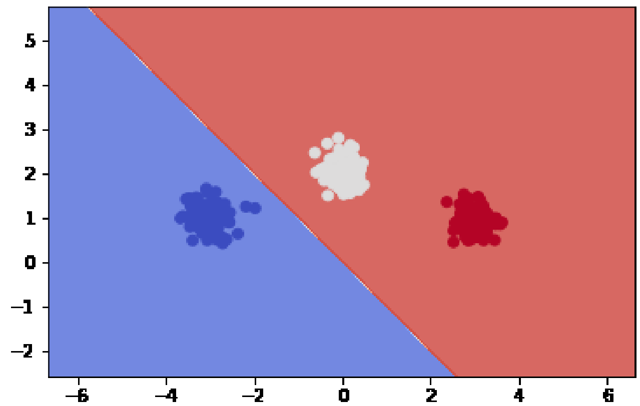
4.1 One-vs-All (or One-vs-Rest)

1. Code is in another repository. The result is shown below.



4.2 Multiclass-SVM

1. Code is in another repository. The result is shown below.

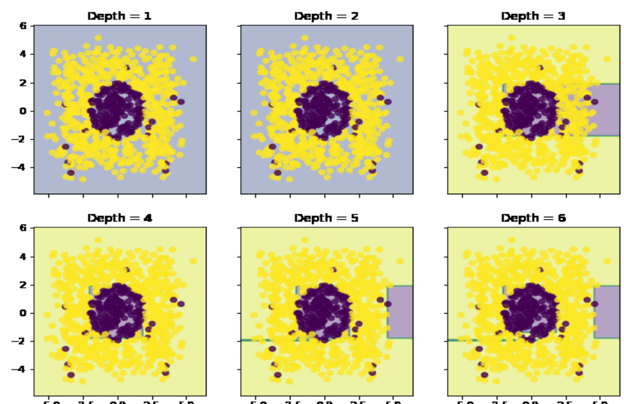


5 Audio Classification

This part is optional, now I just leave it unfinished.

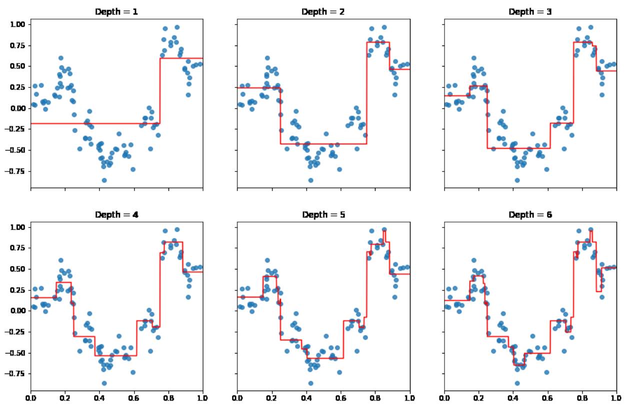
6 Decision Tree Implementation

1. The fitting result of classification decision tree is below:



As the depth's growing, the fitting result is generally becoming better and better, but overfit appears in the last two pictures.

2. The fitting result of regression tree w.r.t. one-dimensional data is below:



Fitting result goes better while overfit to noise aggravates as the depth grows.

7 Gradient Boosting Machine

1. If the prediction function is $f(x)$ and the loss function is given by:

$$l(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2, \hat{y} = f(x)$$

then the partial derivative on (x_i, y_i) w.r.t. the $(m-1)$'th round prediction function $f_{m-1}(x)$ is:

$$\frac{\partial}{\partial f_{m-1}(x_i)} l(y_i, f_{m-1}(x_i)) = f_{m-1}(x_i) - y_i,$$

so the gradient of g_m is:

$$g_m = (f_{m-1}(x_1) - y_1, ...f_{m-1}(x_n) - y_n)^T$$

According to the gradient boosting algorithm, the function selected in m'th round should be:

$$\begin{aligned} h_m &= \arg\min_{h \in \mathcal{F}} \sum_{i=1}^{n} [-g_{m,i} - h(x_i)]^2 \\ &= \arg\min_{h \in \mathcal{F}} \sum_{i=1}^{n} [y_i - f_{m-1}(x_i) - h(x_i)]^2 \end{aligned}$$

Therefore, in every round of iteration, the algorithm always finds the function which fits best the residual in the former round.

2. If the loss function is given by:

$$l(y, f(x)) = \ln(1 + e^{-yf(x)}),$$

the derivative w.r.t. $f_{m-1}(x)$ on (x_i, y_i) is:

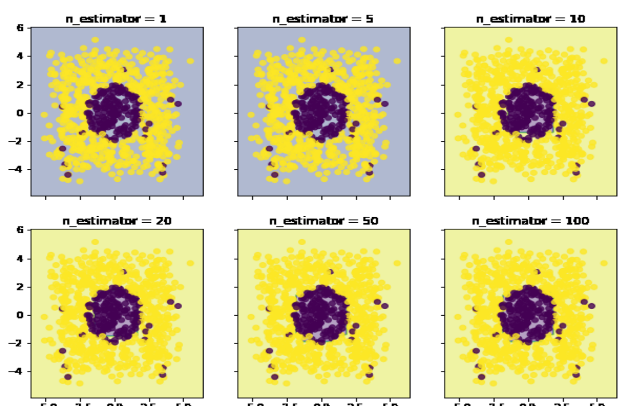
$$\frac{\partial}{\partial f_{m-1}(x_i)} l(y_i, f_{m-1}(x_i)) = -\frac{y_i}{1 + e^{y_i f_{m-1}(x_i)}}$$

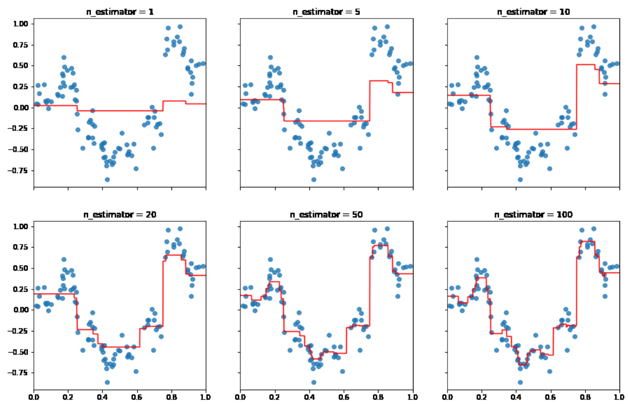
So we have:

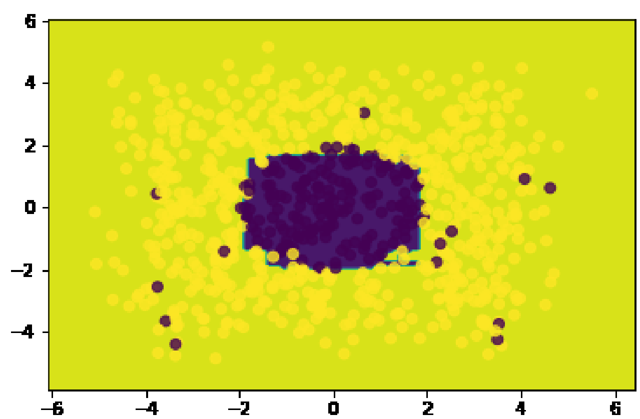
$$h_m = \arg\min_{h \in \mathcal{F}} \sum_{i=1}^{n} [\frac{y_i}{1 + e^{y_i f_{m-1}(x_i)}} - h(x_i)]^2$$

8 Gradient Boosting–Implementataion

1. In this question, an gradient boosting algorithm with l_2 loss and base prediciton function of decision tree is used to perform a classfication task on 2-D data and regression task on 1-D dataset. The result is below:







The last diagram is the prediction result of gradient boosting algorithm in sklearn.