Stochastic Gradient Descent

Este algoritmo se encarga de simplificar el cálculo considerado de una sola muestra antes de obtener las derivadas parciales. Esto supone que las derivadas parciales obtenidas no van a ser óptimas y no van a apuntar en la dirección de mayor descenso de la función de error, pero este problema se compensa con el mayor número de modificaciones a los parámetros que se aplican, lo que lleva a tiempos de entrenamiento mucho menores. Además el algoritmo es un método iterativo en el cual optimiza la función objetivo con propiedades de suavidad como los son la diferenciable o la subdiferenciable, esto puede considerarse como una aproximación estocástica de la optimización del descenso del gradiente ya que este remplazara al gradiente calculado por un conjunto de datos por una estimación del mismo gradiente pero calculado a partir de un subconjunto de datos seleccionados al azar, esto reduce la carga computacional muy alta logrando iteraciones más rápidas a cambio de una menor tasa de convergencia.

El aprendizaje automático y la estimación estadística consideran el problema de minimizar una función objetivo que tiene forma de suma:

$$Q(w) = \frac{1}{n}\sum_{i=1}^{n} Q_i(w)$$

Para el parámetro w que minimiza a la función Q(w) esta por estimar cada función que se va sumando de Q_i y se asocia con la i-esima observación en el conjunto de los datos.

Stochastic Gradient Descent se encarga de seleccionar aleatoriamente una sola muestra del conjunto de datos para realizar cada iteración, a diferencia de otro tipos de optimización como Gradient Descent o Batch Gradient Descent que toman todo el conjunto de datos completos lo que ocasiona que si tenemos demasiados muestras de conjunto de datos tendrán que tomar todos esos para a completar una iteración mientras realiza el descenso de gradiente y esto se tiene que repetir cada iteración por lo tanto se vuele un proceso computacionalmente costoso, lo cual Stochastic Gradient Descent soluciona este problema usando una sola muestra por iteración.

Como el algoritmo utiliza aleatoriamente una muestra del conjunto de datos por cada iteración esto provoca que la ruta que toma el algoritmo para alcanzar el mínimo del gradiente suele ser mas ruidosa que otros tipos de algoritmos de gradiente, pero aun así el algoritmo llega a encontrar los mínimos con un tiempo de entrenamiento significativamente menor.

Algoritmo Stochastic Gradient Descent:

for in range (m):

$$\theta_j = \theta_j - \alpha\left(\check{y}^i - y^i\right)x_j^i$$

Ventajas del Stochastic Gradient Descent:

- Si la función objetivo es la suma de los errores sobre un conjunto muy grande de datos. La muestra suele ser representativa y producir un valor muy cercano al de la población.
- Se reduce el número de cálculos en cada iteración.
- Si la función objetivo es ruidosa el gradiente estocástico permite suavizar la función objetivo y reduce el riesgo de tener una convergencia temprana.

Desventajas del Stochastic Gradient Descent:

- El efecto de los *outliers* en el gradiente de una muestra puede afectar y desviar al algoritmo de su trayectoria de convergencia.

Cross Gradient Booster

Es un modelo el cual está formado por un conjunto de árboles de decisiones individuales los cuales están entrenados de una forma secuencial, de este modo cada árbol mejorar los errores que obtienen los árboles anteriores. La predicción de una nueva observación se obtiene agregando las predicciones de todos los árboles individuales que forman el modelo.

El Gradient Booster fundamenta en la idea de entrenamiento de modelos débiles, esto significa que este tipo de modelo genera arboles de decisión que suelen tener entre 1 y 6 niveles de profundidad y genera que la velocidad en la ejecución del algoritmo sea rápida, posibilitar un aprendizaje lento y progresivo esto significa que sea paso a paso, además de facilitar la detección del overfitting en el proceso.

El algoritmo permite emplear cualquier función de coste siempre que esta sea diferenciable, esto puede ser entendido como un algoritmo de optimización en una función de coste adecuada. El principio principal es entrenar modelos de forma secuencial, de forma que cada tipo de modelo se ajusta a los errores que se obtienen de los modelos anteriores.

El proceso es el siguiente: se ajusta un primer weak learner f_1 con el que se predice la variable respuesta y, y se calculan los residuos $y-f_1(x)$. A continuación, se ajusta un nuevo modelo f_2, que intenta predecir los residuos del modelo anterior, en otras palabras, trata de corregir los errores que ha hecho el modelo f_1.

Entonces se tiene:

$$f_1(x) \approx y$$

$$f_2(x) \approx y - f_1(x)$$

En la siguiente iteración, se calculan los residuos de los dos modelos de forma conjunta $y - f_1(x) - f_2(x)$, los errores cometidos por f_1 y que f_2 no ha sido capaz de corregir, y se ajusta un tercer modelo f_3 para tratar de corregirlos.

$$f_3(x) \approx y - f_1(x) - f_2(x)$$

El proceso se repite n veces de manera que cada nuevo modelo va reduciendo los errores que genera el modelo anterior. Como el objetivo del algoritmo es minimizar los errores en cada iteración este susceptible al overfitting, por lo cual una manera de evitarlo es emplear un learning rate que limite la influencia de cada modelo en el conjunto del ensemble y como consecuencia de esto se debe de emplear mas modelos para formar el ensemble para conseguir mejores resultados.

$$f_1(x) \approx y$$

$$f_2(x) \approx y - \lambda f_1(x)$$

$$f_3(x) \approx y - \lambda f_1(x) - \lambda f_2(x)$$

$$y \approx \lambda f_1(x) + \lambda f_2(x) + \lambda f_3(x) + .. + \lambda f_m(x)$$

Responda las siguientes preguntas:

1. ¿Qué modelo obtiene la mejor “Accuracy”?

El modelo de entrenamiento que obtiene mejor accuracy es el Cross Gradient Booster con un valor de 0.87988, este obtiene un valor alto ya que este tipo de modelo utiliza varios conjuntos de árboles de decisiones ordenados de forma secuencial los cuales van corrigiendo el error que el anterior dejo.

2. ¿Qué configuración de modelo y datos?

La configuración que se utilizo para el modelo Cross Gradient Booster fue un numero de estimaciones de 1000 y un learning rate de 0.05.

3. ¿Utilizando GridSearch+Cross Validation se mejoran los datos?

Si se observa una mejora en los datos ya que se extraen los mejores valores y combinaciones de los parámetros de estos mismos datos, mejorando a los modelos de clasificación.