CS246: Mining Massive Data Sets Problem Set 4

Name: Dat Nguyen
Date: 07/04/2019

By turning in this assignment, I agree by the Stanford honor code and declare that all of this is my own work.

1 Implementation of SVM via Gradient Descent (30 points)

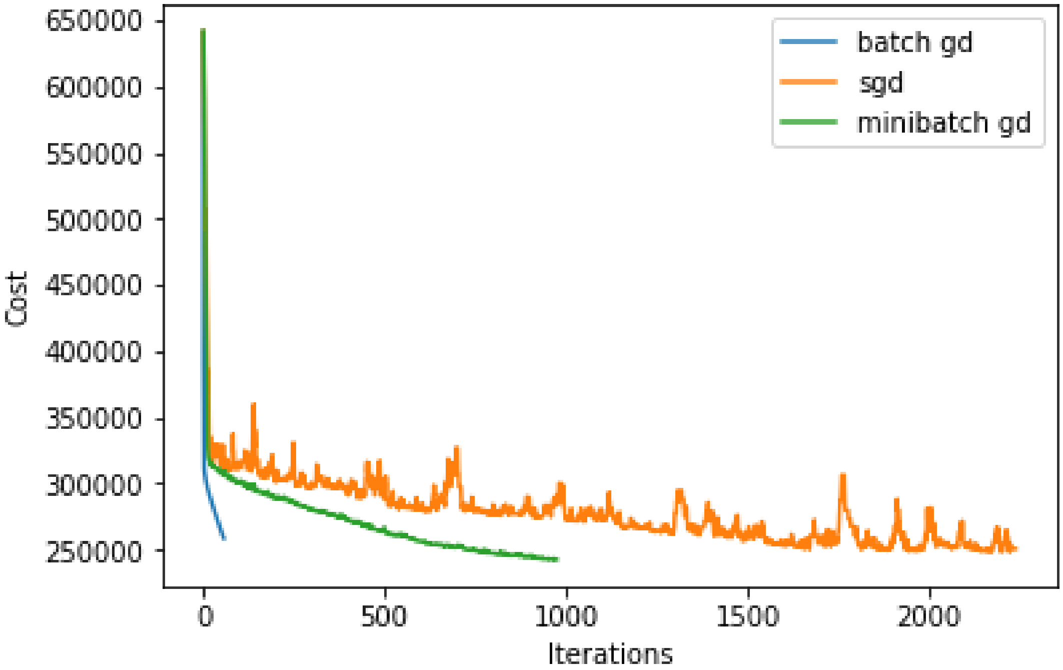
(a)

$$\nabla f_l(\mathbf{w}, b) = C \sum_{i=l*batch\_size+1}^{min(n,(l+1)*batch\_size)} \frac{\partial L(x_i, y_i)}{\partial b}$$

where

$$\frac{\partial L(x_i, y_i)}{\partial b} = \begin{cases} 0 & \text{if } y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 \\ -y_i & \text{otherwise} \end{cases}$$

(b) Plot of $f_k(\mathbf{w}, b)$ vs number of updates.



Total time taken for convergence for batch gradient descent is 0.58s, for stochastic gradient descent is 1.83s, for batch gradient descent is 0.85s.

From the plot we see that the batch gradient descent takes the least number of steps, SGD takes the most number of steps and minibatch gradient descent is in between. Also the plot for batch GD is smoothest, for SGD is roughest and for minibatch is in between. This is because for batch GD the cost is guaranteed to decrease after each step but for SGD the cost can fluctuate since at each step we only update the weight with respect to only one training example. The approach by minibatch GD mediates between two extremes so the cost fluctuation is also the mediation of the two approaches above.
The convergence time for batch GD is the least since it takes less iterations to converge so the number of times to calculate the cost is fewest. SGD is slowest since it takes many iterations to converge and in each iteration we need to evaluate the cost. Minibatch GD takes the time in between since the number of required iterations is also in between.

2 Decision Tree Learning (20 points)

(a) The Gini index of the original sample set is

$$I(D) = 100 \times (1 - 0.6^2 - 0.4^2) = 48$$

Assume that for an attribute items have positive value go to the left tree and have negative value go to the right tree. If we use "likes wine" as the attribute to split, the Gini index of the left and right tree is

$$I(D_L) = 50 \times (1 - 0.6^2 - 0.4^2) = 24$$
$$I(D_R) = 50 \times (1 - 0.6^2 - 0.4^2) = 24$$

So the G value is $G_{wine} = I(D) - (I(D_L) + I(D_R)) = 48 - (24 + 24) = 0$
The Gini index if we use "likes running" as the attribute to split is

$$I(D_L) = 30 \times \left(1 - (\frac{2}{3})^2 - (\frac{1}{3})^2\right) = 13.33$$
$$I(D_R) = 70 \times \left(1 - (\frac{4}{7})^2 - (\frac{3}{7})^2\right) = 34.29$$

Therefore the G value is $G_{running} = 48 - (13.33 + 34.29) = 0.38$
The Gini index if we use "likes pizza" as the attribute to split is

$$I(D_L) = 80 \times \left(1 - (\frac{5}{8})^2 - (\frac{3}{8})^2\right) = 37.5$$
$$I(D_R) = 20 \times \left(1 - (\frac{1}{2})^2 - (\frac{1}{2})^2\right) = 10$$

Therefore the G value is $G_{pizza} = 48 - (37.5 + 10) = 0.5$
Because the gain G for "likes pizza" is largest, we will choose it as the attribute to split the data at the root.

(b) The decision tree will have a_0 as the top node and and all the nodes in each layer will use the same attribute which the more y depends on, the closer to the top that it is. To avoid overfitting we should only keep the top node which uses a_0 as splitting attribute. This is because with only a_0 we already achieved 99% accuracy on training which means that the datapoints depend very strongly on a_0 and very little on other attributes. Therefore it is best to avoid other attributes as they are likely to introduce noise.

3 Clustering Data Streams (20 points)

(a) We have

$$\begin{aligned}
RHS &= 2 \cdot \text{cost}_w(\hat{S}, T) + 2\sum_{i=1}^{l} \text{cost}(S_i, T_i) \\
&= 2\sum_{i=1}^{l}\sum_{j=1}^{k} |S_{ij}| d(t_{ij}, T)^2 + 2\sum_{i=1}^{l}\sum_{j=1}^{k}\sum_{x \in S_{ij}} d(x, t_{ij})^2 \\
&= 2\sum_{i=1}^{l}\sum_{j=1}^{k}\sum_{x \in S_{ij}} d(t_{ij}, T_{ij})^2 + 2\sum_{i=1}^{l}\sum_{j=1}^{k}\sum_{x \in S_{ij}} d(x, t_{ij})^2 \\
&\quad (\text{Let } T_{ij} = \text{min}_{y \in T} d(t_{ij}, y)) \\
&\geq \sum_{i=1}^{l}\sum_{j=1}^{k}\sum_{x \in S_{ij}} d(x, T_{ij})^2 \quad \text{(By triangular inequality for Euclid distance)} \\
&\geq \sum_{i=1}^{l}\sum_{j=1}^{k}\sum_{x \in S_{ij}} d(x, M(x))^2 \\
&\quad (\text{where } M(x) = \text{min}_{y \in T} d(x, y)) \\
&= \sum_{x \in S} d(x, M(x))^2 \\
&= \text{cost}(S, T)
\end{aligned}$$

So we conclude that

$$\text{cost}(S, T) \leq 2 \cdot \text{cost}_w(\hat{S}, T) + 2\sum_{i=1}^{l} \text{cost}(S_i, T_i) \qquad (1)$$

(b) We have

$$\begin{aligned}\sum_{i=1}^{l}\text{cost}(S_i,T_i) &\leq \sum_{i=1}^{l}\alpha\cdot\text{cost}(S_i,T_i^*)\\ &\leq \alpha\cdot\sum_{i=1}^{l}\text{cost}(S_i,T^*)\\ &\quad\text{(if not we just take the centroids in } T^* \text{ to be the corresponding centroids } T_i^* \text{ of } S_i)\\ &= \alpha\cdot\text{cost}(S,T^*)\end{aligned}$$

So we conclude that

$$\sum_{i=1}^{l}\text{cost}(S_i,T_i) \leq \alpha\cdot\text{cost}(S,T^*)(2)$$

(c) Because of ALG approximated algorithm, we have

$$\text{cost}_w(\hat{S},T) \leq \alpha\cdot\text{cost}_w(\hat{S},T^*) \qquad (3)$$

In addition

$$\begin{aligned}&2\sum_{i=1}^{l}\text{cost}(S_i,T_i)+2\cdot\text{cost}(S,T^*)\\ =&2\sum_{i=1}^{l}\sum_{j=1}^{k}\sum_{x\in S_{ij}}d(x,t_{ij})^2+2\sum_{i=1}^{l}\sum_{j=1}^{k}\sum_{x\in S_{ij}}d(x,M(x))^2 \quad (\text{where } M(x)=\min_{y\in T^*}d(x,y))\\ \geq&2\sum_{i=1}^{l}\sum_{j=1}^{k}\sum_{x\in S_{ij}}d(t_{ij},M(x))^2 \qquad \text{(By triangular inequality for Euclid distance)}\\ \geq&2\sum_{i=1}^{l}\sum_{j=1}^{k}\sum_{x\in S_{ij}}d(t_{ij},T_{ij}^*)^2 \quad (\text{where } T_{ij}^*=\min_{y\in T^*}d(t_{ij},y)\ )\\ =&2\sum_{i=1}^{l}\sum_{j=1}^{k}|S_{ij}|d(t_{ij},T_{ij}^*)^2\\ =&\text{cost}_w(\hat{S},T^*)\end{aligned}$$

Therefore

$$\text{cost}_w(\hat{S},T^*) \leq 2\sum_{i=1}^{l}\text{cost}(S_i,T_i)+2\cdot\text{cost}(S,T^*) \qquad (4)$$

From (3) and (4) we have

$$\mathrm{cost}_w(\hat{S}, T) \leq 2\alpha \sum_{i=1}^{l} \mathrm{cost}(S_i, T_i) + 2\alpha \cdot \mathrm{cost}(S, T^*) \tag{5}$$

Plugging (5) and (2) into (1) we have

$$\mathrm{cost}(S, T) \leq (4\alpha^2 + 6\alpha) \cdot \mathrm{cost}(S, T^*)$$

4 Data Streams (30 points)

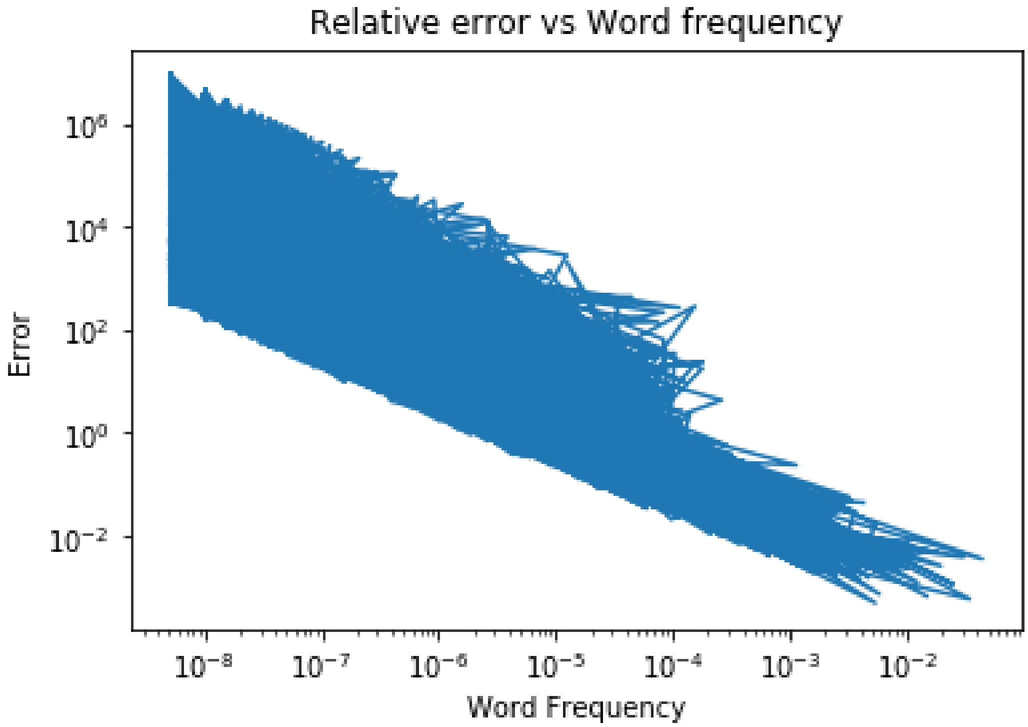
(a) We have

$$\begin{aligned}
& 1 - Pr(\tilde{F}[i] \leq F[i] + \epsilon t) \\
&= Pr(\tilde{F}[i] > F[i] + \epsilon t) \\
&= Pr(\mathrm{min}_j c_{j,h_j(i)} > F[i] + \epsilon t) \\
&= Pr(c_{1,h_1(i)} > F[i] + \epsilon t, \ldots, c_{\lceil \log \frac{1}{\delta} \rceil, h_{\lceil \log \frac{1}{\delta} \rceil}(i)} > F[i] + \epsilon t) \\
&= \prod_{j=1}^{\lceil \log \frac{1}{\delta} \rceil} Pr(c_{j,h_j(i)} > F[i] + \epsilon t) \qquad \text{(Because of the independence of hash functions)} \\
&= \prod_{j=1}^{\lceil \log \frac{1}{\delta} \rceil} Pr(c_{j,h_j(i)} - F[i] > \epsilon t) \\
&\leq \prod_{j=1}^{\lceil \log \frac{1}{\delta} \rceil} \frac{\mathbb{E}[c_{j,h_j(i)} - F[i]]}{\epsilon t} \qquad \text{(From Markov inequality)} \\
&= \prod_{j=1}^{\lceil \log \frac{1}{\delta} \rceil} \frac{\mathbb{E}\big[\mathbb{E}[c_{j,h_j(i)} - F[i] | F[i]]\big]}{\epsilon t} \qquad \text{(Law of total expectation)} \\
&= \prod_{j=1}^{\lceil \log \frac{1}{\delta} \rceil} \frac{\mathbb{E}\big[\frac{t - F[i]}{\lceil \frac{e}{\epsilon} \rceil}\big]}{\epsilon t} \\
&= \prod_{j=1}^{\lceil \log \frac{1}{\delta} \rceil} \frac{t - \mathbb{E}[F[i]]}{\lceil \frac{e}{\epsilon} \rceil \epsilon t} \\
&\leq \prod_{j=1}^{\lceil \log \frac{1}{\delta} \rceil} \frac{t - \mathbb{E}[F[i]]}{e t} \\
&\leq \Big(\frac{1}{e}\Big)^{\log \frac{1}{\delta}} \Big(\frac{t - \mathbb{E}[F[i]]}{t}\Big)^{\log \frac{1}{\delta}} \qquad \text{(Because } \frac{t - \mathbb{E}[F[i]]}{et} \leq 1\text{)} \\
&\leq \delta \qquad \text{(Because } \frac{t - \mathbb{E}[F[i]]}{t} \leq 1\text{)}
\end{aligned}$$

So we conclude that

$$Pr(\tilde{F}[i] \leq F[i] + \epsilon t) \geq 1 - \delta$$

(b) Log-log plot of the relative error as a function of the frequency



The relative error is below 1 for word frequency approximately above $10^{-}4$.