CS246: Mining Massive Data Sets

Assignment number: 4

Fill in and include this cover sheet with each of your assignments. It is an honor code violation to write down the wrong time. Assignments and code are due at 5:00 PM on Scoryst and SNAP respectively. Failure to include the coversheet with you assignment will be penalized by 2 points. Each student will have a total of *two* free late periods. *One late period expires at the start of each class.* (Assignments are due on Thursdays, which means the first late period expires on the following Tuesday at 5:00 PM.) Once these late periods are exhausted, any assignments turned in late will be penalized 50% per late period. However, no assignment will be accepted more than one late period after its due date. (If an assignment is due to Thursday then we will not accept it after the following Thursday.)

Your name:Erli Zhou
Email:erlizhou@stanford.edu
SUNet ID:erlizhou

Collaborators: Jinfeng Huang, Ling-Ling Zhang

I acknowledge and accept the Honor Code.

*(Signed)*Erli Zhou

Answer to Question 1a

A sample training set is:

$\boldsymbol{x}_1 = (x_1^{(1)}, x_1^{(2)}) = (3, 2), y_1 = 1$

$\boldsymbol{x}_2 = (x_2^{(1)}, x_2^{(2)}) = (4, 2), y_2 = 1$

$\boldsymbol{x}_3 = (x_3^{(1)}, x_3^{(2)}) = (2, 4), y_3 = 1$

$\boldsymbol{x}_4 = (x_4^{(1)}, x_4^{(2)}) = (3, 4), y_4 = -1$

$\boldsymbol{x}_5 = (x_5^{(1)}, x_5^{(2)}) = (2, 3), y_5 = -1$

Answer to Question 1b

Suppose $\{(\boldsymbol{x}_i, y_i) | 1 \leqslant i \leqslant n\}$ is the dataset.

We set $\varepsilon_i = \max(0, 1 - y_i \cdot (\boldsymbol{x}_i \cdot \boldsymbol{w} + b)), 1 \leqslant i \leqslant n$

Then the optimization constraints $y_i \cdot (\boldsymbol{x}_i \cdot \boldsymbol{w} + b) \geqslant 1 - \varepsilon_i, \forall i = 1, ..., n$ and $\varepsilon_i \geqslant 0, \forall i = 1, ..., n$ always hold.

Answer to Question 1c

Suppose $\{(\boldsymbol{x}_i, y_i) | 1 \leqslant i \leqslant n\}$ is the dataset.

If the ith observation is misclassified, which is equivalent to $y_i \cdot (\boldsymbol{x}_i \cdot \boldsymbol{w} + b) < 0$. Since $y_i \cdot (\boldsymbol{x}_i \cdot \boldsymbol{w} + b) \geqslant 1 - \varepsilon_i$, the slack variable $\varepsilon_i > 1$.

This shows every error in classification will contribute more than 1 to $\sum_{i=1}^{n} \varepsilon_i$. As a result, $\sum_{i=1}^{n} \varepsilon_i$ is an upper bound to the total number of errors made by the linear classification.

Answer to Question 1d

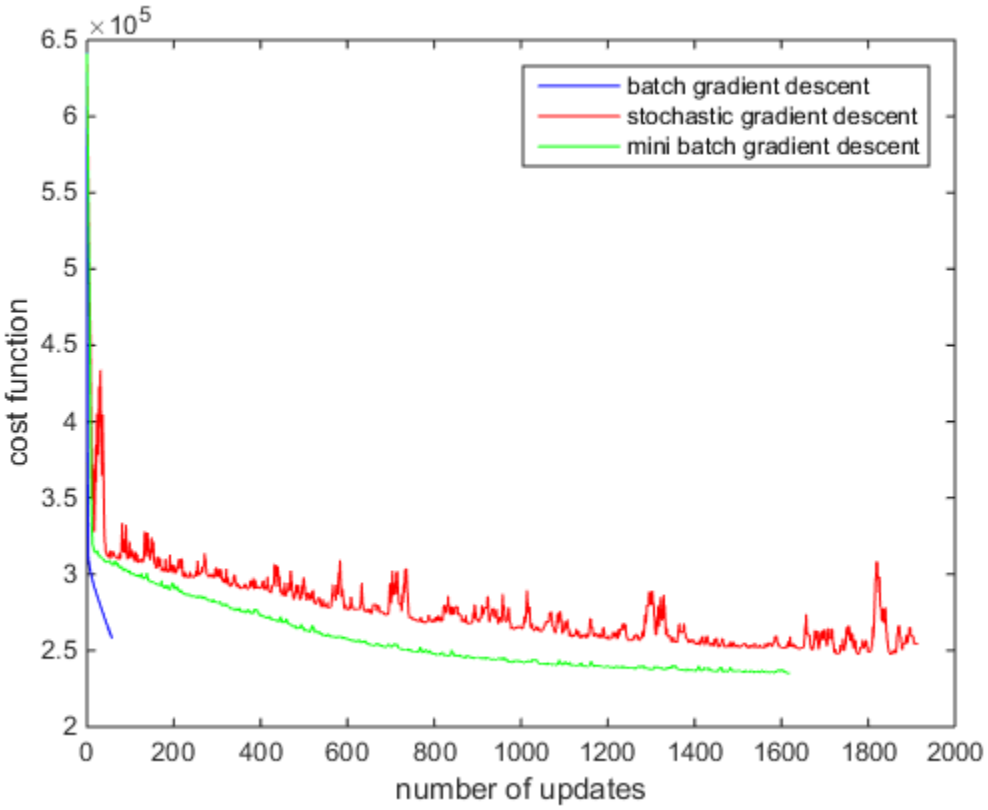
$$\nabla_b f(\boldsymbol{w}, b) = \frac{\partial f}{\partial b}(\boldsymbol{w}, b) = C \sum_{i=1}^{n} \frac{\partial L}{\partial b} L(\boldsymbol{x}_i, y_i)$$

where:

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

Answer to Question 1e

Batch gradient descent converged in 23.24 seconds and 57 iterations.
Stochastic gradient descent converged in 1009.67 seconds and 3020 iterations.
Mini batch gradient descent converged in 489.07 seconds and 1406 iterations.

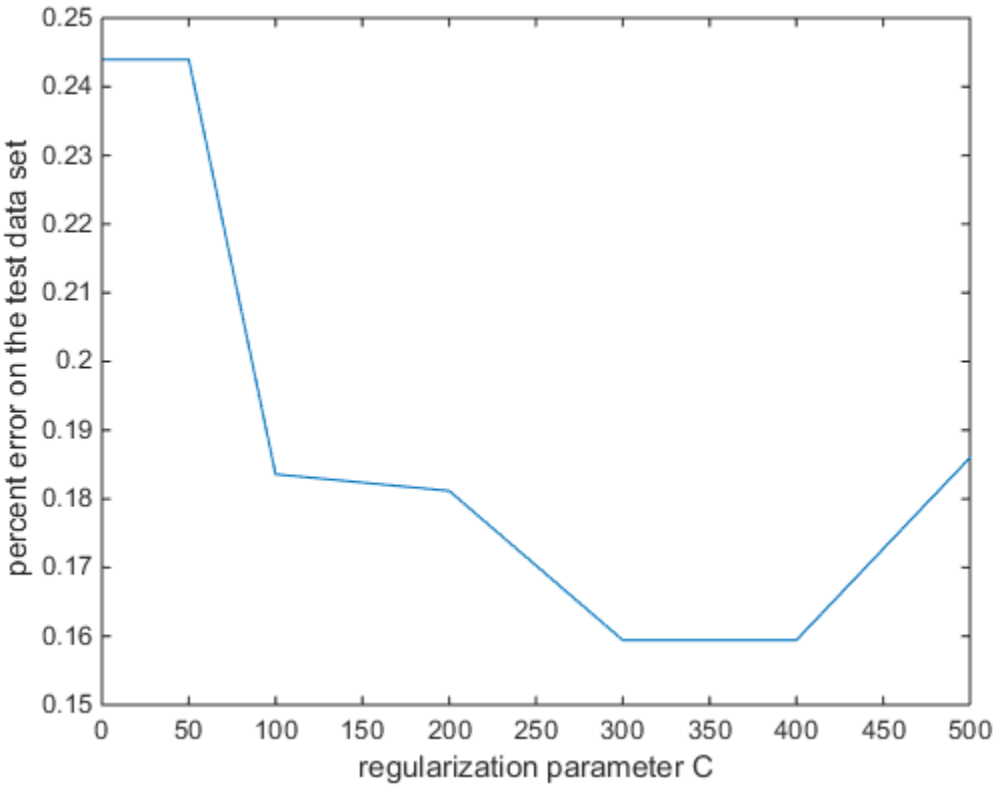


We can see although batch gradient descent converged to a cost function which is higher than stochastic gradient descent or mini batch gradient descent's convergence, it was able to converge in much fewer iterations and in much less time. Besides at the step of its convergence other methods still have a much higher cost function. It is also the only algorithm which decreases the cost function at each iteration.

Compared to the mini batch gradient descent, the stochastic gradient descent results in a cost function which vacillates up and down more than the mini batch gradient descent. Unsurprisingly it also took more iterations to converge, even though its final cost function is slightly higher than mini batch gradient descent's.

As for time per iteration, batch gradient descent takes the longest to run with around 0.41 seconds per iteration. Stochastic gradient descent takes the shortest time to compute with approximately 0.33 seconds per iteration. Mini batch gradient descent takes slightly longer with about 0.35 seconds per iteration.

Answer to Question 1f



The plot means that in general, as regularization parameter increases from 1 to 500, the percent error of the test data set decreases. The plot may be inconsistent at some point(e.g. C = 100) due to randomness of the algorithm but it explains the concept that if we increase our penalty for misclassification, a more precise model will be trained well. The fact that the percentage of error for C = 500 is larger than that of C = 400 maybe because we have overfit the data by setting a high regularization parameter, which causes the test error to increase even though the training error is decreasing.

Answer to Question 2a

$I(D) = 100 * [1 - (\frac{60}{100})^2 - (\frac{40}{100})^2] = 48$

For "likes wine" binary attribute, $|D_L| = |D_R| = 50, I(D_L) = I(D_R) = 50 * [1 - (\frac{30}{50})^2 - (\frac{20}{50})^2] = 24, I(D) - I(D_L) - I(D_R) = 0$

For "likes running" binary attribute, $|D_L| = 30, |D_R| = 70, I(D_L) = 30 * [1 - (\frac{20}{30})^2 - (\frac{10}{30})^2] = 13.33, I(D_R) = 70 * [1 - (\frac{40}{70})^2 - (\frac{30}{70})^2] = 34.29, I(D) - I(D_L) - I(D_R) = 0.38$

For "likes pizza" binary attribute, $|D_L| = 80, |D_R| = 20, I(D_L) = 80 * [1 - (\frac{50}{80})^2 - (\frac{30}{80})^2] = 37.5, I(D_R) = 20 * [1 - (\frac{10}{20})^2 - (\frac{10}{20})^2] = 10, I(D) - I(D_L) - I(D_R) = 0.5$

We will use "likes pizza" binary attribute because it has the highest value of gini index metric G.

Answer to Question 2b

The decision tree classifier will identify a_1 as the most important attribute and use it to split the training examples at the root node. The rest of the attributes will occur in other parts of the tree.

However this decision tree will overfit the data. A desired decision tree will contain just one split on a_1, which label examples as + if $a_1 = 1$ and as - if $a_1 = 0$. This model will avoid overfitting, be easier to understand and also able to predict with high accuracy.

Answer to Question 2c

In the original tree T_0, error $= \dfrac{1+2+1+1+1+2}{1+2+10+2+6+1+7+1+3+1+2+10} = \dfrac{4}{23}$

For tree T_1, we prune the right-most node corresponding to a decision on Z since the node combines the leaves perfectly without increasing the error. We replace it with a leaf containing $6+7=13$ positive examples and $1+1=2$ negative examples. In tree T_1, error $= \frac{1+2+2+1+2}{46} = \frac{4}{23}$

For tree T_2, we prune the left-most node corresponding to a decision on Z since the node increases the apparent error per pruned node the lowest. We replace it with a leaf containing $1+10=11$ positive examples and $2+2=4$ negative examples. In tree T_2, error $= \frac{4+2+1+2}{46} = \frac{9}{46}$

For tree T_3, we prune the left-most node corresponding to a decision on Y since the node combines the leaves perfectly without increasing the error. We replace it with a leaf containing $11+13=24$ positive examples and $4+2=6$ negative examples. In tree T_3, error $= \frac{6+1+2}{46} = \frac{9}{46}$

For tree T_4, we prune the right-most node corresponding to a decision on Y since the node increases the apparent error per pruned node the lowest. We replace it with a leaf containing $3+2=5$ positive examples and $1+10=11$ negative examples. In tree T_4, error $= \frac{6+5}{46} = \frac{11}{46}$

For tree T_5, we prune the X node. We replace it with a leaf containing $24+5=29$ positive examples and $6+11=17$ negative examples. In tree T_5, error $= \frac{17}{46}$

Answer to Question 2d

T_0 is wrong on the second and fourth example of the test set, with a generalization error of $1/2$.

T_1 is wrong on the second and fourth example of the test set, with a generalization error of $1/2$.

T_2 is wrong on the second example of the test set, with a generalization error of $1/4$.

T_3 is wrong on the second example of the test set, with a generalization error of $1/4$.

T_4 is right on all examples of the test set, with a generalization error of 0.

T_5 is wrong on the first and second of the test set, with a generalization error of $1/2$.

So T_3 has the best generalization error.

Answer to Question 3a

Since $\sum_{t_{ij} \in \hat{S}} |S_{ij}| = |S|, \forall x \in S$, suppose $x \in S_i$ and let $z = \arg\min_{t \in T} d(t, x)$.

Using the triangle inequality, $d(x, z) \leqslant d(x, t_{ix}) + d(t_{ix}, z)$ for $t_{ix} = \arg\min_{tix \in T_i} d(x, t_{ix})$

Take squares on each side, we get $d(x, z)^2 \leqslant (d(x, t_{ix}) + d(t_{ix}, z))^2 \leqslant 2d(x, t_{ix})^2 + 2d(t_{ix}, z)^2$ since $(m + n)^2 \leqslant 2m^2 + 2n^2$

Sum over all the elements for $x \in S, \sum_{x \in S} \text{cost}(x, T) = \sum_{i=1}^{\ell} \sum_{x \in S_i} \min_{z \in T} d(x, z)^2$

$\leqslant \sum_{i=1}^{\ell} \sum_{x \in S_i} (2 \min_{t_{ix} \in T_i} d(x, t_{tix})^2 + 2 \min_{z \in T} d(t_{ix}, z)^2) = 2\text{cost}_w(\hat{S}, T) + 2\sum_{i=1}^{\ell} \text{cost}(S_i, T_i)$

Answer to Question 3b

Due to the nature of ALG's design, $\forall |T'| = k$, $\text{cost}(S_i, T_i) \leqslant \alpha \text{cost}(S_i, T')$

Thus, $\sum_{i=1}^{\ell} \text{cost}(S_i, T_i) \leqslant \alpha \sum_{i=1}^{\ell} \text{cost}(S_i, T^*) = \alpha \sum_{i=1}^{\ell} \sum_{x \in S_i} d(x, T^*)^2 = \alpha \sum_{x \in S} d(x, T^*)^2 = \alpha \text{cost}(S, T^*)$

Answer to Question 3c

Let $T_* = \arg\min_{|T'|=k} \text{cost}_w(\hat{S}, T')$, so $\text{cost}_w(\hat{S}, T) \leqslant \alpha{\cdot}\text{cost}_w(\hat{S}, T_*)$

Since $\text{cost}_w(\hat{S}, T^*) \geqslant \text{cost}_w(\hat{S}, T_*)$, sum all elements in S and we get $\alpha{\cdot}\text{cost}_w(\hat{S}, T^*) \geqslant \alpha{\cdot}\text{cost}_w(\hat{S}, T_*) \geqslant \text{cost}_w(\hat{S}, T)$

We then suppose $t_{ij} \in \hat{S}$. Using the triangle inequality, $d(t_{ij}, z) \leqslant d(t_{ij}, x) + d(x, z)$ for $z = \arg\min_{z \in T^*} d(t_{ij}, z) = d(x, T^*)$

Like part b, we deduct $d(t_{ij}, z)^2 \leqslant 2d(t_{ij}, x)^2 + 2d(x, z)^2$

Sum over all the elements for $t_{ij} \in \hat{S}$, $\text{cost}_w(\hat{S}, T^*) = \sum_{t_{ij} \in \hat{S}} |S_{ij}| d(t_{ij}, T^*)^2$

$$\leqslant \sum_{t_{ij} \in \hat{S}} \sum_{x \in S_{ij}} (2d(t_{ij}, x)^2 + 2d(x, T^*)^2)$$

$$= \sum_{i=1}^{\ell} \sum_{x \in S_i} d(t_{ij}, x)^2 + 2\sum_{x \in S} d(x, T^*)^2 = 2\sum_{i=1}^{\ell} \text{cost}(S_i, T_i) + 2\text{cost}(S, T^*)$$

Using all above proofs,

$$\text{cost}(S, T) \leqslant 2\text{cost}_w(\hat{S}, T) + 2\sum_{i=1}^{\ell} \text{cost}(S_i, T_i) \leqslant 2\alpha{\cdot}\text{cost}_w(\hat{S}, T^*) + 2\alpha{\cdot}\text{cost}(S, T^*)$$

$$\leqslant 2\alpha \cdot [2\sum_{i=1}^{\ell} \text{cost}(S, T_i) + 2\text{cost}(S, T^*)] + 2\alpha{\cdot}\text{cost}(S, T^*)$$

$$\leqslant 2\alpha[2\alpha{\cdot}\text{cost}(S, T^*) + 2\text{cost}(S, T*)] + 2\alpha{\cdot}\text{cost}(S, T^*)$$

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

Answer to Question 4a

For each i in the data stream, it will increase $c_{j,h_j(i)}, 1 \leqslant j \leqslant \lceil log(\frac{1}{\delta}) \rceil$ by 1.

So word i occurs $F[i]$ times and contributes $F[i]$ to $c_{j,h_j(i)}$.

There also remains the possibility that other items get hashed to $c_{j,h_j(i)}$. Thus, $c_{j,h_j(i)} \geqslant F[i]$.

Since $\tilde{F[i]} = \min_j c_{j,h_j(i)}, \tilde{F[i]} \geqslant F[i]$.

Answer to Question 4b

For $i \in \{1, 2, ..., n\}$, define a random variable $X_j = c_{j,h_j(i)}$.

We have proved that $X_j \geqslant c_{j,h_j(i)}$ in part a, now we define an indicator random variable $Y_{j,m}$ for $m \neq i$ and $m \in \{1, 2, ..., n\}$ where:

$$Y_{j,m} = \begin{cases} 1 & \text{if } h_j(i) = h_j(m) \\ 0 & \text{otherwise} \end{cases}$$

We now express $X_j = F[i] + \sum_{1 \leqslant m \neq i \leqslant n} Y_{j,m} F[m]$

$E[Y_{j,m}] = Pr[Y_{j,m} = 1] = Pr[h_j(i) = h_j(m)] \leqslant \frac{\epsilon}{e}$

$E[X_j] = F[i] + \sum_{1 \leqslant m \neq i \leqslant n} E[Y_{j,m}] F[m] \leqslant F[i] + \frac{\epsilon}{e} \sum_{1 \leqslant m \neq i \leqslant n} F[m] = F[i] + \frac{\epsilon}{e}(t - F[i])$

So $E[c_{j,h_j(i)}] \leqslant F[i] + \frac{\epsilon}{e}(t - F[i])$

Answer to Question 4c

$Pr[\tilde{F}[i] \leqslant F[i] + \epsilon t] = 1 - Pr[\tilde{F}[i] > F[i] + \epsilon t]$

Since in part a we derived $c_{j,h_j(i)} \geqslant F[i]$,

$Pr[\tilde{F}[i] > F[i] + \epsilon t] \leqslant Pr[c_{j,h_j(i)} > F[i] + \epsilon t], 1 \leqslant j \leqslant \lceil log(1/\delta) \rceil$

Since each hash function is independent of each other,

$$Pr[c_{j,h_j(i)} > F[i] + \epsilon t], 1 \leqslant j \leqslant \lceil log(1/\delta) \rceil = \prod_{j=1}^{\lceil log(1/\delta) \rceil} Pr[c_{j,h_j(i)} > F[i] + \epsilon t]$$

Using Markov's inequality and conclusion from part b,

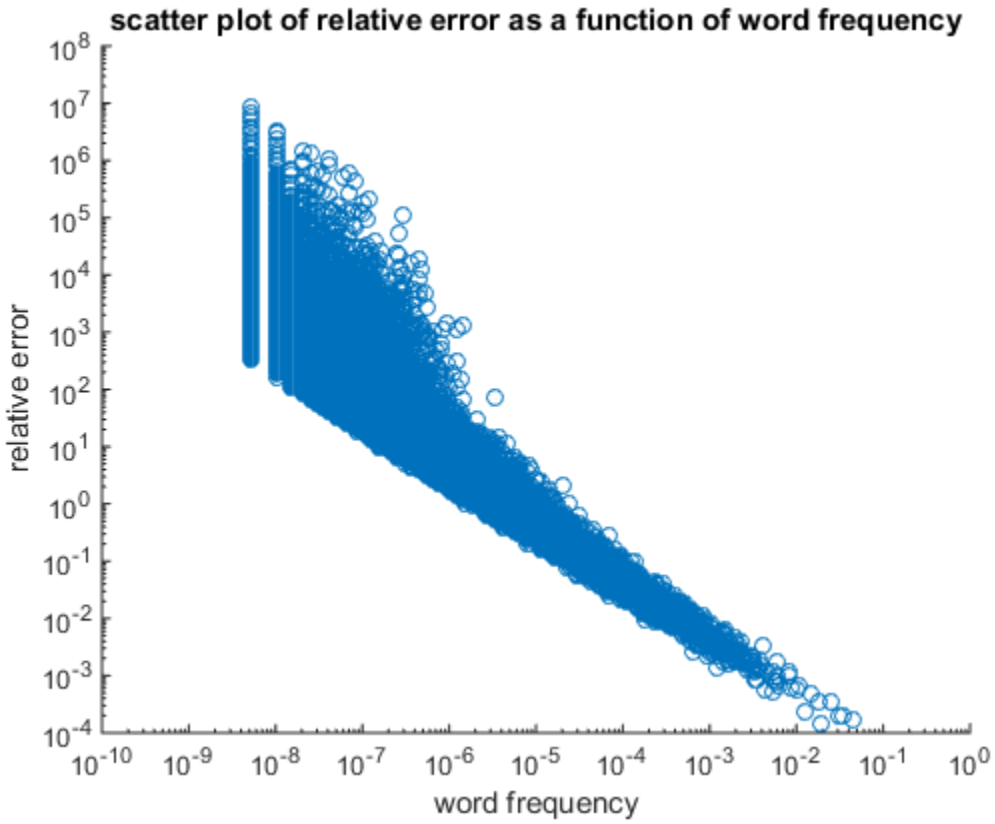
$$Pr[c_{j,h_j(i)} > F[i] + \epsilon t] \leqslant \frac{E[[c_{j,h_j(i)}] - F[i]]}{\epsilon t} \leqslant \frac{t - F[i]}{et} \leqslant \frac{1}{e}$$

As a result, $Pr[\tilde{F}[i] > F[i] + \epsilon t] \leqslant Pr[c_{j,h_j(i)} > F[i] + \epsilon t], 1 \leqslant j \leqslant \lceil log(1/\delta) \rceil \leqslant (\frac{1}{e})^{\lceil log(1/\delta) \rceil}$

$\leqslant (\frac{1}{e})^{log(1/\delta)} = \delta$

So $Pr[\tilde{F}[i] \leqslant F[i] + \epsilon t] \geqslant 1 - \delta$

Answer to Question 4d



For word frequencies larger than 10^{-6}, the relative error falls below 1.