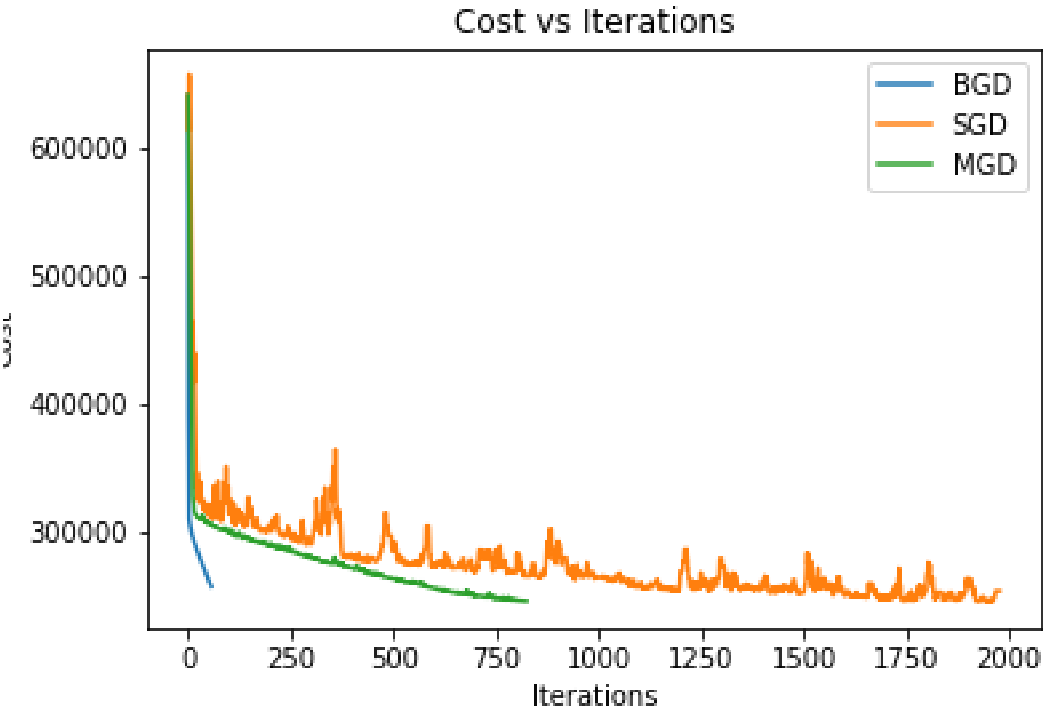
$$\nabla_b f(\mathbf{w}, b) = \frac{\partial f(\mathbf{w}, b)}{\partial b} = C \sum_{i=1}^{n} \frac{\partial L(x_i, y_i)}{\partial b}$$

where

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



Batch gradient descent takes 140 seconds, stochastic gradient descent takes 3.3 seconds, and mini batch gradient descent takes 12.5 seconds. Batch gradient descent needs fewer updates and is more stable, but requires whole dataset and slower than stochastic gradient descent. Stochastic gradient descent evaluates gradient over all examples for each individual training example, it takes less time than batch gradient descent but more unstable. Mini batch gradient descent balances between robustness of batch gradient descent and efficiency of stochastic gradient descent.

Before we do any splitting, the impurity is $I(D) = 100 \times \left(1 - 0.4^2 - 0.6^2\right) = 48$.

1. If we use the wine attribute, 20 out of 50 wine drinkers like beer and 20 of 50 non-wine drinkers like beer. Hence, the impurity of the "like wine" side is $I(D_L) = 50 \times \left(1 - 0.4^2 - 0.6^2\right) = 24$, and the impurity of the "doesn't like wine" side is $I(D_R) = 50 \times \left(1 - 0.4^2 - 0.6^2\right) = 24$. Therefore, $G = I(D) - (I(D_L) + I(D_R)) = 48 - 24 - 24 = 0$. So there is no reduction in impurity.

2. If we use the running attribute, 20 out of 30 runners like beer, and 20 out of 70 non-runners like beer. Therefore, the impurity on the "like running" side is $30 \times \left(1 - 0.66^2 - 0.33^2\right) = 13.33$, and the impurity on the "doesn't like running" side is $70 \times \left(1 - 0.2857^2 - 0.7143^2\right) = 28.5714$ So $G = I(D) - (I(D_L) + I(D_R)) = 48 - 13.33 - 28.5714 = 6.1$.

3. If we use the pizza attribute, 50 out of 80 pizza lovers like beer and 10 out of 20 non-pizza lovers like beer. Therefore, the impurity on the "like pizza" side is $80 \times (1 - 0.375^2 - 0.625^2) = 37.5$, and the impurity on the "doesn't like pizza" side is $20 \times (1 - 0.5^2 - 0.5^2) = 10$. So $G = I(D) - (I(D_L) + I(D_R)) = 48 - 37.5 - 10 = 0.5$.

In conclusion, we should choose the running attribute since it has the largest G.

a_1 will be the root of the tree and left branch denotes $a_1 = 0$, right branch denotes $a_1 = 1$. The desired decision tree which avoids overfitting would have a single decision on the root corresponding to a_1. This is because a_1 is a attribute predictive of the outcome, where the 1% can be considered as noise, and none of the other attributes are predictive of the outcome.

Let $T = \{t_z | z = 1, \ldots, k\}$. Suppose a data point $p \in S_{ij}$ with corresponding clustering center t_{ij}, and $p \in S_z$ with corresponding clustering center t_z. By Triangular Inequality for Euclidean distance, we have

$$||p - t_z||_2 \leq ||p - t_{ij}||_2 + ||t_{ij} - t_z||_2.$$

Then we square the inequality on both sides, we have

$$||p - t_z||_2^2 \leq (||p - t_{ij}||_2 + ||t_{ij} - t_z||_2)^2 \leq 2||p - t_{ij}||_2^2 + 2||t_{ij} - t_z||_2^2.$$

Then we sum up over all $p \in S_{ij}$, we have

$$\sum_{p \in S_{ij}} ||p - t_z||_2^2 \leq 2 \sum_{p \in S_{ij}} ||p - t_{ij}||_2^2 + 2 \sum_{p \in S_{ij}} ||t_{ij} - t_z||_2^2 = 2 \sum_{p \in S_{ij}} ||p - t_{ij}||_2^2 + 2|S_{ij}| \times ||t_{ij} - t_z||_2^2$$

Then we sum up j from 1 to k and i from 1 to l, we have

$$\sum_{i=1}^{l} \sum_{j=1}^{k} \sum_{p \in S_{ij}} ||p - t_z||_2^2 \leq 2 \sum_{i=1}^{l} \sum_{j=1}^{k} \sum_{p \in S_{ij}} ||p - t_{ij}||_2^2 + 2 \sum_{i=1}^{l} \sum_{j=1}^{k} w(t_{ij}) ||t_{ij} - t_z||_2^2,$$

where $w(t_{ij}) = |S_{ij}|$.

The left side of the inequality can be reformulated as

$$\sum_{i=1}^{l} \sum_{j=1}^{k} \sum_{p \in S_{ij}} ||p - t_z||_2^2 = \sum_{p \in S} ||p - t_z||_2^2 = \sum_{p \in S} d(p, T).$$

The right side of the inequality can be reformulated as

$$2 \sum_{i=1}^{l} \sum_{j=1}^{k} w(t_{ij}) ||t_{ij} - t_z||_2^2 + 2 \sum_{i=1}^{l} \sum_{j=1}^{k} \sum_{p \in S_{ij}} ||p - t_{ij}||_2^2 = 2 \sum_{t_{ij} \in \hat{S}} w(t_{ij}) d(t_{ij}, T) + 2 \sum_{i=1}^{l} \sum_{p \in S_i} d(p, T_i)$$

By the definition of the cost function, we have

$$\sum_{p \in S} d(p, T) = \text{cost}(S, T)$$

$$\sum_{t_{ij} \in S} w(t_{ij}) d(t_{ij}, T) = \text{cost}_w(\hat{S}, T)$$

$$\sum_{i=1}^{l} \sum_{p \in S_i} d(p, T_i) = \sum_{i=1}^{l} \text{cost}(S_i, T_i)$$

Therefore, we prove that

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

Since ALG is an α-approximate algorithm, we have

$$\text{cost}(S_i, T_i) \le \alpha \min_{|T'|=k} \{\text{cost}(S_i, T')\} \le \alpha \text{cost}(S_i, T^*)$$

Then we sum up i from 1 to l, we have

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

Since we run ALG on $\hat{S}$ and ALG is a α-approximate algorithm, we have

$$\text{cost}_w(\hat{S}, T) \leq \alpha \cdot \min_{|T'|=k} \{\text{cost}(\hat{S}, T')\} \leq \alpha \cdot \text{cost}_w(\hat{S}, T^*).$$

Hence, we prove the first hint.

Let $T^* = \{t_z^* | z = 1, \ldots, k\}$. Suppose a data point $p \in S_{ij}$ with corresponding clustering center t_{ij} and $p \in S_z^*$ with corresponding clustering center t_z^*. By Triangular Inequality for Euclidean distance, we have

$$||t_{ij} - t_z^*||_2 \leq ||p - t_{ij}||_2 + ||p - t_z^*||_2$$

Then we square the inequality on both sides, we have

$$||t_{ij} - t_z^*||_2^2 \leq (||p - t_{ij}||_2 + ||p - t_z^*||_2)^2 \leq 2 \cdot ||p - t_{ij}||_2^2 + 2 \cdot ||p - t_z^*||_2^2$$

Then we sum up all $p \in S_{ij}$, j from 1 to k, and i from 1 to l, we have

$$\sum_{i=1}^{l} \sum_{j=1}^{k} \sum_{p \in S_{ij}} ||t_{ij} - t_z^*||_2^2 \leq 2 \cdot \sum_{i=1}^{l} \sum_{j=1}^{k} \sum_{p \in S_{ij}} ||p - t_{ij}||_2^2 + 2 \cdot \sum_{i=1}^{l} \sum_{j=1}^{k} \sum_{p \in S_{ij}} ||p - t_z^*||_2^2$$

The left side of the inequality can be reformulated as

$$\sum_{i=1}^{l} \sum_{j=1}^{k} \sum_{p \in S_{ij}} ||t_{ij} - t_z^*||_2^2 = \sum_{i=1}^{l} \sum_{j=1}^{k} |S_{ij}| \cdot ||t_{ij} - t_z^*||_2^2 = \sum_{t_{ij} \in \hat{S}} w(t_{ij}) ||t_{ij} - t_z^*||_2^2 = \sum_{t_{ij} \in \hat{S}} w(t_{ij}) d(t_{ij}, T^*)$$

The right side of the inequality can be reformulated as

$$2 \cdot \sum_{i=1}^{l} \sum_{j=1}^{k} \sum_{p \in S_{ij}} ||p - t_{ij}||_2^2 + 2 \cdot \sum_{i=1}^{l} \sum_{j=1}^{k} \sum_{p \in S_{ij}} ||p - t_z^*||_2^2 = 2 \cdot \sum_{i=1}^{l} \sum_{p \in S_i} d(p, T_i) + 2 \cdot \sum_{p \in S} d(p, T^*)$$

By the definition of the cost function, we have

$$\sum_{t_{ij} \in S} w(t_{ij}) d(t_{ij}, T^*) = \text{cost}_w(\hat{S}, T^*)$$

$$\sum_{i=1}^{l} \sum_{p \in S_i} d(p, T_i) = \sum_{i=1}^{l} \text{cost}(S_i, T_i)$$

$$\sum_{p \in S} d(p, T^*) = \text{cost}(S, T^*)$$

Therefore, we prove the second hint

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

Combine the problem (a), (b), and the two hints, we have

$$\begin{aligned}
\text{cost}(S,T) &\leq 2 \cdot \text{cost}_w(\hat{S},T) + 2\sum_{i=1}^{l} \text{cost}(S_i,T_i) \\
&\leq 2\alpha \cdot \text{cost}_w(\hat{S},T^*) + 2\sum_{i=1}^{l} \text{cost}(S_i,T_i) \\
&\leq 2\alpha \left(2\sum_{i=1}^{l} \text{cost}(S_i,T_i) + 2 \cdot \text{cost}(S,T^*) \right) + 2\sum_{i=1}^{l} \text{cost}(S_i,T_i) \\
&\leq 4\alpha \cdot \text{cost}(S,T^*) + 4\alpha\sum_{i=1}^{l} \text{cost}(S_i,T_i) + 2\sum_{i=1}^{l} \text{cost}(S_i,T_i) \\
&\leq (4\alpha^2 + 6\alpha) \cdot \text{cost}(S,T^*)
\end{aligned}$$

In conclusion, we prove that

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

$$\begin{aligned}
\mathsf{Pr}(\tilde{F}[i] \leq F[i] + \epsilon t) &= 1 - \mathsf{Pr}(\tilde{F}[i] \geq F[i] + \epsilon t) \\
&= 1 - \mathsf{Pr}(\min_j \{c_{j,h_j(i)}\} \geq F[i] + \epsilon t) \\
&= 1 - \mathsf{Pr}(c_{j,h_{j,h_j(i)}} \geq F[i] + \epsilon t, \forall 1 \leq j \leq \left\lceil \log(\frac{1}{\delta}) \right\rceil) \\
&= 1 - \prod_{j=1}^{\lceil \log(\frac{1}{\delta}) \rceil} \mathsf{Pr}(c_{j,h_j(i)} \geq F[i] + \epsilon t)
\end{aligned}$$

The last equality is followed by the independence of hash functions.

By using Markov's inequality, we have

$$\mathsf{Pr}(c_{j,h_j(i)} \geq F[i] + \epsilon t) \leq \frac{\mathsf{E}\left[c_{j,h_j(i)} - F[i]\right]}{\epsilon t}$$

Follow the second property, we have

$$\mathsf{E}[c_{j,h_j(i)}] \leq F[i] + \frac{\epsilon}{e}(t - F[i]) \iff \mathsf{E}\left[c_{j,h_j(i)} - F[i]\right] \leq \frac{\epsilon}{e}(t - F[i]) \leq \frac{\epsilon t}{e}$$

Therefore, we have

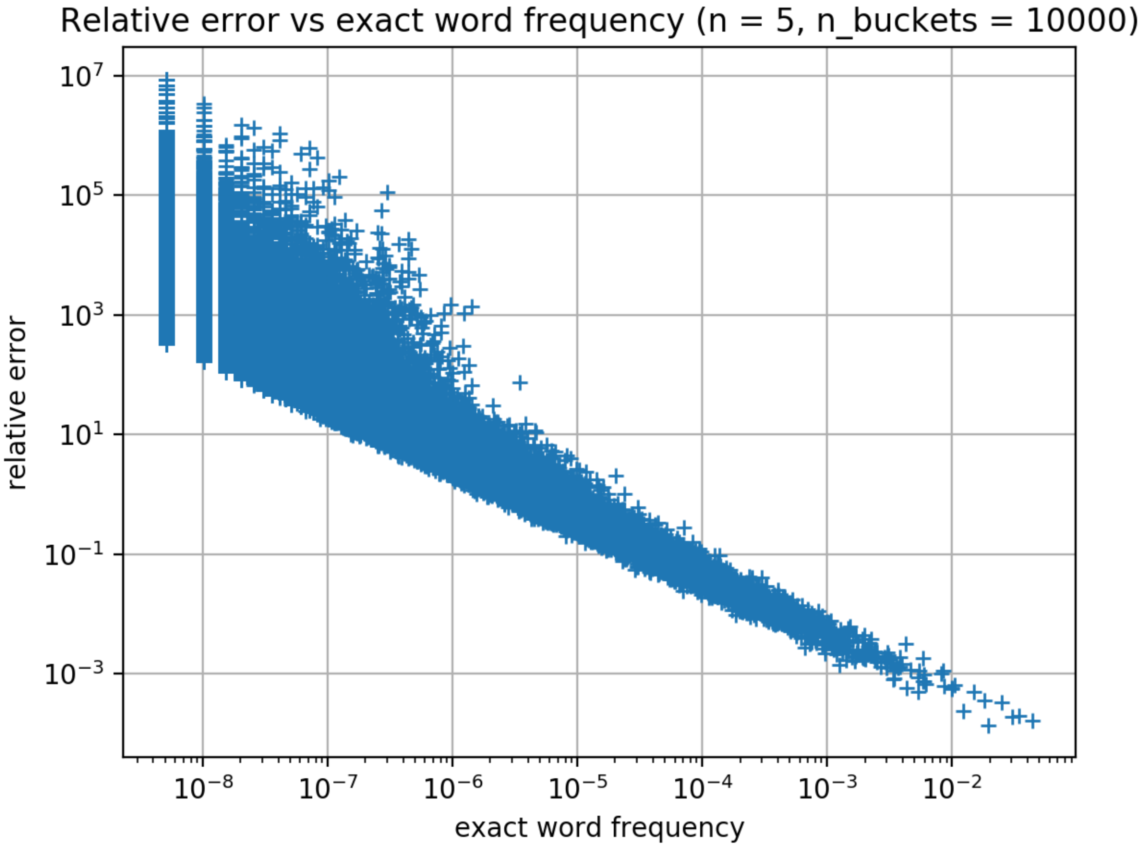
$$\mathsf{Pr}(c_{j,h_j(i)} \geq F[i] + \epsilon t) \leq \frac{\mathsf{E}\left[c_{j,h_j(i)} - F[i]\right]}{\epsilon t} \leq \frac{1}{e}$$

Insert $\mathsf{Pr}(c_{j,h_j(i)} \geq F[i] + \epsilon t) \leq \frac{1}{e}$ into the first equality, we get

$$\mathsf{Pr}(\tilde{F}[i] \leq F[i] + \epsilon t) = 1 - \prod_{j=1}^{\lceil \log(\frac{1}{\delta}) \rceil} \mathsf{Pr}(c_{j,h_j(i)} \geq F[i] + \epsilon t) = 1 - (\frac{1}{e})^{\lceil \log(\frac{1}{\delta}) \rceil} \geq 1 - \delta$$

In conclusion, we prove that

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



Words with frequency over 10^{-5} may have relative error less than 1.

Information sheet
CS246: Mining Massive Data Sets

Assignment Submission Fill in and include this information sheet with each of your assignments. This page should be the last page of your submission. Assignments are due at 11:59pm and are always due on a Thursday. All students (SCPD and non-SCPD) must submit their homework via Gradescope (http://www.gradescope.com). Students can typeset or scan their homework. Make sure that you answer each (sub-)question on a separate page. That is, one answer per page regardless of the answer length. Students also need to upload their code on Gradescope. Put all the code for a single question into a single file and upload it.

Late Homework Policy Each student will have a total of *two* late periods. *Homework are due on Thursdays at 11:59pm PT and one late period expires on the following Monday at 11:59pm PT*. Only one late period may be used for an assignment. Any homework received after 11:59pm PT on the Monday following the homework due date will receive no credit. Once these late periods are exhausted, any assignments turned in late will receive no credit.

Honor Code We strongly encourage students to form study groups. Students may discuss and work on homework problems in groups. However, each student must write down their solutions independently, i.e., each student must understand the solution well enough in order to reconstruct it by him/herself. Students should clearly mention the names of all the other students who were part of their discussion group. Using code or solutions obtained from the web (GitHub/Google/previous year's solutions etc.) is considered an honor code violation. We check all the submissions for plagiarism. We take the honor code very seriously and expect students to do the same.

Your name: Yen-Yu Chang

Email: yenyu@stanford.edu **SUID:** 006350488

Discussion Group: Cheng-Min Chiang, Fang-I Hsiao, Alvin Hou

I acknowledge and accept the Honor Code.

(Signed) Y.Y.Chang