1. (a)

I did this homework with Luning Zhao.

We work on our homework seperately, but we discuss when meeting problems.

Homework is fine.

(b) I certify that all solutions are entirely in my words. and that I have not looked at another student's solutions. I have credited all external sources in this write up.

Siyao Jia

2. (a) $\nabla_{\vec{v}} L(\vec{v}) = \nabla_{\vec{v}} \frac{1}{2}(\vec{y}^T\vec{y} - 2\vec{y}^T\vec{v} + \vec{v}^T\vec{v})$

$= \vec{v} - \vec{y}$

- $\vec{v}^{t+1} = \vec{v}^t - \alpha t \nabla_{\vec{v}} L(\vec{v})|_{\vec{v}=\vec{v}^t}$

$= \vec{v}^t - \alpha t (\vec{v}^t - \vec{y})$

$= (1 - \alpha t)\vec{v}^t + \alpha t \vec{y}$

- $\nabla_{\vec{v}} L(\vec{v})|_{\vec{v}=X\vec{w}} = X\vec{w} - \vec{y}$

- $\vec{v}^{t+1} = (1 - \alpha t) X\vec{w}^t + \alpha t \vec{y}$

- $X\vec{w}^{t+1} = X\vec{w}^t - \alpha t X X^T (X\vec{w}^t - \vec{y})$

The difference between $\vec{v}^{t+1}$ and $X\vec{w}^{t+1}$ is because $\vec{v}$ and $\vec{w}$ lie in different space, so $X\vec{w}^{t+1}$ has an extra XX^T term.

(b) $\tilde{v}^t = \underset{\tilde{v} \in \tilde{V}}{\text{argmax}} \langle -\nabla L(\vec{v}^t), \tilde{v} \rangle$

$= \underset{\tilde{v} = X\vec{w},\ \|\vec{w}\|_2^2 \le 1}{\text{argmax}} \langle -X\vec{w}^t + \vec{y}, \tilde{v} \rangle$

$= X \underset{\|\vec{w}\|^2 \le 1}{\text{argmax}} \langle -X\vec{w}^t + \vec{y}, X\vec{w} \rangle$

$= X \underset{\|\vec{w}\|^2 \le 1}{\text{argmax}} (-\vec{w}^{tT} X^T X\vec{w} + \vec{y}^T X\vec{w})$

Let $\mathcal{L} = -\vec{w}^{tT} X^T X\vec{w} + \vec{y}^T X\vec{w} + \lambda(1 - \vec{w}^T\vec{w})$

$\frac{\partial \mathcal{L}}{\partial \vec{w}} = 0 \Rightarrow -\vec{w}^{tT} X^T X + \vec{y}^T X - 2\lambda\vec{w}^T = 0$

$\Rightarrow \vec{w} \propto -X^T X\vec{w}^t + X^T\vec{y}$

$\Rightarrow \tilde{v}^t = X \underset{\|\vec{w}\|^2 \le 1}{\text{argmax}} (-\vec{w}^{tT} X^T X\vec{w} + \vec{y}^T X\vec{w})$

$\propto XX^T(X\vec{w}^t - \vec{y})$

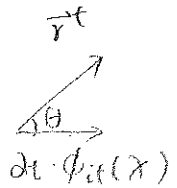
$$\vec{v}^{t+1} = \vec{v}^t + \alpha_t \tilde{v}^t$$
$$= X\vec{w}^t - \alpha_t XX^T(X\vec{w}^t - \vec{y})$$
$$= X\vec{w}^{t+1}$$

(c) $i_t = \underset{i=1,\dots d}{\operatorname{argmax}} \; |\langle \vec{r}^t, \phi_i(x) \rangle|$

$$\vec{v}^t = \underset{\tilde{v} \in \tilde{V}}{\operatorname{argmax}} \; \langle -\nabla L(\vec{v}^t), \tilde{v} \rangle$$
$$= \underset{\tilde{v} \in \tilde{V}}{\operatorname{argmax}} \; \langle -X\vec{w}^t + \vec{y}, \tilde{v} \rangle$$
$$= \underset{\tilde{v} \in \tilde{V}}{\operatorname{argmax}} \; \langle \vec{r}^t, \tilde{v} \rangle$$

$\Rightarrow \tilde{V} = \{\tilde{v} = \pm \phi_i(x), \; i = 1 \dots d\}$

In this case, $\vec{v}^t = \underset{i=1\dots d}{\operatorname{argmax}} \; |\langle \vec{r}^t, \phi_i(x) \rangle|$

(d) 

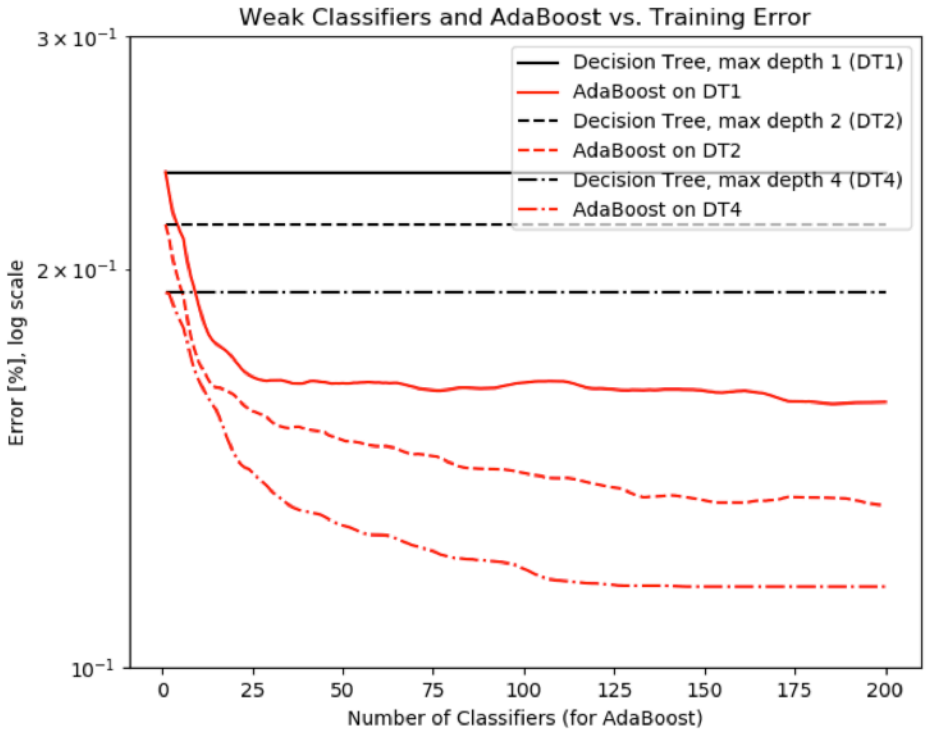
To pick a step-size α that reduces the loss most,
We can project $\vec{r}^t$ into the direction of $\phi_{it}(x)$

Then $\|\vec{r}^t\| \cdot \cos\theta = \alpha_t \|\phi_{it}(x)\|$

$$\alpha_t = \frac{\|\vec{r}^t\| \cdot \langle \vec{r}^t, \phi_{it}(x) \rangle}{\|\vec{r}^t\| \cdot \|\phi_{it}(x)\| \cdot \|\phi_{it}(x)\|} = \frac{\langle \vec{r}^t, \phi_{it}(x) \rangle}{\|\phi_{it}(x)\|^2}$$

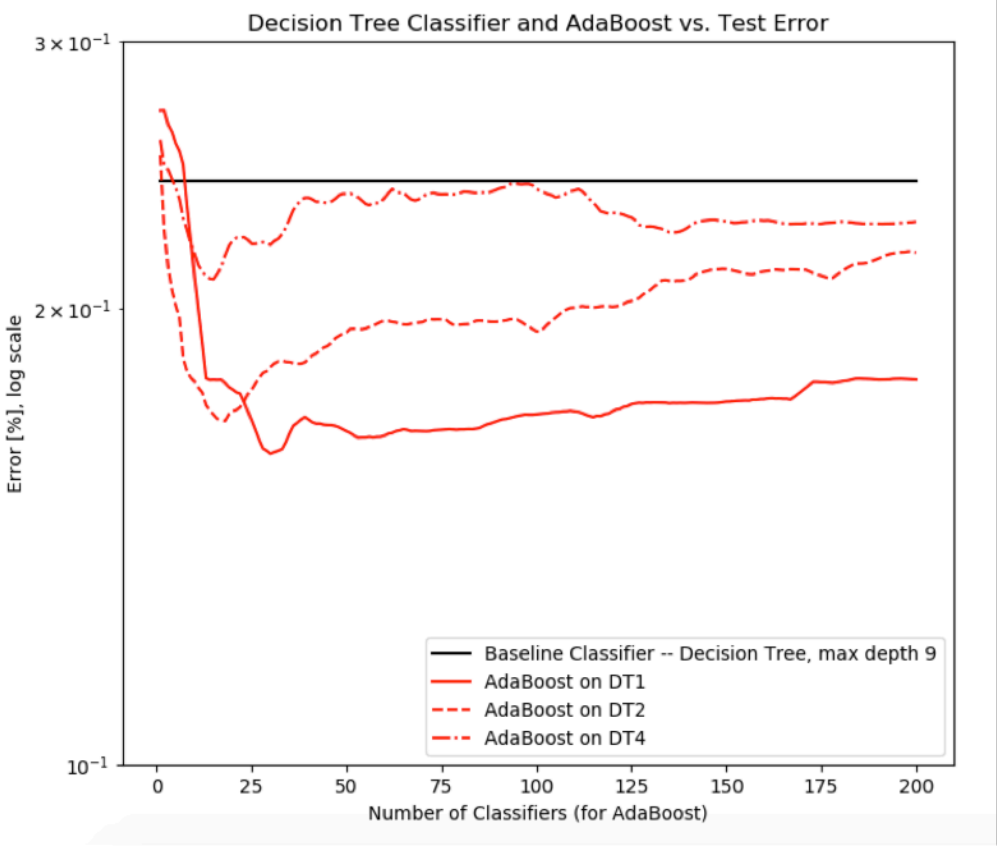
So we can see that obtaining α_t via linesearch is the same as doing MP.

2(h)



Performance is better with deeper trees.
The training error decreases with deeper trees and more classifiers for AdaBoost.
This is because more classifiers give more accurate models, as the trees are improved along more classifiers.

2(i)

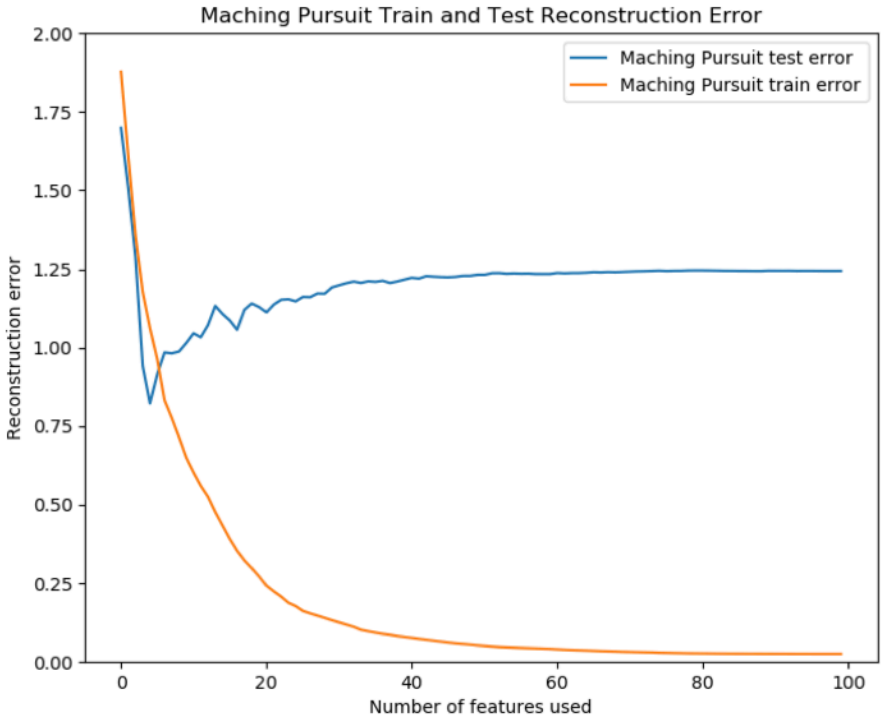


There is a difference between training error and test error.
Training error keeps decreasing when we increase DT, but test error increases from DT1 to DT4.
DT1 works best for AdaBoost.
This is because DT2 and DT4 overfit data.

2 (j)

Limiting the number of base classifiers will help for Adaboost.
From plot in (i), we can run more boosting iterations on base classifier of 20.
Because we have less classifiers, when we use deeper trees, there will be less free parameters, so less likely to have overfitting problem.

2 (k)



Training error decreases with number of features, because we are using more features, so we are fitting the observed y better and better.
Test error is this plot is similar to plot in part(i), also because of overfitting.

4(a) The time complexity for computing distance is $\sim O(dn)$

Finding the k shortest distance using quickselect is $\sim O(n)$.

The time complexity for a single query is $\sim O(dn)$

(b) For 1-d, you need to search 3 cells.

For 2-d, you need to search $3^2 = 9$ cells

For d-d, you need to search 3^d cells

If each cell contains 1 points, the time complexity $O\sim(d\,3^d)$

(c) $\Pr(k\vec{v}^l, \vec{u}| \le |\langle \vec{v}^s, \vec{u}\rangle|)$

$= \Pr(|\langle \vec{v}^l, \vec{u}''\rangle| \le |\langle \vec{v}^s, \vec{u}''\rangle|)$

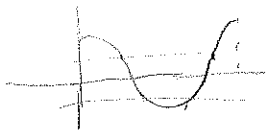
$= \Pr(\|\vec{v}^l\| \|\vec{u}''\| |\cos\theta|) \le \|\vec{v}^s\| \|\vec{u}''\| |\cos\theta^s|)$

$\le \Pr(\|\vec{v}^l\| |\cos\theta| \le \|\vec{v}^s\|)$

$= \Pr(|\cos\theta| \le \|\vec{v}^s\| / \|\vec{v}^l\|)$

(d) Let $\|\vec{v}^s\|/\|\vec{v}^l\| = Vr$.

$|\cos\theta| \le Vr \Rightarrow -Vr \le \cos\theta \le Vr$



$\Rightarrow \theta \in [\cos^{-1} Vr,\ \pi - \cos^{-1} Vr] \cup [\pi + \cos^{-1} Vr,\ 2\pi - \cos^{-1} Vr]$

$$\Pr(|\cos\theta| \le Vr) = \frac{\pi - 2\cos^{-1} Vr \quad + \pi - 2\cos^{-1} Vr}{2\pi}$$

$$= 1 - \frac{2}{\pi}\cos^{-1} Vr$$

$$= 1 - \frac{2}{\pi}\cos^{-1}\frac{\|V^s\|}{\|Vr\|}$$

(e) Pr (failure)

$= Pr$ (at least $\tilde{k}$ points are closer to $\vec{z}$ than $x^{(1)}$ under projection $\vec{u}$)

$\leq \frac{1}{k'} \sum_{i=2}^{N} Pr(\vec{x}^{(i)}$ is closer to $\vec{z}$ than $\vec{x}^{(1)})$

$= \frac{1}{k'} \sum_{i=1}^{k} Pr(|\langle \vec{x}^{(i)} - \vec{z}, \vec{u} \rangle| \leq |\langle \vec{x}^{(1)} - \vec{z}, \vec{u}|)$

$\leq \frac{1}{k'} \sum_{i=2}^{k} (1 - \frac{2}{\pi} \cos^{-1} \frac{\|\vec{x}^{(1)} - \vec{z}\|}{\|\vec{x}^{(i)} - \vec{z}\|})$

$\leq \frac{1}{k'} \sum_{i=2}^{N} \frac{\|x^{(1)} - \vec{z}\|}{\|x^{(i)} - \vec{z}\|}$.

(f) $\|x^{cr^{d'}} - z\| = r$

let $cr^{d'} = i \Rightarrow r = (\frac{i}{c})^{1/d'}$

$\sum_{i=2}^{N} \|x^{(1)} - z\| / \|x^{(i)} - z\| = \sum_{i=2}^{N} \frac{(\frac{1}{c})^{1/d'}}{(\frac{i}{c})^{1/d'}} = \sum_{i=1}^{N} (1/i)^{1/d'}$

(i) Blue: naive exhaustive search.

green: space partitioning

red: DCI

5. Q: prove $P(\bigcup_i A_i) \leq \sum_i P(A_i)$

A: When $i = 1$ $P(A_1) = P(A_1)$

Assume $P(\bigcup_{i=1}^{t} A_i) \leq \sum_{i=1}^{t} P(A_i)$

$P(\bigcup_{i=1}^{t+1} A_i) = P(A_{t+1} \cup \bigcup_{i=1}^{t} A_i)$

$\leq P(A_{t+1}) + P(\bigcup_{i=1}^{t} A_i)$

$\leq P(A_{t+1}) + \sum_{i=1}^{t} P(A_i)$

$= \sum_{i=1}^{t+1} P(A_i)$

So $P(\bigcup_i A_i) \leq \sum_i P(A_i)$