Cover page for answers.pdf
CSE512 Fall 2018 - Machine Learning - Homework 5

Your Name: Sriram Reddy Kalluri

Solar ID: 111878857

NetID email address: skalluri@cs.stonybrook.edu sriram.kalluri@stonybrook.edu

Names of people whom you discussed the homework with:

1) Question 1.

1). Given to prove.

$$\epsilon_{training} = \frac{1}{N}\underbrace{\sum_{j=1}^{N} \delta(H(x^j) \neq y^j)}_{LHS} \leq \frac{1}{N}\underbrace{\sum_{j=1}^{N} \exp(-f(x^j)\, y^j)}_{RHS}.$$

where $\delta(H(x^j) \neq y^j) = 1$ if $H(x^j) \neq y^j$

0 otherwise.

We can write $f(x^j)$ as $\text{sign}\{f(x^j)\}\, |f(x^j)|$

$= H(x^j)\, |f(x^j)|$

Case i): if $H(x^j) \neq y^j$

LHS part $= \delta(H(x^j) \neq y^j) = 1$.

RHS part $= \exp(-f(x^j) \cdot y^j)$.

$= \exp(-H(x^j) \cdot y^j\, |f(x^j)|)$.

$= \exp(|f(x^j)|)$. || $y \in \{-1, 1\}$.

$H(x^j)$ = sign opposite to y^j

LHS $\leq$ RHS.

This is valid for both cases of $y^j = -1$ & $+1$ — (1)

Case 2): if $H(x^j) = y^j$

LHS part $= \delta(H(x^j) \neq y^j) = 0$

RHS part $= \exp(-f(x^j)\, y^j)$

$= \exp(-H(x^j)\, y^j\, |f(x^j)|)$

$= \exp(-|f(x^j)|)$.

From (1), we can say this directly.

exponential is always greater than or equal to 0.

LHS $\leq$ RHS.

Extending this to all the terms, this can be proved for the summation.

So.

$$E_{training} = \frac{1}{N}\sum_{j=1}^{N} \delta(H(x^j) \neq y^j) \leq \frac{1}{N}\sum_{j=1}^{N} \exp(-f(x^j)\, y^j).$$

2). Given the weight for each data point j at step $t+1$ can be defined recursively by.

$$w_j^{(t+1)} = \frac{w_j^{(t)} \exp(-\alpha_t y^j h_t(x^j))}{Z_t}.$$

where Z_t is normalizing constant

$$Z_t = \sum_{j=1}^{N} \exp(-\alpha_t y^j h_t(x^j))$$

$$w_j^{(t+1)} = \frac{1}{Z_t}\left[\frac{w_j^{(t-1)}}{Z_{t-1}} \exp(-\alpha_{t-1} y^j h_{t-1}(x^j))\right] \times \exp(-\alpha_t y^j h_t(x^j)).$$

Similarly expanding the terms till w_j

$$w_j^{t+1} = \frac{w_j}{Z_t Z_{t-1} \cdots Z_1} \exp\left(-y_j \cancel{\sum_{j=1}^{t} \alpha_j h} \left[\alpha_t h_t(x^j) + \alpha_{t-1} h_{t-1}(x^j) + \cdots + \alpha_1 h_1(x^j)\right]\right)$$

we know that from 1.1

$$f(x^j) = \sum_{i=1}^{t} \alpha_i h_i(x^j).$$

$$w_j^{t+1} = \frac{w_j}{\prod_{t=1}^{T} Z_t} \exp\left(-y^j \sum_{i=1}^{t} \alpha_i h_i(x^j)\right).$$

$$w_j^{t+1} = \frac{1}{N \prod_{t=1}^{T} Z_t} \exp\left(-y^j f(x^j)\right).$$ | Initial weights $w_j = \frac{1}{N}$.

For any 'k^{th} update' as the normalization property holds.

$$\sum_{j=1}^{N} w_j^{k} = 1.$$

$$\sum_{j=1}^{N} w_j^{t+1} = 1 = \frac{1}{N \prod_{t=1}^{T} Z_t} \sum_{j=1}^{N} \left(\exp\left(-y^j f(x^j)\right)\right).$$

$$\Rightarrow \prod_{t=1}^{T} Z_t = \frac{1}{N} \sum_{j=1}^{N} \exp\left(-y^j f(x^j)\right).$$

1.3). By combining 1.1 and 1.2, we showed that training error is bounded by $\frac{1}{N} \sum_{j=1}^{N} \exp\left(-y^j f(x^j)\right)$ — ①

$$Z_t = (1-\varepsilon_t)\exp(-\alpha_t) + \varepsilon_t \exp(\alpha_t). \quad \text{— ②}$$

Differentiating w.r.t α_t. and equating $\frac{\partial Z_t}{\partial \alpha_t} = 0$.

$$(1-\varepsilon_t)\exp(-\alpha_t)(-1) + \varepsilon_t \exp(\alpha_t) = 0.$$

$$\Rightarrow \left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) = \left(\exp(\alpha_t)\right)^2 \Rightarrow \left(\exp^{\alpha_t}\right) = \sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}}. \quad \text{— ③}$$

Scanned by CamScanner

Substituting ③ in ②

a)

$$Z_t^{opt} = (1-\varepsilon_t)\sqrt{\frac{\varepsilon_t}{1-\varepsilon_t}} + \varepsilon_t\sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}}$$

$$= 2\sqrt{(1-\varepsilon_t)\varepsilon_t}$$

b) $\boxed{\varepsilon_t = \frac{1}{2} - \gamma_t}$ where $\gamma_t > 0$ implies better than random.

$\gamma_t < 0$ implies worse than random.

Using this in Z_t.

$$Z_t = 2\sqrt{(1-\varepsilon_t)\varepsilon_t} = 2\sqrt{\left(\frac{1}{2}+\gamma_t\right)\left(\frac{1}{2}-\gamma_t\right)}$$

$$= 2\sqrt{\frac{1}{4}-\gamma_t^2}$$

$$= \sqrt{1-4\gamma_t^2}$$

$$\log Z_t = \frac{1}{2}\log(1-4\gamma_t^2).$$

we know that $\log(1-x) \leq -x$ for $0 \leq x < 1$.

Here in our case, we can say. $\Rightarrow$ $0 \leq 4\gamma_t^2 < 1$.

$$\log Z_t = \frac{1}{2}\log(1-4\gamma_t^2) \leq \frac{1}{2}\times(-4\gamma_t^2)$$

$$\leq -2\gamma_t^2.$$

$$Z_t \leq \exp(-2\gamma_t^2).$$

c) If each classifier is better than random;

i.e. $\gamma_t \geq \gamma \quad \forall t$ and $\gamma > 0$.

$$\sum_{t=1}^{T} \gamma_t^2 \geq \sum_{t=1}^{T} \gamma^2$$

$$\geq T\gamma^2.$$

$$\varepsilon_{training} \leq \prod_{t=1}^{T} Z_t \leq \exp\left(-2 \sum_{t=1}^{T} \gamma_t^2\right).$$

$$\leq \exp\left(-2T\gamma^2\right).$$

$$\varepsilon_{training} \leq \exp\left(-2T\gamma^2\right).$$

2.1)
I have raised a piazza question for sum of the squares.

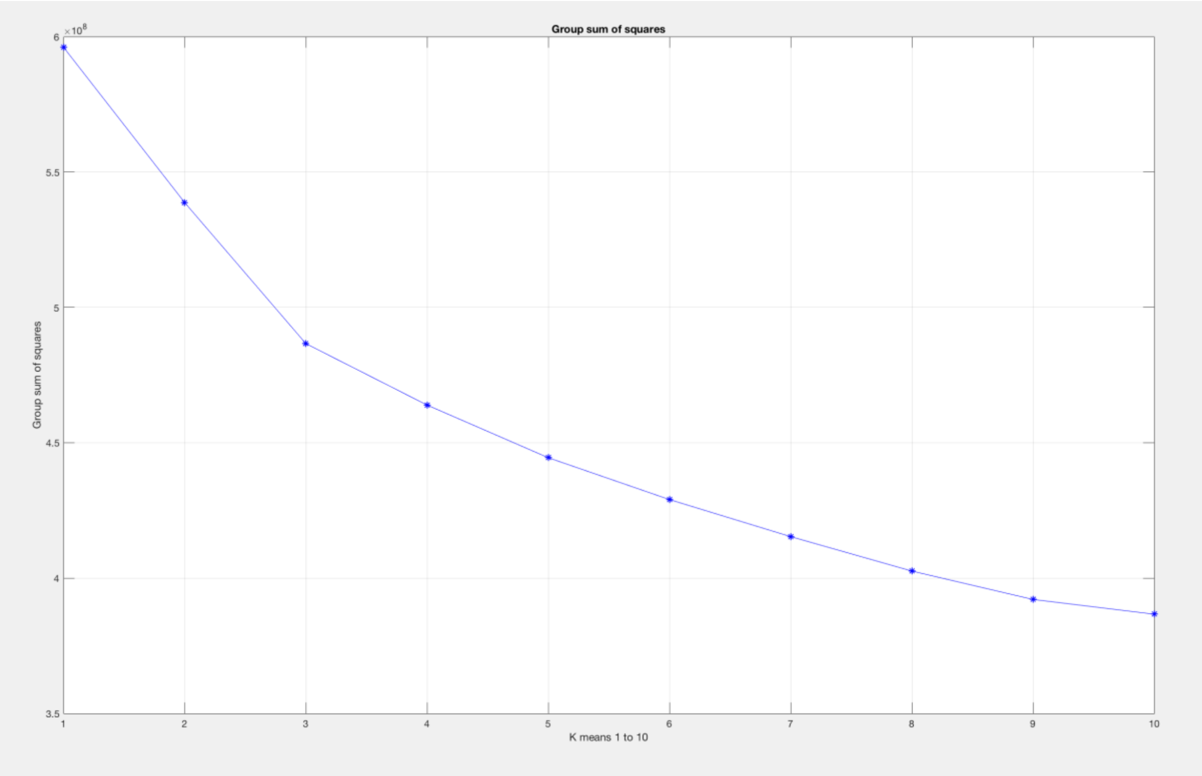
https://piazza.com/class/jltkcjd9q2g34x?cid=209

As per the suggestion, I'm squaring the Euclidean distance and submitting the result accordingly.

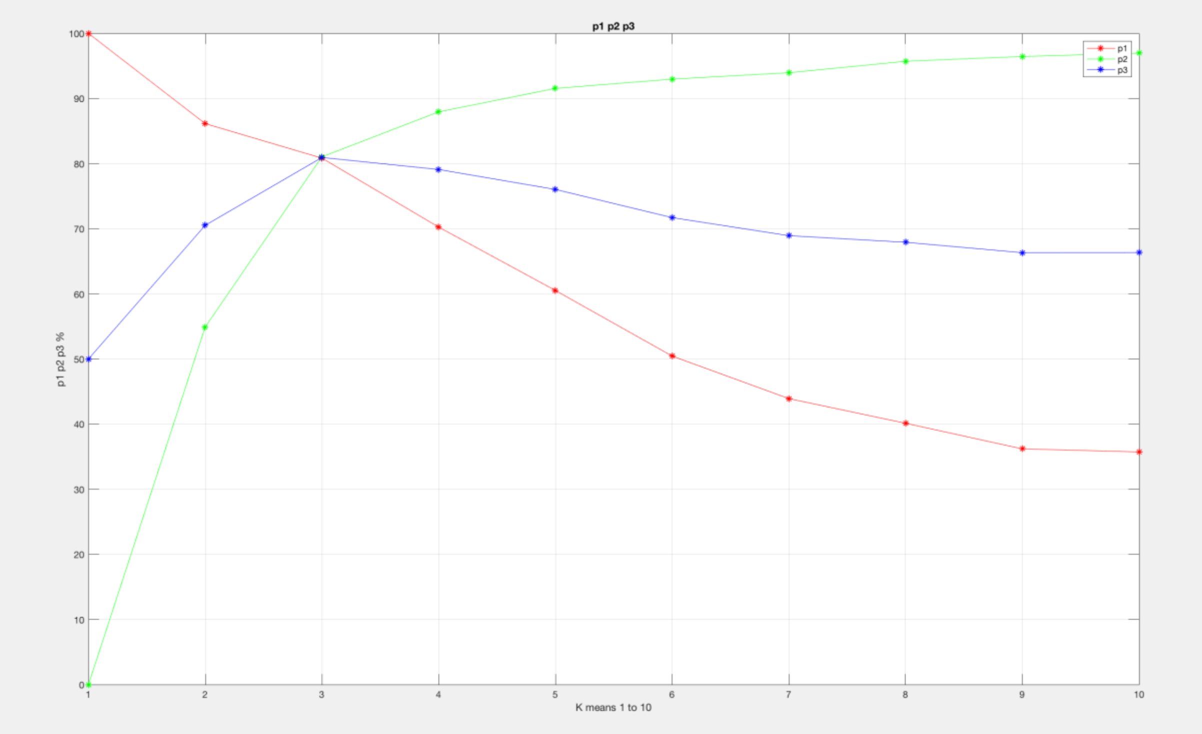
Please use the "q21_final.m" to generate the results.
Please find the results below:
Clusters 2, Iteration break 20, SS(k)=5.364771e+08, p1=79.82, p2=54.81 & p3 67.31
Clusters 4, Iteration break 11, SS(k)=4.611109e+08, p1=67.88, p2=86.83 & p3 77.36
Clusters 6, Iteration break 8, SS(k)=4.313492e+08, p1=55.18, p2=94.43 & p3 74.81

2.2)
I need to get the number of iterations to reach the optimal condition.
8 iterations were needed to optimize the group to 6 clusters.

2.3)
I ran for 10 iterations and then took the average of the 10 clusters.
Individual cluster values have been stored in the file result_q23.csv
Please find the plots below:



2.4)



3.1)
Please find the 5 fold cross validation accuracy below with the default values of C and gamma:

Cross Validation Accuracy = 15.6443%

Please use the following files to generate the results.
check.m KNN.m HW5_BoW1.m to compute the features.
q341_2.m to get the results.

3.2)
Please use "q341_2.m" file for generating the results after loading the train data for 1000 clusters.
Please find the table for the cross validation by varying C and gamma:

	C value					
Gamma value	1	10	100	1000	10000	100000
1	22.2285	57.9629	74.9015	85.5937	87.9572	87.9572
10	57.794	75.5205	86.269	88.4637	88.4637	88.4637
100	75.5768	86.7192	88.2386	88.2386	88.2386	88.2386
1000	78.2217	80.3602	80.3602	80.3602	80.3602	80.3602
10000	26.6179	31.0636	31.0636	31.0636	31.0636	31.0636
100000	15.6443	15.6443	15.6443	15.6443	15.6443	15.6443

3.3)

I have used epsilon =10^-9 in the implementation of chisquare kernel.

	C value					
Gamma value	1	10	100	1000	10000	100000
1	87.4508	93.6972	93.7535	93.7535	93.7535	93.7535
10	68.8801	88.1823	93.5847	**93.8661**	**93.8661**	93.8661
100	15.7006	69.3303	88.1823	93.641	93.7535	93.7535
1000	15.6443	15.7006	69.3303	88.1823	93.641	93.6972
10000	15.6443	15.6443	15.7006	69.3303	88.1261	93.5847
100000	15.6443	15.6443	15.6443	15.7006	69.3303	88.1823

3.4)

I have tried 3 things.

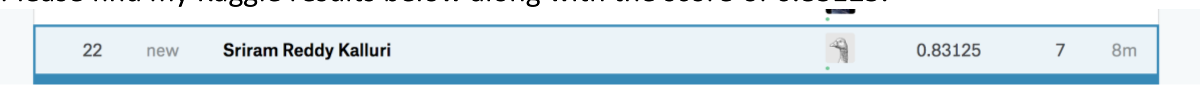
1. By decreasing the features,i.e decreasing the Clusters in K means, it didn't provide any improvement.
2. During the calculation of BowCs , I have increased the samplesize to 1 million from 0.1 million for better robustness.
3. I have changed the HOG features which are generated by using vlfeat.

Please use the below files to generate the results:

check3.m and check5.m

The best values of C and gamma are 1000,10 respectively and the 5 fold cross validation accuracy is 93.8661.

Please find my Kaggle results below along with the score of 0.83125.



22	new	**Sriram Reddy Kalluri**	0.83125	7	8m