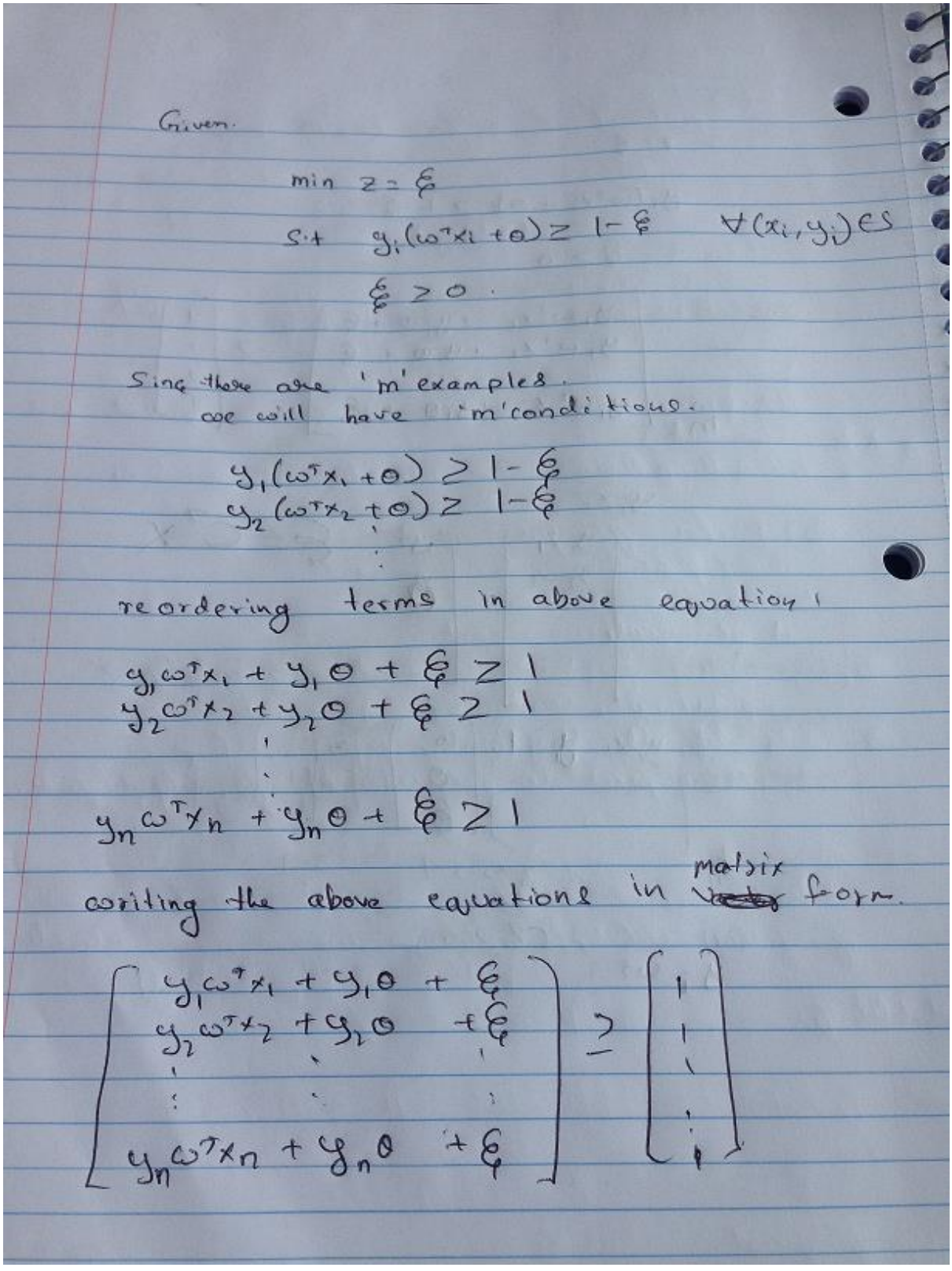
Assignment 3

Submitted by: Seshasai

Problem 1: Classification via Linear Programming



Given.

$$\min \; z = \xi$$

$$s.t \quad y_i(\omega^T x_i + \theta) \geq 1 - \xi \qquad \forall (x_i, y_i) \in S$$

$$\xi \geq 0.$$

Since there are 'm' examples. one will have 'm' conditions.

$$y_1(\omega^T x_1 + \theta) \geq 1 - \xi$$
$$y_2(\omega^T x_2 + \theta) \geq 1 - \xi$$
$$\vdots$$

reordering terms in above equations:

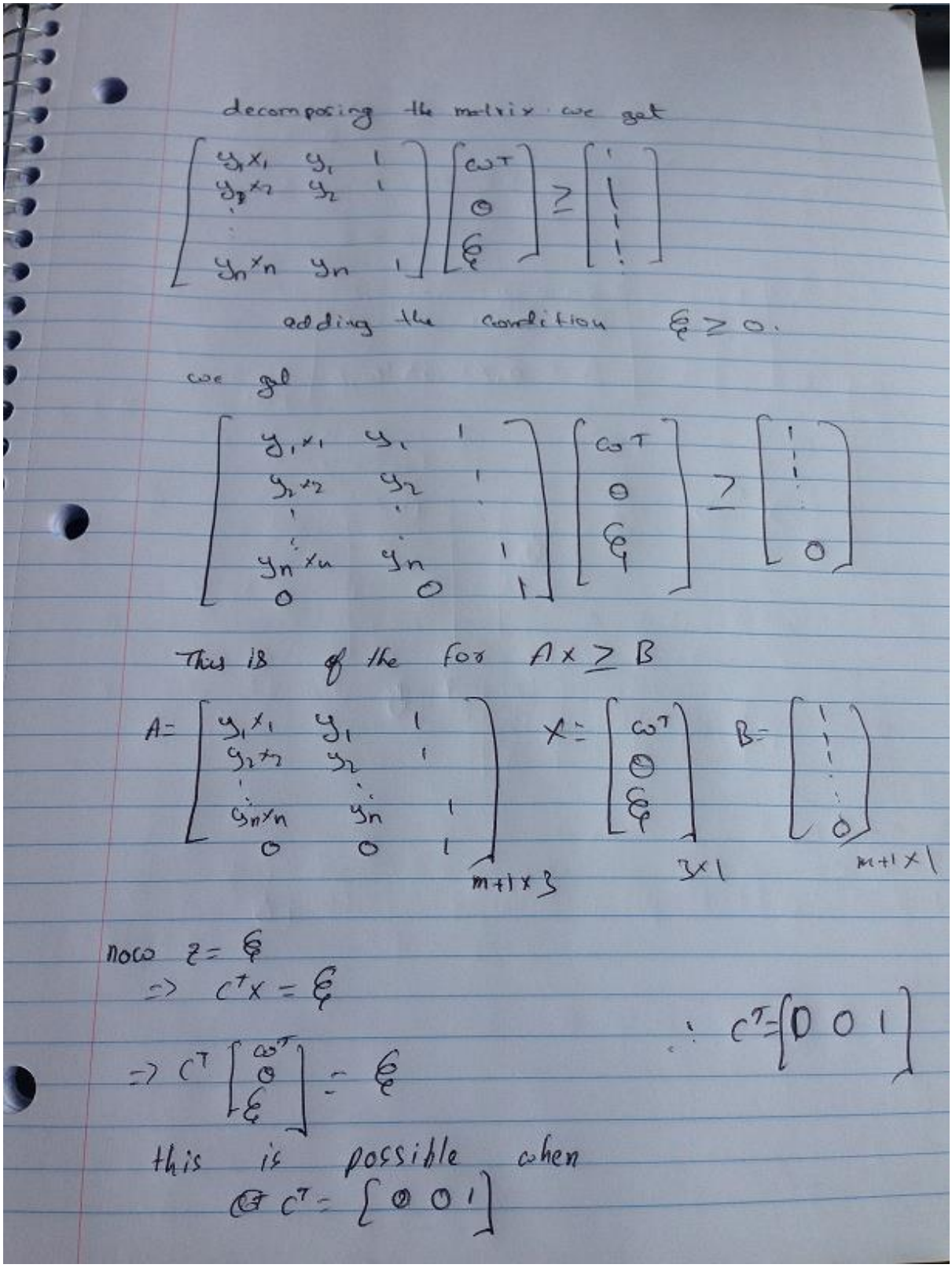
$$y_1 \omega^T x_1 + y_1 \theta + \xi \geq 1$$
$$y_2 \omega^T x_2 + y_2 \theta + \xi \geq 1$$
$$\vdots$$
$$y_n \omega^T x_n + y_n \theta + \xi \geq 1$$

writing the above equations in matrix form.

$$\begin{bmatrix} y_1 \omega^T x_1 + y_1 \theta + \xi \\ y_2 \omega^T x_2 + y_2 \theta + \xi \\ \vdots \\ y_n \omega^T x_n + y_n \theta + \xi \end{bmatrix} \geq \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

Assignment 3

Submitted by: Seshasai



Assignment 3

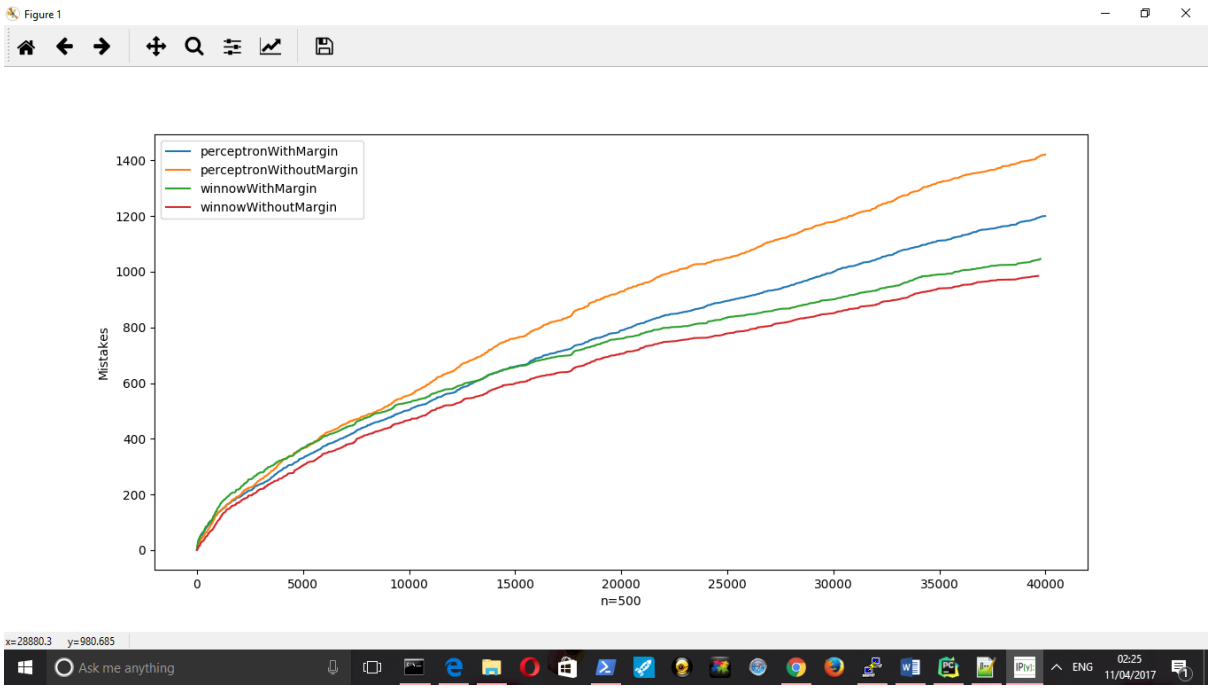
Submitted by: Seshasai

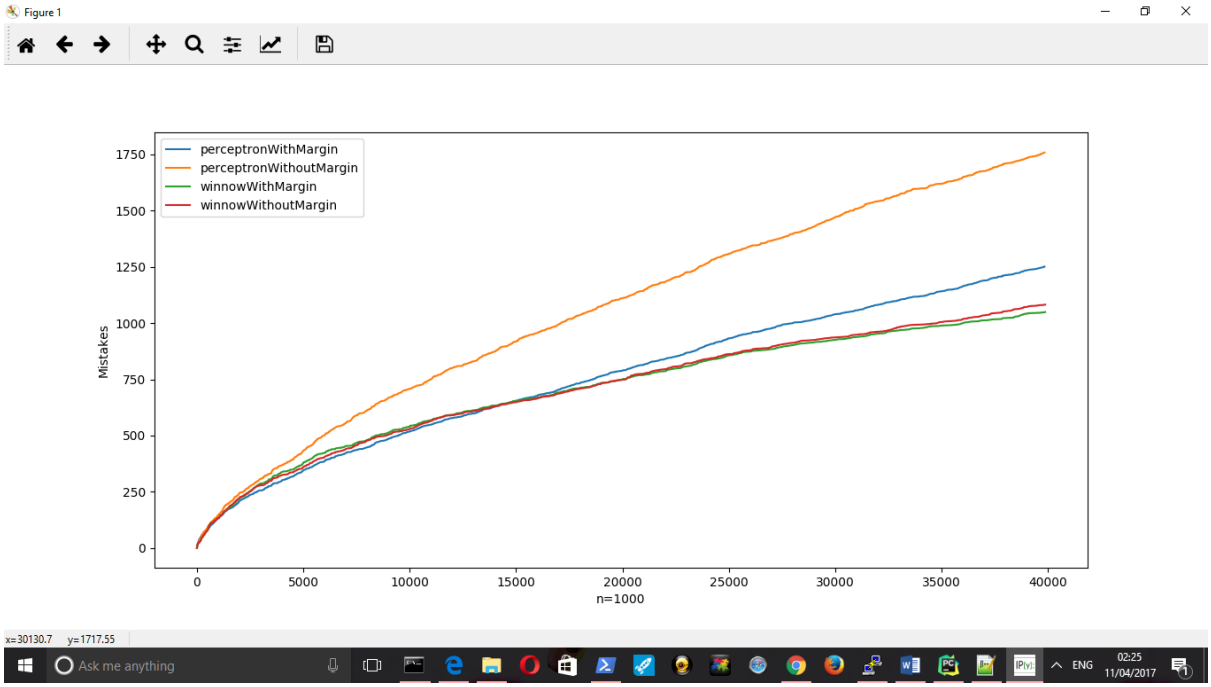
Problem 2: Online Learning

A:

Table 1:

Algorithm	N=500			N=1000		
	Params	Acc(D2)	M	Params	Acc(D2)	M
Perceptron(γ = 0)	0,1	97.18	1421	0,1	97.12	1758
Perceptron(γ =>0)	1,0.001	98.3	1200	1,0.001	97.44	1251
Winnow(γ = 0)	0,1.1	98.12	1061	0,1.1	97.92	1082
Winnow(γ > 0)	.04,1.1	98.34	1061	2,1.1	98.22	1049





Assignment 3

Submitted by: Seshasai

Findings:

1. From the table and from graph we can conclude that Winnow learns a linear classification function much faster than a perceptron.
2. Number of mistakes made by Winnow are comparatively much lower than that of perceptron.
3. Though both the algorithms are online learning algorithms one glaring difference is the way weights are updated. Since we are trying to learn conjunction, learning rule of winnow is suited for this.
4. From the table we can see that Accuracy on D2 is same for both winnow and perceptron this is mainly because for D2 we are just learning on 5000 samples and then predicting. From the graph we can see that for 5000 samples both winnow and perceptron make almost same number of mistakes.
5. So Accuracy on D2 is kind of misleading, primary purpose of D2 was to fix the learning rate and margin. Also from the table we can infer that learning rate for which winnow performs best is largest in the set of learning given.
6. From the graphs it is pretty evident that as winnow sees more examples the number of mistakes it makes is decreasing at a faster rate compared to perceptron.
7. From the graphs it is also clear that for very high dimensional data winnow performs much better than perceptron

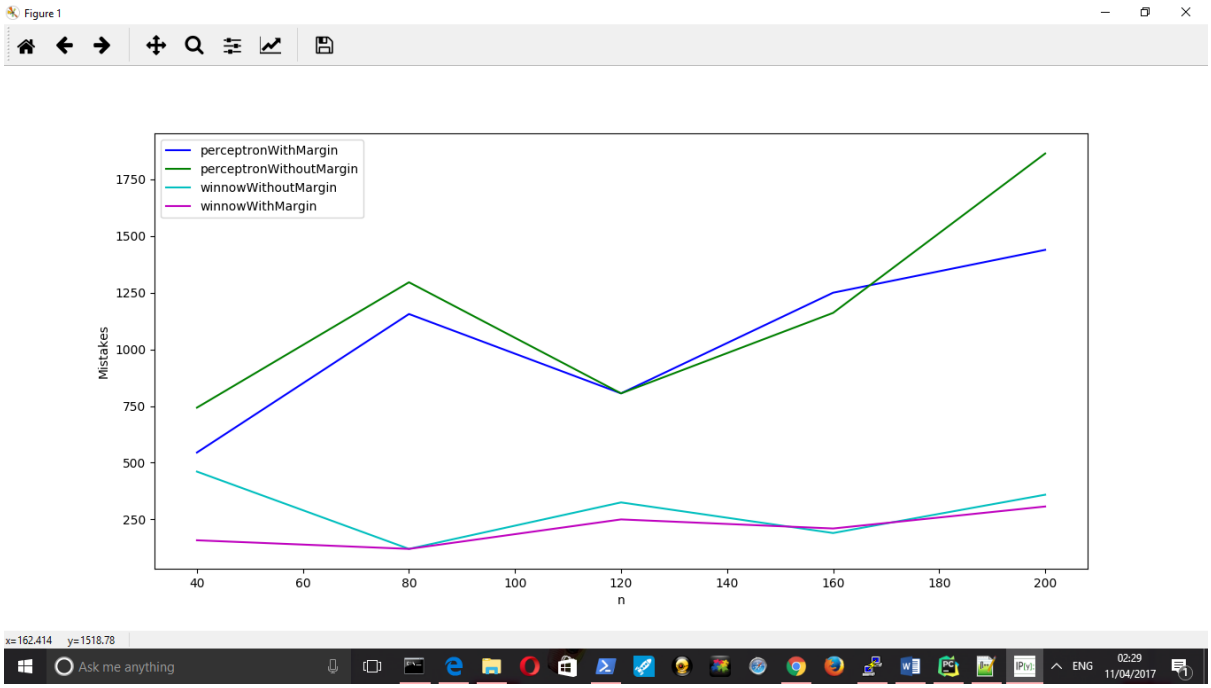
B:

Table 2:

Algorithm		Perceptron	Perceptron(γ)	Winnow	Winnow(γ)
N=40	Params	0,1	1,.25	0,1.1	2,1.01
	Acc(D2)	99.58	99.84	100	100
	M	1314	1194	256	666
N=80	Params	0,1	1,0.25	0,1.1	0.4,1.1
	Acc(D2)	95.82	97.88	99.98	100
	M	1785	1718	422	432
N=120	Params	0,1	1,1.5	0,1.1	0.3,1.1
	Acc(D2)	97.72	97.72	100	100
	M	1897	1897	528	508
N=160	Params	0,1	1,0.03	0,1.1	2,1.1
	Acc(D2)	96.18	97.08	99.92	100
	M	1913	1666	568	544
N=200	Params	0,1	1,0.03	0,1.1	2,1.1
	Acc(D2)	96.68	96.86	99.92	100
	M	1943	1696	622	558

Assignment 3

Submitted by: Seshasai



Findings:

1. From the graph we can infer that as the dimensions of the data increases winnow clearly beats perceptron.
2. The number of mistakes made by winnow is almost less than half mistakes made by perceptron.
3. Including margin during training does reduce the number of mistakes for perceptron but winnow seems stable.
4. The current hypothesis survives(S) that I calculated is 159. S from winnow is more than 1000 for perceptron it is 159.
5. The learning rate for which Winnow performs best from the given set of learning rates is almost always 1.1 which is the highest value. This shows that as learning rate increases winnow's performance increases
6. For data with less dimensions Winnow seems to learn the function in less than 5000 examples of the data.
7. As the number of dimensions increases both the algorithms make more mistakes this means they require more data to learn from.

C:

Table 3:

Algorithm	M=100			M=500			M=1000		
	Params	Acc(D2)	Acc(Test)	Params	Acc(D2)	Acc(Test)	Params	Acc(D2)	Acc(Test)
Perceptron	0,1	81.8	71.25	0,1	82.54	51.5	0,1	84.98	91.13
Perceptron(γ)	1,0.001	87.16	72.29	1,0.001	86.1	51.18	1,0.001	89.22	95.34
Winnow	0,1.1	84.22	67.93	0,1.001	88.04	76.09	0,1.1	86.56	90.71
Winnow(γ)	2,1.1	85.54	79.91	2,1.0001	88.14	76.17	0.04,1.1	86.56	90.71

Assignment 3

Submitted by: Seshasai

Findings:

1. For this experiment the models were trained on noisy and on very high dimensional data(n=1000). The impact of noise is quite evident in the accuracy of algorithms.
2. Winnow seems to be prone to noises in data as compared to perceptron.
3. Since we are using very high dimensional data with may unnecessary attributes winnow and perceptron are trying the find the conjugation function that justifies the data. So it seems like for a very high dimensional data with few important attributes, either we need to provide examples that are positive for the important attributes so that the algorithms learn faster or provide lot of data.
4. In our case we did provide good number of examples but the noise in the data seems to prevent the algorithms from converging. As a result when the algorithms are tested with clean data the low accuracy indicates that the conjugate functions learned by the algorithms is not accurate. The conjugate function learned by the algorithm has accommodated the noise.
5. It maybe that the hyper parameters might be bottleneck. Maybe if we tried to learn with a larger set of hyper parameters then probably the algorithms would have converged.