IST 597: Assignment 1

Shaurya Rohatgi

October 2, 2017

Problem 1

Write a few sentences describing what you learned from the training/model fitting process. Things to discuss: What happens when you change the step-size ? How many epochs did you need to converge to a reasonable solution (for any given step size)?

Solution

The model could have converged earlier by reducing the value of eps. But as it didn't take much time to compute I decided to stretch the epochs until I saw no difference in loss. I print epochs in multiples of 20 for making the console output less verbose.

The results that I got with the following parameter setting

$\alpha = 0.02$
$eps = 0.00001$
$n\_epoch = 960$

Step Size α	Convergence *eps*	epochs	Final loss
0.0001	0.00001	9980	5.48098618
0.001	0.00001	9980	4.51633859
0.015	0.00001	1220	4.47890568
0.02	0.00001	960	4.47836221
0.01	0.00001	1720	4.47986198
0.03	0.00001	1	32.07273388

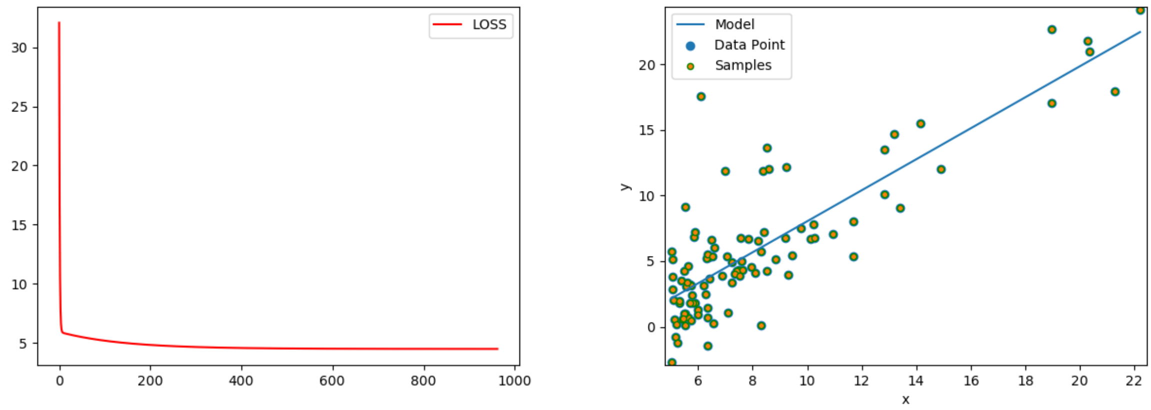
Figure 1: Problem 1 - Varying the Step Size

I think these are the best parameter settings because -

- it converges to the minimum loss.
- it takes the minimum number of epochs to converge.

Problem 2

a) With respect to the feature mapping, what is a potential problem that the scheme we have designed above might create? What is one solution to fixing any potential problems created by using this scheme (and what other problems might that solution induce)?



(a) Loss vs Epochs

(b) Learned Model with data points

Figure 2: Problem 1

b) What do you observe as the capacity of the model is increased? Why does this happen?

c) Refit your 15th order polynomial regressor to the same data but this time vary the meta-parameter using the specific values $\{0.001, 0.01, 0.1, 1.0\}$ and produce a corresponding plot for each trial run of your model. What do you observe as you increase the value of β ? How does this interact with the general model fitting process (such as the step size α and number of epochs needed to reach convergence)?

d) How many steps it then took with this early halting scheme to reach convergence. What might be a problem with a convergence check that compares the current cost with the previous cost (i.e., looks at the deltas between costs at time t and $t-1$), especially for a more complicated model? How can we fix this?

Solution

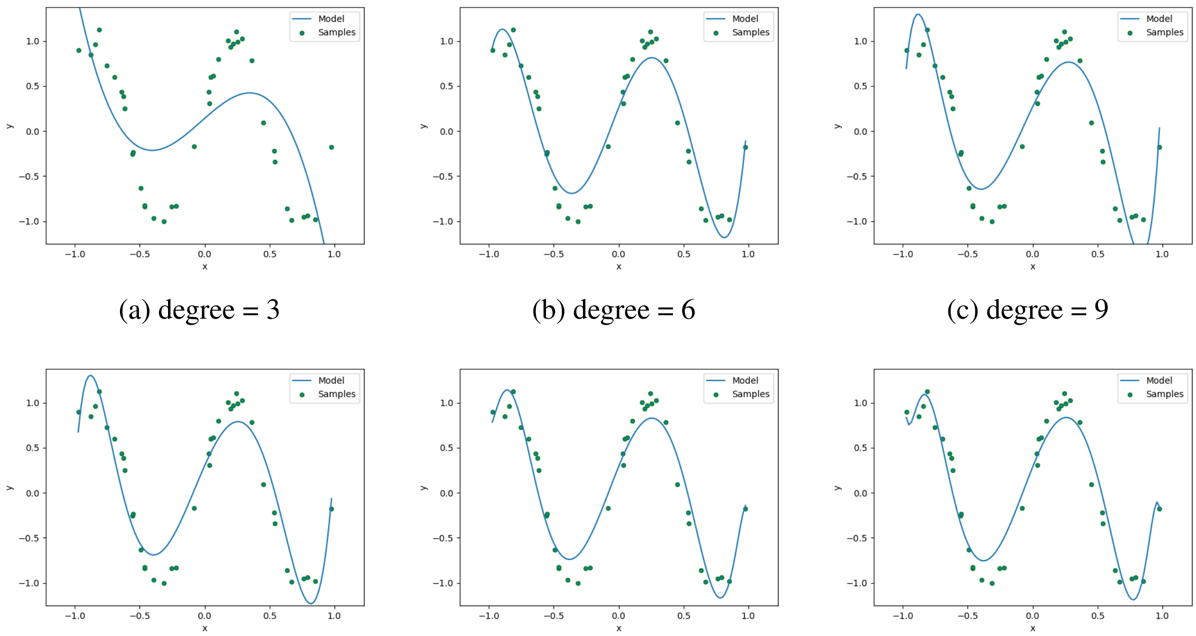
After plotting lots and lots of models and loss functions the following are the best set of parameters which I have finalised for this solution -

$degree = 15$
$beta = 0.0001$
$alpha = 1.6$ # yes, I kept increasing it until something bad happened to the model
$eps = 0.00001$
$n\_epoch = 10000$ # early stops at usually 2216$^{\text{th}}$ epoch

I know α should ideally be in the range $[0, 1]$ but I was playing around with it and found my model to converge faster with a high step size! There may be other parameter settings which are also acceptable but I will justify in the following points what let me to these.

a) Increasing the degree of the polynomial may lead to overfitting. Regularization is the potential fix to this problem. Potential problem induced here is that we now have an extra meta-parameter to tune or learn.

b) As the capacity of the model is increased there are more convexes and concaves in the model. By increasing the capacity we are increasing the flexibility of the model to curve and fit the data. See Fig. 3 for more details.



(a) degree = 3 (b) degree = 6 (c) degree = 9

(d) degree = 12 (e) degree = 15 (f) degree = 20

Figure 3: Problem 2b - Increasing Capacity of the model while keeping other parameters fixed

c) Increasing β -

- reduces the epochs for "convergence". See Fig 5.
- the model starting underfitting. See Fig. 4.

Changing α keeping rest parameters fixed. See Fig. 6

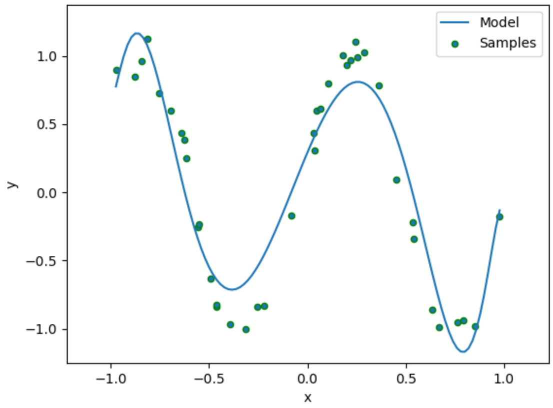
- I think the least number of epochs and the best fit to the data comes when $\alpha = 1.6$. This is when other parameters are kept constant, defined at the start of the problem.

d) For the current convergence criteria used it took 2216 *epochs* to converge ($eps = 0.0001$). The problem with this convergence check is that -

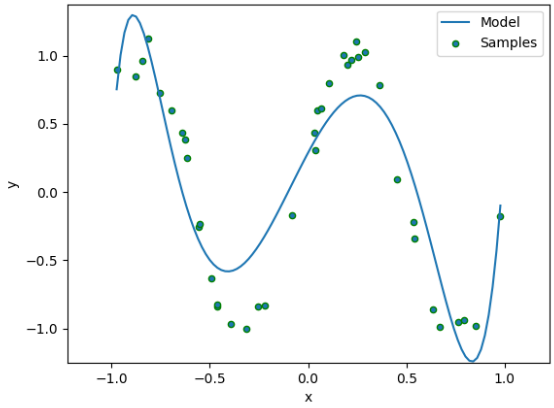
- the error may increase after an iteration
- if the loss is stuck at a saddle point in the local minima and we stop there. Maybe the loss is trying to come out of the saddle point, the loss might increase and then converge to an even lower minima, which may be the global minima.

There is no golden rule to fix this. But we can try the following -

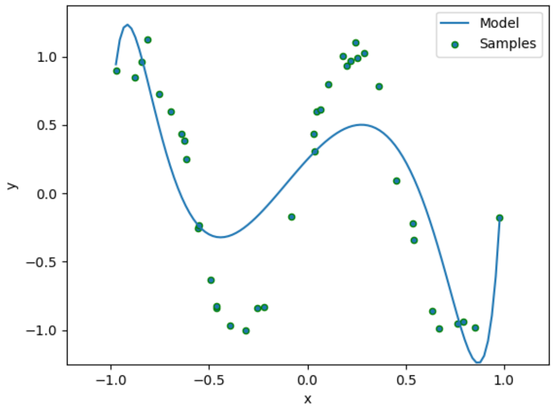
- plot Validation loss with the training loss. It will give a good idea if the model has a high bias or high variance and when we need to stop.
- run for a long as you can "afford" to run. Afford accounts for 2 things, time and computation costs.



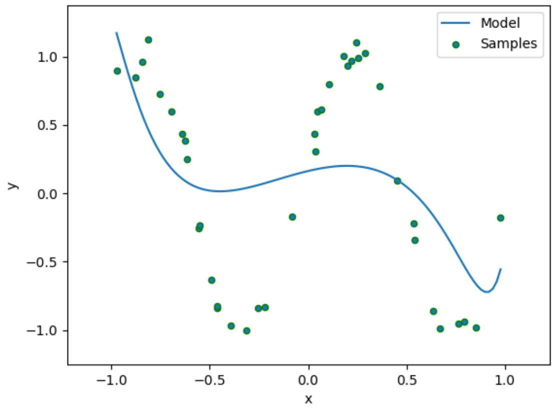
(a) beta = 0.001



(b) beta = 0.01



(c) beta = 0.1



(d) beta = 1

Figure 4: Problem 2c - Different Values of Beta while keeping other parameters fixed

Problem 3

a) Keeping β fixed at zero (which means no regularization), tuning only the number of epochs and your learning rate α.

b) Re-fit your logistic regression but this time with $\beta = \{0.1, 1, 10, 100\}$, again, tuning the number of epochs and the learning rate. Copy each of the resultant contour plots to your answer sheet and report your classification error for each scenario. Comment (in the answer sheet) how the regularization changed your models accuracy as well as the learned decision boundary. Why might regularizing our model be a good idea if it changes what appears to be such a well-fit decision boundary?

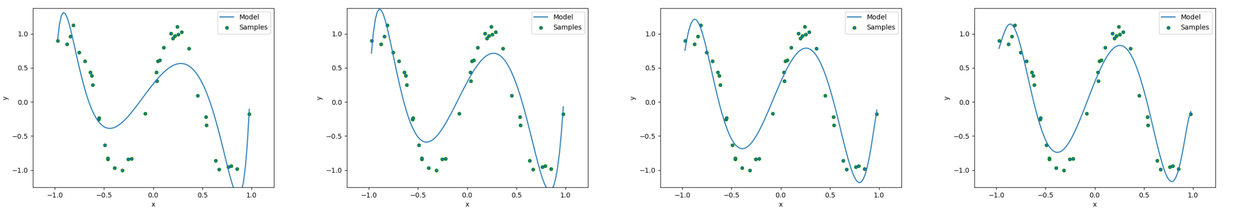
Solution

a) Keeping β fixed and changing α. We can see when $\alpha = 1.5$ the model is not acceptable. See Fig. 7 α.

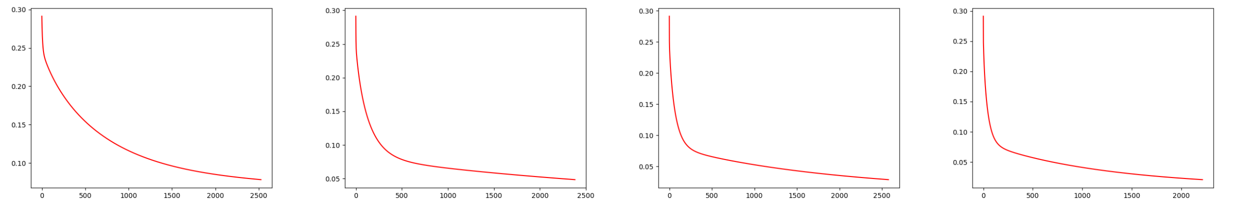
b) Final parameter settings for this problem

Figure 5: Problem 2c - Varying β

Step Size α	Degree	Beta β	eps	epochs	Final loss
1.6	15	0.001	0.00001	2026	0.02643295
1.6	15	0.01	0.00001	1099	0.05640311
1.6	15	0.1	0.00001	240	0.11667098
1.6	15	1	0.00001	55	0.20485019



(a) alpha = 0.1 (b) alpha = 0.5 (c) alpha = 1.0 (d) alpha = 1.6



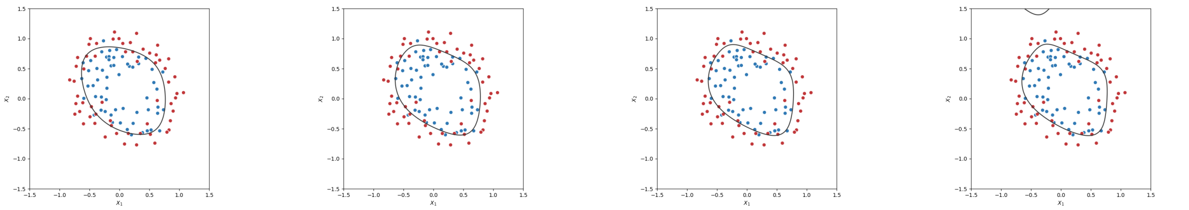
(e) alpha = 0.1 (f) alpha = 0.5 (g) alpha = 1.0 (h) alpha = 1.6

Figure 6: Problem 2c - Model and Loss with different Values of α while keeping other parameters fixed

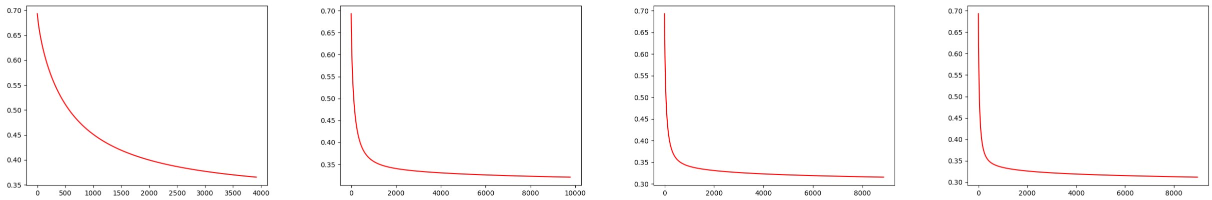
$$degree = 6$$
$$beta = 1$$
$$alpha = 1.5$$
$$n_epoch = 100000$$
$$eps = 0.00001$$

The following are my comments and observations for this problem

- I tried for different values of beta. See Fig. 9. Increasing β increased the classification error percentage.
- Here are the other values of classification error percentage and the final loss for these values - Fig. 8
- The model seems to shift from the ideal decision boundary for the dataset as the value of β is increased.
- I have also included the loss functions for these β values. See Fig. 10
- As we can see the model has fit the the decision boundary well enough but still regularization is a good idea because the model might have slightly overfit and regularization will help to balance out the bias and variance.



(a) alpha = 0.1 (b) alpha = 0.5 (c) alpha = 1.0 (d) alpha = 1.5

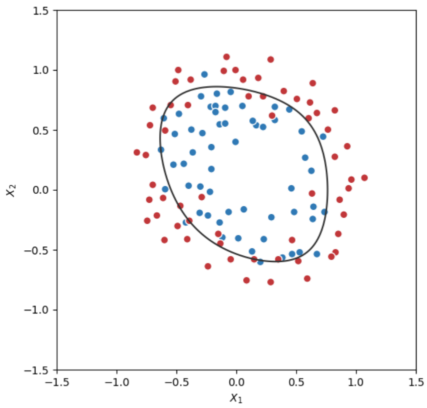


(e) alpha = 0.1 (f) alpha = 0.5 (g) alpha = 1.0 (h) alpha = 1.5

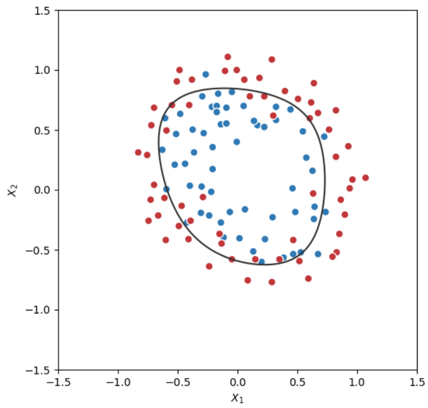
Figure 7: Problem 3a - Model and Loss with different Values of α while keeping $\beta = 0$

Step Size α	Degree	Beta β	eps	epochs	Error%	Final loss
1.5	6	0.1	0.00001	485	16.95%	0.396444173
1.5	6	1	0.00001	140	16.95%	0.529263587
1.5	6	10	0.00001	24	25.42%	0.64825241
1.5	6	100	0.00001	5	36.44%	0.64825241

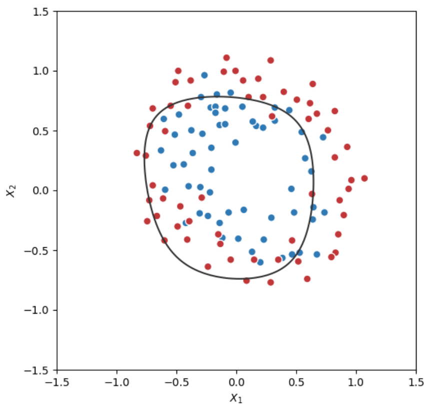
Figure 8: Problem 3b - Varying β



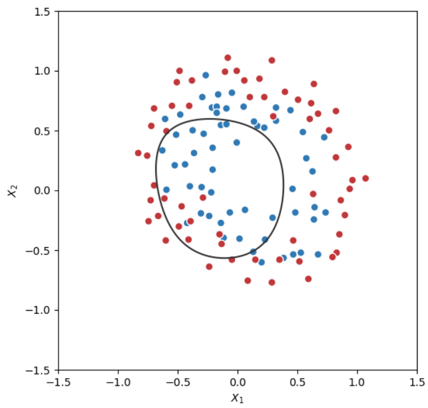
(a) beta = 0.1



(b) beta = 1

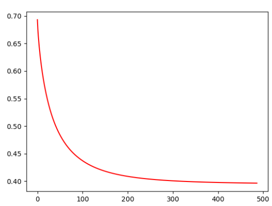


(c) beta = 10

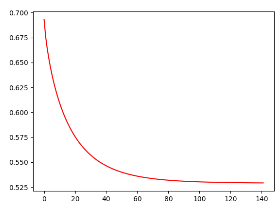


(d) beta = 100

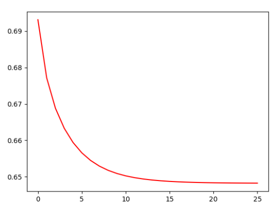
Figure 9: Problem 3b - Different Values of β while keeping other parameters fixed



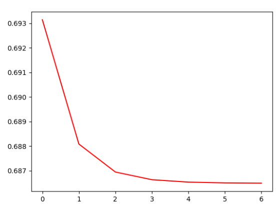
(a) beta = 0.1



(b) beta = 1



(c) beta = 10



(d) beta = 100

Figure 10: Problem 3b - Loss vs Epochs for different values of β