University of Central Florida

CAP6676

Project 2 Report

Submitted To:
Guo-Jun Qi
Asst. Professor
Computer Science Department

Submitted By :
Pooya Abolghasemi
Fall 2015

Contents

The dataset I chose the Statlog(Heart) dataset. This database contains 13 attributes (which have been extracted from a larger set of 75). It contains different types of attributes(Real, Binary, Nominal and Ordered). The 13 features are:

1. age
2. sex
3. chest pain type (4 values)
4. resting blood pressure
5. serum cholesterol in mg/dl
6. fasting blood sugar more than 120 mg/dl
7. resting electrocardiograph results (values 0,1,2)
8. maximum heart rate achieved
9. exercise induced angina
10. oldpeak = ST depression induced by exercise relative to rest
11. the slope of the peak exercise ST segment
12. number of major vessels (0-3) colored by flourosopy
13. thal: 3 = normal; 6 = fixed defect; 7 = reversible defect

The dataset contains 270 observations and no missing values. The data types as recorded in the Machine Learning Repository is multivariate. The default task is classification and attribute types are categorical and real.
For applying logistic regression we need to convert the categorical attributes to a set of boolean attributes. In this dataset we have 3 categorical attribute (number 4, 7 and 13). After converting them to boolean we will end up 20 attributes.

1 Logistic Regression

For implementing gradient ascend logistic regression we need the conditional log likelihood function

$$l(W) = \sum_l Y^l(w_0 + \sum_i^n w_i X_i^l) - ln(1 + exp(w_0 + \sum_i^n w_i X_i^l)) \tag{1}$$

and we need to calculate the derivative of log likelihood function for gradient ascend algorithm.

$$\frac{\partial l(W)}{\partial w_i} = \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, W)) \tag{2}$$

Based on these equation we are going to update the logistic regression weights based on these equation:

$$w_i \leftarrow w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, W) \tag{3}$$

In my setting for running batch gradient ascent I repeat updating the weights until the log likelihood difference between two successive iterations goes below a certain threshold. However, for running stochastic gradient ascent I used the maximum value for the number of iteration. Because log likelihood difference between two successive iteration in stochastic setting might get very small in early iterations because we are not considering all the data. I am going go through the details in the following chapters.

2 Logistic Regression - Batch Gradient Ascent

For this part I considered 65% of the dataset as training set and the remaining as test set. The learning rate was set to 0.015. Figure1 shows the ascending of log likelihood in successive iterations for batch gradient ascending. Figure2 shows the train and test error for batch gradient ascent. The error is simply the difference between the model prediction and the real label for each of the samples divided by number of samples. I could not see any over-fitting in the process even when I let the algorithm to perform 1000 iterations. The only time that I saw over-fitting was when I decrease training data dramatically. I decreased the training data to 10% and you can see the train and test error in figure3. In this case obviously lack of training data caused the over-fitting. I needed this case to see whether my regularization works or not, since without over-fitting I could not see any benefit when I added the regularization term to the likelihood function.

Figure 1: log likelihood vs iteration

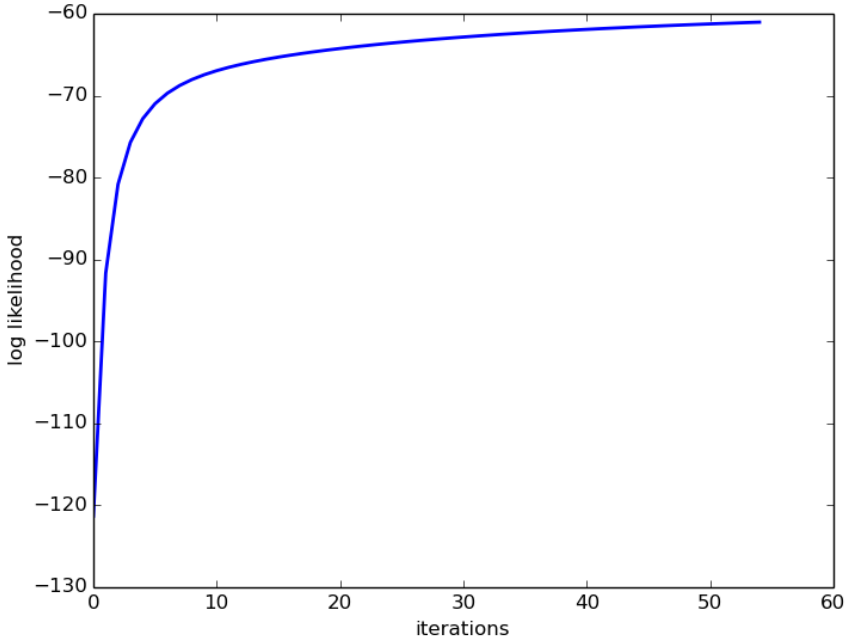


Figure 2: train error(Blue) and test error(red) for batch gradient ascent

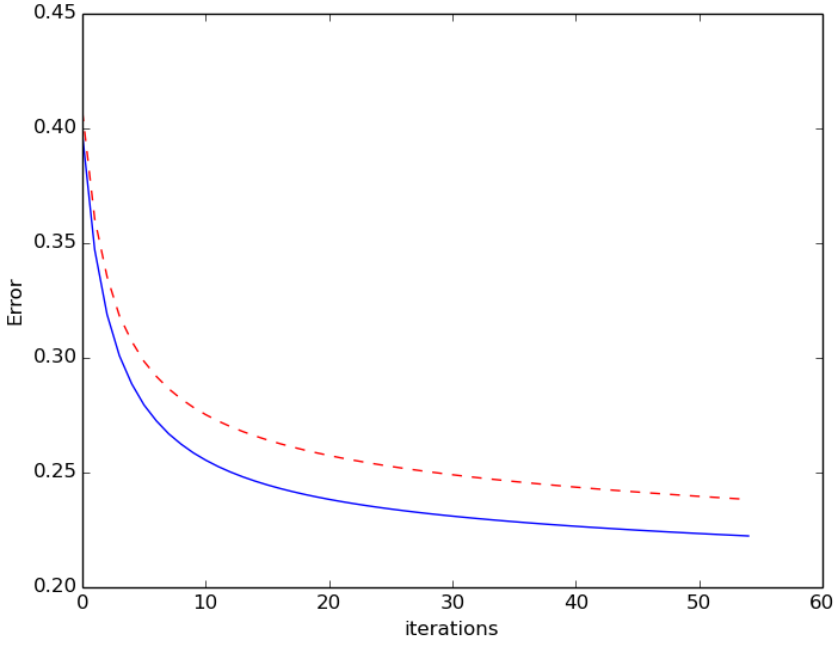
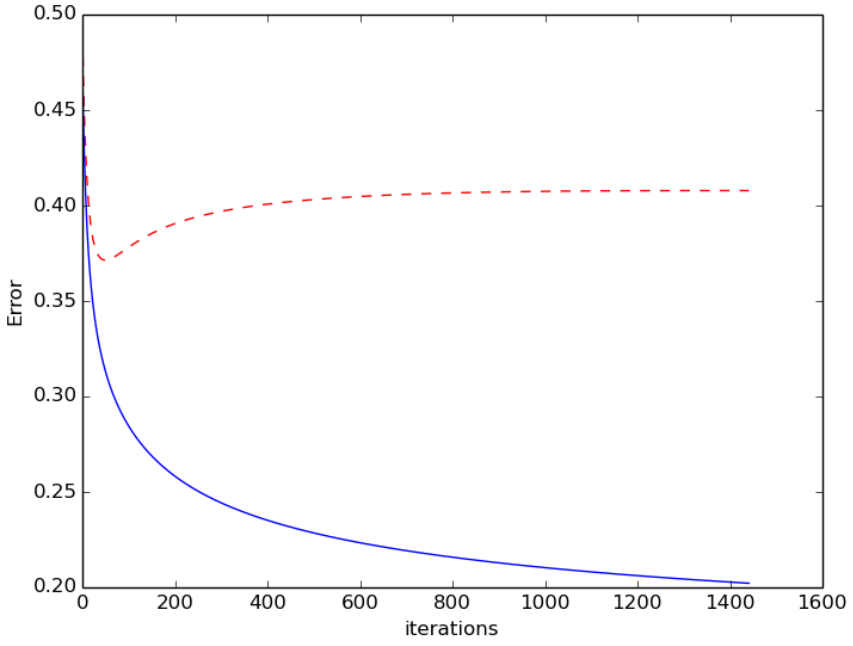


Figure 3: train error(Blue) and test error(red) for batch gradient ascent



3 Logistic Regression - Stochastic Gradient Ascent

For stochastic gradient ascending we need to choose a proper size of randomly chosen subset. To do so I test the process with different subset sizes. In figure4 you can see train and test error with different values for the subset size. After observing results for different subset sizes between 1 to 100 I set the subset size to 40. Since subset sizes bigger than 40 doesn't improve the performance very much. As it is shown in the graphs bigger subset size gives us better performance but as a result the process take more time especially when subset size is very big. When the subset size is not big enough and there is lots of outliers or inaccurate measurements for the attributes in the dataset, stochastic gradient ascent might take a long time to converge. However, stochastic gradient ascent is useful for the datasets with lots of samples, since performing batch gradient ascent will take lots of time. In figure 5 you can see the log likelihood of stochastic gradient ascent with 40 subset size.

Figure 4: train error(Blue) and test error(red) for stochastic gradient ascent. Subset size grow from left to right, top to bottom

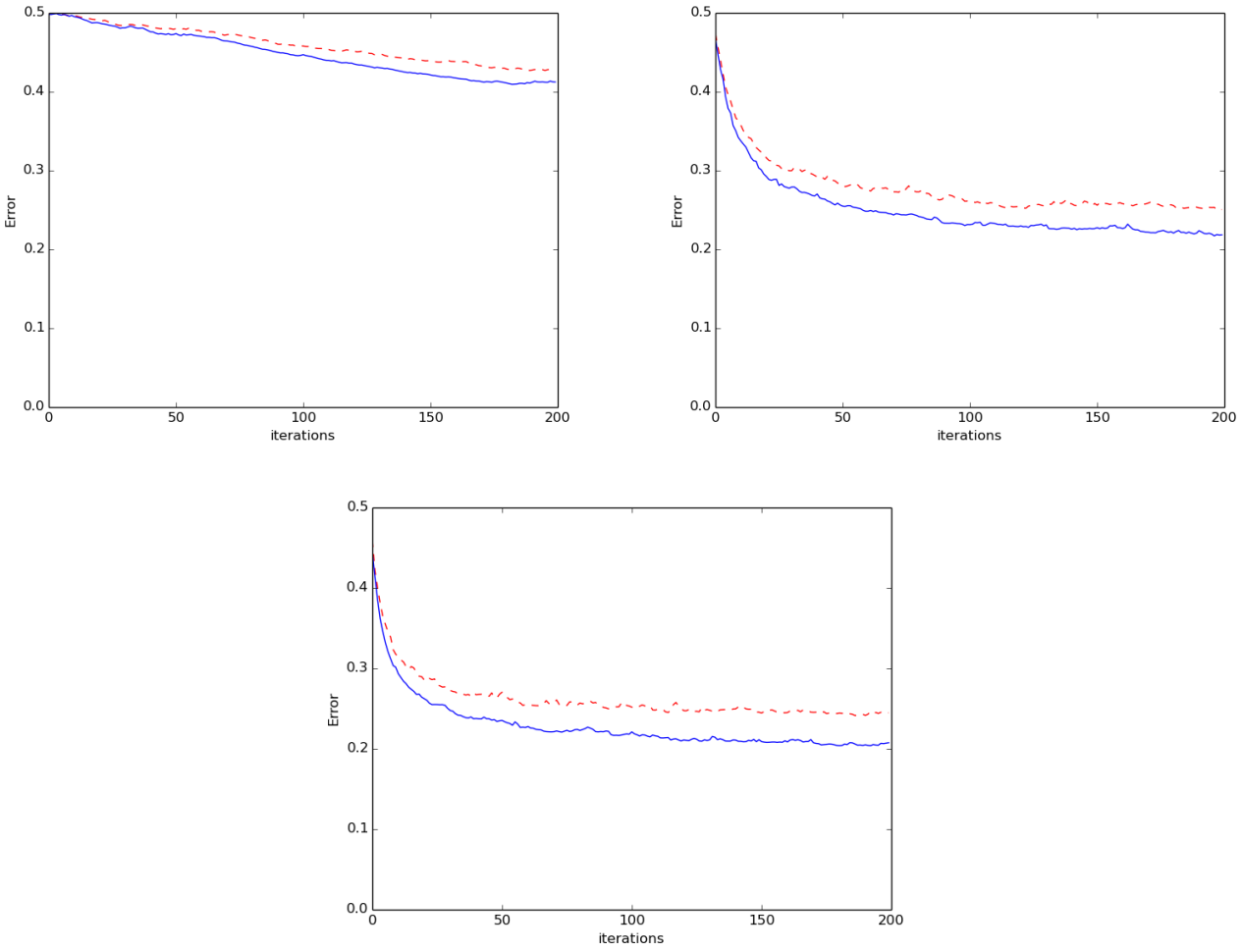
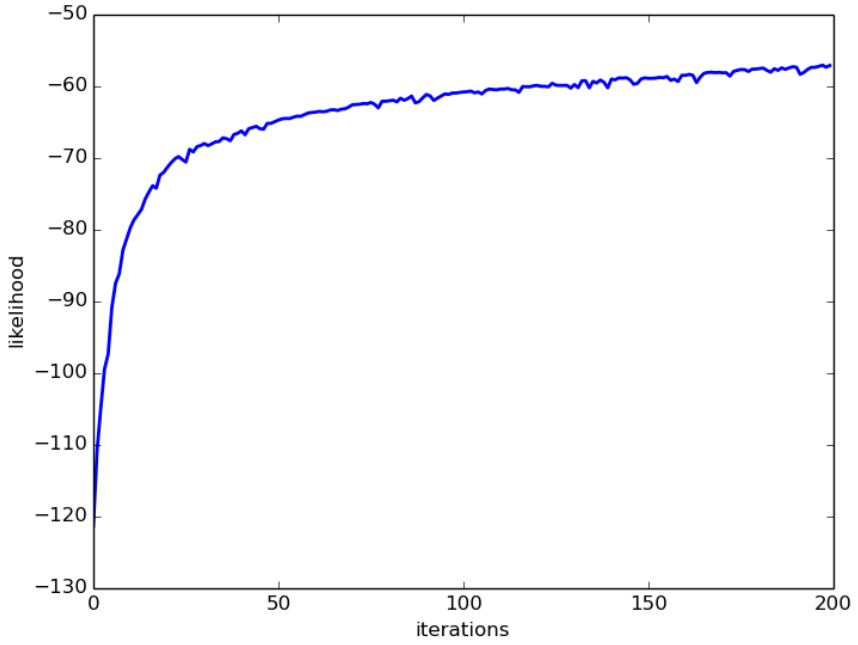


Figure 5: likelihood vs iterations for stochastic gradient ascent with 40 subset size



4 Logistic Regression - Batch Gradient Ascent Regularized

For regularizing logistic regression we need to add the term $-\frac{1}{2\sigma^2}||w||^2$. So our equations will look like this:

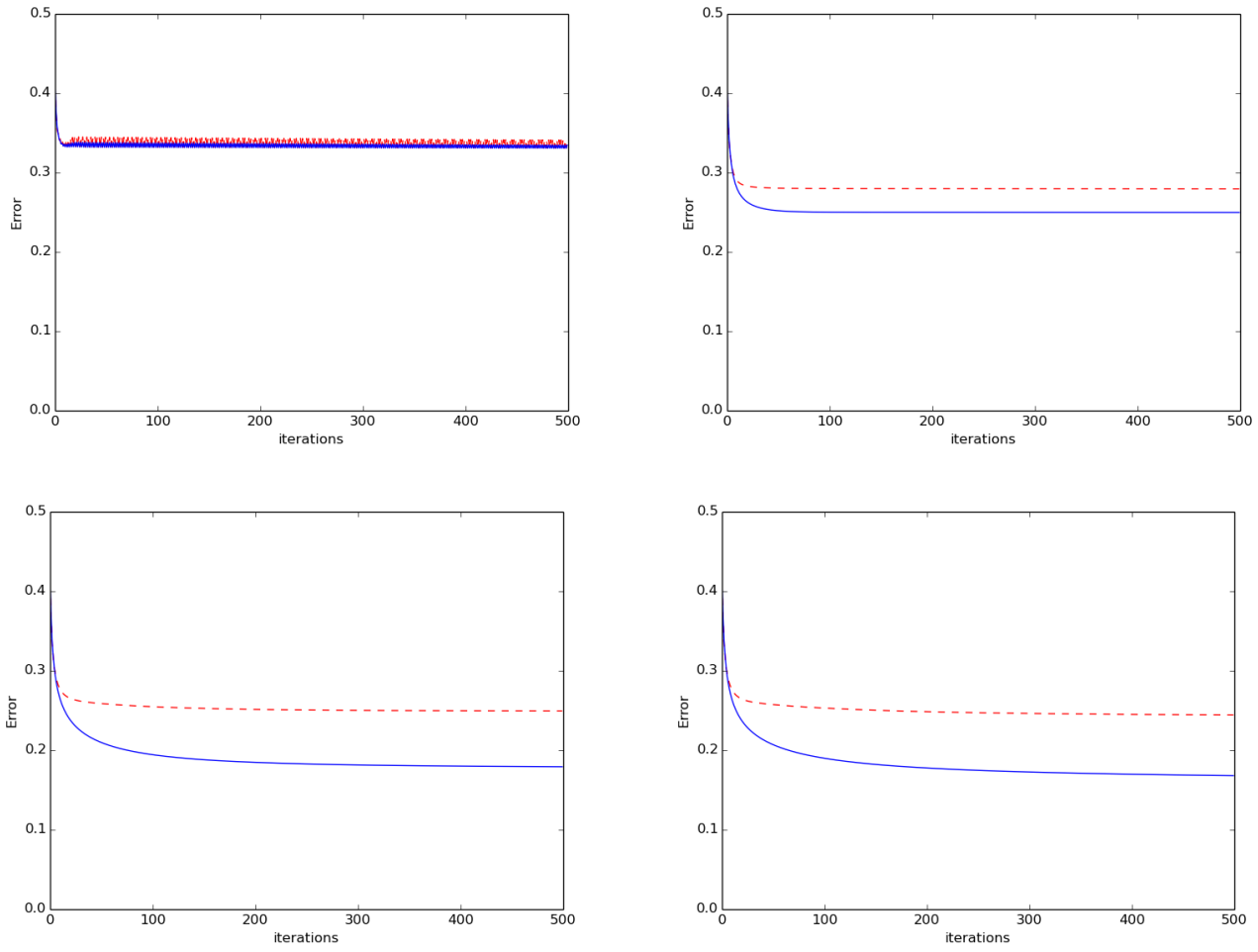
$$l(W) = \sum_l Y^l(w_0 + \sum_i^n w_i X_i^l) - ln(1 + exp(w_0 + \sum_i^n w_i X_i^l)) - \frac{1}{2\sigma^2}||w||^2 \tag{4}$$

$$\frac{\partial l(W)}{\partial w_i} = \sum_l X_i^l(Y^l - \hat{P}(Y^l = 1|X^l, W)) - \frac{w_i}{\sigma} \tag{5}$$

$$w_i \leftarrow w_i + \eta \left[\sum_l X_i^l(Y^l - \hat{P}(Y^l = 1|X^l, W) - \frac{w_i}{\sigma}\right] \tag{6}$$

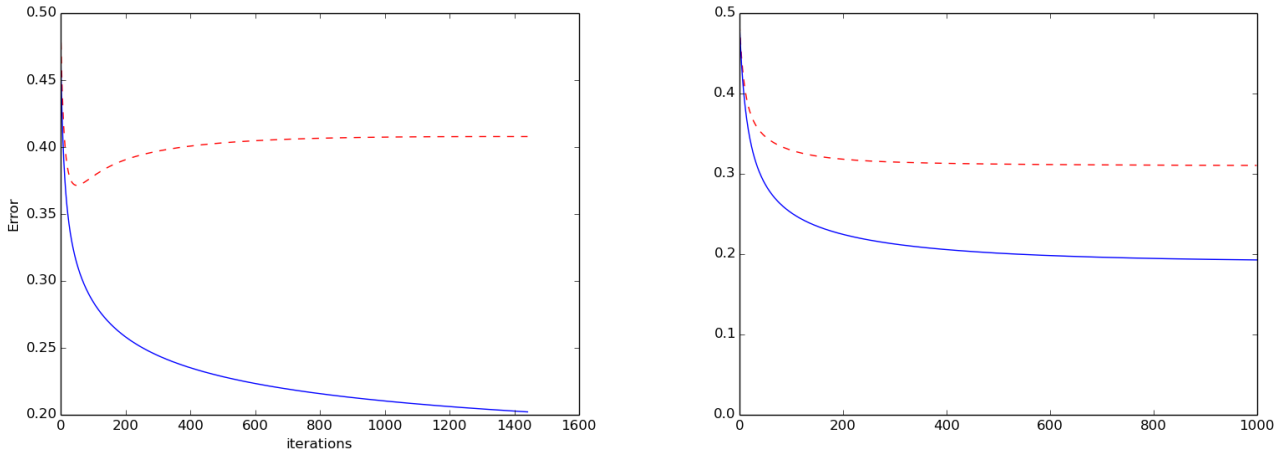
For the regularization section I divided the dataset into three groups: train set(60%), validation set(20%) and test set(20%). I tried different values for σ. it is set to $[0.1, 0.5, 1, 2, 10, 1000]$. The purpose was to tune σ over validation set. However, Since I did not have any over-fitting on my model, setting σ to bigger values had a better result for my model. I also test the the regularization in the case that I set the training set to be very small, just to see the regularization effect.
These figures show regularized classifiers with different values for σ. bigger value for the σ will reduce the effect of regularization. As it is illustrated in the graphs since we did not have any over-fitting

Figure 6: batch gradient ascent with regularization with different values for σ. From left to right and top to bottom σ is set to $0.1, 0.5, 10, 100000$.



on the data regularization does not help us that much, it just make the graphs more smooth. In the figure with σ set to 0.1, since η/σ is equal to 0.2 it means that we subtract $0.2w_i$ from it each iteration and this have a severe impact on the results and the test error seem to fluctuate. Now I am going to investigate the case when the training set is set to 10% and we have over-fitting. In figure 7 train(blue) and test(red) error is shown for cases with and without regularization. Obviously, lack of training samples caused over-fitting in this case. I am describing this just to show the regularization effect. In the right graph (with regularization) you can see that test error remain at same level (around 0.3) compare to left graph (without regularization) which test error goes up (around 0.4) by performing more iterations.

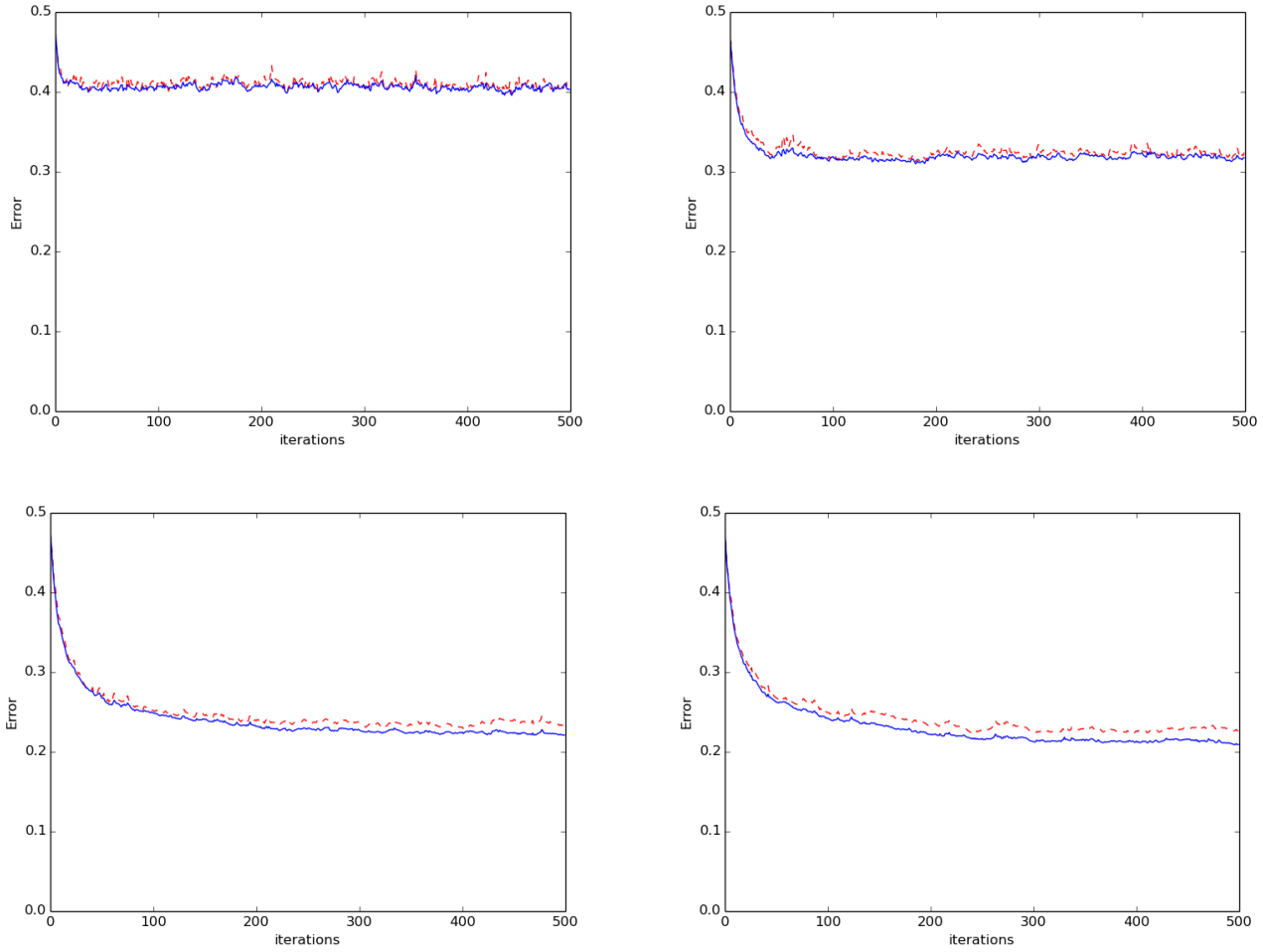
Figure 7: batch gradient ascent with(right) and without(left) regularization



5 Logistic Regression - Stochastic Gradient Ascent Regularized

Figure 8 shows regularized stochastic gradient ascent with different values for σ. As discussed in the previous section since the model does not have over-fitting, regularization with higher values for seems to have a better performance.

Figure 8: stochastic gradient ascent with regularization with different values for σ. From left to right and top to bottom σ is set to $0.1, 0.5, 10, 1000$.



6 Compare With Decision Tree

Results suggests that logistic regression have a better performance on this dataset. This might be due to the fact that this datasets' feature are more discriminative when the model consider them together and our decision tree did not consider attributes correlation in classification. Logistic regression can capture the correlation between the attributes and this might be the reason that it worked better on this dataset.

Test Fold Index	Accuracy	Precision	Recall	F1-Measure
Decision Tree	70.92%	69.36%	61.7%	0.64
Batch Gradient Ascent	88.42%	80%	91.42%	0.85
Stochastic Gradient Ascent	82.1%	80.5%	74.3%	0.77