UNIVERSITY OF SOUTHERN CALIFORNIA, COMPUTER SCIENCE

Machine Learning: HomeWork 2

Rohit Kondekar
740-581-9473

October 7, 2014

1 Question 1

1.1 Regression with heterogenous noise

Solution a:
As the noise is independently distributed but donot have to be identically distributed, This implies the presence of either heteroscedasticity, or correlations i.e. the covariance matrix Σ may be a diagonal matrix.

As noise is normally distributed with $N(0, \sigma_n^2)$:

$$P(\varepsilon_n) = (2\Pi)^{\frac{-n}{2}} |\Sigma|^{\frac{-1}{2}} exp(\frac{-1}{2}\varepsilon_n \Sigma^{-2} \varepsilon_n)$$

As they are independently distributed -

$$\begin{aligned}
L(\varepsilon) = \prod_{i=1}^{n} P(\varepsilon_n) = (2\Pi)^{\frac{-n^2}{2}} |\Sigma|^{\frac{-n}{2}} exp(\frac{-1}{2}\sum_{i=1}^{n}[(y_i - x_i^T\beta)^T\Sigma^{-1}(y_i - x_i^T\beta)])\\
l(\varepsilon) = log(L)\\
l(\varepsilon) = const - \frac{n}{2}log(|\Sigma|) - \frac{1}{2}\sum_{i=1}^{n}[(y_i - x_i^T\beta)^T\Sigma^{-1}(y_i - x_i^T\beta)]
\end{aligned}$$

Solution b:

$$\begin{aligned}
\hat{\beta} &= argmin_b (Y - X\beta)\Sigma^{-1}(Y - X\beta)\\
&= (Y^T - \beta^T X^T)\Sigma^{-1}(Y - X\beta)\\
&= (Y^T\Sigma^{-1} - \beta^T X^T \Sigma^{-1})(Y - X\beta)\\
&= Y^T\Sigma^{-1}Y - \beta^T X^T\Sigma^{-1}Y - Y^T\Sigma^{-1}X\beta + \beta^T X^T \Sigma^{-1} X\beta
\end{aligned}$$

Properties used:

$$\begin{aligned}
\nabla_\Theta(b^T\Theta) = b\\
\nabla_\Theta(\Theta^T A \Theta) = (A + A^T)\Theta
\end{aligned}$$

$$\frac{dl(\varepsilon)}{d\beta} = -2(X^T\Sigma^{-1}Y) + [X^T\Sigma^{-1}X + (X^T\Sigma^{-1}X)^{-1}]\beta = 0$$

$$\begin{aligned} 2X^T\Sigma^{-1}X\beta &= 2X^T\Sigma^{-1}Y \\ \beta &= (X^T\Sigma^{-1}X)^{-1}X^T\Sigma^{-1}Y \end{aligned}$$

As $X^T\Sigma^{-1}X$ is a symmetric matrix.

1.2 Smooth Coefficients

Solution a:
To define this regularizer, a special constant matrix C p×p is defined such that $C_{ij} = 1$ whenever $i = j$, $C_{ij} = 0$ when $j = n$ and $C_{ij} = -1$ when $j - i = 1$. For eg. if β is 4×1 then :

$$C = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C\beta = \begin{bmatrix} \beta_1 - \beta_2 \\ \beta_2 - \beta_3 \\ \beta_3 - \beta_4 \\ 0 \end{bmatrix}$$

Therefore the new regularized equation can be stated as:

$$\begin{aligned} RSS(\beta) &= \frac{1}{2}\sum_{i=1}^{m}(y_i - \theta^T x_i)^2 + \lambda_1||C\beta||_2^2 + \lambda_2||\beta||_2^2 \\ &= \frac{1}{2}(Y - \beta X)^T(Y - \beta^T X) + \lambda_1(C\beta)^T(C\beta) + \lambda_2\beta^T\beta \end{aligned}$$

Solution b:

$$\begin{aligned} \hat{\beta} &= argmin_b\frac{1}{2}(Y - \beta X)^T(Y - \beta^T X) + \lambda_1\beta^T C^T C\beta + \lambda_2\beta^T\beta \\ \frac{dRSS(\beta)}{d\beta} &= 0 \\ 0 &= X^T X\beta - X^T Y + \lambda_1[C^T C + C^T C]\beta + \lambda_2\beta \\ X^T X\beta + 2\lambda_1 C^T C\beta + \lambda_2\beta &= X^T Y \\ \beta &= (X^T X + 2\lambda_1 C^T C + \lambda_2 I)^{-1}X^T Y \end{aligned}$$

1.3 Linearly Constrained Linear Regression

Solution:
This problem can be solved using langrange multiplier by adding $\lambda(A\beta - b)$ as it has a non-empty set of solutions.
Also, originally without this constraint $\hat{\beta}$ was given by:

$$\hat{\beta} = (X^T X)^{-1}X^T Y \tag{1.1}$$

Now with the given constraint we have to minimize :

$$\hat{\beta}' = argmin_\beta\frac{1}{2}(Y - \beta^T X)^T(Y - \beta^T X) + \lambda(A\beta - b) \tag{1.2}$$

Where $\hat{\beta}'$ is the constrained Maximum likelihood estimator. With constraint : $A\beta - b = 0$

Therefore differentiating the equation we get:

$$X^T X\beta - X^T Y + A^T\lambda = 0 \tag{1.3}$$
$$A\beta - b = 0 \tag{1.4}$$

To estimate λ : multiplying minimizing Eq. by $A(X^T X)^{-1}$ we get:

$$\begin{aligned} A(X^T X)^{-1}X^T X\beta - A(X^T X)^{-1}X^T Y + A(X^T X)^{-1}A^T\lambda = 0 \\ A\beta - A\hat{\beta} + A(X^T X)^{-1}A^T\lambda = 0 \\ A(X^T X)^{-1}A^T\lambda = A\hat{\beta} - b \end{aligned}$$

$$\lambda = (A(X^T X)^{-1}A^T)^{-1}(A\hat{\beta} - b) \tag{1.5}$$

Now solving for β:

$$X^T X\beta = X^T Y - A^T[A(X^T X)^{-1}A^T)^{-1}(A\hat{\beta} - b)] \tag{1.6}$$
$$\beta = (X^T X)^{-1}(X^T Y - A^T[A(X^T X)^{-1}A^T)^{-1}(A\hat{\beta} - b)]) \tag{1.7}$$

2 Question: Online Learning

Solution: As we know that whenever perceptron correctly classifies a datapoint: $y_n W_i x_n > 0$ else when incorrectly classifies $y_n W_i x_n <= 0$.
Therefore our optimization equation boils down to solving [Assumnig L2norm squared]:

$$\begin{aligned} \text{Minimize } \frac{1}{2}||w_{i+1} - w_i||_2^2 \\ \text{With constraint } y_n(w_{i+1}x_n) > 0 \end{aligned}$$

Therefore writing it in terms of lagrange operator:

$$\begin{aligned} \text{Minimize } \frac{1}{2}(w_{i+1} - w_i)^T(w_{i+1} - w_i) + \lambda[y_n(w_{i+1}x_n)] \\ \text{Minimize } \frac{1}{2}w_{i+1}^T w_{i+1} - w_{i+1}^T w_i + \lambda[y_n w_{i+1}x_n] \end{aligned}$$

Taking Derivative:

$$\begin{aligned} -w_{i+1} - w_i + \lambda y_n x_n = 0 \\ w_{i+1} = w_i - \lambda y_n x_n \end{aligned}$$

To obtain the value of lambda differentiate the minimizing equation wrt lambda. We get:

$$y_n w_{i+1} x_n = 0$$

Substituting w_{i+1} from last equation:

$$\begin{aligned} y_n[w_i - \lambda y_n x_n]x_n = 0 \\ \lambda = \frac{y_n w_i x_n}{x_n^2} \end{aligned}$$

3 Question: Kernels

3.1

Solution a:
We can show that, K3 is indeed a positive definite kernal function if :

$X^T(a_1K_1 + a_2K_2)X >= 0$

It can be written as:

$a_1X^TK_1X + X^Ta_2K_2)X >= 0$

As $X^TK_1X >= 0$ and $X^TK_2X >= 0$ as K1 and K2 are PSDs and $a_1 >= 0, a_2 >= 0$ which shows that:

$a_1X^TK_1X + X^Ta_2K_2)X >= 0$

Therefore it is Positive SemiDefinite Matrix.

Solution b:
Any matrix of the form:

$$K_3 = \begin{bmatrix} f(x_1)f(x_2) & f(x_1)f(x_2) & ... & f(x_1)f(x_n) \\ ... & ... & ... & ... \\ f(x_n)f(x_1) & f(x_n)f(x_2) & ... & f(x_n)f(x_n) \end{bmatrix}$$

Can be broken down to: FF^T where

$$F = \begin{bmatrix} f(x_1) \\ f(x_2) \\ . \\ . \\ f(x_n) \end{bmatrix}$$

So We have to show that $X^TFF^TX >= 0$

$$\begin{aligned} X^TFF^TX &= (F^TX)^TF^TX \\ &= ||F^TX||_2^2 \\ ||F^TX||_2^2 &>= 0 \\ X^TFF^TX &>= 0 \end{aligned}$$

Solution c:
This is bascically the Hadamard product of two positive semidefinite matrices, which is always positive definite. We can show that using Eigen Value Decomposition.

Let K1 = $\sum \mu_i k1_i k1_i^T$ and K2 = $\sum \nu i k2_i k2_i^T$

$$\begin{aligned} K1 \circ K2 &= \sum \mu_i k1_i k1_i^T \circ \sum \nu j k2_j k2_j^T \\ &= \sum_{ij} \mu_i \nu j (k1_i \circ k2_j)(k1_i \circ k2_j)^T \end{aligned}$$

To show that $(k1_i \circ k2_j)(k1_i \circ k2_j)^T$ is positive:

$$X(k1_i \circ k2_j)(k1_i \circ k2_j)^T X = (\sum_k k1_{i,k} k2_{j,k} x_k)^2 \geq 0$$

This shows that it is positive definite matrix.

4 Question: Bias-Variance Tradeoff

5 Question: Programming Assignment

***Please Note:
Log taken is Natural Log: Matlabs Log() function.

(1) top 3 words that occur most frequently?

(enron,600) (will,351) (please,291)

(2)Updating equation for w and b for batch gradient descent

Without Regularization:

$$\begin{aligned} &\text{Repeat } \{ \\ &\quad w_j^{t+1} = w_j^t - \eta[\sum_{i=1}^m (h(x^i) - y^i) x_j] \\ &\quad b^{t+1} = b^t - \eta \sum_{i=1}^m (h(x^i) - y^i) \\ &\} \end{aligned}$$

In Matlab this can be written as:

weights = weights + η*((y-hypothesis)'*x)';
b = b + η*(sum(y-hypothesis));

Note: Here h(x) is a sigmoid function.

With Regularization:

$$\begin{aligned} &\text{Repeat } \{ \\ &\quad w_j^{t+1} = w_j^t - \eta[\sum_{i=1}^m (h(x^i - y^i) x_j - \lambda w_j^t)] \\ &\quad b^{t+1} = b^t - \eta \sum_{i=1}^m (h(x^i) - y^i) \\ &\} \end{aligned}$$

In Matlab this can be written as:

weights = weights + η*((y-hypothesis)'*x+lambda*w')';
b = b + η*(sum(y-hypothesis));

Note: Here h(x) is a sigmoid function.

	L2 norm without regularization	0.001	0.01	0.05	0.1	0.5
(3)(b)	EmailSpam	2.5903	7.7726	26.4429	51.2757	253.4047
	Ionosphere	1.5111	4.7440	19.1862	39.9547	189.6151

	L2 norm $\eta = 0.01$	0	0.05	0.1	0.15	0.2	0.25	0.30	0.35	0.4	0.45	0.50
(4)(b)	EmailSpam	2.602	2.596	2.590	2.584	2.578	2.572	2.565	2.559	2.553	2.547	2.541
	Ionosphere	1.501	1.498	1.495	1.492	1.489	1.486	1.483	1.480	1.477	1.475	1.472

(5)Updating equation for w and b for Newton's method

In Newton's Method the basic update equation is given by:

Without Regularization

$$w^{t+1} \leftarrow w^t - (H^t)^{-1} \bigtriangledown \varepsilon(w^t)$$

Where H^t is given by:

$$\text{For Weights} \quad H(w_j^t) = \sum_{i=1}^{m} (h(x^i)(1 - h(x^i)x_j^i(x_j^i)^T))$$

$$\text{For Bias} \quad H = \sum_{i=1}^{m} (h(x^i)(1 - h(x^i))$$

And $\bigtriangledown \varepsilon(w^t)$ is given by:

$$\text{For Weights} \quad \bigtriangledown \varepsilon(w_j^t) = \sum_{i=1}^{m} (h(x^i) - y^i)x_j^i$$

$$\text{For Bias} \quad \bigtriangledown \varepsilon(b) = \sum_{i=1}^{m} (h(x^i) - y^i)$$

In Matlab this can be achieved by:

(x' * (hypothesis-y)) * (x' * diag(hypothesis) * diag(1-hypothesis) * x)

With Regularization:

$$w^{t+1} \leftarrow w^t - (H^t)^{-1} \bigtriangledown \varepsilon(w^t)$$

Where H^t is given by:

$$\text{For Weights} \quad H(w_j^t) = \sum_{i=1}^{m} (h(x^i)(1 - h(x^i)x_j^i(x_j^i)^T)) + 2\lambda diag(1)$$

$$\text{For Bias} \quad H = \sum_{i=1}^{m} (h(x^i)(1 - h(x^i))$$

And $\bigtriangledown \varepsilon(w^t)$ is given by:

$$\text{For Weights} \quad \bigtriangledown \varepsilon(w_j^t) = \sum_{i=1}^{m} (h(x^i) - y^i)x_j^i + 2\lambda w_j$$

$$\text{For Bias} \quad \nabla \varepsilon(b) = \sum_{i=1}^{m} (h(x^i) - y^i)$$

(6b) L2 norm of w for EmailData = 4.0437e+33
(6b) L2 norm of w for IonoData = Infinity (NaN)

(7)(b)

L2 norm Newtons	0	0.05	0.1	0.15	0.2	0.25	0.30	0.35	0.4	0.45	0.50
EmailSpam	4e+33	12.5	10.39	9.25	8.482	7.91	7.462	7.09	6.78	6.52	6.29
Ionosphere	NaN	16.006	12.227	10.454	9.362	8.600	8.029	7.578	7.210	6.900	6.635

Note: Column 1 with lambda = 0 gives value for unregularized version.

(7)(c): Cross Entropy value after 50 iterations for regularized at different lambdas and unregularized

CrossEntropy 50 Steps	0	0.05	0.1	0.15	0.2	0.25	0.30	0.35	0.4	0.45	0.50
EmailSpam	NaN	15.76	23.0	28.33	32.65	36.32	39.55	42.43	45.05	47.46	49.69
Ionosphere	NaN	7.79	10.80	12.87	14.49	15.81	16.95	17.94	18.82	19.61	20.33

(8) Explanation for result in 3 and 4

For Email and Ionosphere data, in case of unregularized we can see that, as step size size increases there are more spikes (zig-zag) but it still manages to converge (get lower cross-entropies) to some point. For lower step sizes the curve is much more smoother but the change in cross-entropy is much lower between subsequent steps.

In Case of regularized version, the cross entropy achieved by the end is much higher for higher regularizers because the model is more simpler. Also the model is not able to converge at higher step sizes for regularized version, and shows a zigzag nature, in gradient descent. This suggests that, larger the regularization, smaller the step size should be.

(9) Explanation for result in 4 and 7

From results, we can see that, Newton's method converges faster (in 4-5 iterations here) as compared to gradient descent. But it is slower computationaly as it has to compute the inverse of Hessian Matrix. For many dimensions, this would be significantly slower. So for higher dimension, it is better to use gradient descent, but for lower dimensions, Newton's method is much more faster.

Figure 5.1: Cross-Entropy wrt Steps: Email Train Data Unregularized

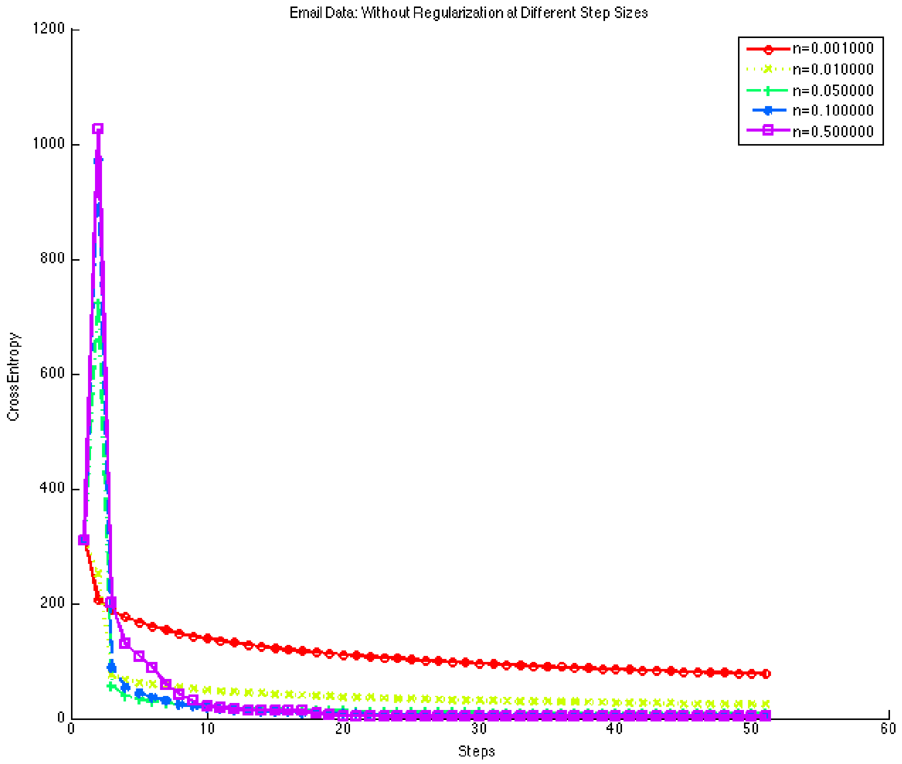


Figure 5.2: Cross-Entropy wrt Steps: Ionosphere Train Data Unregularized

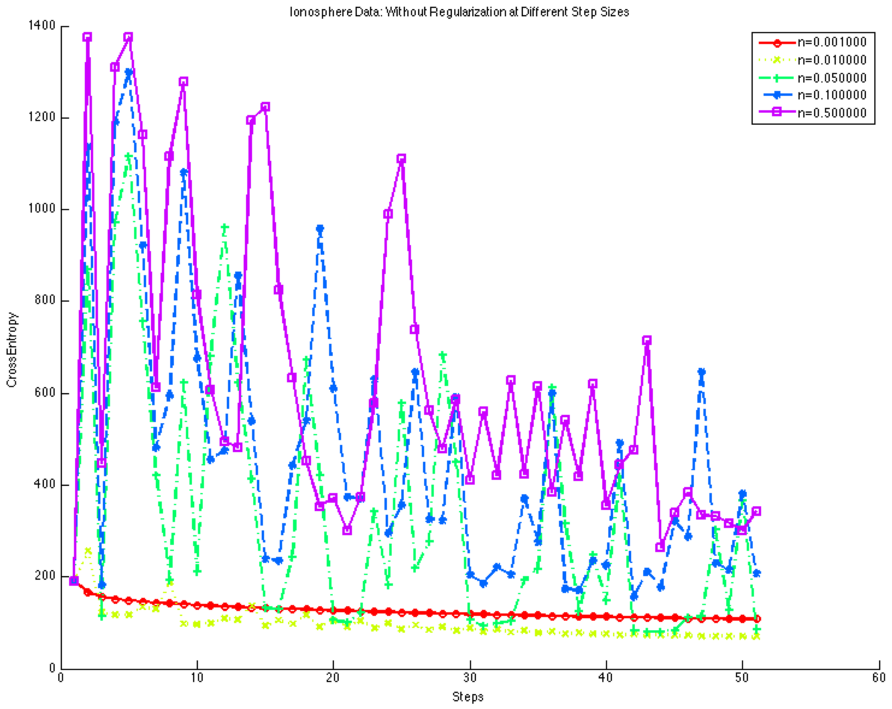


Figure 5.3: Cross-Entropy wrt Steps: Email Train Data Regularized

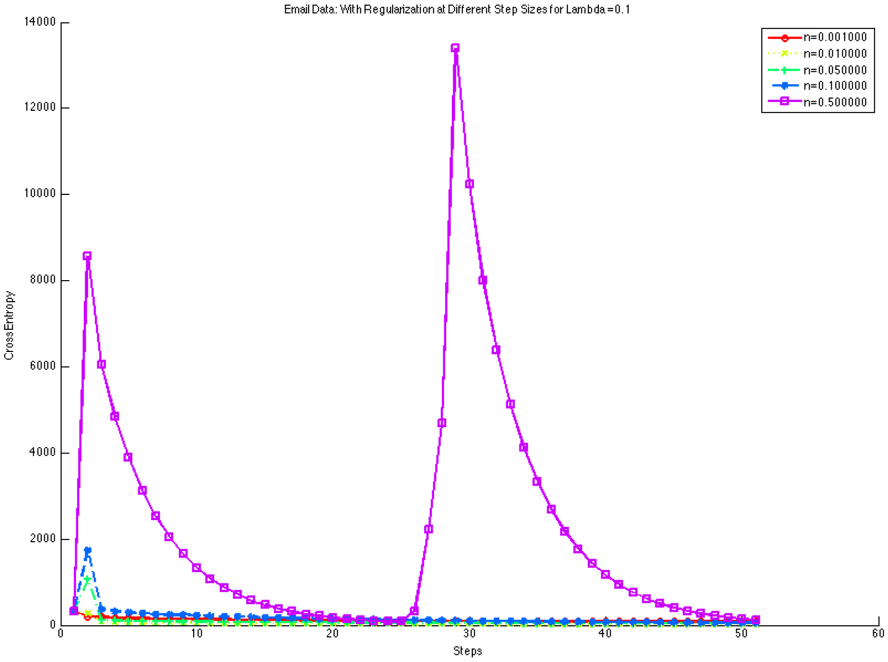


Figure 5.4: Cross-Entropy wrt Steps: Ionosphere Train Data Regularized

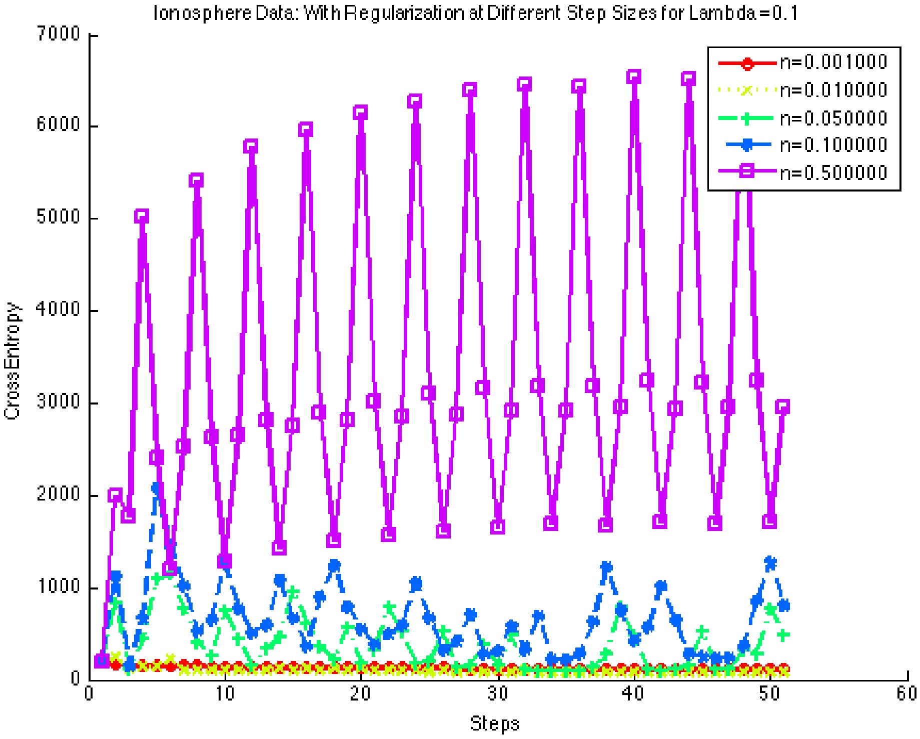


Figure 5.5: Cross-Entropy vs Regularizing Coefficients: Email: Step Size = 0.001

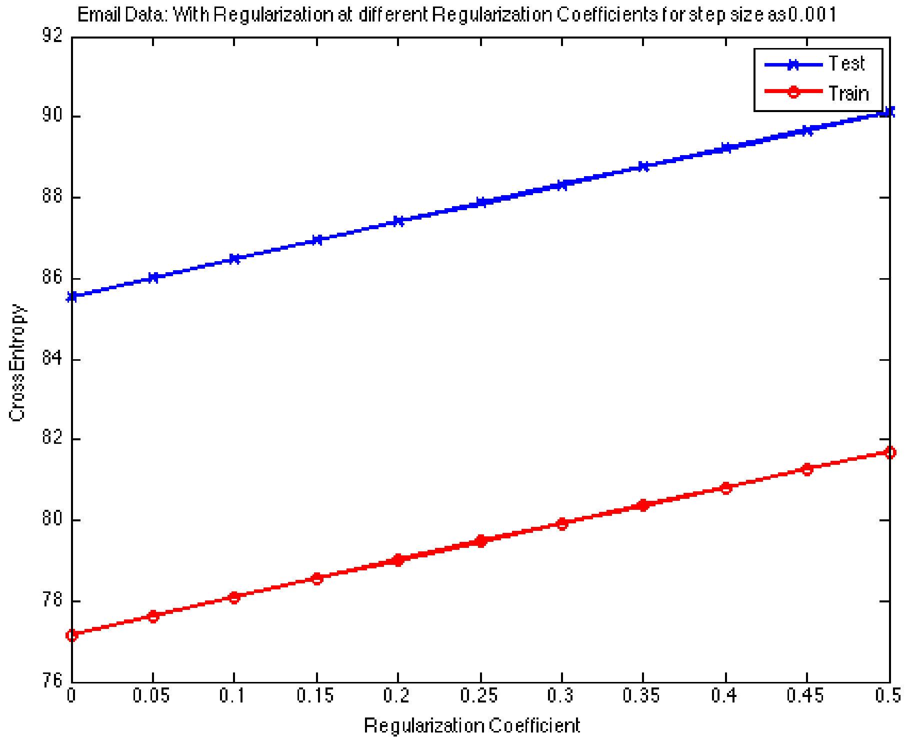


Figure 5.6: Cross-Entropy vs Regularizing Coefficients: Email: Step Size = 0.01

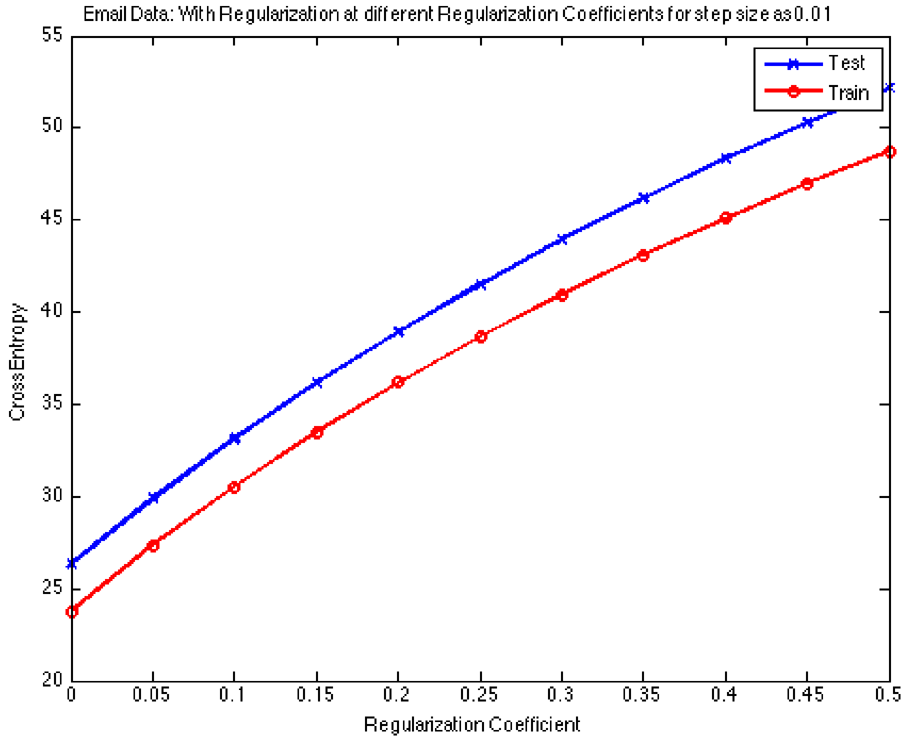


Figure 5.7: Cross-Entropy vs Regularizing Coefficients: Email: Step Size = 0.05

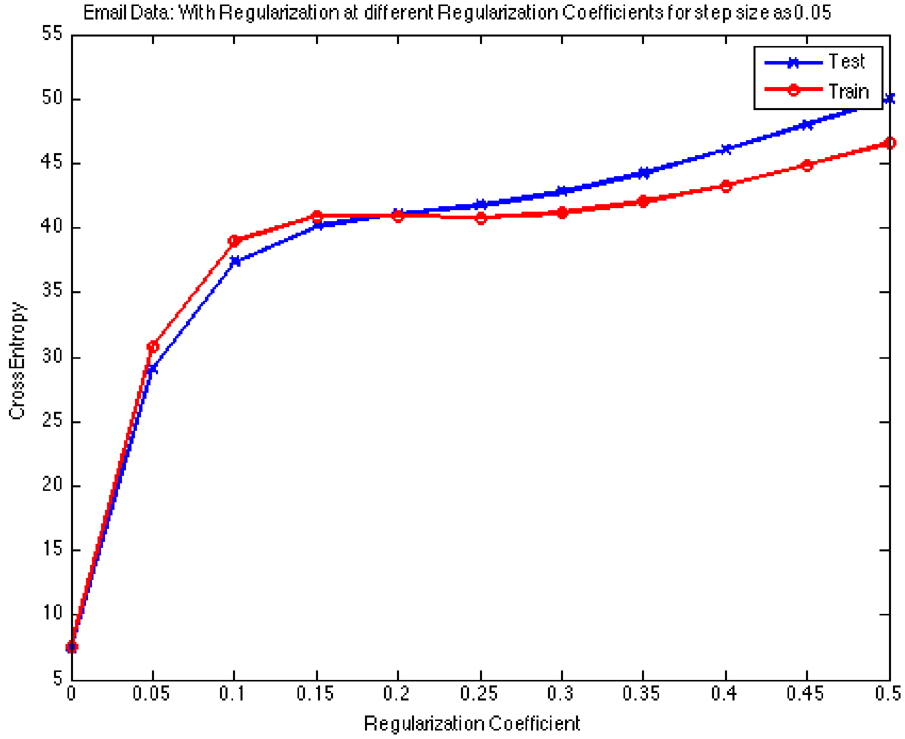


Figure 5.8: Cross-Entropy vs Regularizing Coefficients: Email: Step Size = 0.1

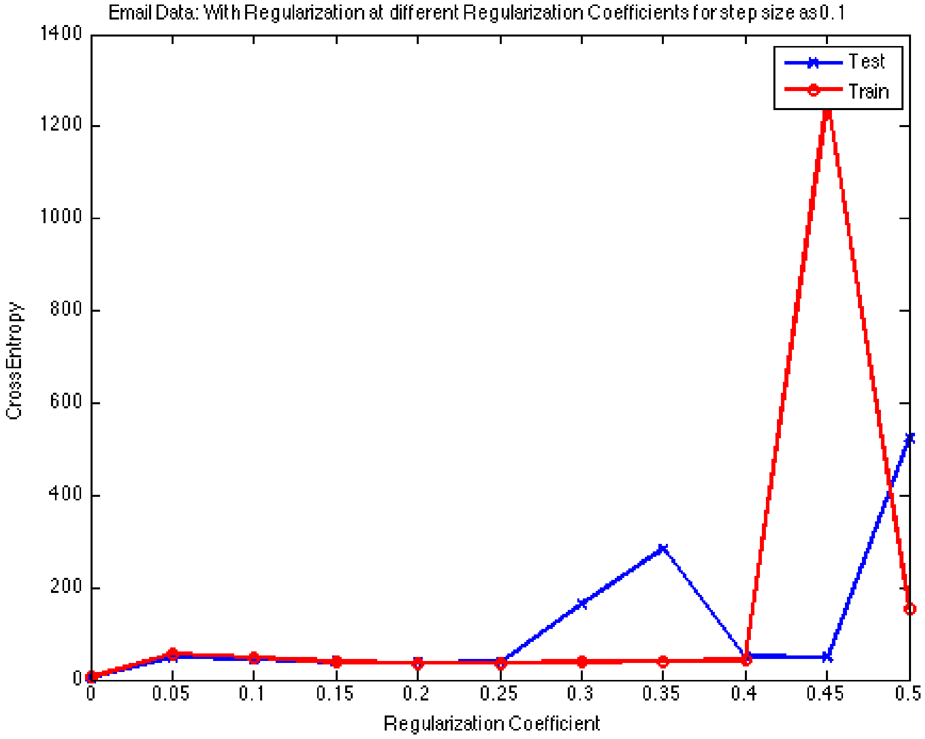


Figure 5.9: Cross-Entropy vs Regularizing Coefficients: Email: Step Size = 0.5

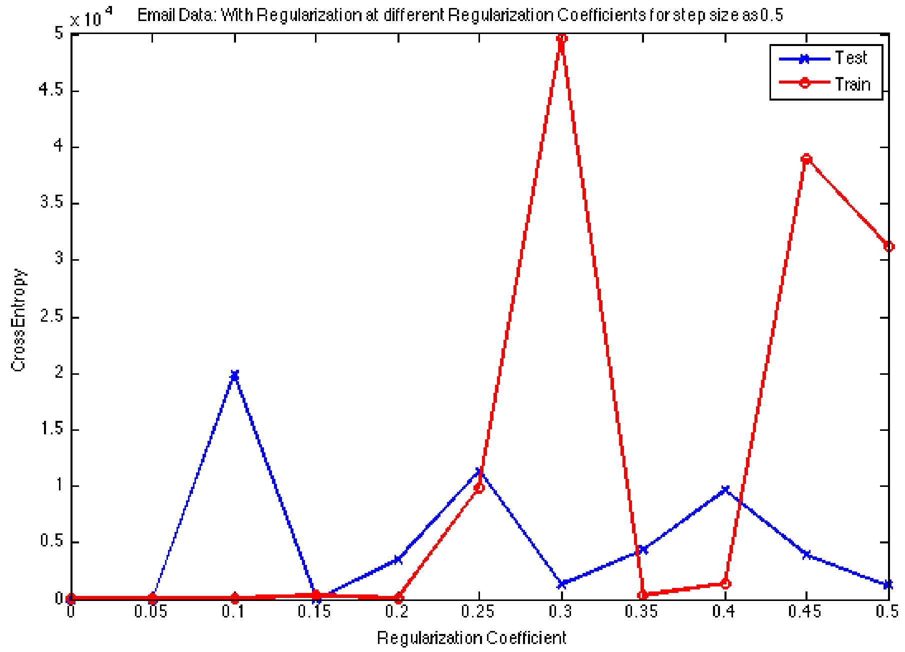


Figure 5.10: Cross-Entropy vs Regularizing Coefficients: Ionosphere: Step Size = 0.001

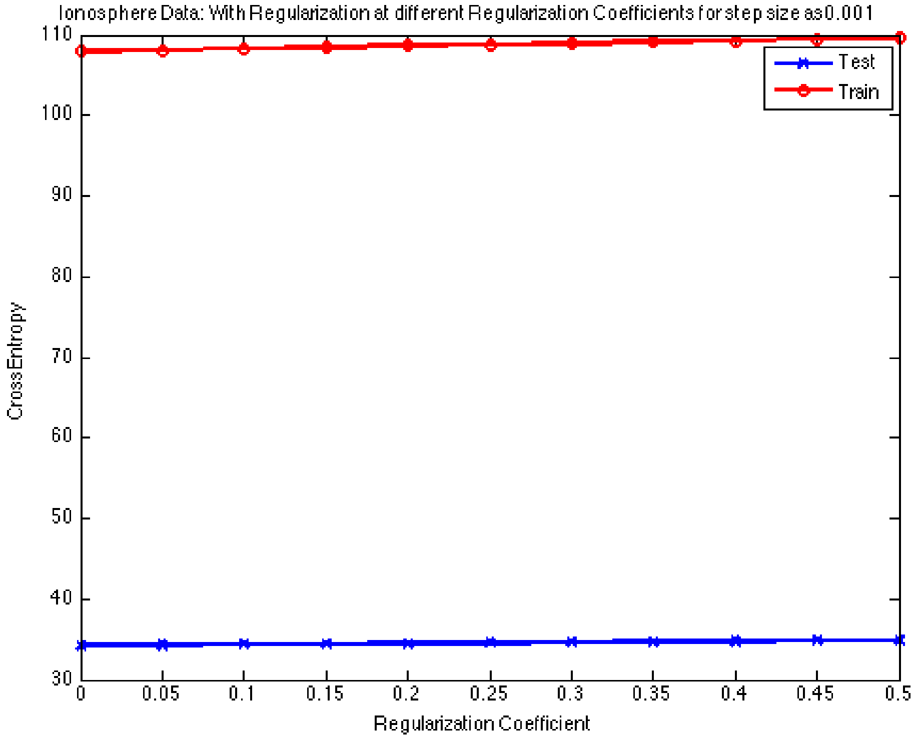


Figure 5.11: Cross-Entropy vs Regularizing Coefficients: Ionosphere: Step Size = 0.01

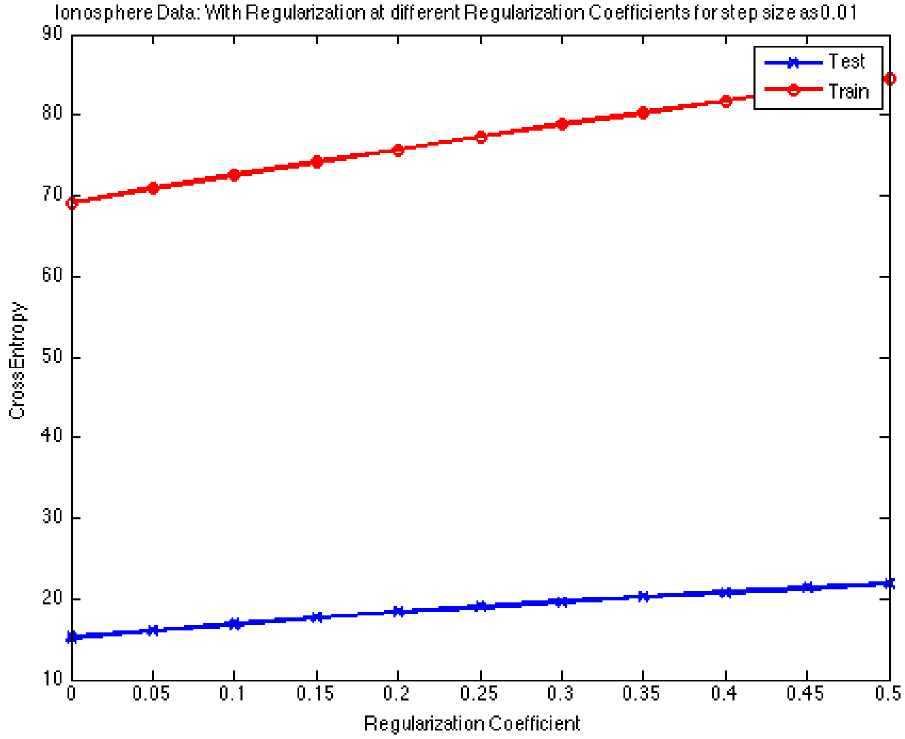


Figure 5.12: Cross-Entropy vs Regularizing Coefficients: Ionosphere: Step Size = 0.05

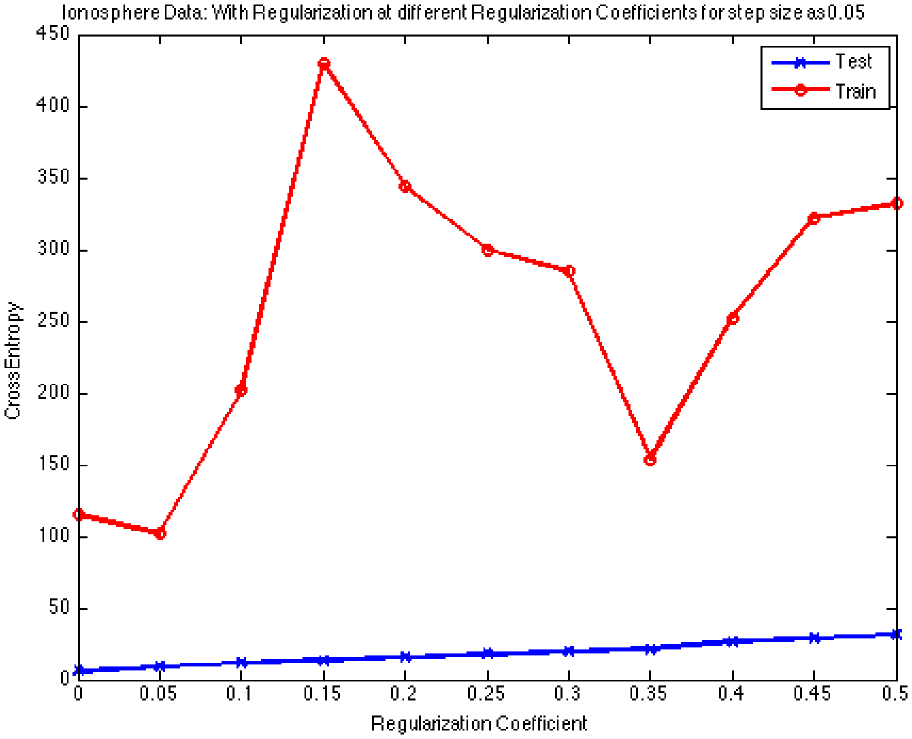


Figure 5.13: Cross-Entropy vs Regularizing Coefficients: Ionosphere: Step Size = 0.1

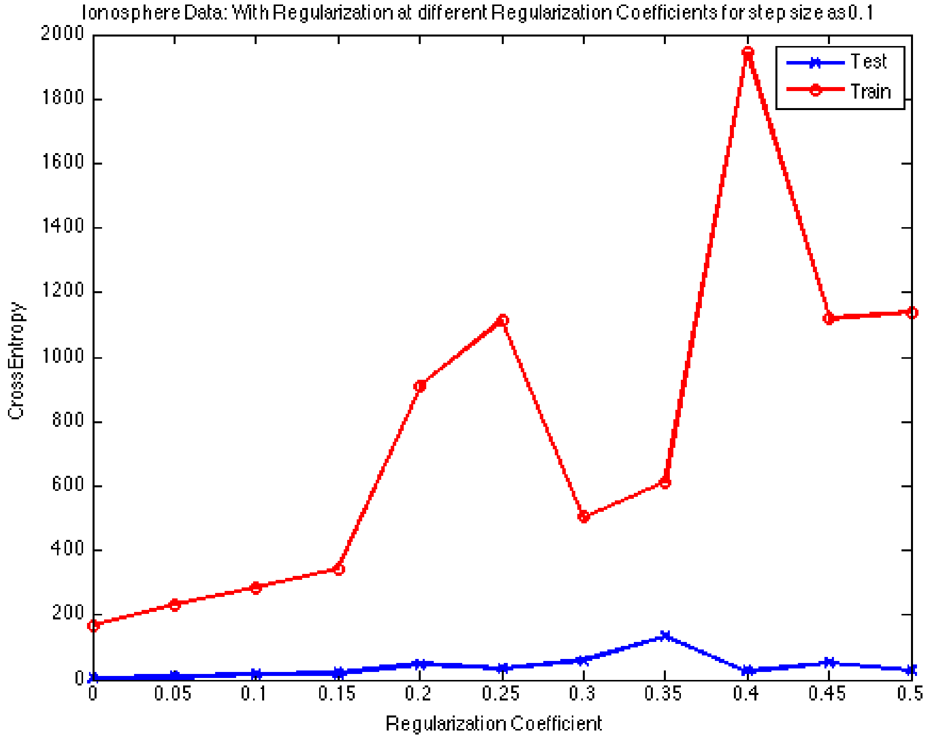


Figure 5.14: Cross-Entropy vs Regularizing Coefficients: Ionosphere: Step Size = 0.5

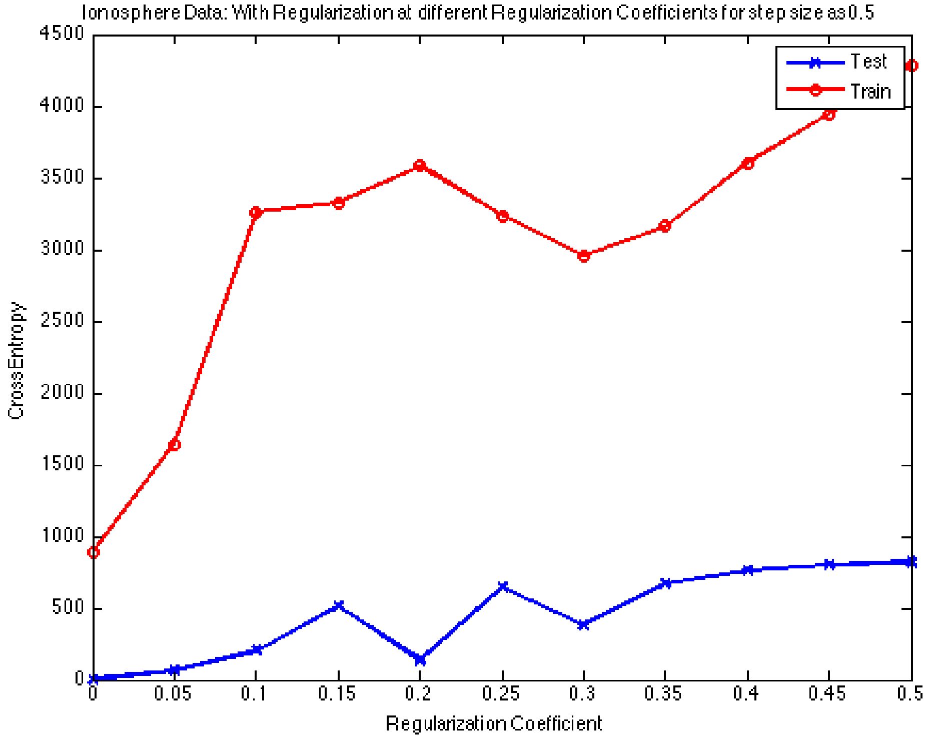


Figure 5.15: Cross-Entropy wrt Steps: Email Train Data Unregularized

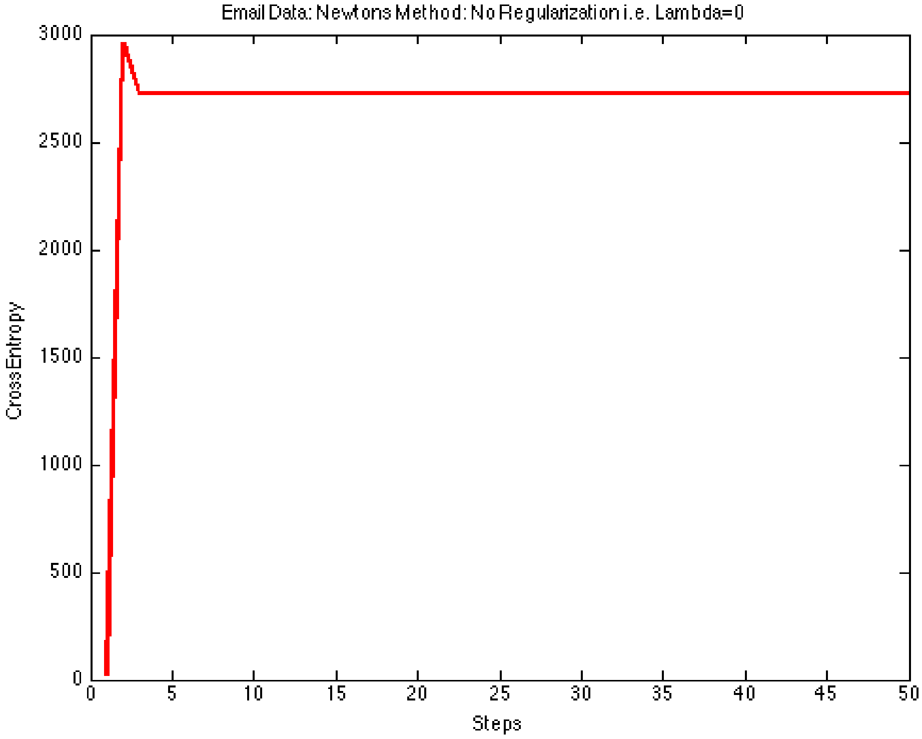


Figure 5.16: Cross-Entropy wrt Steps: Email Train Data Regularized

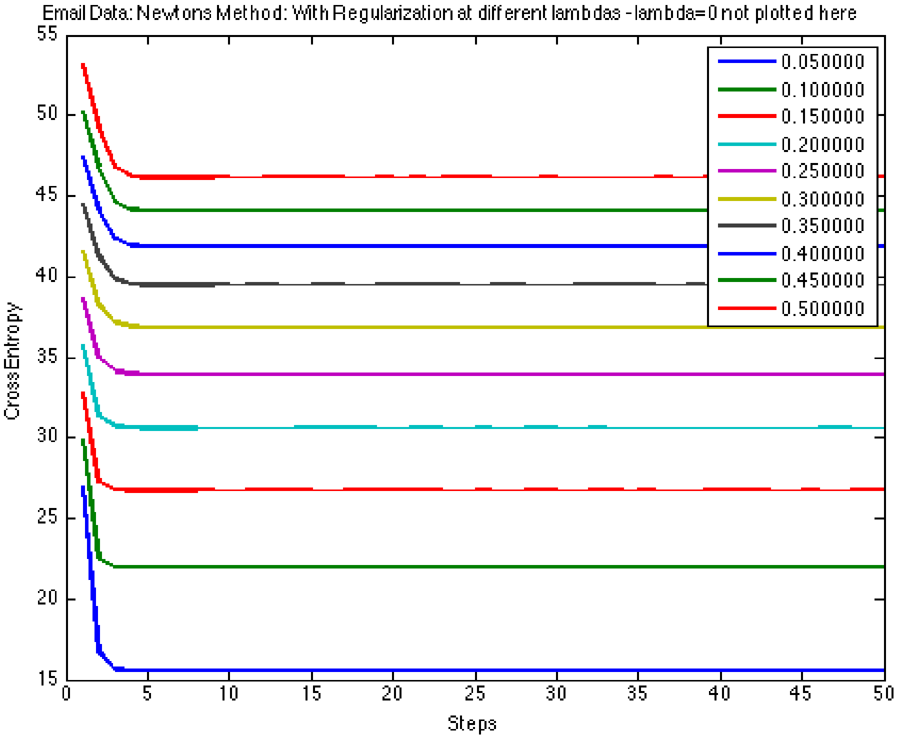


Figure 5.17: Cross-Entropy wrt Steps: Ionosphere Train Data Unregularized

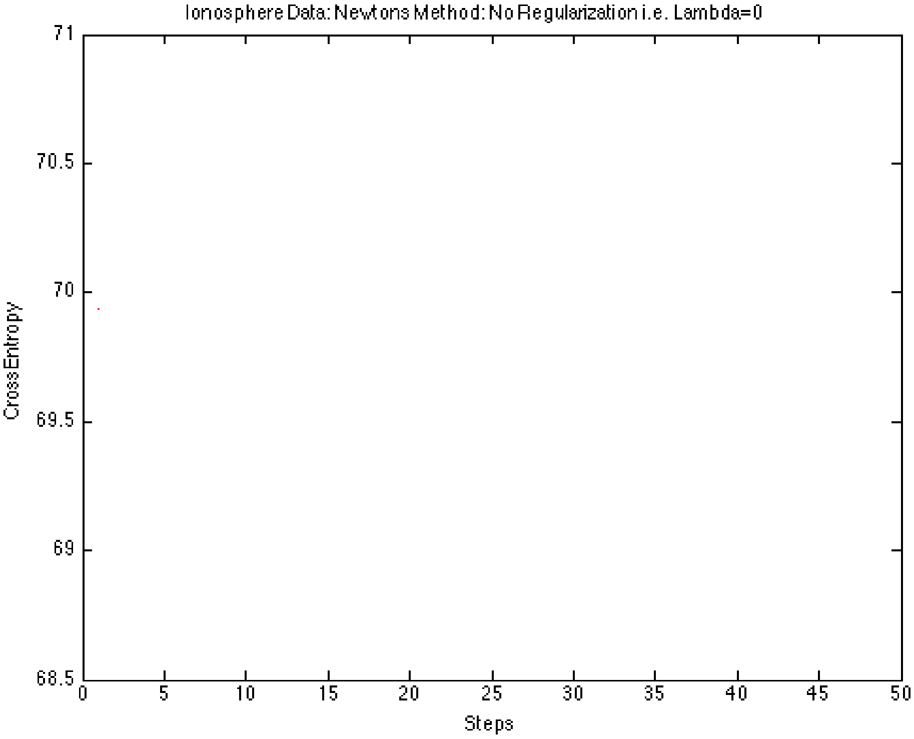
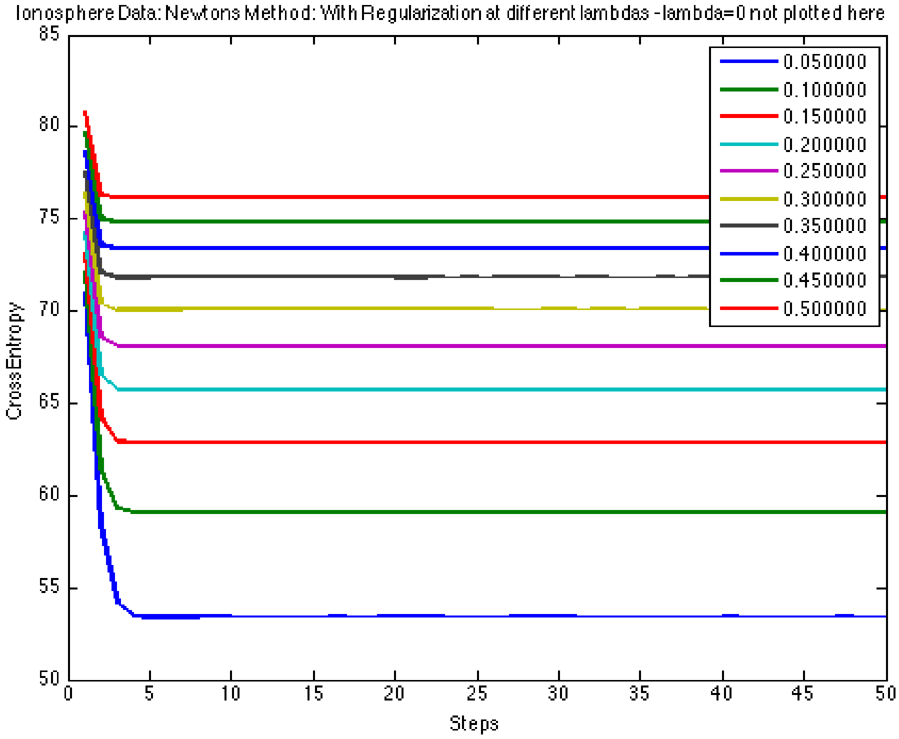


Figure 5.18: Cross-Entropy wrt Steps: Ionosphere Train Data Regularized


Ionosphere Data: Newtons Method: With Regularization at different lambdas -lambda=0 not plotted here
Cross Entropy
Steps