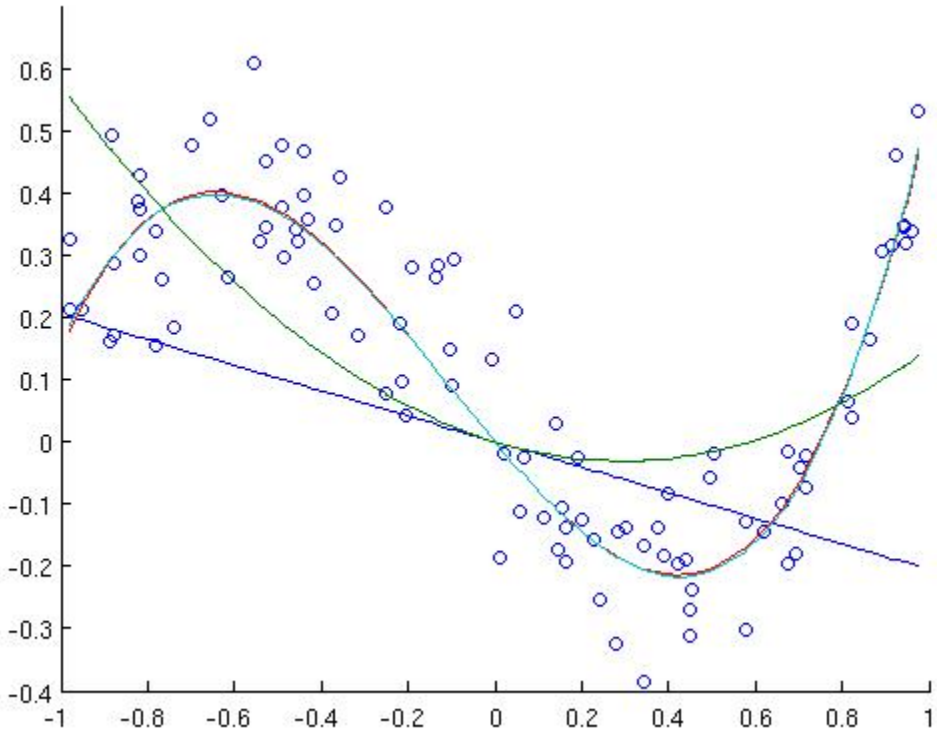
CS189 HW3
Sunil Srinivasan 23038238
Chaaru Dingankar 22580441
02/25/2014

Q1:
a: *d*=3 made the cost function the smallest for the training data, and this is most likely because the data appear to be cubic; this is most likely the best choice for *d* as the data appears cubic and any greater degrees would cause overfitting.

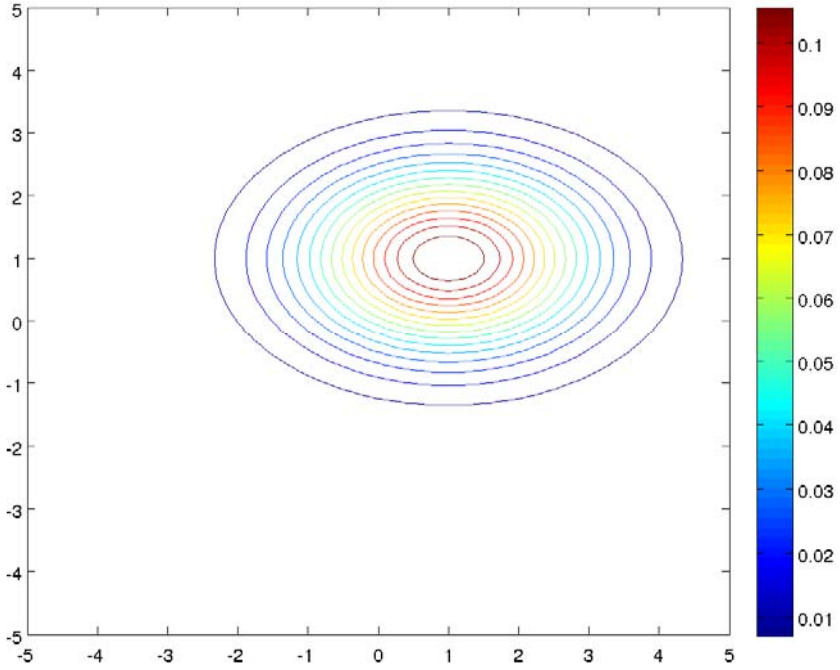
In the figure, the third- and fourth-degree polynomial models are hard to distinguish because they are so similar. The data is represented as a scatter plot and the models are lines, with the straight line for *d*=1, the quadratic curve for *d*=2, and the last curve for *d*=3 and *d*=4.



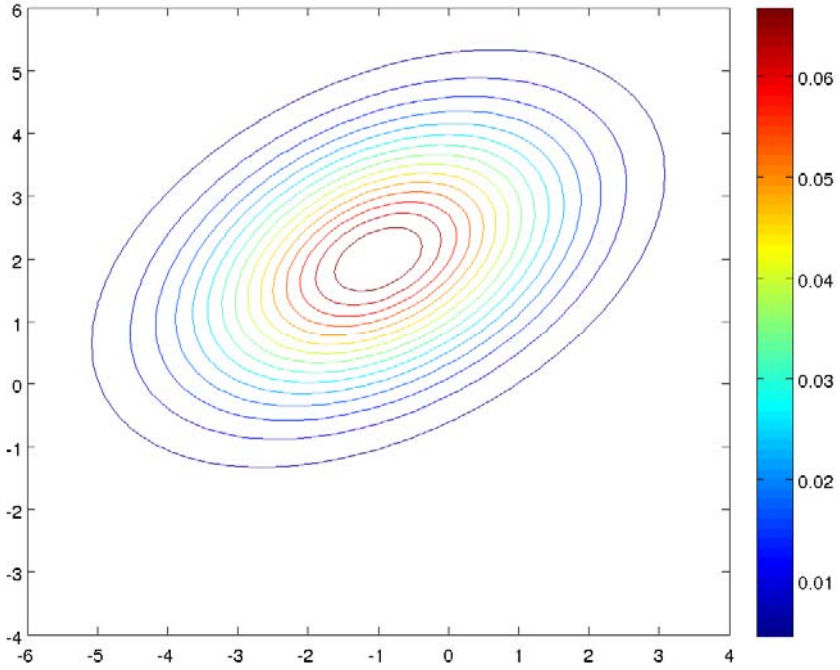
b: The prediction error for $d = 3$ is 4.8148 and for $d = 10$ is 4.7738. We can infer that the 10^{th} degree polynomial does not do too much better at fitting to a curve than the 3rd-degree polynomial. This is true of all polynomial models with degree greater than 3: their improvement over the prediction error from the 3^{rd} degree model is tiny.

CS189 HW3
Sunil Srinivasan 23038238
Chaaru Dingankar 22580441

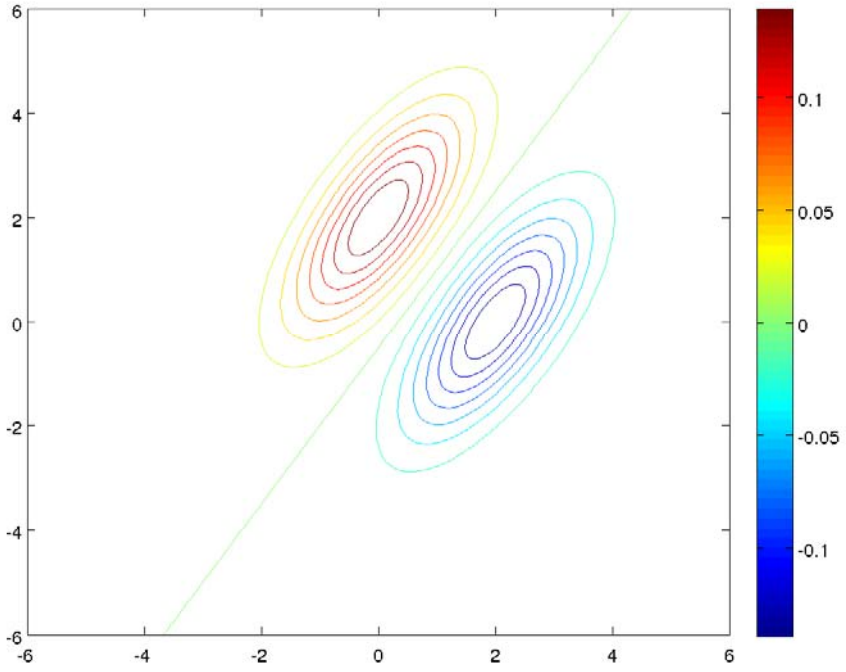
Q2:
i)



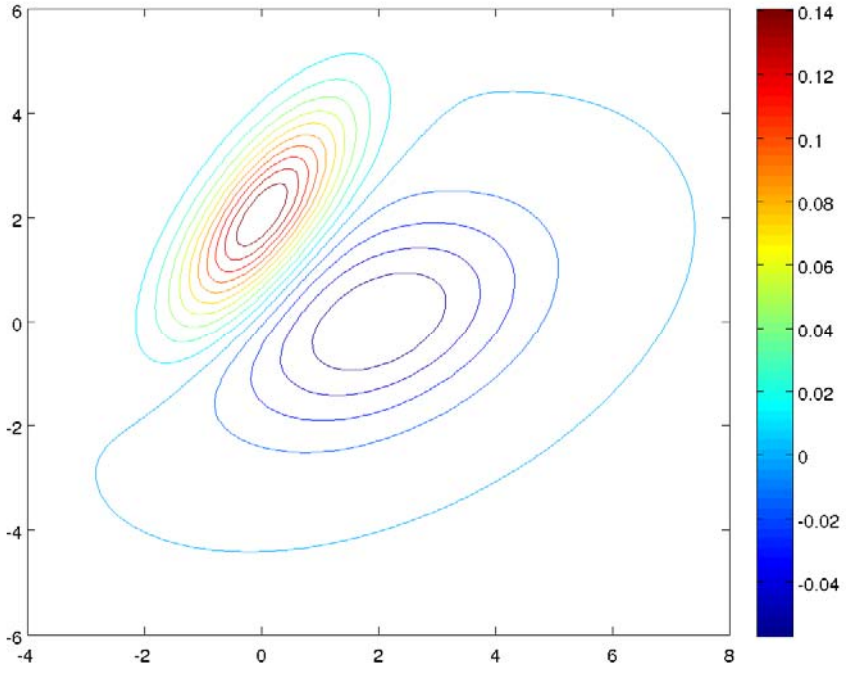
ii)



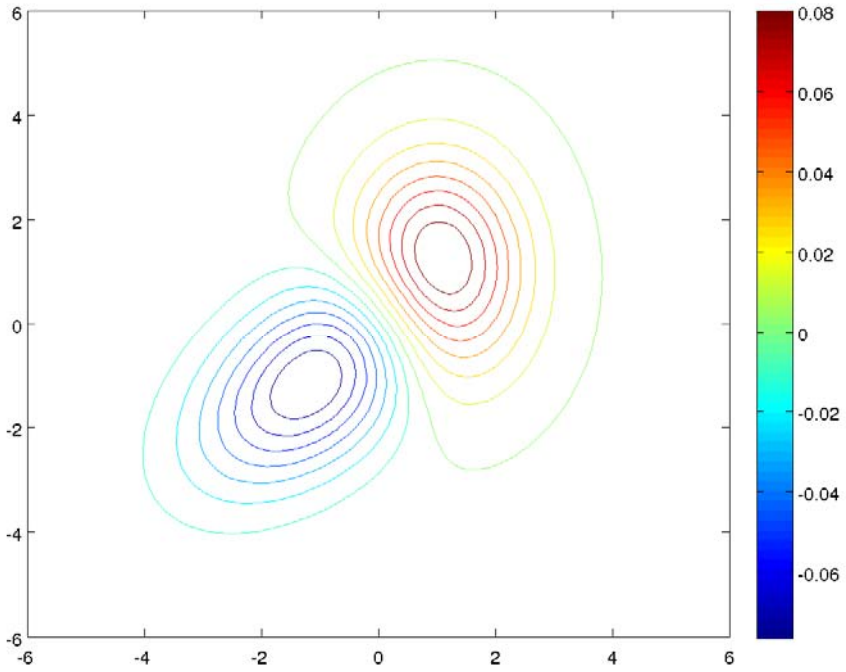
CS189 HW3
Sunil Srinivasan 23038238
Chaaru Dingankar 22580441
iii)



iv)



CS189 HW3
Sunil Srinivasan 23038238
Chaaru Dingankar 22580441
v)



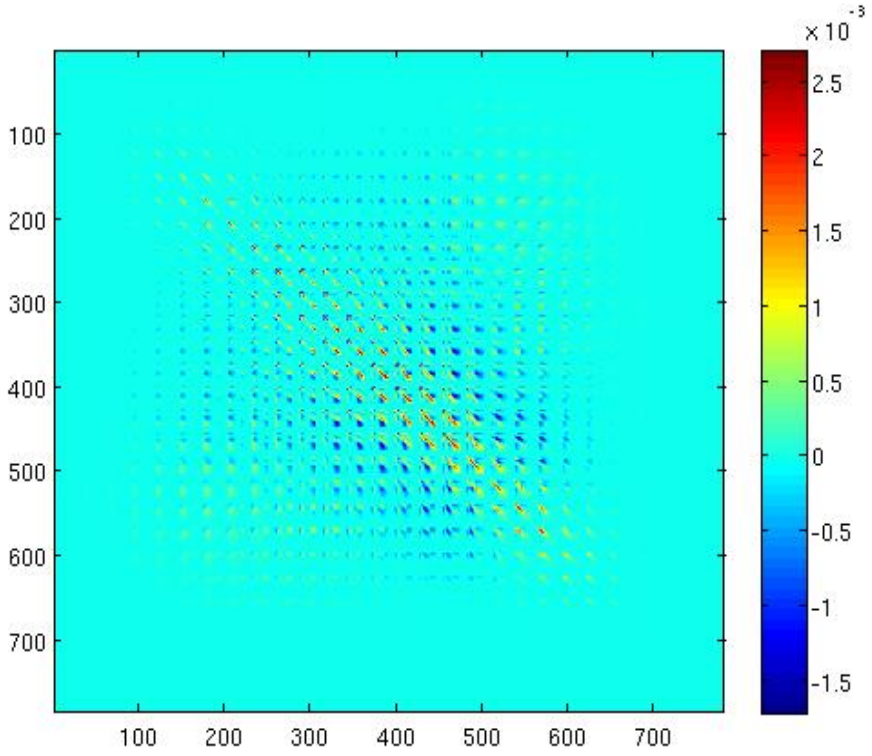
Q3:

i. The MLEs for each digit class in each test set are calculated by the code in q3.m, using the MLEs for a Gaussian distribution:
$\mu = (1/n)(\sum x_i)$
$cov = (1/n)(\sum (x_i - \mu)(x_i - \mu)^T)$
The mean is an unbiased estimator and the covariance is a biased estimator.

ii. We model the priors simply by noting how many of the samples are labeled as each digit class in the training set out of the total number of samples in the training set. Our code in q3.m computes the priors for each digit class in each training set as the array named "priors". In the case of the 10,000-example training set, for example, the prior for the class '0' is 0.0988, the prior for the class '3' is 0.1022, and the prior for the class '9' is 0.0991 (to select three examples at random).

CS189 HW3
Sunil Srinivasan 23038238
Chaaru Dingankar 22580441

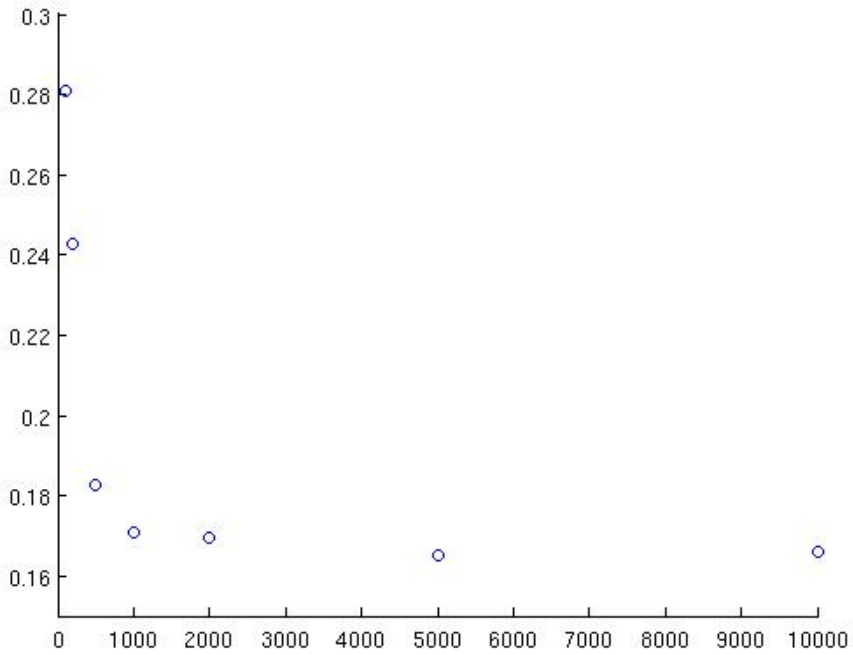
iii. Here is the covariance matrix for the digit class '7' with 10,000 training examples:



We can notice some structure: that there are patches of positive correlation along the diagonal, and negative correlation patches in regions just off the diagonal. This symbolizes that if a pixel is shaded, it is likely that a pixel nearby is shaded as well, but unlikely that a pixel farther away is shaded. The fading to zero towards the edges of the covariance matrix means that a pixel being shaded does not signify anything about pixels near the edge of an image – likely because those pixels at the edge are hardly ever shaded.

CS189 HW3
Sunil Srinivasan 23038238
Chaaru Dingankar 22580441

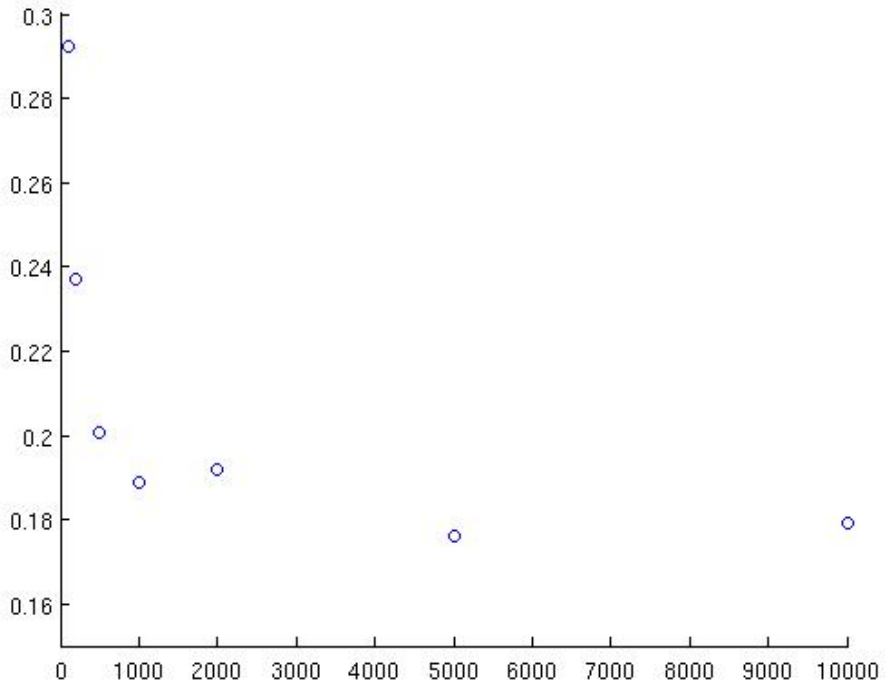
iv. a: In this case, the decision boundary was linear since the covariance matrices were the same for each class (making distance to the mean the primary factor in classifying). Below is the plot of number of training samples vs. error rates:



The error rates are, in order, 0.2808, 0.2429, 0.1825, 0.1707, 0.1696, 0.1652, and 0.1658.

CS189 HW3
Sunil Srinivasan 23038238
Chaaru Dingankar 22580441

b: The decision boundary is a curve, since the covariance matrices are not the same for each class. Below is the plot of the number of training samples vs. error rates:



The errors are, in order, 0.2923, 0.2369, 0.2005, 0.18890, 0.1921, 0.1763, and 0.1792.

c: Our error rates from part (a) are across the board a bit smaller than our rates from part (b). However, according to Piazza, the results from part (b) should be in the range of 5-10% for the last training set, and the results from part (a) should be 10-15% for that same set. Our results are between 15 and 20% for both sets, and instead of (a) being worse than (b), our (a) slightly outperforms (b). We believe this is due to the way we weighted our covariance matrices (so that they were invertible).

Sunil Srinivasan - 23038238
Chaaru Pinyankar - 22580441

CS189 HW 3 #4

$\nabla J = \frac{\partial J}{\partial w_0} + \frac{\partial J}{\partial w}$ Find solutions to $\frac{\partial J}{\partial w_0} = 0$ and $\frac{\partial J}{\partial w} = 0$ to optimize J

$J(w, w_0) = (y - Xw - w_0 1)^T (y - Xw - w_0 1) + \lambda w^T w$

$w_0^2 1^T 1$ ↑

For w_0, rearrange as such:

$J(w, w_0) = (y - Xw)^T (y - Xw) - (y - Xw)^T w_0 1 - w_0 1^T (y - Xw) + n w_0^2 + \lambda w^T w$

$\frac{\partial J}{\partial w_0} = 2nw_0 - (y - Xw)^T 1 - 1^T (y - Xw)$

$0 = 2nw_0 - (y - Xw)^T 1 - 1^T (y - Xw)$ ← $(y - Xw)^T 1 = 1 \cdot (y - Xw) = 1^T (y - Xw) = \sum_i y_i - x_i^T w$

$2nw_0 = 2 \cdot 1^T (y - Xw)$

$nw_0 = \sum_i y_i - x_i^T w$

$x_i^T w = x_i \cdot w = w \cdot x_i$

$nw_0 = \left(\sum_i y_i\right) - w \cdot \sum_i x_i$

$w_0 = \frac{1}{n} \sum_i y_i - \frac{w}{n} \cdot \sum_i x_i$

$\boxed{\hat{w}_0 = \bar{y} - 0 = \bar{y}}$

For w, rearrange J as such:

$J(w, w_0) = (y - w_0 1)^T (y - w_0 1) - (y - w_0 1)^T Xw - (Xw)^T (y - w_0 1) + (Xw)^T Xw + \lambda w^T w$

$(Xw)^T (y - w_0 1) = w^T X^T (y - w_0 1)$; $(Xw)^T Xw = w^T X^T Xw$; $\lambda w^T w = \lambda w^T I w$

$\frac{\partial J}{\partial w} = 0 - X^T (y - w_0 1) - X^T (y - w_0 1) + 2X^T Xw + 2\lambda I w$

$\frac{\partial (w^T X^T X w)}{\partial w} = 2X^T Xw$ since $\frac{\partial x^T A x}{\partial x} = 2Ax$ for symmetric A

$0 = -2X^T (y - w_0 1) + 2(X^T X + \lambda I) w$

$(X^T X + \lambda I) w = X^T (y - w_0 1)$

$X^T 1 = \sum_i x_i = 0$

$(X^T X + \lambda I) w = X^T y - w_0 X^T 1$

$(X^T X + \lambda I) w = X^T y$

$\boxed{\hat{w} = (X^T X + \lambda I)^{-1} X^T y}$

Chaaru Dingankar -22580441
Sunil Srinivasan - 23038238

Hw 3 Problem 5

Show the MLE: $y_i|x_i \sim \mathcal{N}(w_0 + w_1 x_i, \sigma^2)$

$$P(y_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - w_0 - w_1 x_i)^2}{2\sigma^2}\right)$$

$$P(Y|\theta) = \mathcal{L}(\theta|Y) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - w_0 - w_1 x_i)^2}{2\sigma^2}\right)$$

$$l = \sum_{i=1}^{n} \ln\left(\frac{1}{\sqrt{2\pi}\sigma}\right) + \sum_{i=1}^{n}\left(-\frac{(y_i - w_0 - w_1 x_i)^2}{2\sigma^2}\right)$$

$$l = n \ln \frac{1}{\sqrt{2\pi}\sigma} - \frac{n}{2\sigma^2} \sum_{i=1}^{n} (y_i - w_0 - w_1 x_i)^2$$

$$\frac{\delta l}{\delta w_0} = -\frac{n}{\sigma^2} \sum_{i=1}^{n} (-y_i + w_0 + w_1 x_i)$$

$$0 = \sum_{i=1}^{n} (-y_i + w_0 + w_1 x_i)$$

$$\sum_{i=1}^{n} y_i - w_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} w_0 = n w_0$$

$$\frac{1}{n} \sum_{i=1}^{n} y_i - \frac{w_1}{n} \sum_{i=1}^{n} x_i = w_0$$

$$\bar{y} - w_1 \bar{x} = w_0$$

$$\frac{\delta l}{\delta w_1} = \sum_{i=1}^{n} (-y_i x_i + w_0 x_i + w_1 x_i^2)$$

$$0 = -\sum_{i=1}^{n} y_i x_i + \sum_{i=1}^{n} w_0 x_i + \sum_{i=1}^{n} w_1 x_i^2$$

$$0 = -\sum_{i=1}^{n} y_i x_i + \sum_{i=1}^{n} (\bar{y} - w_1 \bar{x}) x_i + \sum_{i=1}^{n} w_1 x_i^2$$

$$\sum_{i=1}^{n} y_i x_i - \sum_{i=1}^{n} \bar{y} x_i = w_1 \left(-\sum_{i=1}^{n} \bar{x} x_i + \sum_{i=1}^{n} x_i^2\right)$$

$$\frac{\sum_{i=1}^{n} y_i x_i - \sum_{i=1}^{n} \bar{y} x_i \cdot \frac{n}{n}}{-\sum_{i=1}^{n} \bar{x} x_i \cdot \frac{n}{n} + \sum_{i=1}^{n} x_i^2} = w_1$$

$$\frac{\sum_{i=1}^{n} y_i x_i - n\bar{y}\bar{x}}{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2} = w_1 \approx \frac{\text{cov}(X,Y)}{\text{var}(X)}$$