CIS 520, Machine Learning, Fall 2020
Homework 2
Due: Monday, September 28th, 11:59pm
Submit to Gradescope

Chun Chang

1 Programming: Least Squares Regression

1.1 Data Set 1 (synthetic 1-dimensional data)

This data set contains 100 training examples and 1000 test examples, all generated i.i.d. from a fixed probability distribution. For this data set, you will run unregularized least squares regression.

1. **Learning Curve.**

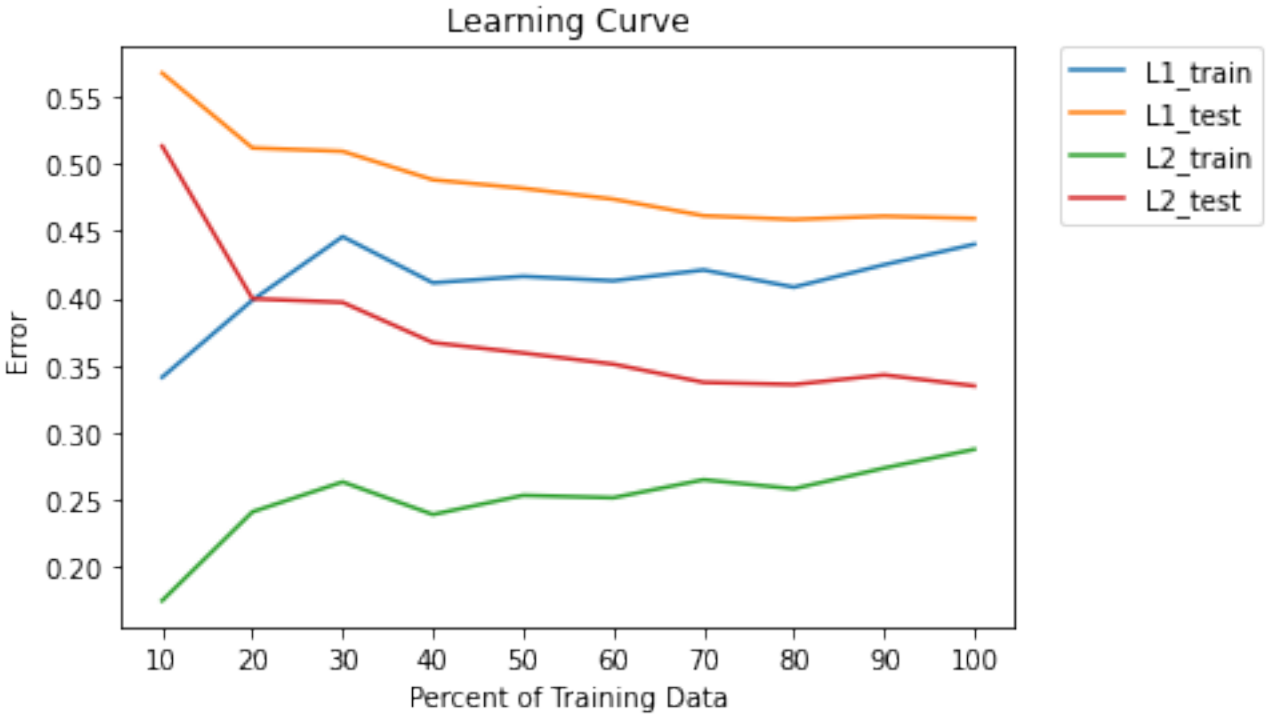

Figure 1: your plot here

2. **Analysis of model learned from full training data.**

Figure 2: your plot here

w = [[-2.23442304]] // b: [1.16717388]// L2 test errors: 0.3347596869735334 // L2 train errors: 0.2876339586619918 //

1.2 Data Set 2 (real 8-dimensional data)

This is a real data set that involves predicting median house value from the location coordinates, demographics and the number of rooms and bedrooms in the houses in total in the block (or district). The data set is a subset of a larger dataset curated for a research conducted by Pace,R. Kelly and Ronald Barry in 1997.This subset has 960 training examples and 240 test examples. For this data set, you will run both unregularized least squares regression and L_2-regularized least squares regression (ridge regression).

1. **Regression on 5% of the training data.**

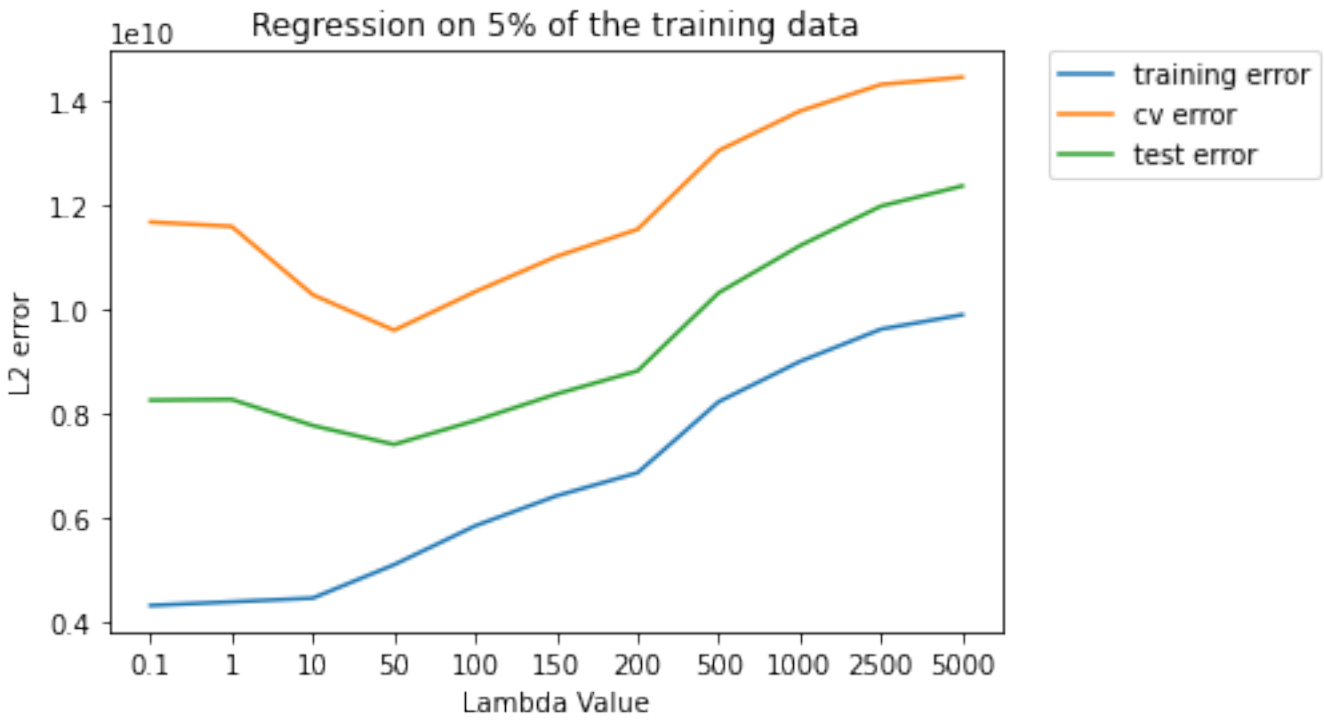

Figure 3: your plot here

2. **Regression on 100% of the training data.**

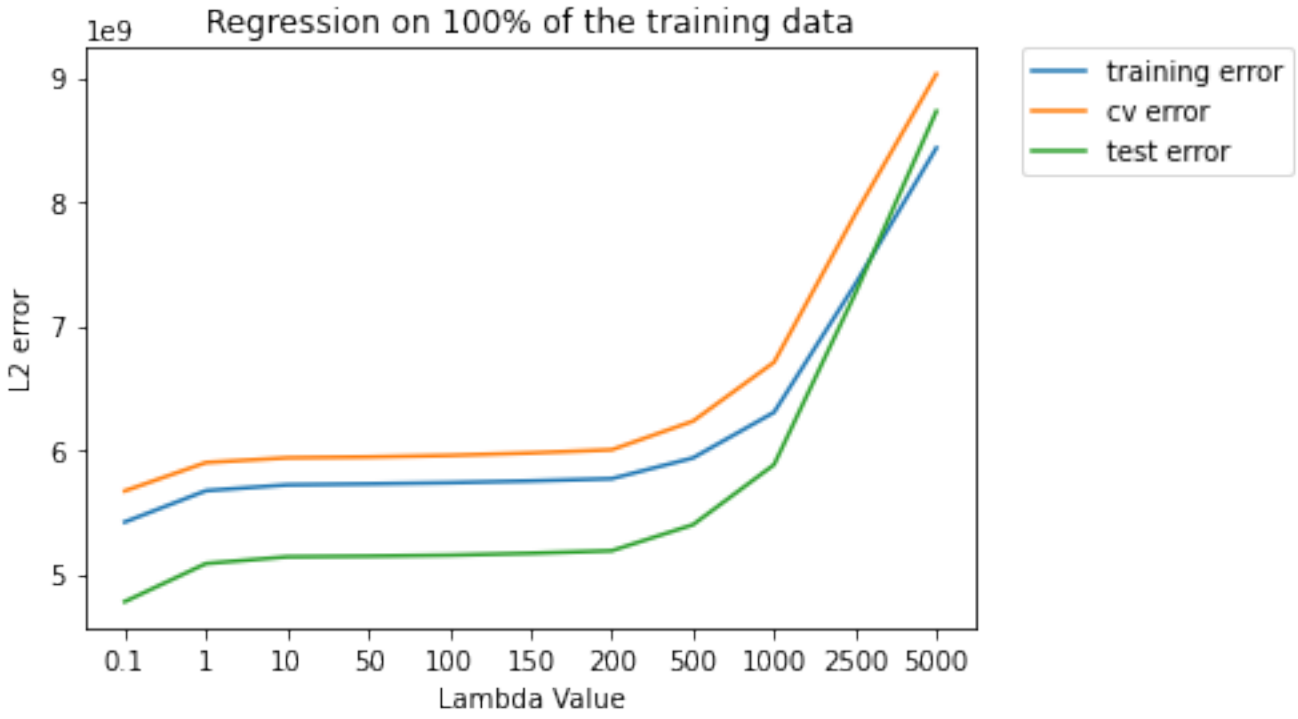

Figure 4: your plot here

3. **Comparison of models learned by two methods** For each of the two training sets considered above (5% and 100%), compare the training and test errors of the models learned using ridge regression. What can you conclude from this about the value of regularization for small and large training sets?

   The models worked poorly when the lambda went to high because all features were penalized too much. And the models had higher bias and lower variance to predict the data; therefore lost the sensitivities.

   The error of predictions of model trained by small dataset tended to have larger magnitude because the ridge regression overfitted the data.

4. **Theoretical Value of λ.** For each of the two training sets considered above (5% and 100%), Which λ should be larger by theory?why? Do those values align with the conclusion you made in part 1.3?

$$\sum_{i=1}^{N}(y_i - w^T X)^2 + \lambda * \sum_{j=1}^{P} w_i^2$$

   In general, the small dataset should have smaller lambda and large dataset should have bigger lambda to compensate the weakening effect caused by the increase of the data.

   The results of previous test didn't align with the theory: lambda trained by the small dataset was larger than that trained by the large dataset. And that is possibly due to overfitting on the noise small set of data.

2 Programming: Batch Gradient Descent

1. **OLS runtime.** Time the closed-form unregularized linear regression implementation you wrote in previous section on the full training data for Data Set 1. Write down the weight and bias terms, $\hat{w}$ and $\hat{b}$, learned from the full training data, as well as the L_2 error on the test data, and the time it took to run the full process.

$$\begin{aligned} \hat{w} &= [[-2.23442304]] \\ \hat{b} &= [1.16717388] \\ L2 \text{ error} &= 0.3347596869735334 \\ \text{time} &= 0.003926992416381836 \end{aligned}$$

2. **Gradient descent runtime.** Time the gradient descent implementation you just wrote on the full training data for Data Set 1 with iterations from range {10, 100, 1000}, and a learning rate of 0.01. Write down the weight and bias terms, $\hat{w}$ and $\hat{b}$, learned from the full training data, as well as the L_2 error on the test data, and the time it took to run the full process.

$$
\begin{aligned}
iterations &: 100 \\
\hat{w} &= -0.26961946 \\
\hat{b} &= 0.10995921 \\
L2 \text{ error} &= 0.5897357152996778 \\
\text{time} &= 0.008636236190795898
\end{aligned}
$$

$$
\begin{aligned}
iterations &: 1000 \\
\hat{w} &= -1.57947694 \\
\hat{b} &= 0.81379186 \\
L2 \text{ error} &= 0.3484509604983921 \\
\text{time} &= 0.053145647048950195
\end{aligned}
$$

$$
\begin{aligned}
iterations &: 2000 \\
\hat{w} &= -2.04105781 \\
\hat{b} &= 1.06284194 \\
L2 \text{ error} &= 0.33135516029916695 \\
\text{time} &= 0.10153245925903325
\end{aligned}
$$

3. **Comparison of algorithms.** Which algorithm runs faster? Why might that be the case? Why would we ever use gradient descent linear regression in practice while a closed form solution exists?

   The ridge regression is faster. If the number of features is

   $$p$$

   and size of training data is

   $$N$$

   , the time complexities of closed form regression is

   $$O(p^3)$$

   and the space complexity is

   $$O(p^2)$$

   , and for the SGD, it is

   $$O(p * N^2)$$

   for runtime and

   $$O(p)$$

   for the space. So if our dataset has lots of features and it doesn't have too many data points, we need SGD to store the training information and still have similar results to the exact solution by the closed form of regression.

3 Regression Models and Squared Errors

Regression problems involves instance spaces $\mathcal{X}$ and labels, and the predictions, which are real-valued as $\mathcal{Y} = \hat{\mathcal{Y}} = \mathbb{R}$. One is given a training sample $S = ((x_1, y_1), ..., (x_m, y_m)) \in (\mathcal{X} \times \mathbb{R})^m$, and the goal is to learn a regression model $f_S : \mathcal{X} \to \mathbb{R}$. The metric used to measure the performance of this regression model can vary, and one such metric is the squared loss function. The questions below ask you to work with regression problems and squared error losses.

1. The squared error is given by $\mathbb{E}_{(x,y)\sim p(X,Y)}[(f(x)-y)^2]$, where the examples are drawn from a joint probability distribution $p(X,Y)$ on $\mathcal{X}\times\mathbb{R}$. Find the lower bound of the expression $\mathbb{E}_{(x,y)\sim p(X,Y)}[(f(x)-y)^2]$. From this lower bound, what is the optimal expression of $f(x)$, in terms of x and Y?

$$
\begin{aligned}
&\mathbb{E}_{(x,y)\sim p(X,Y)}[(f(x)-y)^2] \\
=&\mathbb{E}_{(x,y)\sim p(X,Y)}[(f(x)-\hat{y}+\hat{y}-y)^2] \\
=&\mathbb{E}_{(x,y)\sim p(X,Y)}[(f(x)-\hat{y})^2+2*(f(x)-\hat{y})(\hat{y}-y)+(\hat{y}-y)^2] \\
=&\mathbb{E}_{(x,y)\sim p(X,Y)}[(f(x)-\hat{y})^2]+\mathbb{E}_{(x,y)\sim p(X,Y)}[2*(f(x)-\hat{y})(\hat{y}-y)]+\mathbb{E}_{(x,y)\sim p(X,Y)}[(\hat{y}-y)^2]
\end{aligned}
$$

$$
\begin{aligned}
&\mathbb{E}_{(x,y)\sim p(X,Y)}[2*(f(x)-\hat{y})(\hat{y}-y)] \\
=&2*\mathbb{E}_x[\mathbb{E}_{y|x}[(f(x)-\hat{y})(\hat{y}-y)]],
\end{aligned}
$$

where

$$f(x)-\hat{y}$$

are dependent on sample S

$$=2*\mathbb{E}_x[(f(x)-\hat{y})\mathbb{E}_{y|x}[(\hat{y}-y)]]$$

When

$$
\begin{aligned}
&\hat{y}=\mathbb{E}_{y|x}[y] \\
&\mathbb{E}_{(x,y)\sim p(X,Y)}[2*(f(x)-\hat{y})(\hat{y}-y)]=0
\end{aligned}
$$

$$
\begin{aligned}
&\mathbb{E}_{(x,y)\sim p(X,Y)}[(f(x)-y)^2] \\
=&\mathbb{E}_{(x,y)\sim p(X,Y)}[(f(x)-\hat{y})^2]+\mathbb{E}_{(x,y)\sim p(X,Y)}[(\hat{y}-y)^2] \\
&\geq \mathbb{E}_{(x,y)\sim p(X,Y)}[(\hat{y}-y)^2]
\end{aligned}
$$

this is the lowerbound

when optimal

$$f(x)=\mathbb{E}_{y|x}[y]$$

, the mean squared error reached lowerbound

$$\mathbb{E}_{(x,y)\sim p(X,Y)}[(f(x)-y)^2]=\mathbb{E}_{(x,y)\sim p(X,Y)}[(\hat{y}-y)^2]$$

2. With this result, complete the following two problems. Consider the regression task in which instances contain two features, each taking values in $[0,1]$, so that the instance space is $\mathcal{X}=[0,1]^2$, and with label and prediction spaces belonging to the real space. Suppose examples $(\mathbf{x},y)$ are drawn from the joint probability distribution D, whose marginal density on $\mathcal{X}$ is given by

$$\mu(\mathbf{x})=2x_1,\quad \forall\mathbf{x}=(x_1,x_2)\in\mathcal{X}$$

and the conditional distribution of Y given $\mathbf{x}$ is given by

$$Y|X=\mathbf{x}\sim\mathcal{N}(x_1-2x_2+2,1)$$

What is the optimal regression model $f^*(X)$ and the minimum achievable squared error for D?

$$f(x) = \mathbb{E}_{y|x}[y] = \int yPr * (y \mid x)dy = x_1 - 2x_2 + 2$$

$$L_D[f^*] = \mathbb{E}_{(x,y)\sim p(X,Y)}[(\hat{y} - y)^2] = 1$$

3. Suppose you give your friend a training sample $S = ((\mathbf{x}_1, y_1), ..., (\mathbf{x}_m, y_m))$ containing m examples drawn i.i.d from D, and your friend learns a regression model given by

$$f_S(\mathbf{x}) = x_1 - 2x_2, \quad \forall \mathbf{x} = (x_1, x_2) \in \mathcal{X}$$

Find the squared error of f_S with respect to D.

$$L_D[f_S] = \mathbb{E}_{(x,y)\sim p(X,Y)}[(f(x) - \hat{y})^2] + \mathbb{E}_{(x,y)\sim p(X,Y)}[(\hat{y} - y)^2]$$
$$= \mathbb{E}_{(x,y)\sim p(X,Y)}[(x_1 - 2x_2 - (x_1 - 2x_2 + 2))^2] + \mathbb{E}_{(x,y)\sim p(X,Y)}[(\hat{y} - y)^2] = 5$$

4. Consider a linear model of the form

$$f(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{P} w_i x_i$$

together with a sum of squares error function of the form

$$L_P(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} (f(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n)^2$$

where P is the dimensionality of the vector $\mathbf{x}$, N is the number of training examples, and $\mathbf{t}$ is the ground truth target . Now suppose that the Gaussian noise ϵ_i with zero mean and variance σ^2 is added independently to each of the input variables x_i. By making use of $\mathbb{E}[\epsilon_i] = 0$ and $\mathbb{E}[\epsilon_i \epsilon_j] = \delta_{ij}\sigma^2$, show that minimizing L_P averaged over the noise distribution is equivalent to minimizing the sum of squares error for noise-free input variables L_P with the addition of a weight-decay regularization term, in which the bias parameter w_0 is omitted from the regularizer.

$$f_{noise}(\mathbf{x}_n, \mathbf{w}) = w_0 + \sum_{i=1}^{P} w_i x_i + \sum_{i=1}^{P} w_i \epsilon_i = f(\mathbf{x}_n, \mathbf{w}) + \epsilon$$

where

$$\epsilon \sim \mathcal{N}(\sum_{i=1}^{P} w_i \mu_i = 0, \sum_{i=1}^{P} w_i^2 \sigma^2)$$

$$
\begin{aligned}
\mathbb{E}[\tilde{L}] = \ & \mathbb{E}[\frac{1}{2}\sum_{n=1}^{N}(f_{noise}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n)^2] \\
&= \mathbb{E}[\frac{1}{2}\sum_{n=1}^{N}(f(\mathbf{x}_n, \mathbf{w}) + \epsilon_n - \mathbf{t}_n)^2] \\
&= \frac{1}{2}\mathbb{E}[\sum_{n=1}^{N}(((f(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n) + (\epsilon_n))^2] \\
&= \frac{1}{2}\mathbb{E}[\sum_{n=1}^{N}((f(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n)^2 + \epsilon_n^2 + 2 * (f(\mathbf{x}_n, \mathbf{w}) - t_n) * \epsilon_n)] \\
&= \frac{1}{2}\sum_{n=1}^{N}(\mathbb{E}[(f(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n)^2] + \mathbb{E}[\epsilon_n^2] + \mathbb{E}[2 * (f(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n) * \epsilon_n]) \\
&= \frac{1}{2}\sum_{n=1}^{N}(\mathbb{E}[(f(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n)^2] + \mathbb{E}[\epsilon_n^2] + 2 * \mathbb{E}[(f(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n)] * \mathbb{E}[\epsilon_n]) \\
&= \frac{1}{2}\sum_{n=1}^{N}(\mathbb{E}[(f(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n)^2] + \sum_{i=1}^{P} w_i^2\sigma^2)
\end{aligned}
$$

4 Maximum Likelihood Estimation

1. (a) Suppose that there are blue (B) and red (R) balls in a box and the frequency of blue balls is θ. That is, a random draw from the basket will result in drawing a blue ball with a probability θ and drawing a red ball with a probability $1 - \theta$. Let's say each person draws two balls replacing the ball after each draw. So a person can draw either one of these three combinations (BB, BR, RR). What are the probabilities of each of the outcomes?

$$
\begin{aligned}
P(BB) &= \theta^2 \\
P(BR) &= 2 * \theta * (1 - \theta) \\
P(RR) &= (1 - \theta)^2
\end{aligned}
$$

   (b) Suppose that in the population p_1 people draw BB, p_2 people draw BR and p_3 people draw RR. What is the log likelihood function $LL(P(D/\theta)$ Find the Maximum Likelihood estimate of θ?

$$
\begin{aligned}
LL(P(D/\theta)) &= log(\frac{(p1 + p2 + p3)!}{p1!p2!p3!}\theta^{2p1} * 2 * \theta^{p2} * (1 - \theta)^{p2} * (1 - \theta)^{2p3}) \\
&= log(\frac{(p1 + p2 + p3)!}{p1!p2!p3!}) + (2p1 + p2)log(\theta) + (2p3 + p2)log * (1 - \theta) \\
MLE(\theta) &= \frac{\partial LL(P(D/\theta))}{\partial\theta} \\
&= \frac{2p1 + p2}{\theta} - \frac{2p3 + p2}{1 - \theta} = 0
\end{aligned}
$$

$$
\theta_{MLE} = \frac{2p1 + p2}{2(p1 + p2 + p3)}
$$

   (c) Suppose that out of 100 people, 50 draw BB combination, 10 draw BR combination and 40 draw RR combination. What is the MLE estimate of θ (fraction of blue balls)?

$$\hat{\theta} = \frac{110}{200}$$

2. We have a dataset with N records in which the i^{th} record has one real-valued input attribute x_i and one real-valued output attribute y_i. The model has one unknown parameter w to be learned from data, and the distribution of y_i is given by

$$y_i \sim \mathcal{N}(\log(wx_i), 1)$$

Suppose you decide to do a maximum likelihood estimation of w. What equation does w need to satisfy to be a maximum likelihood estimate?

$$\begin{aligned} LL &= \log(\prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}} exp \frac{-(y_i - log(wx_i))^2}{2}) \\ &= -\frac{N}{2} log(2\pi) - \frac{1}{2} \sum_{i=1}^{N} (y_i - log(wx_i))^2 \\ w_{mle} &= w(\frac{\partial LL(P(D/w))}{\partial w} = 0) \\ &= w(\frac{\partial LL(P(D/w))}{\partial w} = \sum_{i=1}^{N} \frac{(y_i - log(wx_i))}{w} = 0) \end{aligned}$$

$$w = exp \frac{\sum_{i=1}^{N} y_i - log(x_i)}{N}$$

3. Consider a linear basis function regression model for a multivariate target variable $\mathbf{t}$ having a Gaussian distribution of the form

$$p(\mathbf{t}|\mathbf{W}, \mathbf{\Sigma}) = \mathcal{N}(\mathbf{t}|f(\mathbf{x}, \mathbf{W}), \mathbf{\Sigma})$$

where

$$f(\mathbf{x}, \mathbf{W}) = \mathbf{W}^T \phi(\mathbf{x})$$

together with a training data set comprising input basis vectors $\phi(\mathbf{x}_n)$ and corresponding target vectors $\mathbf{t}_n$ with $n = 1, 2, \ldots, N$. Show that the maximum likelihood solution $\mathbf{W}^*$ for the parameter matrix $\mathbf{W}$ has the property that each column is given by the solution to a univariate target variable. Note that this is independent of the covariance matrix $\mathbf{\Sigma}$.

Also, give the maximum likelihood solution for $\mathbf{\Sigma}$ – feel free to use standard results for the MLE

solution of Σ in your answer.

$$
\begin{aligned}
LL &= \log(\prod_{i=1}^{N} \frac{1}{\sqrt{2\pi^p \Sigma}} exp \frac{-(t_n - w^T\phi(x_n))^T \Sigma^{-1}(t_n - w^T\phi(x_n))}{2}) \\
&= -\frac{N}{2} log(2\pi) + \frac{N}{2} log(\Sigma^{-1}) - \Sigma^{-1} * \frac{1}{2} \sum_{i=1}^{N} (t_n - w^T\phi(x_n))^2 \\
\mathbf{W}^* &= W(\frac{\nabla \partial LL(P)}{\partial w} = 0 \\
&= \sum_{n=1}^{N} (t_n - w^T\phi(x_n)) * \phi(x)^T
\end{aligned}
$$

$$
\begin{aligned}
0 &= \sum_{n=1}^{N} t_n \phi(x_n)^T - w^T (\sum_{n=1}^{N} \phi(x_n)\phi(x_n)^T) \\
w_{ML} &= (\Phi^T\Phi)^{-1}\Phi^T t
\end{aligned}
$$

where

$$\Phi_{np} = \phi_p(x_n)$$

$$
\begin{aligned}
\mathbf{\Sigma} &= \Sigma(\frac{\partial LL(P)}{\partial \Sigma} = 0) \\
&= \frac{N}{2}(\frac{\partial log(\Sigma^{-1})}{\partial \Sigma}) - \frac{1}{2} \sum_{n=1}^{N} (t_n - w_{ML}^T \phi(x_n))^2 \\
\mathbf{\Sigma} &= \frac{\sum_{n=1}^{N} (t_n - w_{ML}^T \phi(x_n))^2}{N}
\end{aligned}
$$