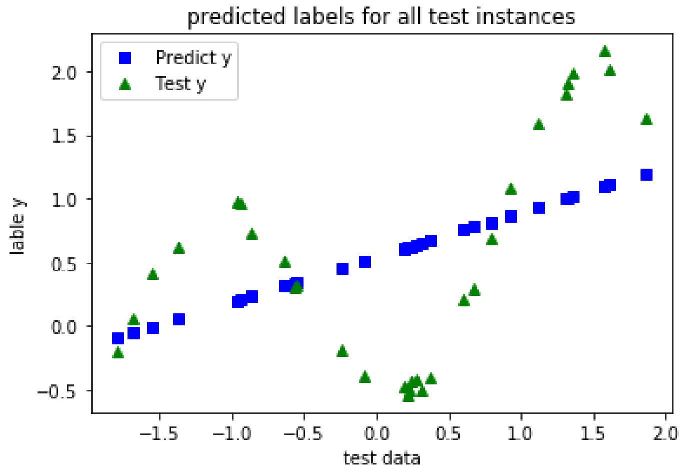
EECS 545 Homework 2

Di Lu (qfdlu)   Feb 12, 2018

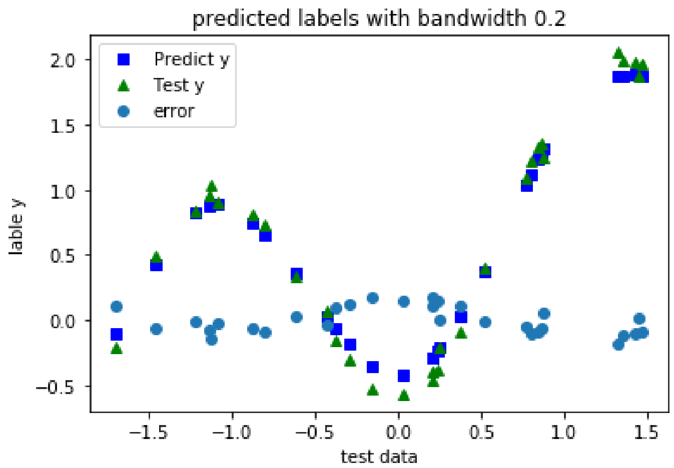
1. Locally weighted linear regression

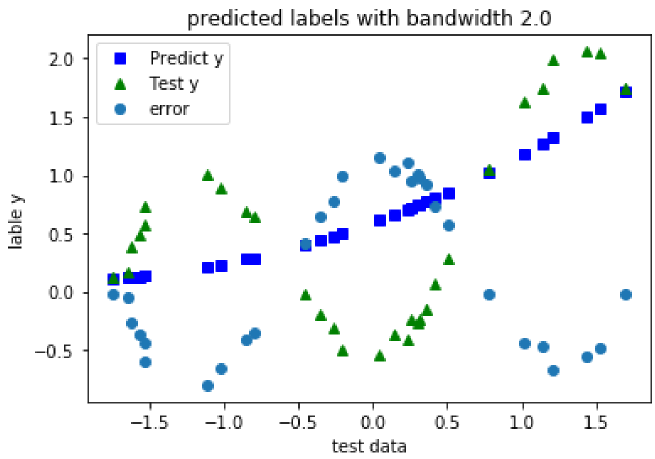
(a) Normalize the features in training set with the mean and variance of training set, also normalize the features in test set with the mean and variance of training set and then add bias term to the featurs.

Use the closed form solution for the linear regression model, we fit the blue line(below) to test set, and get the test error, which is mean squared error, to be 0.573435.



(b) Fit a locally weighted linear regression model to the training set with a Gaussian kernel with kernel width tau. At tau = 0.2, we get the fit on the left, at tau = 2.0, we get the fit on the right. At these two case, the test error are 0.010222 and 0.466592, respectively.



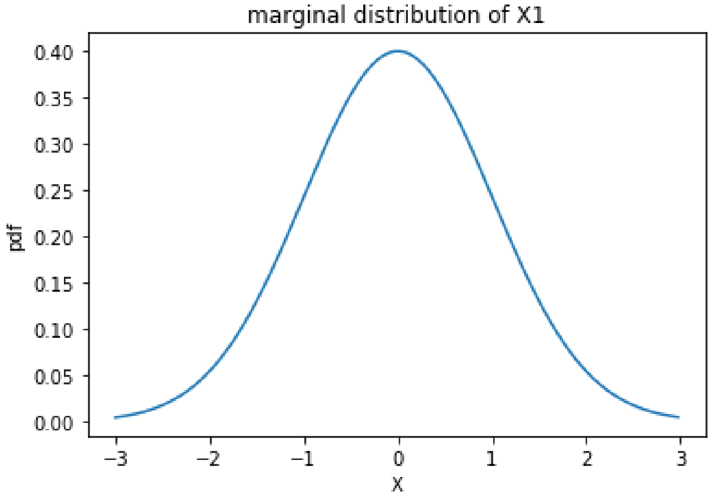


2. Plot marginal distributions and conditional distributions for multivariate Gaussians.

(a) Complete the function marginal_distribution
As in the code prob2.py.

(b) X1 will follow normal distribution with mean = 0, variance = 1, which has pdf

$$f(x1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x1^2}{2}}$$

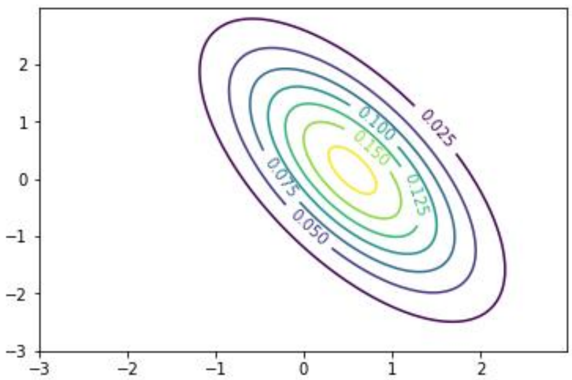


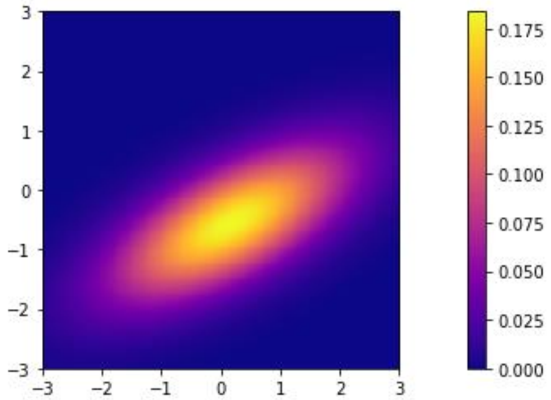
(c) Complete the function conditional_distribution.
As in the code prob2.py.

(d) P(X1;X4;X2 = 0:1;X3 = -0.2) will follow joint gaussian distribution with

$$\mu(X1, X4) = [0.55, 0.15]^T$$

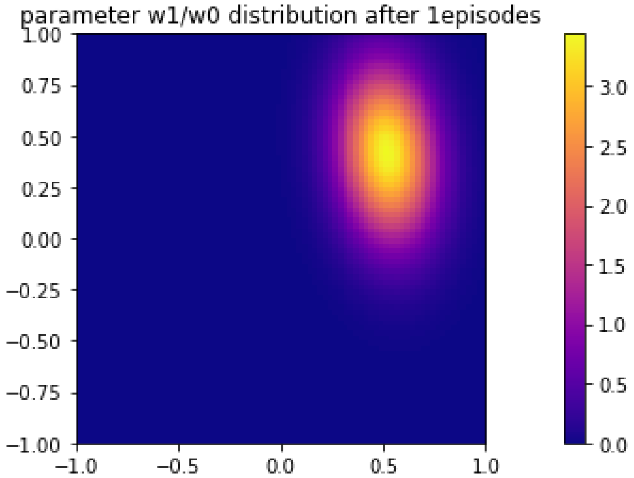
$$\Sigma(X1,\ X4) = \begin{bmatrix} 0.75 & -0.75 \\ -0.75 & 1.75 \end{bmatrix}$$

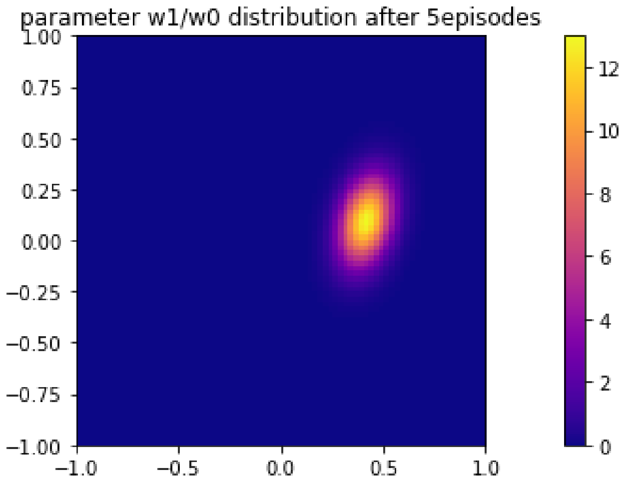


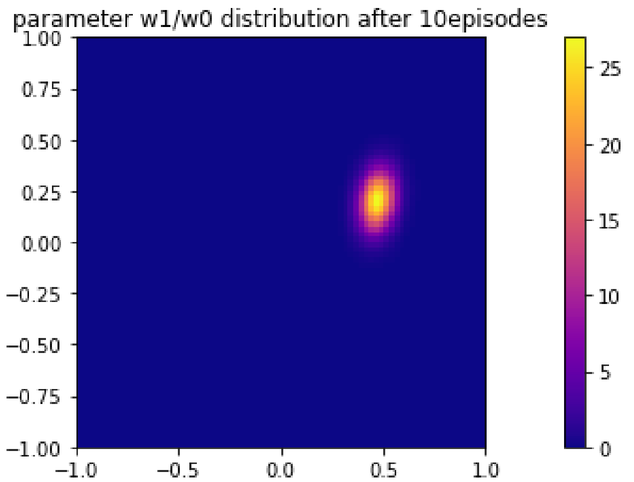


3. Sequential Bayesian linear regression algorithm

Run the algorithm in prob3.py and plot the initial distribution of w and the distributions after 1, 5, 10 instances as below: As number of iteration increase, we can see the learned parameter is getting closer to w1 = 0:5, w0 = - 0:3, and variance is getting very small. At 10 instances, the w = [0.48394096 -0.25854919]







4. Gaussian process (GP) regression model

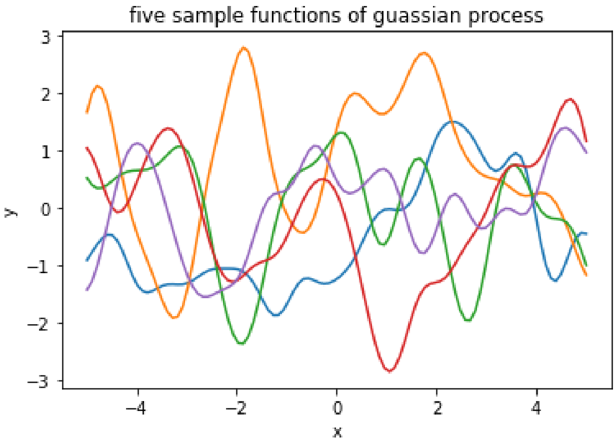
(a)

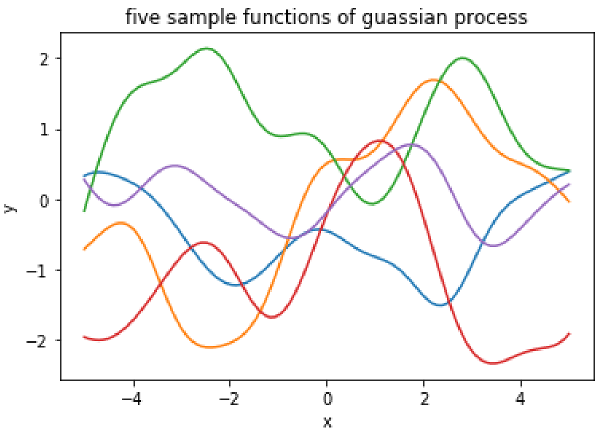
(i) $[y(x1);\ \ y(x2), \ldots, y(xn)]^T$ is a Gaussian vector with

Mean = $[0,0, \ldots, 0]^T$,

$$Cov(y_i, y_j) = k(x_i - x_j) = exp(-\frac{(x_i - x_j)^2}{2\sigma^2}).$$

(ii) With sigma = 0.3, 0.5, 1.0(respectively), the sample from GP are as follows:





(b)

(i) From x1, …, x100 in (-5,5), and x101 = -1.3, x102 = 2.4, x103 = -2.5, x104 = -3.3, x105 = 0.3, they follow joint Gaussian distribution.

They have mean: $\mu = (\mu_a,\ \mu_b)$

Covariance: k = $\begin{bmatrix} \sum aa & \sum ab \\ \sum ba & \sum bb \end{bmatrix}$

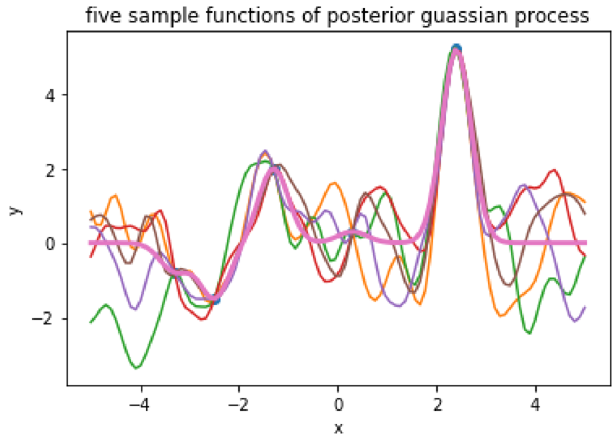
Then we can get the conditional distribution of a = (x1, …, x100) as follows:

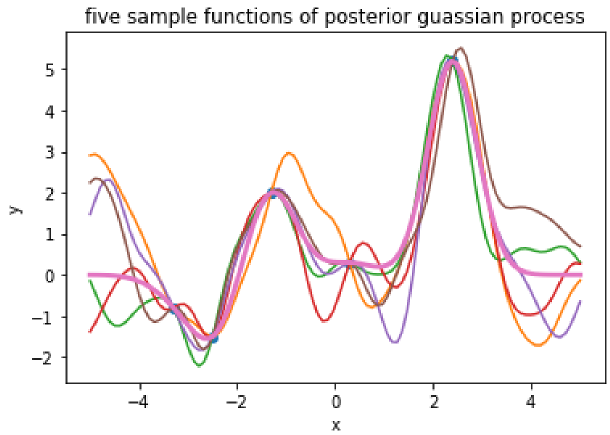
$$\mu_{a|b} = \mu_a + \sum ab \sum bb^{-1} (X_b - \mu_b)$$

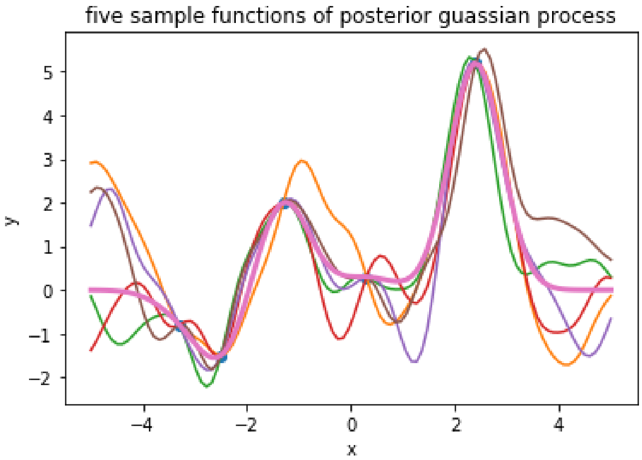
$$\sum a|b = \sum aa - \sum ab \sum bb^{-1} \sum ba$$

Where, $X_b = (2{,}2, -1.5, -0.8{,}0.3)$

(ii) With sigma = 0.3, 0.5, 1.0(respectively), the sample from posterior GP are as follows:





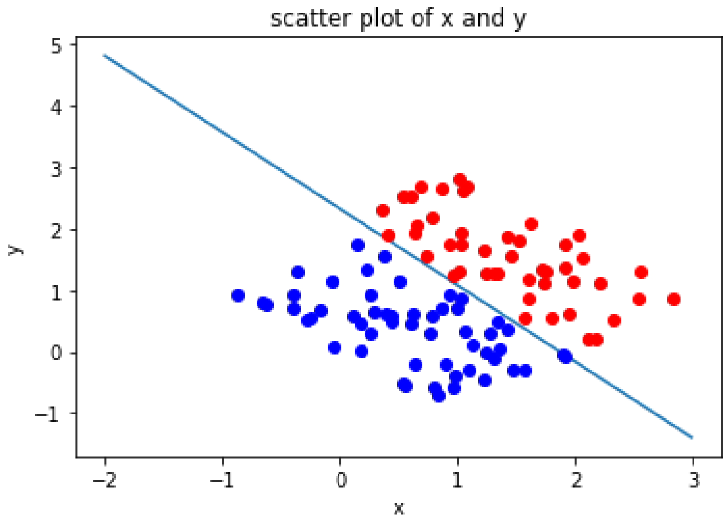

five sample functions of posterior guassian process
y
x
5
4
3
2
1
0
−1
−2
−4
−2
0
2
4

5. Perceptron Algorithm

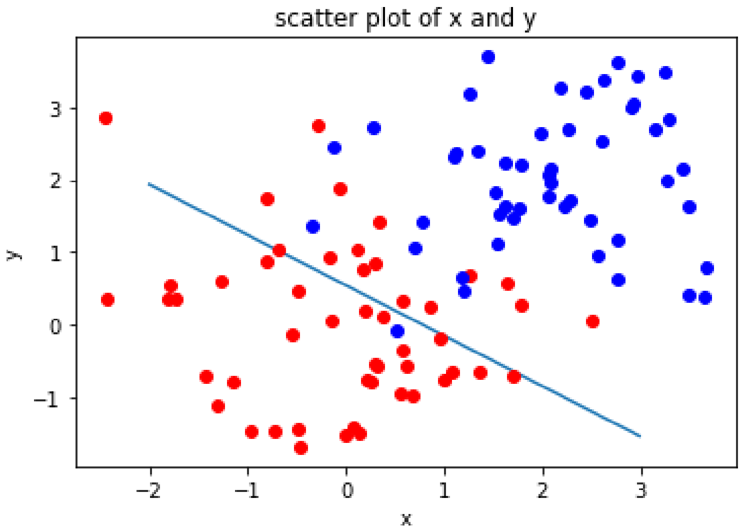
When dataset is linearly separatable, I initialized w vector to zeros, and iterate through the dataset 10 times, each time updating the w vector for each datum via the perceptron update rule.

The learned w vector is [1.6039718, 1.28827177, -3.   ],

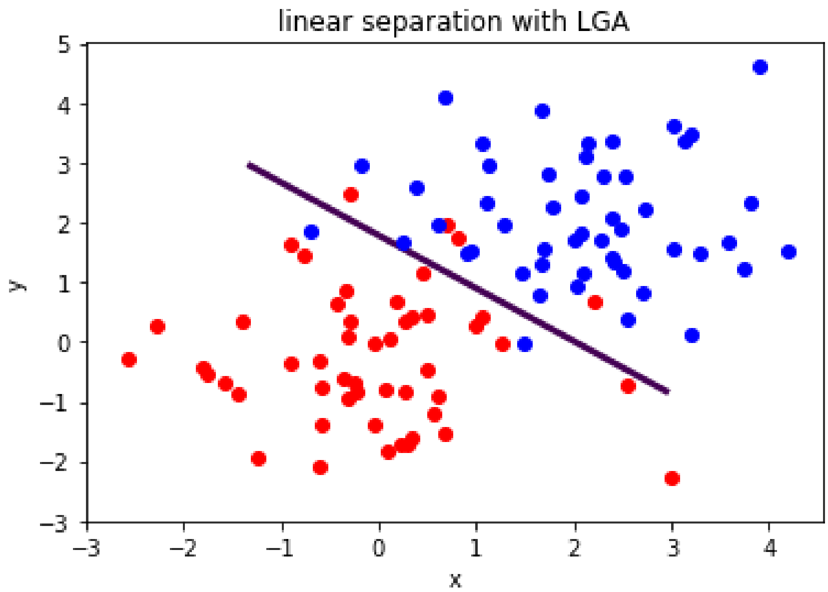
$w^T \ x \ = 0$ is the separation line.



When dataset is not linearly separatable, the perceptron algorithm does not give a good separation.



We apply linear descrimitive analysis with two Gaussian distribution with shared covariance matrix and get linear separation as below.



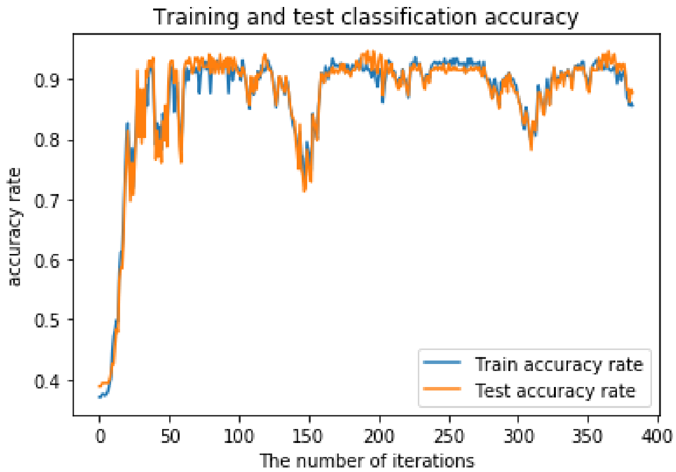
linear separation with LGA
5
4
3
2
1
0
−1
−2
−3
y
−3
−2
−1
0
1
2
3
4
x

6. logistic regression

I initialized each element of w randomly between -1 and 1, and perform stochastic gradient descent method with a learning rate of 10^(-2). After iterating through each example in the training set once in random order, we get plot of

(a) training and test classification accuracy vs. the number of iterations of SGD

(b) training and test error vs. the number of iterations of SGD



(c) learned parameter vector

w = [-1.515935965420990572e-01
4.107699853156790426e-02
3.783916245810290957e-01
-7.215346610076075473e-01
7.299575817125587562e-01
-9.050310253973313790e-01
-8.881645222182247279e-01
-1.686345700035273842e-01
2.317169432194860768e-01
9.673367247019082696e-03
-1.084607524947478918e+00
-3.921934673228703039e-01
5.409435455708458962e-01
5.837463290333984034e-01
-6.052481436847230167e-01
2.777280269285515302e-01
2.140674046423806098e-01
6.748762787015731845e-01
-8.655680275434579629e-01
3.112423201080868895e-01

3.753622069952970014e-02
-8.964038989213080066e-01
9.644134700052998943e-02
2.949421874677267408e-02
-8.237111752605961179e-01
-1.963869096539768988e-01
-6.949073289199492187e-01
-5.180620488723818307e-01
-9.304696038741960828e-01
7.695582997153413984e-01]

(d) final training and test cross-entropy: -172.0611324916054, -84.25563694121533

(e) final training and test classification accuracy: 0.855643044619, 0.877659574468