**Programming Exercise 8:Anomaly Detection and Recommender Systems第一部分**

2016年03月17日 20:29:36

大家好，我是Mac Jiang,今天和大家分享Coursera-Stanford University-Machine Learning-Programming Exercise 8:Anomaly Detection and Recommender System的第一部分：Anomaly Detection。第二部分Recommender System的代码我将会在接下来的博客中给出。我的代码虽然是通过了系统的测试，但不一定是最好的，如果博友有更好的想法，请留言联系，谢谢！希望我的博客可以为您带来一些学习上的帮助！

这部分的主要内容是Anomaly Detection，即异常点检测。异常点检测的作用是找出那些明显偏离的样本，在生产中他能帮助找到不合格的产品，在生活中他可以帮助检测用户的银行卡是否被盗刷，在机器学习中有着广泛的应用。异常点检测的一般算法是利用高斯分布函数实现的，利用已有样本训练适合的高斯分布函数（可高维），并设定阈值epsilon，如果样本p(x)小于epsilon则认为它是异常点。

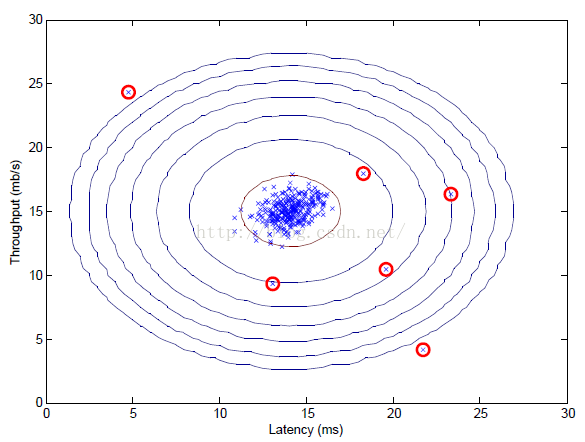
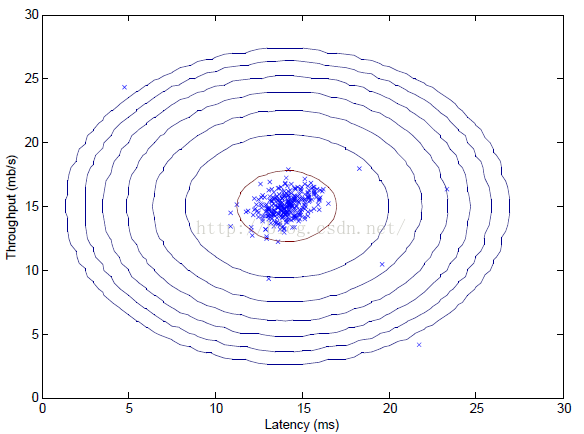
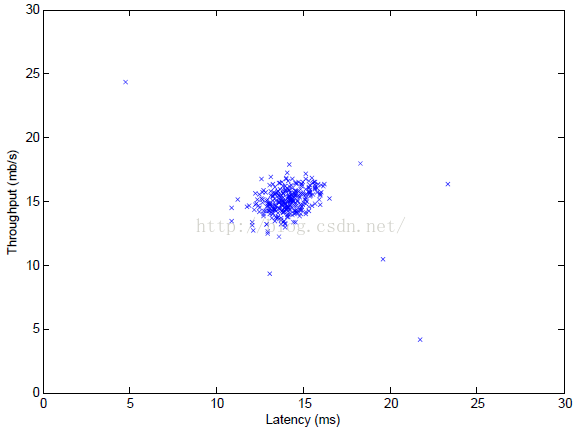
由于正常点的样本占绝大多数，只有极少数的异常样本，所以这是一个偏斜类。对于偏斜类好坏的评价，我们采用F1 score，因此我们利用F1 score决定到底选取那个epsilon。

这次实验是对一个二维数据集进行异常检测。具体过程为：

1.首先输入二维数据，建立高斯分布函数拟合这些样本。

2.由于我们不知道具体阈值epsilon的最优取值，我们取多个epsilon的值，分别计算他的F1 score并比较，取最大的F1 score对应的epsilon为最优阈值。

3.根据得到的epsilon，标出样本中p(x)小于epsilon的点，即为异常点。



**1.数据集和文件说明**

数据集：

ex8data1.mat---即上面所述的二位数据集合，二位数据可以可视化

  ex8data2.mat----用于异常检测的多维数据集合，实际上它的维度为11维，不可以可视化

文件：

ex8.m---控制异常检测进行的过程的控制文件

  multivariateGuassian.mat---利用已有的均值mu和方差sigma2建立多位高斯分布函数

  estimateGusssian.m---对于输入的样本X，分别对每一维计算均值mu和方差sigma2，保存在向量中，需要完善代码！

  selectThreshold.m---对于多个输入epsilon分别计算F1 score，去最大F1 score对应的epsilon为最优解，需要完善代码！

**2.ex8.m过程说明**

1. %% Initialization
2. clear ; close all; clc
3. %% ================== Part 1: Load Example Dataset ===================
4. % We start this exercise by using a small dataset that is easy to
5. % visualize.
6. %
7. % Our example case consists of 2 network server statistics across
8. % several machines: the latency and throughput of each machine.
9. % This exercise will help us find possibly faulty (or very fast) machines.
10. %
11. fprintf('Visualizing example dataset for outlier detection.\n\n');
12. % The following command loads the dataset. You should now have the
13. % variables X, Xval, yval in your environment
14. load('ex8data1.mat');
16. % Visualize the example dataset
17. plot(X(:, 1), X(:, 2), 'bx');
18. axis([0 30 0 30]);
19. xlabel('Latency (ms)');
20. ylabel('Throughput (mb/s)');
21. fprintf('Program paused. Press enter to continue.\n');
22. pause
24. %% ================== Part 2: Estimate the dataset statistics ===================
25. % For this exercise, we assume a Gaussian distribution for the dataset.
26. %
27. % We first estimate the parameters of our assumed Gaussian distribution,
28. % then compute the probabilities for each of the points and then visualize
29. % both the overall distribution and where each of the points falls in
30. % terms of that distribution.
31. %
32. fprintf('Visualizing Gaussian fit.\n\n');
33. % Estimate my and sigma2
34. [mu sigma2] = estimateGaussian(X);
35. % Returns the density of the multivariate normal at each data point (row)
36. % of X
37. p = multivariateGaussian(X, mu, sigma2);
38. % Visualize the fit
39. visualizeFit(X, mu, sigma2);
40. xlabel('Latency (ms)');
41. ylabel('Throughput (mb/s)');
42. fprintf('Program paused. Press enter to continue.\n');
43. pause;
45. %% ================== Part 3: Find Outliers ===================
46. % Now you will find a good epsilon threshold using a cross-validation set
47. % probabilities given the estimated Gaussian distribution
48. %
49. pval = multivariateGaussian(Xval, mu, sigma2);
50. [epsilon F1] = selectThreshold(yval, pval);
51. fprintf('Best epsilon found using cross-validation: %e\n', epsilon);
52. fprintf('Best F1 on Cross Validation Set: %f\n', F1);
53. fprintf(' (you should see a value epsilon of about 8.99e-05)\n\n');
54. % Find the outliers in the training set and plot the
55. outliers = find(p < epsilon);
56. % Draw a red circle around those outliers
57. hold on
58. plot(X(outliers, 1), X(outliers, 2), 'ro', 'LineWidth', 2, 'MarkerSize', 10);
59. hold off
61. fprintf('Program paused. Press enter to continue.\n');
62. pause;
64. %% ================== Part 4: Multidimensional Outliers ===================
65. % We will now use the code from the previous part and apply it to a
66. % harder problem in which more features describe each datapoint and only
67. % some features indicate whether a point is an outlier.
68. %
69. % Loads the second dataset. You should now have the
70. % variables X, Xval, yval in your environment
71. load('ex8data2.mat');
72. % Apply the same steps to the larger dataset
73. [mu sigma2] = estimateGaussian(X);
74. % Training set
75. p = multivariateGaussian(X, mu, sigma2);
76. % Cross-validation set
77. pval = multivariateGaussian(Xval, mu, sigma2);
78. % Find the best threshold
79. [epsilon F1] = selectThreshold(yval, pval);
80. fprintf('Best epsilon found using cross-validation: %e\n', epsilon);
81. fprintf('Best F1 on Cross Validation Set: %f\n', F1);
82. fprintf('# Outliers found: %d\n', sum(p < epsilon));
83. fprintf(' (you should see a value epsilon of about 1.38e-18)\n\n');
84. pause

Part1:Load Example Dataset---导入数据并可视化

Part2:Esitimate the dataset statistics---调用estimateGuassian.m对每一维度分别计算均值mu(i),方差sigma2(i),之后利用得到数据建立多位高斯分布，并调用selectThreshold.m计算最有epsilon

Part3:Find Outliers---利用上面训练得到的高斯分布和epsilon值找到那些异常点

Part4:Multidimensional Outliers---调用ex8data2.mat中的数据，这是11维的数据，对这个数据进行异常检测，得到异常点。注意：11维的数据不能可视化

**2.estimateGaussian.m实现过程**

1. function [mu sigma2] = estimateGaussian(X)
2. %ESTIMATEGAUSSIAN This function estimates the parameters of a
3. %Gaussian distribution using the data in X
4. % [mu sigma2] = estimateGaussian(X),
5. % The input X is the dataset with each n-dimensional data point in one row
6. % The output is an n-dimensional vector mu, the mean of the data set
7. % and the variances sigma^2, an n x 1 vector
8. %
9. % Useful variables
10. [m, n] = size(X);
11. % You should return these values correctly
12. mu = zeros(n, 1);
13. sigma2 = zeros(n, 1);
14. % ====================== YOUR CODE HERE ======================
15. % Instructions: Compute the mean of the data and the variances
16. % In particular, mu(i) should contain the mean of
17. % the data for the i-th feature and sigma2(i)
18. % should contain variance of the i-th feature.
19. %
20. %这里当然也可以采取for循环的方式计算，但是向量法的计算速度快一些
21. mu = sum(X)' / m; %sum是对X按列求和的意思
22. temp = X' - repmat(mu,1,m); %repmat(A,m,n)，把矩阵A复制m\*n份，行为m，列为n
23. %这里也可用temp = bsxfun(@minus, X', mu)
24. sigma2 = sum(temp.^2,2) / m; %sum(A,2)对矩阵按列求和

注意：可能有些同学会用for循环的方法编写代码，但是一般来说利用向量的方法会比循环的方法快很多，所以如果能利用向量方法的最好利用向量

**3.selectThreshold.m实现过程**

1. function [bestEpsilon bestF1] = selectThreshold(yval, pval)
2. %SELECTTHRESHOLD Find the best threshold (epsilon) to use for selecting
3. %outliers
4. % [bestEpsilon bestF1] = SELECTTHRESHOLD(yval, pval) finds the best
5. % threshold to use for selecting outliers based on the results from a
6. % validation set (pval) and the ground truth (yval).
7. %
8. bestEpsilon = 0;
9. bestF1 = 0;
10. F1 = 0;
11. stepsize = (max(pval) - min(pval)) / 1000;
12. for epsilon = min(pval):stepsize:max(pval)
14. % ====================== YOUR CODE HERE ======================
15. % Instructions: Compute the F1 score of choosing epsilon as the
16. % threshold and place the value in F1. The code at the
17. % end of the loop will compare the F1 score for this
18. % choice of epsilon and set it to be the best epsilon if
19. % it is better than the current choice of epsilon.
20. %
21. % Note: You can use predictions = (pval < epsilon) to get a binary vector
22. % of 0's and 1's of the outlier predictions
23. %yval表示的是它是不是异常点，是为1不是为0。
24. %pval表示的是利用已经训练出的系统对xval计算p(x)的值，若小于ipsilon我们就认为它是异常点
25. %tp:true possitive，即它实际为1我们又预测它为1的样本数
26. %fp：false possitive即他实际为0我们预测他为1的样本数
27. %fn:false negative,即它实际为1我们预测他为0的样本数
28. %prec = tp/（tp+fn）查准率；rec = tp/（tp+fn）回收率；F1 srcore = 2\*prec\*rec/(prec+rec)
29. cvPrediction = pval < epsilon; %yval小于ipsilon则置1否则置0
30. tp = sum((cvPrediction == 1) & (yval == 1)); %cvPrediction为我们的预测值，yval为实际值
31. fp = sum((cvPrediction == 1) & (yval == 0));
32. fn = sum((cvPrediction == 0) & (yval == 1));
33. prec = tp / (tp + fp);
34. rec = tp / (tp + fn);
35. F1 = 2 \* prec \* rec / (prec + rec);
36. % =============================================================
37. if F1 > bestF1
38. bestF1 = F1;
39. bestEpsilon = epsilon;
40. end
41. end
42. end

http://blog.csdn.net/a1015553840/article/details/50913824