17.5.8（异常检测实现）

异常检测

$$p(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

高斯概率函数

You can estimate the parameters, (μ_i, σ_i^2), of the i-th feature by using the following equations. To estimate the mean, you will use:

$$\mu_i = \frac{1}{m}\sum_{j=1}^{m} x_i^{(j)}, \qquad (1)$$

and for the variance you will use:

$$\sigma_i^2 = \frac{1}{m}\sum_{j=1}^{m} (x_i^{(j)} - \mu_i)^2. \qquad (2)$$

mean求均值，var求方差，matlab中var函数求方差除的是（m-1）要把它变成除以（m）。
代码实现如下：

```
mu = mu + mean(X)';

sigma2 = sigma2 + (var(X)*(m-1)/m)';
```

求得高斯概率，就可以和阀值进行比较判断异常与否。下面求得最好的阀值，

```
stepsize = (max(pval) - min(pval)) / 1000;
for epsilon = min(pval):stepsize:max(pval)
```

最大预测概率减去最小预测概率分成一千份，用for循环一千次，选择F1分数最大的作为最好的阀值。

The F_1 score is computed using precision ($prec$) and recall (rec):

$$F_1 = \frac{2 \cdot prec \cdot rec}{prec + rec}, \qquad (3)$$

You compute precision and recall by:

$$prec = \frac{tp}{tp+fp} \qquad (4)$$

$$rec = \frac{tp}{tp+fn}, \qquad (5)$$

上面是求F1 score的思路，TP 就是true positives的个数，也就是猜测正样本，实际正样本的个数，FP 就是 false positives的个数，也就是猜测正样本，实际负样本的个数，FN 就是 false negatives 的个数，也就是猜测负样本，实际正样本的个数。PREC 就是查准率，REC 就是查全率。具体代码实现如下：

```
pred = (pval < epsilon);
fp = sum((pred == 1) & (yval == 0));
tp = sum((pred == 1) & (yval == 1));
fn = sum((pred == 0) & (yval == 1));

prec = tp/(tp+fp);
rec = tp/(tp+fn);
F1 = 2*prec*rec/(prec+rec);
```

以往的分类算法：一个X对应一个Y，推荐系统是一个X对应多个Y，这时的X特征和theta都是需要学习得到的，所以优化目标就是同时和X与theta 有关，这时就需要用到协同过滤算法，先初始化X特征，然后用梯度下降优化得到theta ，然后用得到的theta优化X，反复如始，最小化cost
推荐系统的cost function 和 gradient
cost function公式：

$$J(x^{(1)},\dots,x^{(n_m)},\theta^{(1)},\dots,\theta^{(n_u)}) = \frac{1}{2}\sum_{(i,j):r(i,j)=1}((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \left(\frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}(\theta_k^{(j)})^2\right) + \left(\frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}(x_k^{(i)})^2\right).$$

代码实现：

```
h = (X*Theta'-Y).^2;
J = J + sum(h(R==1))/2 + sum(sum(X.^2))*lambda/2 + sum(sum(Theta.^2))*lambda/2;
```

gradient 公式：

$$\frac{\partial J}{\partial x_k^{(i)}} = \sum_{j:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})\theta_k^{(j)} + \lambda x_k^{(i)}$$

$$\frac{\partial J}{\partial \theta_k^{(j)}} = \sum_{i:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})x_k^{(i)} + \lambda \theta_k^{(j)}.$$

代码实现：

```
te = X*Theta'-Y;
te(R==0) = 0;
X_grad = X_grad + te*Theta + lambda*X;
Theta_grad = Theta_grad + te'*X + lambda*Theta;
```