计算机应用数学实验报告

徐进(11621002)

May 30, 2016

1 实验题目

1.1 Curve Fitting

Implement polynomial curve fitting in python.

1. sample the function curve of y=sin(x) with Gaussian noise
2. fit degree 3 and 9 curves in 10 samples
3. fit degree 9 curves in 15 and 100 samples
4. fit degree 9 curve in 10 samples but with regularization term

1.2 PCA

Represent digits '3' in 2D.

1. convert data from the UCI
2. perform PCA over all digit '3' with 2 components
3. plot the PCA results

1.3 2D MOG

implement MOG in 2D case.

1. Generate 2D Gaussian distribution
2. E-M method

1.4 L-M Algorithm

Implement the Levenberg-Marquardt method

1.5 2D SVM

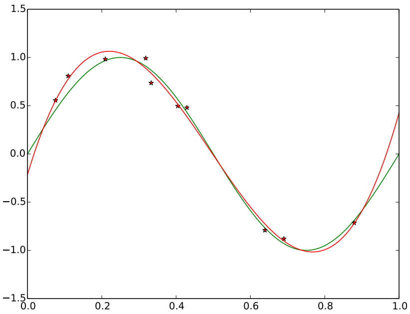
Implement (simplified) SVM method.

1. input 2D data and their label (in two classes)
2. implement quadratic programming
3. output classification results

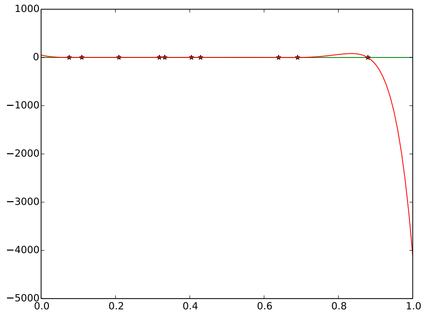
2 算法及实验结果

2.1 Curve Fitting

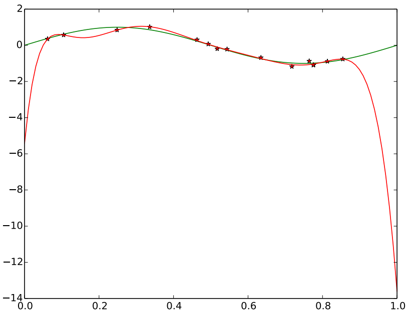
随机产生x的值，然后根据x值对y=sin(x)进行采样，得到的值再加上方差为0.1的高斯噪声，得到的结果作为样本。



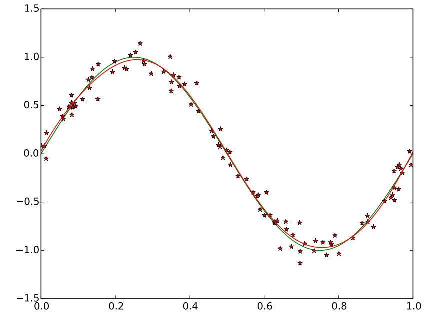
(a) 3次多项式对10个样本进行拟合



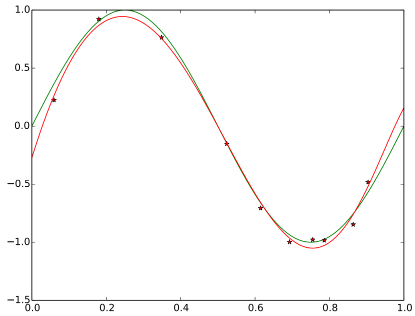
(b) 9次多项式对10个样本进行拟合



(c) 9次多项式分别对15个样本进行拟合



(d) 9次多项式分别对100个样本进行拟合



(e) 9次多项式（含正则项）对10个样本进行拟合

Figure 1: 对函数y=sin(x)进行多项式拟合

2.2 PCA

使用的数据是数字的训练样本，选择数字3的训练样本。每一个样本作为一个列向量，所有的样本组成一个矩阵，对该矩阵进行SVD分解得到U，S，V矩阵，取V矩阵前两行分别作为x,y坐标，画出散点图，如Figure 2(a)所示。只取S矩阵的前两个特征向量，对数据进行还原，得到的结果如Figure 2(b)所示。

2.3 2D MOG

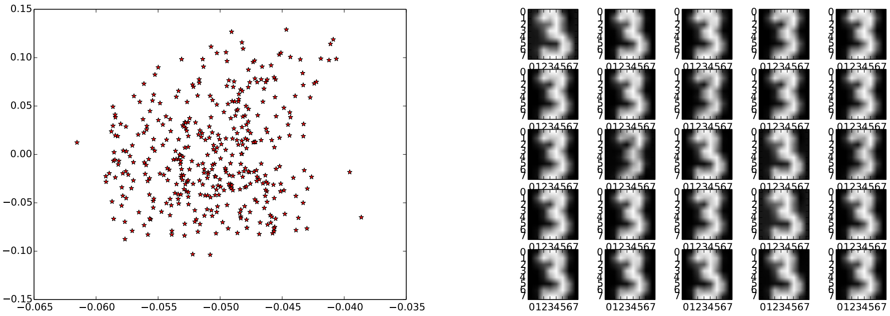
产生2D高斯分布：分别产生坐标x,y的高斯分布，然后使用$x = \Sigma^{\frac{1}{2}} + \mu$生成指定的2D高斯分布。

EM算法：

E-step:

根据样本估算$\hat{p}(j|i)$

$$\hat{p}(j|i) \leftarrow P(y_i = j|\boldsymbol{x_i}, \theta) forall j = 1, 2 and i = 1, ..., n \tag{2.1}$$



(a) PCA降维后的散点图　　(b) 只取2维对数据进行还原后的效果

Figure 2: PCA

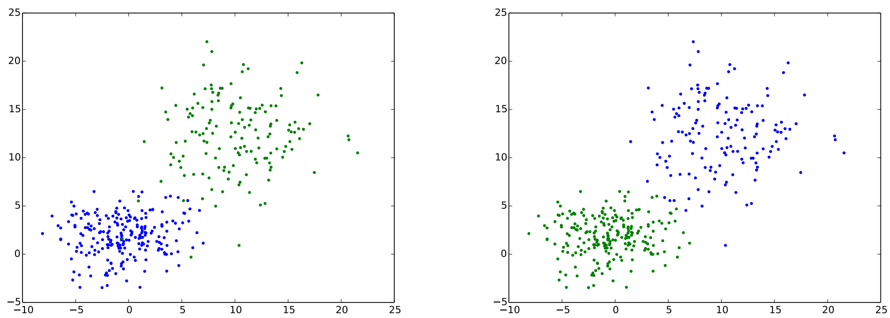
$$P(y_i = k|\boldsymbol{x_i}, \theta) = \frac{P(y=k)\cdot p(\boldsymbol{x}|y=k)}{\Sigma_{j=1,..,n} P(y=j)\cdot p(\boldsymbol{x}|y=j)} \tag{2.2}$$

M-step:

根据$\hat{p}(j|i)$ 重新计算2D高斯分布的参数

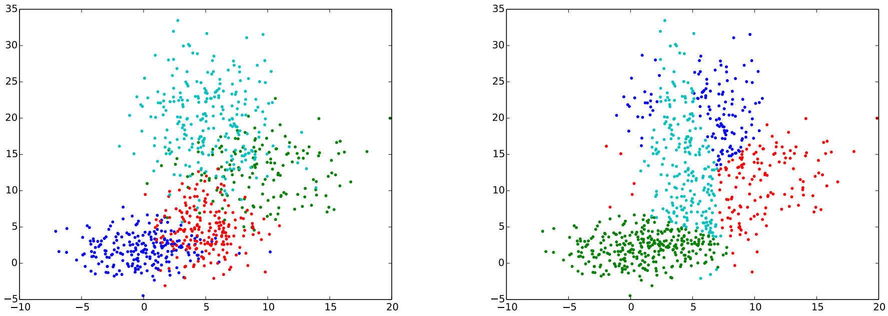
$$\begin{aligned}
\hat{n}_j &\leftarrow \sum_{i=1}^{n} \hat{p}(j|i) \\
\hat{p}_j &\leftarrow frac\hat{n}_j n \\
\hat{\mu}_j &\leftarrow frac1\hat{n}_j \sum_{i=1}^{n} \hat{p}(j|i)\boldsymbol{x}_i \\
\hat{\Sigma}_j &\leftarrow frac1\hat{n}_j \sum_{i=1}^{n} \hat{p}(j|i)(\boldsymbol{x}_i - \hat{\mu}_j)(\boldsymbol{x}_i - \hat{\mu}_j)^T
\end{aligned} \tag{2.3}$$

重复迭代n次，得到最终的2D高斯分布。然后根据$\hat{p}(j|i)$对样本点进行分类。Figure 3 (a)是2个2D高斯分布,$\mu_1 = [0,2], \mu_2 = [10,13], \Sigma_1 = [3,2], \Sigma_2 = [4,4]$，将这些散点图作为输入，根据EM算法得到结果如Figure 3 (b)所示。$\mu_1 = [0.13, 1.90], \mu_2 = [10.41, 13.14], \Sigma_1 = [2.77, 1.98], \Sigma_2 = [3.98, 4.17], p_1 = 0.41, p_2 = 0.59$。Figure 4 (a)是4 个2D 高斯分布，$\mu_1 = [0,2], \mu_2 = [10,13], \mu_3 = [5,5], \mu_4 = [5,20], \Sigma_1 = [3,2], \Sigma_2 = [4,4], \Sigma_3 = [2,3], \Sigma_4 = [3,5]$将这些散点图作为输入，根据EM 算法得到结果如Figure 4 (b) 所示。$\mu_1 = [0.82, 2.54], \mu_2 = [8.86, 12.88], \mu_3 = [4.79, 14.44], \mu_4 = [5.50, 14.96], \Sigma_1 = [3.24, 2.01], \Sigma_2 = [5.23, 5.79], \Sigma_3 = [2.03, 8.61], \Sigma_4 = [3.56, 8.10], p_1 = 0.32, p_2 = 0.23, p_3 = 0.22, p_4 = 0.23$。



(a) 原2D高斯分布　　(b) EM聚类后的结果

Figure 3: EM算法(2个2D高斯分布)



(a) 原2D高斯分布 (b) EM聚类后的结果

Figure 4: EM算法(4个2D高斯分布)

2.4 L-M Algorithm

1. 给定初始点$x_0, \mu_0, k=1$
2. 计算g_k和G_k
3. 若$\|g_k\| < \varepsilon$，停止。
4. 分解$G_k + \mu_k I$,若不正定，置$u_k = 4u_k$,重复4直到正定。
5. 解$(G_k + \mu_k I)s = -g_k$,求出s_k
6. 求$f(x_k + s_k), q^{(k)}$和$r_k = \frac{\Delta f_k}{\Delta q^{(k)}}$
7. 若$r_k < 0.25$,置$u_{k+1} = 4u_k$;若$r_k > 0.75$,置$u_{k+1} = u_k/2$;否则，$u_{k+1} = u_k$
8. 若$r_k < 0$,置$x_{k+1} = x_k$;否则,置$x_{k+1} = x_k + s_k$
9. 令$k = k+1$,转2。

对下面的函数使用LM算法求最小值：

$f = (1 - x - 1)^2 + 105 * (x_2 - x_1^2)^2$**:**
$x_1 = 1.00, x_2 = 1.00$的时候取到最小值$f = 5.98 * 10^{-13}$。

$f = e^{x_1^2 + x_2}$**:**
$x_1 = 2.86, x_2 = -20.15$的时候取到最小值$f = 6.45 * 10^{-6}$。

$f = -2 * e^{-x} * sin(x)$**:**
$x = 0.79$的时候取到最小值$f = -0.64$。

2.5 2D SVM

2D SVM我做的例子比较简单，只有两类。

$$y_i = \begin{cases} 1 & \text{如果}x_i\text{在第一类;} \\ -1 & \text{如果}x_i\text{在第二类.} \end{cases} \tag{2.4}$$

SVM分类就相当于求一个超平面（在2D的情况下就是一条直线或曲线）使得这个超平面把样本点分割开，并且两边到这个超平面的距离最远。在2D情况下，线性SVM的超平面就是一条直线，定义为$\omega^T x + b = 0$,对于样本点满足以下公式:

$$\begin{array}{ll} (\omega^T x_i) + b > 0 & y_i = 1 \\ (\omega^T x_i) + b < 0 & y_i = -1 \end{array} \tag{2.5}$$

需要求ω, b使得$\omega^T x + b = \pm 1$这两条直线间的距离最远。这样就变成了一个规划问题

$$min_{\omega,b} \quad \frac{1}{2}\omega^T \omega \tag{2.6}$$

$$y_i((\omega^T x_i) + b) \geq 1, \quad i = 1, ..., l. \tag{2.7}$$

使用sklearn中datasets里的数据iris，取前2维属性和前2类数据。然后使用线性2D SVM对数据进行分类，得到的结果如Figure 5所示。

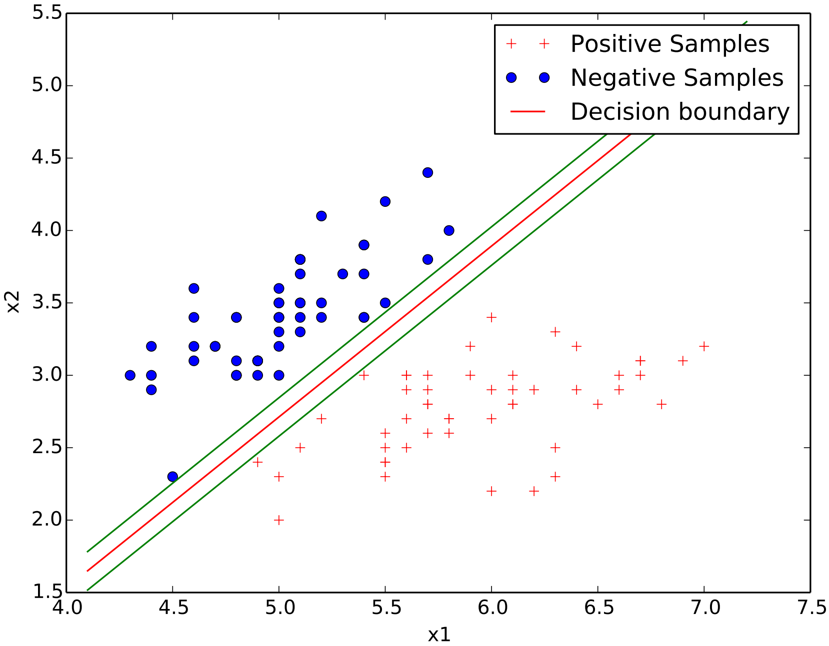


Figure 5: 线性2D SVM