计算机应用数学课程作业

华俊豪 (11331007)

2014 年 5 月 12 日

1　多项式拟合 (Curve Fitting)

1) 生成带高斯噪声的 $y = \sin(x)$ 的数据样本（$N = 10$），并用 3 阶多项式拟合。
2) 10 个样本，用 9 阶多项式拟合。
3)15 个样本，9 阶多项式拟合。
4)100 个样本，9 阶多项式拟合。
5) 依然 10 个样本，加入正则项的 9 阶多项式拟合
如图 (1.1)
所以代码都为 python 程序，详见https://github.com/huajh/csmath

2　主成分分析 (PCA)

对某一个数字 digit, 先从 optdigits.tra 训练样本中到所有 label 为 digit 的样本。该文本文件中，每一行为一个样本，每一行的最后一个数字为 label，$[0 : -1]$ 是长度为 64 的样本（向量），将样本归一化，用 *linalg.eig* 函数得到其特征值与特征向量。取特征值最大的两个特性向量，便可画出 2 维的特征空间，如 (2.1,2.2)。

然后在 2 维特征空间中，找出离网格节点上最近的点，于此得到具有代表性的一系列点。这些有代表性的点对应的灰度图见图 (2.3);

3　高斯混合模型与 EM 算法

用 numpy 中的 random.multivariate_normal 生成多变量高斯分布样本数据。如图 (3.1) 左图，生成了三个二维的高斯分布，共 200 的样本点，从左到右的混合系数, 均值，协方差分别为

$$\begin{aligned}
\alpha &= [0.33, 0.33, 0.34] \\
\mu &= [0, 0], [2, 2], [4, 4] \\
\Sigma &= \begin{bmatrix} 0.1 & 0 \\ 0 & 1.5 \end{bmatrix}, \begin{bmatrix} 1.5 & 0 \\ 0 & 0.1 \end{bmatrix}, \begin{bmatrix} 0.1 & 0 \\ 0 & 1.5 \end{bmatrix}
\end{aligned}$$

现在只有这么一组数据，需要从数据估计得到内在的模型，即高斯混合模型。假设混合个数已知 $k = 3$，用 EM 算法估计模型参数 (N 维)，初始化过程用 k-means，EM 更新步骤见 PRML[1] 的 438-439 页。

k-means 共迭代了 100 次，EM 迭代不超过 10 次便可收敛（对数似然值变化幅度不超过 0.00001%），最终结果为图 (3.1) 右图,

$$\begin{aligned}
\tilde{\alpha} &= [0.3237, 0.3375, 0.3388] \\
\tilde{\mu} &= [-0.0662, -0.0946], [1.9289, 2.0051], [4.0002, 3.8716] \\
\tilde{\Sigma} &= \begin{bmatrix} 0.1238 & -0.0151 \\ -0.0151 & 1.3352 \end{bmatrix}, \begin{bmatrix} 1.4624 & 0.0164 \\ 0.0164 & 0.1054 \end{bmatrix}, \begin{bmatrix} 0.1151 & -0.0999 \\ -0.0999 & 1.2982 \end{bmatrix}
\end{aligned}$$

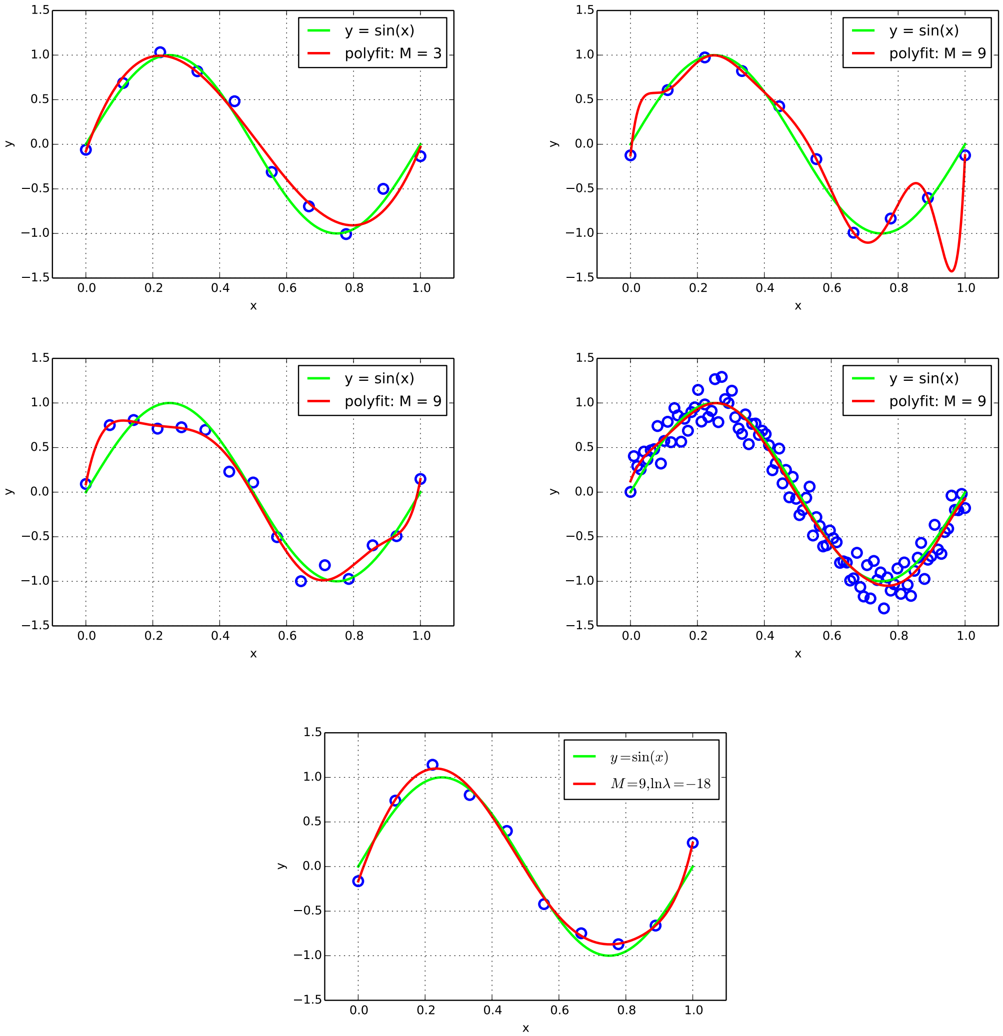

图 1.1: 多项式拟合 N=10,15,100,M=3,9

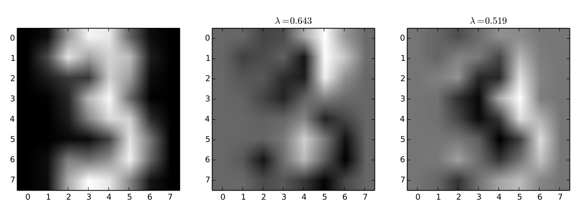

图 2.1: 主成分可视化

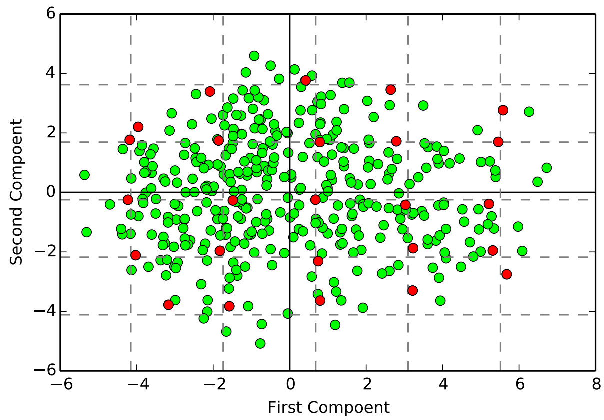

图 2.2: 特征空间可视化

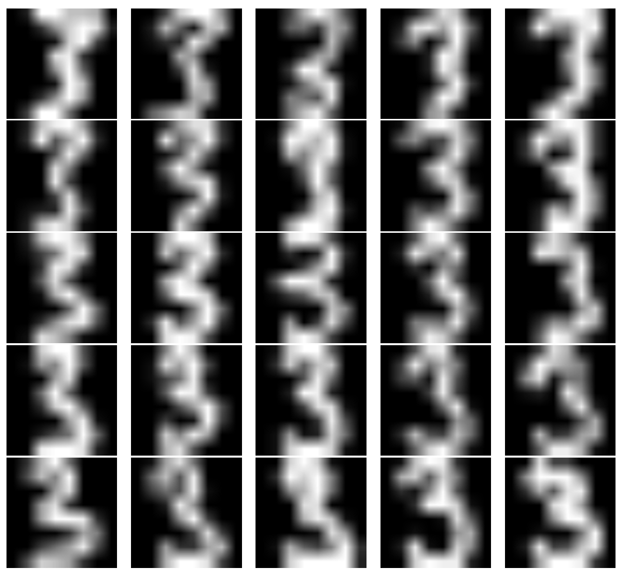

图 2.3: 灰度图

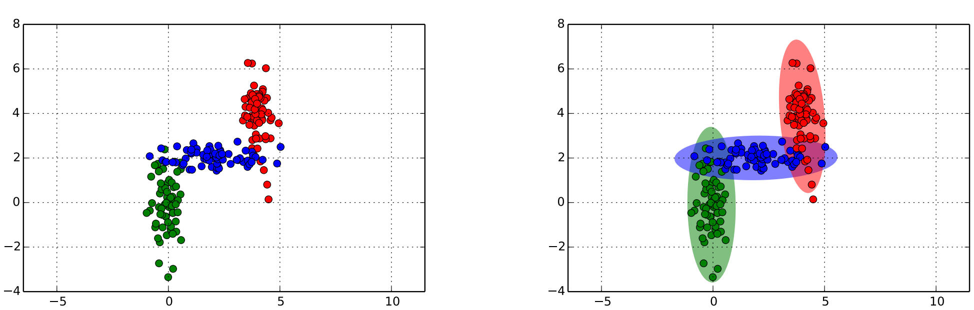

图 3.1: 左图：数据样本; 右图：高斯混合模型聚类结果

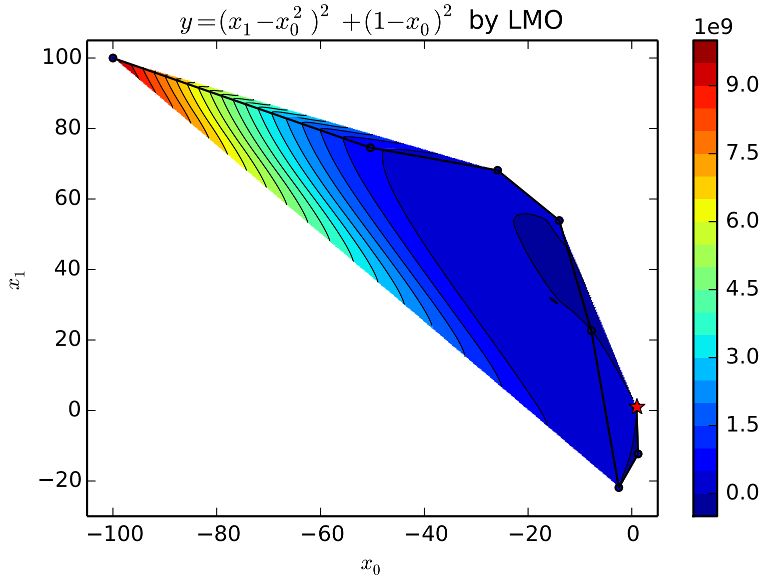

图 4.1: Levenberg-Marquardt 算法迭代过程

4 Levenberg-Marquardt 算法

根据 Machine Learning: An Algorithmic Perspective[2] 一书第 11 章节的 LM 算法例子改写，算法流程按照课件 csmath-07-nonliear，此外还参考了 [4, 3] 两篇 tutorial.

例子:$f(x_0, x_1) = 100(x_1 - x_0^2)^2 + (1 - x_0)^2$, 初始点设为 $(-100, 100)$, 最优点为 $(1, 1)$, 迭代过程见图 (4.1)。

5 simplified SVM 与二次规划（QP）

根据文献 [5] 与 prml 第七章实现了 Least Square SVM。该方法将原本 norm1 约束修改为 norm2, 将原问题由二次规划问题转化为线性方程求解问题，极大降低了问题复杂性，不足之处在于支持向量不再稀疏。这里是实现最早的 LSSVM，对改进的稀疏 LSSVM 未做尝试。内积核使用了线性核，多项式核，径向基函数核，指数核。在如图 (5.1) 线性不可分的例子，用线性核无法正确分类，其他三个都可正确分类，其中指数核分类精度最高，如图 (5.2) 所示。

References

[1] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[2] Stephen Marsland. *Machine learning: an algorithmic perspective*. CRC Press, 2011.

[3] Ananth Ranganathan. The levenberg-marquardt algorithm. *Tutoral on LM Algorithm*, 2004.

[4] Sam Roweis. Levenberg-marquardt optimization. *Notes, University Of Toronto*, 1996.

[5] Johan AK Suykens, Tony Van Gestel, Jos De Brabanter, Bart De Moor, Joos Vandewalle, JAK Suykens, and T Van Gestel. *Least squares support vector machines*, volume 4. World Scientific, 2002.

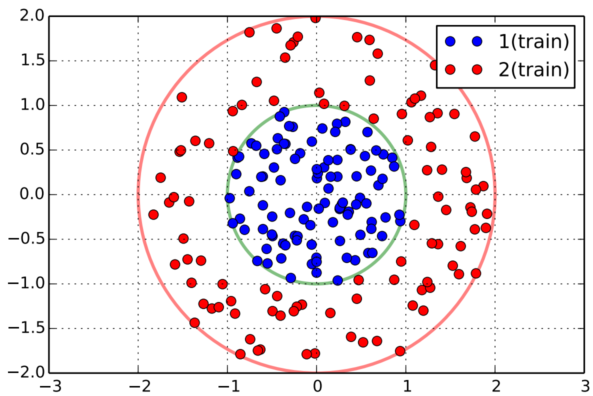

图 5.1: 训练样本

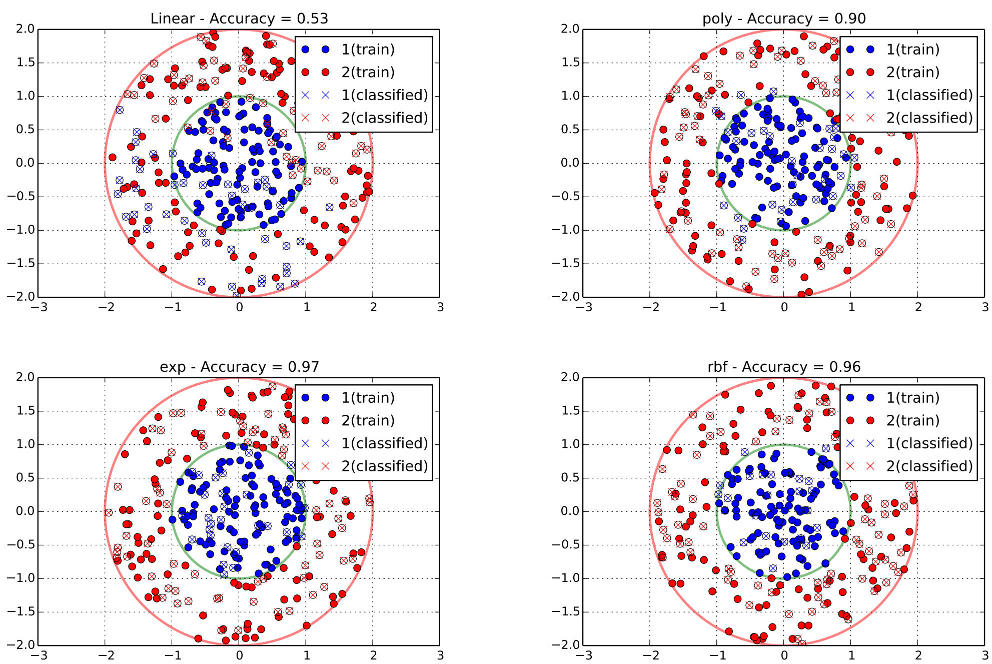

图 5.2: 用四个不同基得到的分类结果