计算机应用数学课程实验报告

张丹(11621004)

May 29, 2016

1 回归分析

1.1 方法概述

回归分析（regression analysis)是确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法，其本质是一个函数估计的问题，找出因变量和自变量之间的因果关系。在回归分析问题中，按照涉及的变量的数量的多少，可以分为一元回归和多元回归问题；按照自变量和因变量之间的关系类型，可以分为线性回归分析和非线性回归分析。这里我们以多项式拟合问题为例，结合实验展示其求解过程。

当我们用多项式作为自变量和因变量之间的映射关系时，我们称这类回归问题为多项式拟合。

1.2 实验陈述

具体实验以一维数据为样本。对于给定的数据集$< x_i, t_i >, i = 1, 2, 3, ...$，我们的目标是得到一个从$x$ 到$t(x)$的映射函数，这里的映射函数类型即为多项式，记为

$$t(x) \approx f(x, \mathbf{w}) = \sum_{i=0}^{k} w_i x^k \tag{1.1}$$

其中的权重系数$\mathbf{w}$是需要求解的参数。最终问题可以描述为：

$$\mathbf{w}^* = \underset{w}{\operatorname{argmin}} \sum_j [t(x_j) - \sum_{i=0}^{k} w_i x^i]^2 \tag{1.2}$$

这是最小化误差平方和问题，一种无约束的最优化问题，通常的解决思路是：梯度下降法和牛顿迭代法。本次实验采用最直接的解法—最小二乘法。最小二乘法可以用来求解线性回归问题的理论依据来源于极大似然估计。我们把上述问题写成矩阵、向量的形式，可以得到利用最小二乘法的求解的一般形式：

$$\mathbf{w} = (X^T X)^{-1} X^T \mathbf{y} \tag{1.3}$$

其中，$\mathbf{w} = [w_0, \ldots, w_k]^T$ 为参数向量，X是数据矩阵，其中每一行对应一个数据样本，可表示为$X[i] = [x^0, x^1, \ldots, x^k]$，$\mathbf{y}$是数据样本的label组成的向量。这种形式的求解，往往会产生过拟合的现象。为了解决这个问题，我们使用带正则化项的最小二乘法，其求解的一般形式为：

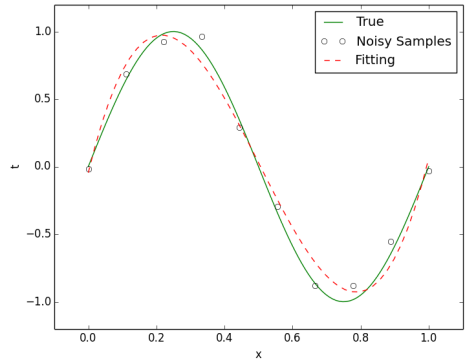
$$\mathbf{w} = (X^T X + \lambda I)^{-1} X^T \mathbf{y} \tag{1.4}$$

其中，λ为正则项的权重系数，I为单位阵。

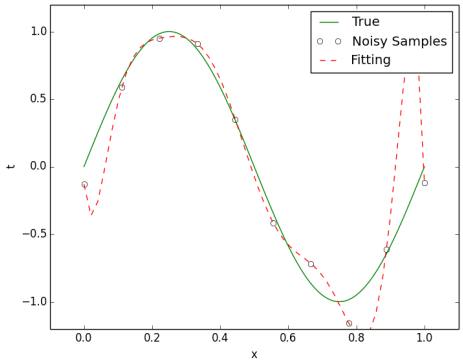
1.3 实验结果

本次实验中，样本数据是从添加了高斯噪声的正弦函数$y = \sin(x) + n$中采样得到的。我们分别用3次和9次多项式对10个样本数据进行拟合；并用9次多项式对15和100个数据样本进行拟合；另外用9次多项式并加上正则项对10个数据点进行拟合。实验效果如图1所示。

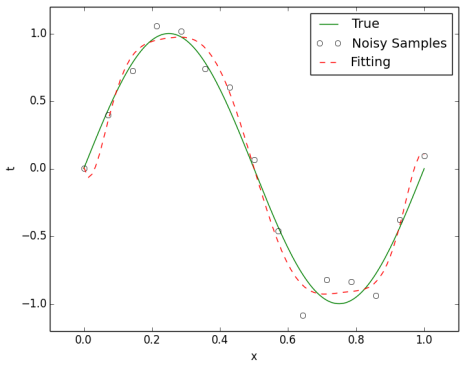
从图中可以看出，同样的样本数据，当多项式的次数越高时，拟合效果越精确，乃至过拟合；样本数据越丰富，能够有效避免产生过拟合；另外，在样本数据不足的情况下，通过添加正则项，也很好地缓解了过拟合现象。



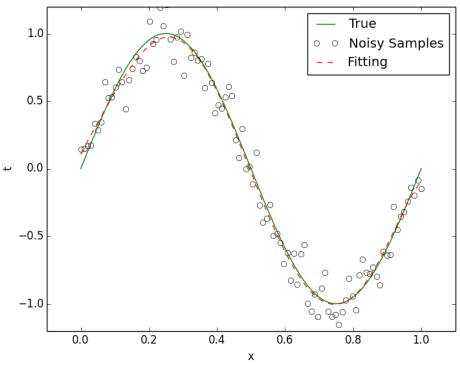
(a) M=3 & N=10



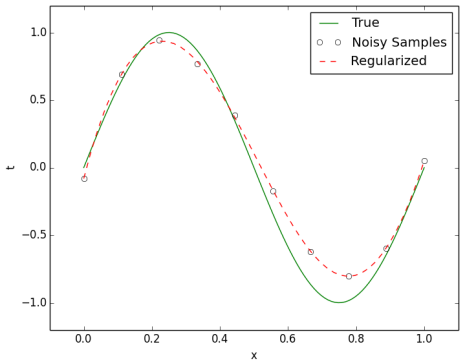
(b) M=9 & N=10



(c) M=9 & N=15



(d) M=9 & N=100



(e) M=9 & N=100

Figure 1: Result of Polynomial Fitting

1.4 小结

最小二乘法，简单直接，能够较好地解决回归问题；但是我们可以看到在求解的过程中，是存在矩阵求逆的计算的。当数据的维度很高，数据集很大时，求逆变得低效，这种情况下，可以利用梯度下降等相关求解方法。

2 PCA(Principal Component Analysis, 主成分分析)

2.1 方法概述

PCA(Principal Component) [3]是一种常用的数据分析方法，是一种掌握事物主要矛盾的统计分析方法，它可以从多元事物中解析出主要影响因素，揭示事物的本质，简化复杂的问题，其目的通常是将高维数据投影到较低维空间。给定n个变量的m 个观察值，形成一个m*n的数据矩阵，n通常比较大。对于一个由多个变量描述的复杂事物，通常我们对事物的全局难以有一个直观的概念，所以需要找到事物主要特性，并以此进行详细分析。但是，在一般情况下，并不能直接找出这样的关键变量。这时我们可以通过对原有变量进行线性变换来寻找事物的主要特性，PCA就是这样一种分析方法。其主要可以用来降维，进行数据可视化。

现对PCA做一个简要的解释分析。我们在现实应用中获得的数据往往是高维数据，甚至是成千上万维度的，因为高维数据能够更详细地描述事物。在这种情况下，硬件资源消耗是不可接受的，因此我们必须对数据进行降维。降维当然意味着信息的丢失，但是因为实际数据本身往往存在的相关性，所以我们可以在降维的同时想办法将信息的损失尽量降低，或者说我们主观上希望最大限度地降低数据维度并减少信息损失。那么，如何衡量信息的相关性、如何度量丢失信息的多少，又如何选择具体降低维度？

假设我们的数据由m个n维样本数据构成一个数据矩阵$X \in \mathbb{R}^{n\times m}$，降维实际上是将高维数据向低维空间进行投影，而这个投影变换，可以用一个矩阵$P \in \mathbb{R}^{k\times n}$来表达，一般$k \leq n$；这个投影矩阵本质上是一组低维空间的基。我们希望投影后的数据能够尽可能多的保留原有信息，也就是说希望投影后投影值尽可能分散，而这种分散程度，可以用数学上的方差来表述。关于pca的“最小化平方误差理论”的解释，可参考[2]。既然我们希望投影后各特征属性能够尽可能多的保留原始信息，也就是说让它们之间尽可能不相关，而相关性，在数学上可以用协方差来表示，于是我们的降维问题有了最终的目标：

将n维数据矩阵X降为k维，这里$k \leq n$，其目标是选择一组正交基，使得原数据经过基变换后得到的数据，各属性之间协方差为0，而属性本身的方差尽可能大。

对于上述问题，均值处理后的数据矩阵对应的协方差矩阵$C = \frac{1}{n-1}XX^T$有着很重要的作用。我们很容易看到C是一个对称矩阵，其对角线分别是各个属性特征的方差，而第i行j列和j行i列元素相同，表示i和j两个特征属性间的协方差。由[1]中的相关推导，我们知道，我们要找的投影矩阵P，其实就是能让原始协方差矩阵对角化的矩阵。数学上有很多方法可以对角化矩阵C，我们依照[1]中的推导，选择$P \equiv E^T$，这里E是协方差矩阵C的特征向量单位化后按列排列出的矩阵。如果我们只取前k个降序排列后的特征值对应的特征向量，则可以得到我们需要的投影矩阵P，用它乘上原始数据矩则完成降维操作。

2.2 实验陈述

由上述简介，我们可以知道，对于m个n维样本数据构成的数据矩阵$X \in \mathbb{R}^{n\times m}$，PCA的算法步骤可总结如下：

- 将X的每一行减去该行的均值；
- 求协方差矩阵$C = \frac{1}{n-1}XX^T$；
- 求协方差矩阵的特征值以及特征向量；
- 将特征值降序排列，取前k个特征值对应的特征向量，按行组成矩阵P；
- $Y = PX$即为降维后的数据。

2.3 实验结果

此次实验，数据集为手写数字数据集[4] “optdigits.tra”，包含0-9是个数字。实验以其中的所有数字“3”构成的数据矩阵为对象，按照以上步骤，进行PCA操作，最终选择最大的两个特征值对应的特征向量组成投影矩阵；实验结果如图2所示。

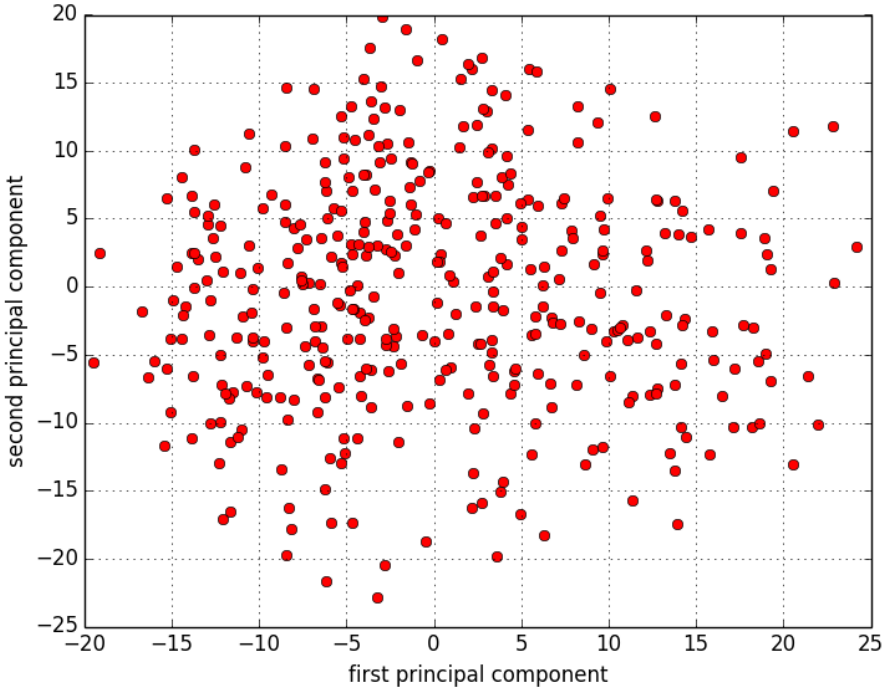


Figure 2: Result of PCA

2.4 小结

PCA本质上是将方差最大的方向作为主要特征，并且在各个正交方向上将数据“离相关”，也就是让它们在不同正交方向上没有相关性。因此，PCA也存在一些限制，例如它可以很好的解除线性相关，但是对于高阶相关性就不适用。另外，PCA假设数据各主特征是分布在正交方向上，如果在非正交方向上存在几个方差较大的方向，PCA的效果就大打折扣。PCA便于通用实现，但是本身无法个性化的优化。

3 MOG(Mixtures of Gaussians，高斯混合模型)

3.1 方法概述

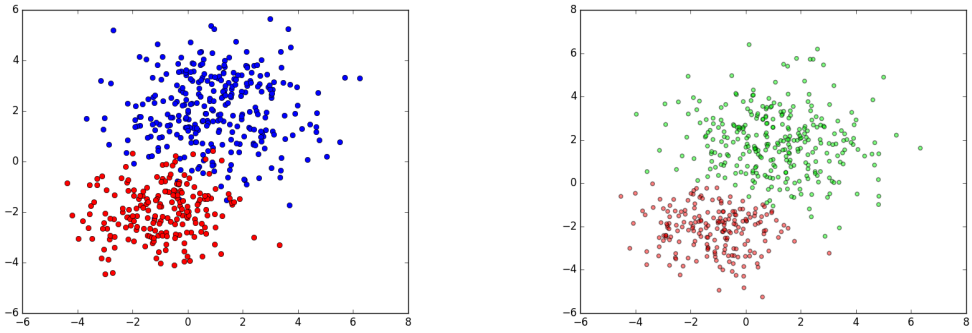
高斯混合模型[5] [6]是指对样本的概率密度分布进行估计，而估计采用的模型是一系列高斯模型的加权和。也就是说，每一个MOG模型(也称GMM, Gaussian Mixed Model)由K个Gaussian分布(常为多元高斯分布)组成，其中的每一个Gaussian称为一个“Component”，这些”Component”线性加成在一起就组成了MOG的概率密度函数，记为：

$$\begin{aligned} p(x) &= \sum_{k=1}^{K} p(k)p(x|k) \\ &= \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k), \end{aligned} \tag{3.1}$$

其中，π_k表示选择第k个Component的概率，也叫权值因子；$N(x|\mu_k, \Sigma_k)$表示第k个Component的均值为μ_k, 协方差矩阵为Σ_k。

MOG是一种聚类算法，每个Component就是一个聚类中心。与K-means聚类算法相比，MOG在聚类时引入了概率，也就是说，K-means会直接将数据点划分到某一个具体的类中，而MOG则会给出划分到该类的概率。当数据样本的类别已知时，我们可以通知直接统计样本的特征来确定MOG模型的参数；但现实中的样本数据往往是不知道类别的，这样我们在用最大似然求解模型参数时，由于“类别”这个潜变量未知，而面临无法求解的困境，最终我们利用EM(Expectation Maximization, 期望最大化)算法来求解模型参数。

Expectation Maximization算法[7] [8]是统计学中用来给带隐含变量的模型做最大似然或者最



(a) Original Data Points (b) After Clustering by MoG

Figure 3: Result of MoG

大后验概率的一种方法。用于求解MOG模型参数的EM算法步骤如下：

初始化相关参数，

循环下列步骤，直到收敛{

(1)E步：计算每个数据样本由每个Component生成的概率，记为$\gamma(i,k)$，

$$\gamma(i,k) = \frac{\pi_k N(x_i|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_i|\mu_j, \Sigma j)} \tag{3.2}$$

(2)M步：按照以下公式更新模型参数，

$$N_k = \sum_{i=1}^{N} \gamma(i,k) \tag{3.3}$$

其中，N为样本个数；

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma(i,k) x_i \tag{3.4}$$

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma(i,k)(x_i - \mu_k)(x_i - \mu_k)^T \tag{3.5}$$

$$\pi_k = \frac{N_k}{N} \tag{3.6}$$

}。

3.2 实验陈述

此次实验，以二维高斯分布为例，先对密度函数进行采样，获得样本数据点，再已这些样本数据点为对象，用MOG模型对其进行建模，并利用EM算法求解模型参数，最终将最初的样本点的分布情形与聚类后的效果进行了对比。

3.3 实验结果

具体实验中，共有500个数据点，分别对两个二维高斯分布进行采样，采样点空间位置如图3a 所示；两个二维高斯分布的参数为：

$$\mu_1 = [1,2]^T, \Sigma_1 = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}$$

$$\mu_2 = [-1,-2]^T, \Sigma_2 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

利用训练好MOG模型对数据点进行聚类后的效果如图3b所示。

f(x)	params	grad	nu
8.20397585679	[0.993 0.738]	6.12337574941	40.0515121106 0.001
6.12337574941	[0.999 0.999]	4.80378573012e-07	55.0134606244 0.0001
4.80378573012e-07	[1.000 1.000]	7.64212029422e-12	0.00968081329974 1e-05
7.64212029422e-12	[1.000 1.000]	3.64573358183e-22	6.12759674617e-05 1e-06
3.64573358183e-22	[1.000 1.000]	1.23259516441e-32	1.95584566103e-11 1e-07

Figure 4: Result of LM

3.4 小结

通过实验效果效果图，我们可以看到，MOG模型的分类结果更为合理，某些在空间上更接近其他中心点的数据点在MOG模型下，给出了一个概率值，而不是像k-means一样直接分到了空间上最为接近的类。当然，因为使用EM算法求解模型参数，当数据量较大时，速度会受到影响。

4 Levenberg-Marquardt 算法

4.1 方法概述

Levenberg-Marquardt 算法(下文简称LM算法) [9] [10]是介于牛顿法与梯度下降法之间的一种非线性优化方法，同时具有牛顿法和梯度法的优点，对于过参数化问题不敏感，能有效处理冗余参数问题，使代价函数陷入局部极小值的机会大大减小，这些特性使得LM算法在计算机视觉等领域得到广泛应用。

4.2 实验陈述

LM一开始是用来解决最小二乘曲线拟合的优化问题，其核心公式为：

$$(J^T J + \lambda I)\delta = J^T[y - f(\beta)] \tag{4.1}$$

其中，J为雅克比矩阵，β为带求解参数向量，$\delta \geq 0$为阻尼系数，是个可变量。当λ很小时，类似于牛顿法；而当λ很大时，类似于梯度下降法。

整个算法过程为：

1.初始化参数x_0

2.当$J^T[y - f(\beta)] > tolerance$并且没有达到最大迭代次数时，重复以下过程：

- 算出位移量δ
- 计算更新值：$x_new = x + \delta$
- 计算目标函数的真实减少量与预测减少量的比值ρ
- 如果$0 < \rho < 0.25$，接受更新值
- 如果$\rho > 0.25$，接受更新值，并减小阻尼系数
- 否则，目标函数在变大，拒绝更新值，增加阻尼系数

4.3 实验结果

此次实验，我们以Roseenbrock's 函数研究对象：

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \tag{4.2}$$

令$\mathbf{r} = (10(x_2 - x_1^2), 1 - x_1)^T$, 则$f(x_1, x_2) = \mathbf{r^T r}$,其对应的雅克比矩阵为：

$$J = \begin{pmatrix} -20x_1 & 10 \\ -1 & 0 \end{pmatrix} \tag{4.3}$$

我们随机初始化参数值，依图4所示，算法很快收敛。

4.4 小结

LM算法属于一种“信赖域法”，牛顿法本质上也可以看做一种信赖域法，即利用局部信息对函数进行建模近似，求取局部最小值。所谓的信赖域法，就是从初始点开始，先假设一个可以信赖的最大位移s(牛顿法中该位移无穷大)，然后在以当前点为中心，以s为半径的区域内，通过寻找目标函数的一个近似函数的最优点，来求解得到真正的位移。在得到了位移之后，再计算目标函数值，如果其使目标函数值的下降满足了一定条件，那么就说明这个位移是可靠的，则继续按此规则迭代计算下去；如果其不能使目标函数值的下降满足一定的条件，则应减小信赖域的范围，再重新求解。

5 SVM(Support Vector Machine，支持向量机)

5.1 方法概述

SVM(Support Vector Machine，支持向量机) [11] [12] 是Vapnik等人在多年研究统计学习基础上提出的一种二类分类模型，其基本模型定义为特征空间上的间隔最大的线性分类器，其学习策略便是间隔最大化，最终可转化为一个凸二次规划问题的求解。

支持向量机学习的基本想法是求解能够正确划分训练数据集并且几何间隔最大的分离超平面。对现行可分的训练数据集而言，线性可分分离超平面有无穷多个，但是集合间隔最大的分离超平面是唯一的。对训练数据集找到集合间隔最大的超平面意味着以充分大的确信度对训练数据进行分类。也就是说，不仅将正负实例点分开，而且对最难分的实例点(离超平面最近的点)也有足够大的确信度将它们分开。这个超平面可以表示为：

$$w^T\mathbf{x}+\mathbf{b}=0 \tag{5.1}$$

现在我们来看点到超平面的距离，定义为：

$$\tilde{\gamma}=y\frac{w^T\mathbf{x}+\mathbf{b}}{\|w\|} \tag{5.2}$$

对一个数据点进行分类，当超平面离数据点的“间隔”越大，分类的确信度（confidence）也越大。所以，为了使得分类的确信度尽量高，需要让所选择的超平面能够最大化这个“间隔”值。于是最大间隔分类器（maximum margin classifier）的目标函数可以定义为：

$$\begin{aligned}&\max\frac{1}{\|w\|}\\&s.t.,y_i(w^Tx_i+b)\geq 1,i=1,\ldots,n\end{aligned} \tag{5.3}$$

转化到这种形式后，我们用过Lagrange Duality 转到对偶变量的优化问题，寻求冯家有效的方法来进行求解，并且结合核函数，将问题推广到线性不可分的情形。引入拉格朗日变量：

$$\mathcal{L}(\sqsupseteq,\lfloor,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}n\alpha_i(y_i(w^Tx_i+b)-1) \tag{5.4}$$

并令：

$$\theta(w)=\max_{\alpha_i\geq 0}\mathcal{L}(\sqsupseteq,\lfloor,\alpha) \tag{5.5}$$

对于线性不可分的问题，我们使用一个非线性映射将数据变换到一个特征空间，然后在这个特征空间使用线性学习分类器，常用的核函数包括多项式核函数和高斯核函数。

5.2 实验陈述

此次实验主要是针对数据点线性不可分的情形。先从4个高斯分布中采样数据点，各分布参数为：

$$\begin{aligned}mean1=[-1,2],\quad mean2=[-2,2]\\mean3=[5,-5],\quad mean4=[-6,6]\end{aligned}$$

所有高斯分布的协方差矩阵取为相同的：

$$cov=\begin{bmatrix}2&1\\1&2\end{bmatrix}$$

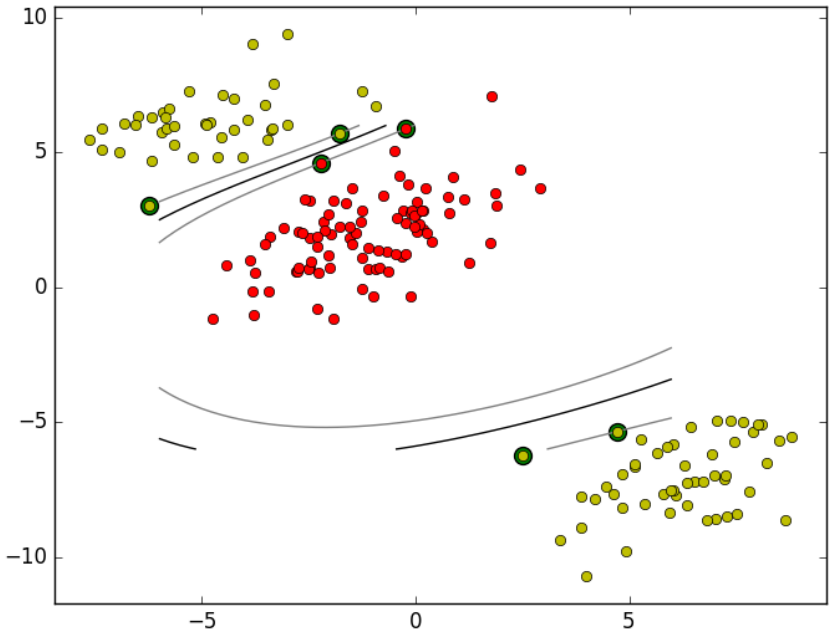


Figure 5: Result of SVM

5.3 实验结果

我们将参数为$mean1, mean2$生成的而数据分为一组，将$mean3, mean4$生成的数据点分为另外一组。分别从两组数据中选择若干数据形成训练集和测试集，二者数据不重复。将训练集经过核函数的映射，找到支持向量，解出参数，然后在测试集上测试。实验结果如图5所示。

5.4 小结

SVM在小样本训练集上能够得到比其它算法好很多的结果。支持向量机之所以成为目前最常用，效果最好的分类器之一，在于其优秀的泛化能力，这是是因为其本身的优化目标是结构化风险最小，而不是经验风险最小，因此，通过margin的概念，得到对数据分布的结构化描述，因此减低了对数据规模和数据分布的要求。

参考文献

[1] Shlens, Jonathon. ”A tutorial on principal component analysis.” arXiv preprint arXiv:1404.1100 (2014).

[2] http://www.cnblogs.com/jerrylead/archive/2011/04/18/2020209.html

[3] Wold, Svante, Kim Esbensen, and Paul Geladi. ”Principal component analysis.” Chemometrics and intelligent laboratory systems 2.1-3 (1987): 37-52.

[4] http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits

[5] Anzai, Yuichiro. Pattern Recognition & Machine Learning. Elsevier, 2012.

[6] Murphy, Kevin P. Machine learning: a probabilistic perspective. MIT press, 2012.

[7] http://www.cnblogs.com/jerrylead/archive/2011/04/06/2006936.html

[8] http://blog.pluskid.org/?p=39

[9] 张鸿燕，狄征. Levenberg-Marquardt算法的一种新解释. 计算机工程与应用，2009，45(19),5-8.

[10] Lourakis, Manolis IA. ”A brief description of the Levenberg-Marquardt algorithm implemented by levmar.” Foundation of Research and Technology 4 (2005): 1-6.

[11] Suykens, Johan AK, and Joos Vandewalle. ”Least squares support vector machine classifiers.” Neural processing letters 9.3 (1999): 293-300.

[12] http://blog.csdn.net/v_jul_v/article/details/7624837