深度学习与自然语言处理第二次报告

郭子龙

1216199336@qq.com

摘要

EM（Expectation-Maximum）算法也称期望最大化算法，曾入选“数据挖掘十大算法”中，在机器学习、数据挖掘中具有显著的影响力。EM算法是最常见的隐变量估计方法，在机器学习中有极为广泛的用途，例如常被用来学习高斯混合模型（Gaussian mixture model，简称GMM）中的参数。本文将会基于EM算法估计所给男女身高数据的高斯分布参数。

1. 介绍

由数据产生过程可知，本文的高斯混合模型应当为两个高斯分布加权而得，假设模型如下式所示：

$$N = p_0 N(\mu_0, \sigma_0) + p_1 N(\mu_1, \sigma_1)$$

通过训练集和EM算法对相关参数进行迭代，待相关参数收敛后，对其结果进行测试与评估

2. 方法

M1. 数据库生成模块

```python
# 定义高斯分布的参数
mean1, std1 = 164, 3
mean2, std2 = 176, 5

# 从两个高斯分布中生成各50个样本
data1 = np.random.normal(mean1, std1, 500)
data2 = np.random.normal(mean2, std2, 1500)
data = np.concatenate((data1, data2), axis=0)

# 将数据写入 CSV 文件
df = pd.DataFrame(data, columns=['height'])
df.to_csv('height_data.csv', index=False)

# 绘制数据的直方图
plt.hist(data, bins=20)
plt.xlabel('Height (cm)')
plt.ylabel('Count')
plt.title('Distribution of Heights')
plt.show()
```

通过随机数生成相应的身高数据库，其中女生数据为500条，男生为1500条。

M2.数据库分块模块

```
#测试集和训练集
girl = data[:500]
boy = data[500:]
girl_train = girl[:400]
girl_test = girl[400:]
boy_train = boy[:1200]
boy_test = boy[1200:]
```

依照比例生成测试集和训练集，比例为4:1。

M3.定义高斯函数模块与测试模块

```
def gauss_pdf(x, miu, sigma):#高斯函数的概率密度函数  pdf表示概率密度
    pdf = 1 / np.sqrt(2 * np.pi) / sigma * np.exp(-0.5 * ((x - miu) / sigma) ** 2)
    return pdf
```

计算数据的高斯密度函数。

```
def test(test_data, miu0, miu1, sigema0, sigema1):   #测试男女概率
    prob_girl = gauss_pdf(test_data, miu0, sigema0)
    prob_boy = gauss_pdf(test_data, miu1, sigema1)
    count_boy = 0
    count_girl = 0
    for i in range(len(prob_girl)):
        if prob_girl[i] >= prob_boy[i]:
            count_girl += 1
        else:
            count_boy += 1
    return count_boy, count_girl
```

对输出结果进行测试，验证正确率。

M4.通过EM算法更新参数

```
#E
gama0 = p0 * gauss_N(data_train, miu0, sigema0) / \
(p0 * gauss_pdf(data_train, miu0, sigema0) + p1 * gauss_pdf(data_train, miu1, sigema1))
gama1 = 1 - gama0
```

```
#M 更新参数
lenn0 = np.sum(gama0)
lenn1 = lenn - lenn0
miu0 = gama0.dot(data_train) / lenn0
miu1 = gama1.dot(data_train) / lenn1
sigema0 = np.sqrt(gama0.dot((data_train - miu0) ** 2) / lenn0)
sigema1 = np.sqrt(gama1.dot((data_train - miu1) ** 2) / lenn1)
p0 = lenn0 / lenn
p1 = lenn1 / lenn
```

M5.结果输出与绘图模块

```
        break
print("男生：均值=", miu1, "; 标准差=", sigema1, "; 权重=", p1)
print("女生：均值=", miu0, "; 标准差=", sigema0, "; 权重=", p0)
girl_ic, girl_c = test(girl_test, miu0, miu1, sigema0, sigema1)
boy_c, boy_ic = test(boy_test, miu0, miu1, sigema0, sigema1)
girl_ac = ('%.2f' % (girl_c / len(girl_test) * 100))
boy_ac = ('%.2f' % (boy_c / len(boy_test) * 100))
print("女生测试集正确率：", girl_ac, '%')
print("男生测试集正确率：", boy_ac, '%')

#绘图
x=data_train
x = np.linspace(150, 195, 5000)
y = p0 * gauss_pdf(x, miu0, sigema0) + p1 * gauss_pdf(x, miu1, sigema1)
plt.hist(data, bins=50, density=True, alpha=0.5)
plt.plot(x, y, 'r-', linewidth=2)
plt.show()
```

3.结果

```
男生：均值= 175.73601622687798 ; 标准差= 4.815668821304205 ; 权重= 0.756543091930987
女生：均值= 163.93107367539525 ; 标准差= 2.863464559073651 ; 权重= 0.24345690806901296
女生测试集正确率： 92.00 %
男生测试集正确率： 92.67 %
迭代次数： 157
```

迭代结果的参数和正确率如上所示，从结果可以看出，系统能在较少的循环次数内收敛，并达到较高的正确率，验证了EM算法在高斯混合模型中的有效性。

参考文献

[1] Blömer J, Bujna K. Simple methods for initializing the EM algorithm for Gaussian mixture models[J]. CoRR, 2013.
[2] Chen F. An Improved EM algorithm[J]. arXiv preprint arXiv:1305.0626, 2013.
[3] Kwedlo W. (2013) A New Method for Random Initialization of the EM Algorithm for Multivariate Gaussian Mixture Learning. In: Burduk R., Jackowski K., Kurzynski M., Wozniak M., Zolnierek A. (eds) Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013. Advances in Intelligent Systems and Computing, vol 226. Springer, Heidelberg.