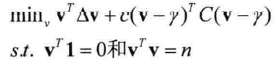
1. （必填）自己提出的问题的理解（罗列全部）：
2. 提出的问题1：如何选择半监督算法，有哪些可借鉴的原则？

讨论后的理解： 这在现实中可能很困难。尽管如此，可以尝试以下条件:这些类能很好地生成集群数据吗?如果是，生成混合模型的EM算法可能是一个不错的选择;这些特征会自然地分成两组吗?如果是，可能需要co training算法;两个特征相似的点是否会在同一类?如果是，可以使用基于图的方法;已经使用支持向量机? 转导SVM是一种扩展算法;现有的监督分类器是否复杂且难以修改?自训练是一种实用且普遍适用的算法思想。

提出的问题2：P132公式（7）是什么意思



讨论后的理解：为了减小最小割倾向于选择不平衡的问题的影响，谱图直推器提出最小化标准割，其中割的值被两个集合边的数目的乘积所标准化。为了简化运算，引入图的组合拉普拉斯算子，为了使所有已标注数据都能合适的分类，在代价函数中加入加号后面的限制条件。

1. （必填）别人提出的问题的理解（选择几个问题罗列，并给出理解）：
2. 问题3：co-training算法要求数据具有至少两个属性集且相互独立，实际数据集很难满足，有没有什么办法可以放松这种假设条件？

自己的理解：为了进一步放松条件独立性假设,Balcan等定义了阿尔法- 膨胀性(Expansion)。在一定程度上为实际任务中视图条件独立性假设、甚至弱依赖性假设都不成立时,基于分歧的多视图半监督学习方法仍可能奏效的原因提供了解释。 他们对学习器的能力进行了约束,要求每个视图上的分类器均能可信正确地标记正样本;这为实际任务中使用强基学习器的惯例提供了理论依据。

问题4：Em算法能够保证必定收敛嘛？

自己的理解： 要证收敛，只需证对数似然函数的值一直在增大（单调有界可以推收敛，联合概率必小于等于1，对数似然函数必小于等于零），即

由

令

两式相减得

上式分别取和，相减得

现在证明上式非负，由于使极大，因此

而

|  |  |  |
| --- | --- | --- |
|  |  | (3) |
|  |  | (4) |
|  |  | (5) |

其中（4）式是Jenson不等式的逆用，（5）式是概率之和为1。

综上，

得证，参数收敛于某一局部最优解，如果是凸的，则可以收敛到全局最优

1. （必填）读书计划

1、本周完成的内容章节：刘兵书5.1

2、下周计划：李航书第一章

四、（选做）读书摘要及理解或伪代码的具体实现（读书摘要、伪代码的具体实现代码等可以写到这个部分）

1、读书摘要及理解（选做）

1.Self-training algorithm(自训练算法)

这个是最早提出的一种研究半监督学习的算法，也是一种最简单的半监督学习算法．用预测的结果作为新的训练数据。

2.Multi-view algorithm(多视角算法)

一般多用于可以进行自然特征分裂的数据集中．考虑特殊情况（每个数据点表征两个特征）：每一个数据点看成是两个特征的集合，然后利用协同训练(Co-training algorithm)进行处理．协同训练（co-training）算法，此类算法隐含地利用了聚类假设或流形假设，它们使用两个或多个学习器，在学习过程中，这些学习器挑选若干个置信度高的未标记示例进行相互标记，从而使得模型得以更新。

3.Generative Models(生成模型)

以生成式模型为分类器，将未标记示例属于每个类别的概率视为一组缺失参数，然后采用EM算法来进行标记估计和模型参数估计，此类算法可以看成是在少量有标记示例周围进行聚类，是早期直接采用聚类假设的做法。EM算法的贪心本质使其容易陷入局部极值，因此算法对初始值的选择具有很强的依赖性．常用的解决方法是采用多组初值进行重复运算，并从中选择最好的一组解，或者通过复杂的优化算法(如分裂合并EM算法)获取参数的优化解．这些做法尽管降低了对初始值选择的敏感性，但却引入了过多的运算负担。

4.Transductive SVM(转导SVM)

强调一点，根据Vapnik的Statistical Learning Theory中提出统计学习中考虑两种不同类型的推理：归纳推理（Inductive inference）和转导推理(Transductive inference)．转导推理的目的是估计某一未知预测函数在给定兴趣点上的值（而不是在该函数的全部定义域上的值）．关键是，通过求解要求较低的问题，可以得到更精确的解．

传统的推理方法是归纳－演绎方法，人们首先根据用已有的信息定义一个一般规则，然后用这个规则来推断所需要的答案．也就是说，首先从特殊到一般，然后从一般到特殊．但是在转导模式中，我们进行直接的从特殊到特殊的推理，避免了推理中的不适定部分．

5.Graph-Based Algorithms(基于图的算法)

是基于图正则化框架的半监督学习算法，此类算法直接或间接地利用了流形假设，它们通常先根据训练例及某种相似度度量建立一个图，图中结点对应了（有标记或未标记）示例，边为示例间的相似度，然后，定义所需优化的目标函数并使用决策函数在图上的光滑性作为正则化项来求取最优模型参数

2.代码实现

import numpy as np

import math

import copy

import matplotlib.pyplot as plt

isdebug = True

# 指定k个高斯分布参数，这里指定k=2。注意2个高斯分布具有相同均方差Sigma，均值分别为Mu1,Mu2。

def init\_data(Sigma,Mu1,Mu2,k,N):

global X

global Mu

global Expectations

X = np.zeros((1,N))

Mu = np.random.random(k)

Expectations = np.zeros((N,k))

for i in xrange(0,N):

if np.random.random(1) > 0.5:

X[0,i] = np.random.normal(Mu1, Sigma)

else:

X[0,i] = np.random.normal(Mu2, Sigma)

if isdebug:

print("\*\*\*\*\*\*\*\*\*\*\*")

print("初始观测数据X：")

print(X )

# EM算法：步骤1，计算E[zij]

def e\_step(Sigma, k, N):

global Expectations

global Mu

global X

for i in xrange(0,N):

Denom = 0

Numer = [0.0] \* k

for j in xrange(0,k):

Numer[j] = math.exp((-1/(2\*(float(Sigma\*\*2))))\*(float(X[0,i]-Mu[j]))\*\*2)

Denom += Numer[j]

for j in xrange(0,k):

Expectations[i,j] = Numer[j] / Denom

if isdebug:

print("\*\*\*\*\*\*\*\*\*\*\*")

print("隐藏变量E（Z）：")

print(Expectations)

# EM算法：步骤2，求最大化E[zij]的参数Mu

def m\_step(k,N):

global Expectations

global X

for j in xrange(0,k):

Numer = 0

Denom = 0

for i in xrange(0,N):

Numer += Expectations[i,j]\*X[0,i]

Denom +=Expectations[i,j]

Mu[j] = Numer / Denom

# 算法迭代iter\_num次，或达到精度Epsilon停止迭代

def run(Sigma,Mu1,Mu2,k,N,iter\_num,Epsilon):

init\_data(Sigma,Mu1,Mu2,k,N)

print("初始<u1,u2>:", Mu)

for i in range(iter\_num):

Old\_Mu = copy.deepcopy(Mu)

e\_step(Sigma,k,N)

m\_step(k,N)

print(i,Mu)

if sum(abs(Mu - Old\_Mu)) < Epsilon:

break

if \_\_name\_\_ == '\_\_main\_\_':

sigma = 6 # 高斯分布具有相同的方差

mu1 = 40 # 第一个高斯分布的均值 用于产生样本

mu2 = 20 # 第二个高斯分布的均值 用于产生样本

k = 2 # 高斯分布的个数

N = 1000 # 样本个数

iter\_num = 1000 # 最大迭代次数

epsilon = 0.0001 # 当两次误差小于这个时退出

run(sigma,mu1,mu2,k,N,iter\_num,epsilon)

plt.hist(X[0,:],50)

plt.show()