

期末综合大作业

**课程名称：**大数据挖掘

**综合题目：svm实现兵王问题的分类**

**学 号：** y21208058

**姓 名：宋天池**

**班 级：210803**

**专 业：人工智能与大数据**

**指导教师：**田秀霞

**实验日期：**

## 论文背景+环境分析（详细）

结合数据挖掘技术，无线传感器网络(WSN)主要面临网络中不同传感器节点之间数据包传输的安全问题。为了克服这一挑战，提出了一种有效的聚类技术，即自适应鸡群优化算法来选择聚类头。这种自适应方法在很大程度上降低了网络的时间消耗，同时提高了网络的生存期和可扩展性。此外，在入侵检测系统(IDS) 利用基于确认的方法报告恶意传感器节点中提出了一种二分类技术，即自适应SVM分类一种监督学习技术。在此基础上，结合入侵检测系统(IDS)对DOS、probe、U2R、R2L等不同类型的攻击进行了检测。一旦被检测到，就会向其他传感器节点提供一种高级的安全机制和入侵响应，从而在不同的传感器节点之间进行安全的数据包传输。该方法在python平台上实现，与已有方法进行了比较，取得了较好的效果。

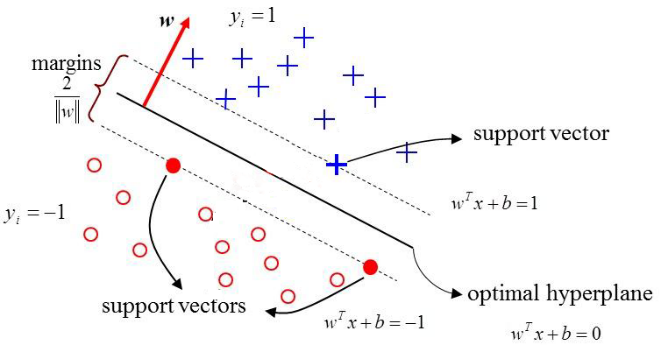
无线传感器网络(WSNs)本质上是无基础设施、分布式和动态的。无线传感器网络的丰富能力向新兴技术领域转变。雾计算就是一个很好的例子。为了满足移动支持、地理分布、位置感知以及物联网应用程序的低延迟需求，Fog节点为用户执行物联网应用程序提供了便利。由于无线传感器网络脆弱的本质，这些网络总是暴露在严重类型的威胁中，这些威胁可以破坏其整体功能。身份验证协议和安全路由协议实现了对加密密钥的使用，以确保数据的安全传输，但不能对内部攻击(即被动攻击)提供保护。这些协议的争夺。有价值的数据来自试图从外部访问它们的入侵者，但来自内部节点的被动攻击不能避免。

数据安全被认为是当今世界最重要的问题之一。数据容易受到各种类型的入侵攻击，这些攻击可能会降低任何网络或系统的效用。识别和防止此类攻击被称为入侵检测系统，它是最具挑战性的任务之一。入侵检测系统是一种针对计算机和网络的安全管理系统。它收集和分析计算机或网络中各个区域的信息，以识别可能的安全漏洞，其中包括入侵(来自组织外部的攻击)和误用(来自组织内部的攻击)。ID使用脆弱性评估，开发来评估计算机系统或网络的安全性。数据被认为是任何组织中最重要的方面。如果组织的数据是安全的，那么只有这样它才能成功地执行其操作。为了在短时间内准确地检测入侵，本文引入了一种基于数据挖掘概念的高效分类器。

## 二、论文采用算法（功能）原理描述（中英文）

Svm分类：

1. 线性可分-硬间隔



SVM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

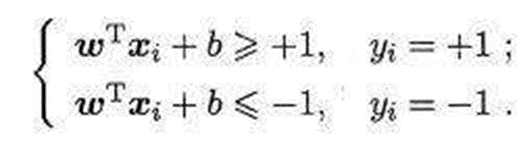
最大的划分超平面。如图所示， 即为划分超平面，对于线性可分

的数据集来说，这样的超平面有无穷多个，但是几何间隔最大的分

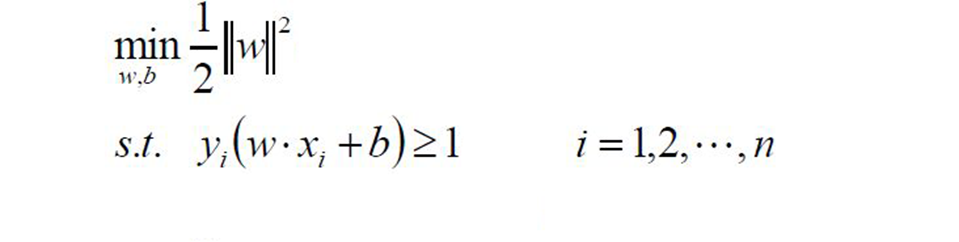
离超平面却是唯一的。而这种线性可分SVM中的划分超平面的间隔也

叫硬间隔

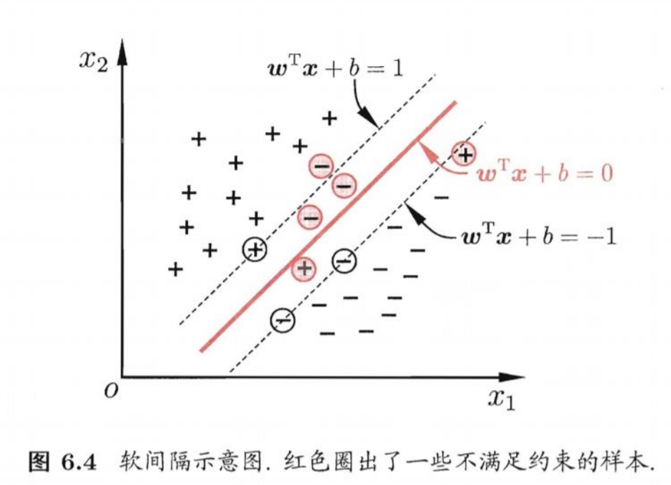
诺超平面能将训练样本正确划分，则有：



SVM基本型：



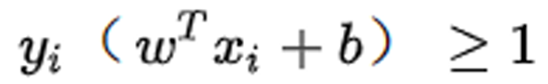
1. 非线性可分-软间隔



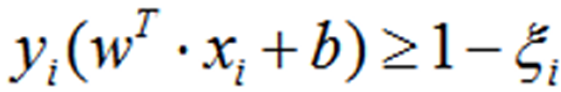
硬间隔的问题：

线性可分svm可通过构造超平面令硬间隔最大化，要正负例完美分开，从而求得最好的分隔超平面,实际情况的数据一半都不会太理想，总在或多或少的噪声数据，根据目标函数最优化我们无法找到一组w和b, 也就是我们就很难找到一个泛化能力很好的超平面了，即使找到了，也有可能是过拟合的。

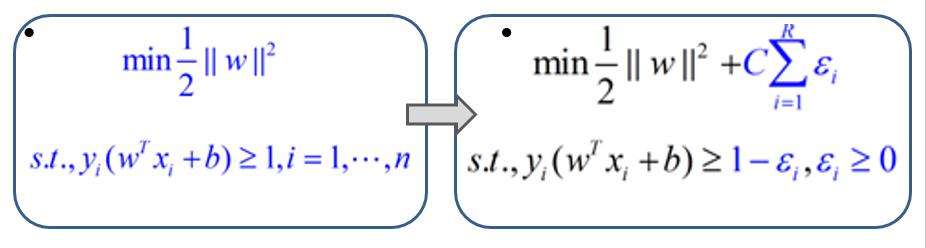
硬间隔支持向量机要求所有的样本均被最佳超平面正确划分，要严格满足条件：



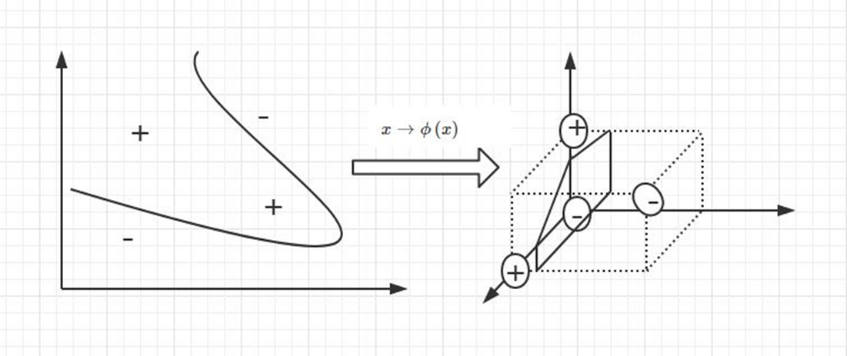
而软间隔支持向量机允许某些样本点不满足间隔大于等于1的条件。也就是说在上边这个约束条件上加了一个松弛变量ξi≥0（每个数据点自己有一个ξi）约束变成：



目标函数优化

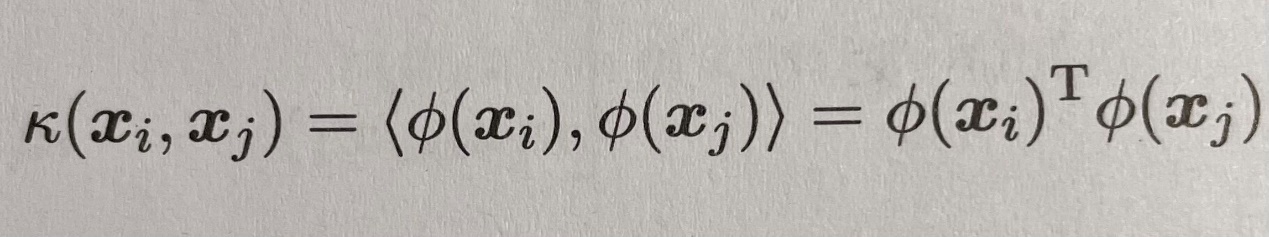


1. 非线性-核函数

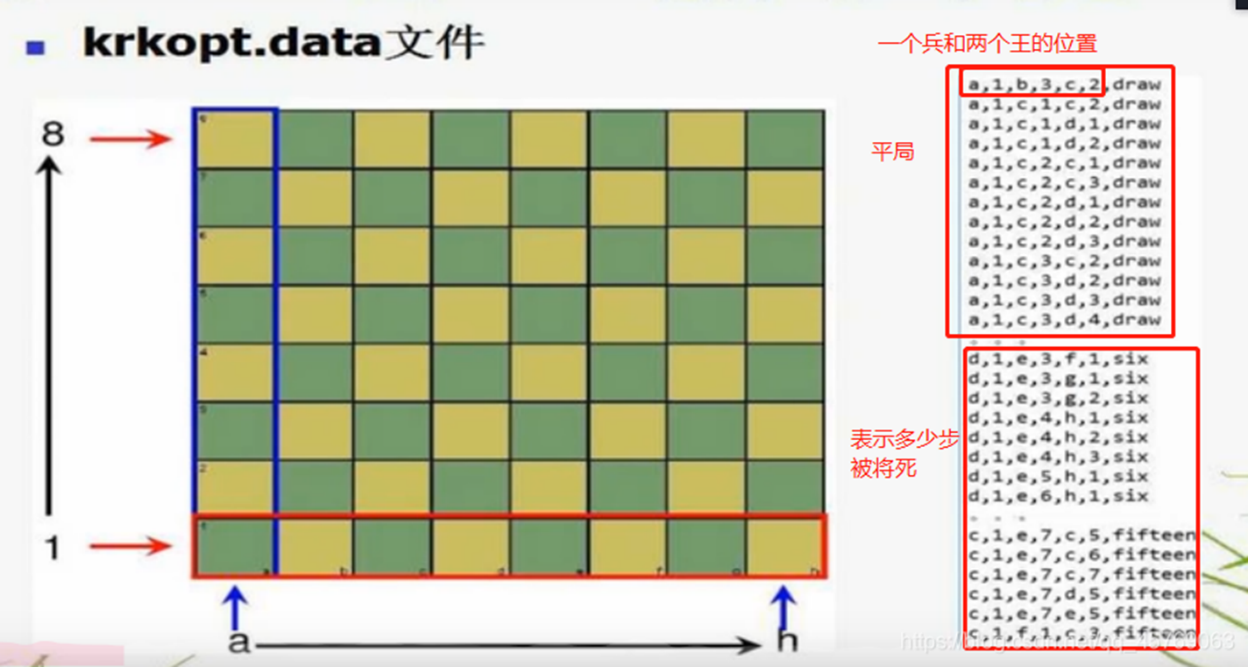


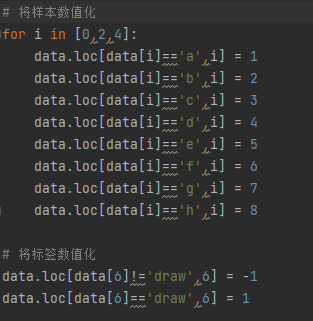
而对于非线性的情况，SVM 的处理方法是选择一个核函数 κ(⋅,⋅)，通过将数据映射到高维空间，来解决在原始空间中线性不可分的问题。此外，因为训练样例一般是不会独立出现的，它们总是以成对样例的内积形式出现，而用对偶形式表示学习器的优势在为在该表示中可调参数的个数不依赖输入属性的个数，通过使用恰当的核函数来替代内积，可以隐式得将非线性的训练数据映射到高维空间，而不增加可调参数的个数(当然，前提是核函数能够计算对应着两个输入特征向量的内积)。

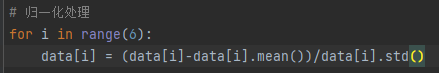
在线性不可分的情况下，支持向量机首先在低维空间中完成计算，然后通过核函数将输入空间映射到高维特征空间，最终在高维特征空间中构造出最优分离超平面，从而把平面上本身不好分的非线性数据分开。



## 三、数据特征抽取、分析以及和算法的结合







# 拆分训练集和测试集  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(data.iloc[:,:6],data[6].astype(int),test\_size=0.82178500142572)

## 四、结合算法的应用设计原理（包括流程图等）



## 五、结合算法的应用设计实现核心代码以及自己改进代码

核心代码：

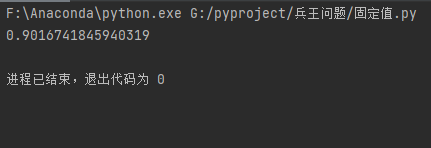
# 读取数据  
data = pd.read\_csv('krkopt.DATA',header=None)  
data.dropna(inplace=True)  
  
# 将样本数值化  
for i in [0,2,4]:  
 data.loc[data[i]=='a',i] = 1  
 data.loc[data[i]=='b',i] = 2  
 data.loc[data[i]=='c',i] = 3  
 data.loc[data[i]=='d',i] = 4  
 data.loc[data[i]=='e',i] = 5  
 data.loc[data[i]=='f',i] = 6  
 data.loc[data[i]=='g',i] = 7  
 data.loc[data[i]=='h',i] = 8  
# 将标签数值化  
data.loc[data[6]!='draw',6] = -1  
data.loc[data[6]=='draw',6] = 1  
# 归一化处理  
for i in range(6):  
 data[i] = (data[i]-data[i].mean())/data[i].std()  
# 拆分训练集和测试集  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(data.iloc[:,:6],data[6].astype(int),test\_size=0.82178500142572)  
  
#将确定好的参数重新建立svm模型  
model = SVC(kernel = 'rbf', C=1,gamma=100)  
model.fit(X\_train, y\_train)  
pre = model.predict(X\_test)  
model.score(X\_test,y\_test)  
print(model.score(X\_test,y\_test))

改进后的代码：

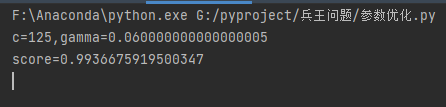
# 寻找C和gamma的粗略范围  
CScale = [i for i in range(100,201,10)]  
gammaScale = [i/10 for i in range(1,11)]  
cv\_scores = 0  
for i in CScale:  
 for j in gammaScale:  
 model = SVC(kernel = 'rbf', C = i,gamma=j)  
 scores = cross\_val\_score(model,X\_train, y\_train,cv =5,scoring = 'accuracy')#5折交叉验证集  
 if scores.mean()>cv\_scores:  
 cv\_scores = scores.mean()  
 savei = i  
 savej = j\*100  
  
# # 找到更精确的C和gamma  
CScale = [i for i in range(savei-5,savei+5)]  
gammaScale = [i/100+0.01 for i in range(int(savej)-5,int(savej)+5)]  
cv\_scores = 0  
for i in CScale:  
 for j in gammaScale:  
 model = SVC(kernel = 'rbf', C = i,gamma=j)  
 scores = cross\_val\_score(model,X\_train, y\_train,cv =5,scoring = 'accuracy')  
 if scores.mean()>cv\_scores:  
 cv\_scores = scores.mean()  
 savei = i  
 savej = j

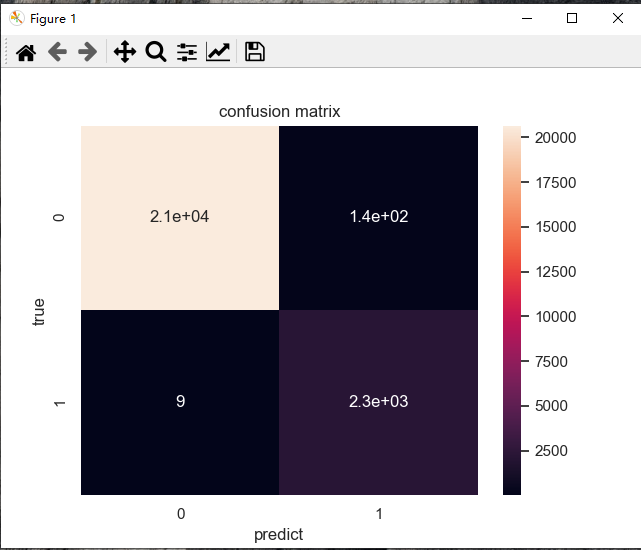
## 六、改进前和改进后实验结果对比分析

改进前：



改进后：





## 七、实验中遇到的问题以及具体解决方法（要具体到调试中的问题）

训练集拆分时数据集与标签分离，数据集无法全部拆分,标签类型报错

解决办法：通过iloc[; ;]纯整数位置索引进行位置选择，冒号前代表索引行，后一位代表索引列，最后一位为步长。

通过astype(int)强制转换数据类型为整型。

## 八、用户使用手册