

人工智能

题目: 对比SVM和GNN两类算法的结果

|  |  |  |  |
| --- | --- | --- | --- |
| 作 者 | 袁博（12202400203） | | |
| 院 别 | 信息科学与工程学院 | 班 级 | 计科20-3BJ |
| 指导教师 | 孙树平 | | |
| 完成时间 | 2023年6月27日 | | |

# 

# 摘 要

身处于SVM 和 GNN 都是常用的分类算法，但它们本质上是不同的。SVM 通过找到最优分离超平面来实现分类，而 GNN 则是一种图神经网络，可以对图像和非结构化数据进行分类。下面分别介绍 SVM 多分类和 GNN 分类的基本原理和应用场景。

SVM 多分类算法可以通过 One-vs-All 方法来实现。One-vs-All 是指将多分类问题看作多个二分类问题，将每个类别和其他类别划分开来，分别训练一个二分类器。在测试时，将测试样本分别送到每个分类器中进行分类，最终将分类器得到的结果汇总，选择置信度最高的分类器的结果作为最终分类结果。

GNN 是一种神经网络，它可以对图像和非结构化数据进行分类。GNN 利用对图像中节点之间的关系的理解，并利用这些关系来学习和分类非结构化数据。例如，当用于分类人体运动姿势数据时，GNN 可以利用节点之间的关系来学习和分类数据。

在实际应用中，SVM 多分类可以应用于文本分类、图像分类、语音识别等任务，而 GNN 分类可以应用于社交网络分析、图像分类、语义分割等场景。因此，在选择使用哪种分类算法时，需要根据具体的数据类型和任务要求进行选择。

关键词：算法、人工智能、GNN、SVM

# 

# Abstract

In t Both SVM and GNN are commonly used classification algorithms, but they are fundamentally different. SVM implements classification by finding the optimal separation Hyperplane, while GNN is a graph neural network that can classify images and unstructured data. Below are the basic principles and application scenarios of SVM multi classification and GNN classification.

The SVM multi classification algorithm can be implemented through the One vs All method. One vs All refers to treating multi classification problems as multiple binary classification problems, dividing each category and other categories, and training a binary classifier separately. During testing, the test samples are sent to each classifier for classification, and the results obtained by the classifier are summarized. The result of the classifier with the highest confidence is selected as the final classification result.

GNN is a neural network that can classify images and unstructured data. GNN utilizes an understanding of the relationships between nodes in images and utilizes these relationships to learn and classify unstructured data. For example, when used to classify human motion posture data, GNN can utilize the relationships between nodes to learn and classify the data.

In practical applications, SVM multi classification can be applied to tasks such as text classification, image classification, speech recognition, etc., while GNN classification can be applied to scenarios such as social network analysis, image classification, semantic segmentation, etc. Therefore, when choosing which classification algorithm to use, it is necessary to choose based on specific data types and task requirements.

**Key words:**Algorithms, artificial intelligence, GNN, SVM

目 录

[摘 要 2](#_Toc5530)

[Abstract 3](#_Toc18102)

[第1章 题目 5](#_Toc15372)

[1.1 具体题目 5](#_Toc19829)

[1.2 题目需求分析 5](#_Toc25255)

[第2章 具体作答 6](#_Toc11707)

[2.1 打印的图片 6](#_Toc14260)

[2.2 代码及注释 6](#_Toc30364)

[第3章 综合布线工程规划 10](#_Toc2078)

[3.1 工程概述 10](#_Toc31445)

[3.2 设计概述 10](#_Toc14123)

**第1章 题目**

**1.1 具体题目**

SVM 将图中给定 5 类数据分为 3 类，并在原图中画出 3 类别分类曲线，然后再计算分类器精度。 其中分类器按照 85-15%总体数据进行学习和测试，用模糊矩阵表示分类结论并对分类结果进行说明。

**1.2 题目需求分析**

1、读取 fig 文件中数据，并按照自己的理解方式重新制作 MATLAB 图形。

2、利用𝑓𝑖𝑡𝑐𝑠𝑣𝑚函数实现多分类（二分类算法）

3、如何利用𝑝𝑙𝑜𝑡𝑐𝑜𝑛𝑓𝑢𝑠𝑖𝑜𝑛函数实现多分类结果表示并计算出分类精度（混淆矩阵）

4、如何利用 GMM 学习算法实现以上分类效果。（GNN算法分类）

**第2章 具体作答**

**2.1 打印的图片**

1、读取 fig 文件中数据，并按照自己的理解方式重新制作 MATLAB 图形。

2、利用𝑓𝑖𝑡𝑐𝑠𝑣𝑚函数实现多分类（二分类算法）

3、如何利用𝑝𝑙𝑜𝑡𝑐𝑜𝑛𝑓𝑢𝑠𝑖𝑜𝑛函数实现多分类结果表示并计算出分类精度（混淆矩阵）

4、如何利用 GMM 学习算法实现以上分类效果。（GNN算法分类）

**2.2 代码及注释**

fig\_handle = open('Fig\_DataSetM.fig'); % 打开一个Fig文件

plot\_children = get(gca, 'Children'); % 获取当前图形对象子元素

new\_fig = figure; % 新建一个图形窗口

colormap = [0 0 1; 1 0 0; 0 1 0; 0 1 1; 1 0 1];% 定义颜色映射数组

plotmarkers = ['s', 'd', 'v', 'p', 'h','o', 'x', '+', '\*', '.']; % 定义绘制符号

plota = [];

for i = 1:length(plot\_children) % 遍历子元素

childe = plot\_children(i);

marker = plotmarkers(mod(i-1, length(plotmarkers)) + 1); % 获取符号数组中的符号

if strcmp(get(childe, 'Type'), 'hggroup') % 如果是分组图形

groupn = get(childe, 'Children'); % 获取子元素中的组对象

else

groupn = childe;

end

for j = 1:length(groupn) % 遍历组对象

groupild = groupn(j);

xvalues = get(groupild, 'XData'); % 获取X轴数据

yvalues = get(groupild, 'YData'); % 获取Y轴数据

scatter(xvalues, yvalues, [], colormap(i,:), marker); % 绘制散点图

hold on;

plota = [plota; [xvalues(:), yvalues(:), repmat(i, length(xvalues), 1)]];

% 将数据存储到pltoa数组中

end

end

title('图1'); % 添加标题

legend('Location', 'northeast'); % 添加图例

hold off; % 取消保持绘图状态

feat = plota(:, 1:2); % 从pltoa数组中提取特征

lab = plota(:, 3); % 从pltoa数组中提取标签

original\_labels = [1 2 3 4 5]; % 定义旧标签

new\_labels = [2 2 3 2 1]; % 定义新标签

for i = 1:length(original\_labels) % 遍历旧标签

lab(lab == original\_labels(i)) = new\_labels(i); % 将旧标签改为新标签

end

cvp = cvpartition(lab, 'Holdout', 0.1); % 分割数据集

train\_feat = feat(training(cvp), :); % 获取训练特征

train\_lab = lab(training(cvp)); % 获取训练标签

test\_feat = feat(test(cvp), :); % 获取测试特征

test\_lab = lab(test(cvp)); % 获取测试标签

SVMModel = fitcecoc(train\_feat, train\_lab); % 训练分类器

pred\_lab = predict(SVMModel, test\_feat); % 预测标签

figure;

gscatter(train\_feat(:,1), train\_feat(:,2), train\_lab, 'rgb'); % 绘制散点图

hold on; d = 0.8;

[x1Grid, x2Grid] = meshgrid(min(train\_feat(:,1)):d:max(train\_feat(:,1)),... min(train\_feat(:,2)):d:max(train\_feat(:,2))); xGrid = [x1Grid(:), x2Grid(:)]; % 生成一个网格，用于绘制决策边界

for i = 1:length(SVMModel.BinaryLearners) % 遍历二元分类器

[~, scores] = predict(SVMModel.BinaryLearners{i}, xGrid);

contour(x1Grid, x2Grid, reshape(scores(:,2), size(x1Grid)),[0 0],'r'); % 绘制决策边界

end

title('决策边界'); % 添加标题

hold off;

figure;

plotconfusion(categorical(pred\_lab),categorical(test\_lab)); % 画混淆矩阵

title('矩阵'); % 添加标题

numComponents = 3; % 定义高斯混合模型的模型数量

options = statset('MaxIter',600); % 定义最大迭代次数

GMModels = cellfun(@(x) fitgmdist(train\_feat(train\_lab == x, :), numComponents,'Options',options), num2cell(1:3)', 'UniformOutput', false); % 对训练集使用高斯混合模型进行拟合

x1 = min(test\_feat(:,1)):0.01:max(test\_feat(:,1)); % 计算决策边界X1轴的范围

x2 = min(test\_feat(:,2)):0.01:max(test\_feat(:,2)); % 计算决策边界X2轴的范围

[X1, X2] = meshgrid(x1, x2); % 生成决策边界坐标数组

XGrid = [X1(:), X2(:)]; Z = zeros(length(X1(:)), 3);

for i = 1:3 % 对测试集预测标签，并将标签reshape成X1、X2形式

Z(:,i) = pdf(GMModels{i}, XGrid);

end

[~, Z] = max(Z, [], 2);

Z = reshape(Z, size(X1));

figure;

contourf(X1, X2, Z, 'LineColor', 'none'); % 绘制等高线图

colormap([0.8 0.8 0.8; 0.4 0.4 0.4; 0 0 0])

hold on;

gscatter(test\_feat(:,1), test\_feat(:,2), test\_lab, 'rgb', 'o', 10); % 绘制测试集散点图

title('决策边界'); % 添加标题

hold off; scores = cell2mat(cellfun(@(model) pdf(model, XGrid), GMModels, 'UniformOutput', false)); % 计算等高线上的分数

scoresGrid = reshape(scores, size(X1, 1), size(X2, 2), 3);

figure;

hold on;

for i = 1:3 % 绘制每个类别的等高线图

contour(X1, X2, scoresGrid(:,:,i), 'LineColor', colors{i});

end

hold off; title('等高线图'); % 添加标题

xlabel('X1');

ylabel('X2');

# 第3章 综合布线工程规划

**3.1 工程概述**

工程名称：湖南理工大学网络规划

地理位置：湖南理工大学

建筑物数量：n栋

信息点总数估计：总计1w+

综合布线目的：达到所有线路均部署在PVC管槽里面，不能外露。布线方案合理，节约，美观。

**3.2 设计概述**

综合布线系统是建筑物或建筑群内的信息传输系统。它使话音和数据通信设备、交换机设备、信息管理系统及设备控制系统、安全系统彼此相连，也使这些设备与外部通信网络相连接。它包括建筑物到外部网络或电话局线路上的连线、与工作区的话音或数据终端之间的所有电缆及相关联的布线部件。布线系统由不同系列的部件组成，其中包括：传输介质、线路管理硬件、连接器、插座、插头、适配器、传输电子线路、电器保护设备和支持硬件。

建筑物结构化综合布线网是由六个独立的子系统组成：

工作区子系统：由工作区内的终端设备连接到信息插座的连接电缆组成。常用设备是计算机，电话，传真机等设备。

管理子系统：由交叉连接、直接连接配线的连接硬件等设备所组成。实现配线管理，其设计很完善，使用颜色编码，很容易追踪和跳线，体积小比传统配线箱节省45％空间。

水平子系统：由每一个工作区的信息插座开始，经水平布置一直到管理区的内侧配线架的线缆所组成。实现信息插座和管理子系统间(跳线架)的连接，常用三类和五类双绞线实现这种连接。

主干线子系统：由建筑物内所有的干线多对数线缆组成，即多对数铜缆，同轴电缆和多模多芯光纤以及将此线缆连接到其他地方的相关支撑硬件所组成。实现计算机设备、程控机PBX和各管理子系统间的连接。常用通信介质是光纤，使系统传输率达到100MBPS。

设备间子系统：由设备间的线缆、连接器和相关支撑硬件组成。实现布线系统与设备的连接，主要为配合不同设备有关的适配器。

建筑群子系统：将一个建筑物中的线缆延伸到建筑物群，实现楼宇之间布线，连接到另一些建筑物中的通讯设备和装置上，它由电缆、光缆和入楼处线缆上过流过压的电器保护设备等相关硬件组成。

综合布线系统本身具有很高的兼容性，根据用户要求，本方案为开放式结构，能支持话音及多种计算机数据和图像传输系统。系统能兼容话音、数据、图像的传输，并可与外部公用网络进行连接。

本综合布线系统工程具体内容包括招标文件所覆盖的数据网络通讯，语音网络系统的综合布线部分。