

東南大學

毕业设计（论文）报告

题目 论文题目

软件 院（系） 专业 专 业

学 号 你的学号

学生姓名 你的名字

指导教师 导师名字

顾问老师 校内导师名字

起止日期 毕设起止日期

设计地点

东南大学毕业（设计）论文独创性声明

本人声明所呈交的毕业（设计）论文是我个人在导师指导下进行的研究工作及取得的科研成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得东南大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

论文作者签名：_____ 日期：_____年____月____日

东南大学毕业（设计）论文使用授权声明

东南大学有权保留本人所送交毕业（设计）论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权东南大学教务处办理。

论文作者签名：_____ 导师签名：_____

日期：_____年____月____日 日期：_____年____月____日

摘 要

这里是中文摘要

关键词：关键字

title

ABSTRACT

英文摘要

KEY WORDS: key word

目 录

摘 要	I
ABSTRACT	II
目 录	III
第一章 数学公式	1
1.1 行内数学公式	1
1.2 数学公式	1
1.3 带有大括号的数学公式	1
第二章 图片	2
2.1 普通图片	2
2.2 子图	2
第三章 算法	3
第四章 列表	4
4.1 有序列表	4
4.2 无序列表	4
第五章 表格	5
第六章 论文引用	6
参考文献	7
致谢	12

第一章 数学公式

这里可以写点东西

1.1 行内数学公式

这是一段话，包含了一个行内公式 $\sum_{t=1}^T k_t = \frac{1}{2}\gamma$

1.2 数学公式

这是一个普通的数学公式

$$q_{\pi}(s, a) = \mathcal{R}_s^a + \gamma \sum_{s' \in S} \mathcal{P}_{ss'}^a \sum_{a' \in A} \pi(a'|s') q_{\pi}(a', s') \quad (1.1)$$

1.3 带有大括号的数学公式

这是一个带有大括号的数学公式

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha} [\eta_{ij}]^{\beta}}{\sum_{k \in A} [\tau_{ik}(t)]^{\alpha} [\eta_{ik}]^{\beta}} & k \in Available \\ 0 & else \end{cases} \quad (1.2)$$

第二章 图片

2.1 普通图片



图 2.1 测试图片

2.2 子图



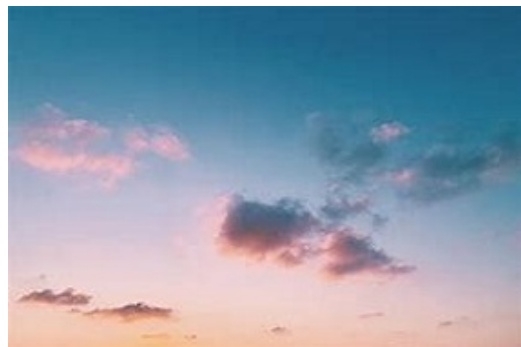
(a) 测试图片 1



(b) 测试图片 2



(c) 测试图片 3



(d) 测试图片 4

图 2.2 带有子图的图片

第三章 算法

算法 1 ACO 算法解决 TSP

输入: 图 $G(V, E)$, d_{ij} 城市 i 与城市 j 的距离, 蚂蚁的集合 M

输出: 最短路径, 最短路径的长度

```

1: function constructRoutes
2:   for  $i$  in  $1, \dots, |V| - 1$  do
3:     for  $\forall k \in M$  do
4:       选择下一个城市  $s_k$ 
5:       增加边  $edge(r_k, s_k)$ 
6:        $r_k \leftarrow s_k$ 
7:     end for
8:   end for
9:   for  $\forall k \in M$  do
10:    增加边  $edge(r_k, s_k)$ 
11:   end for
12: end function
13: function updatePheromones
14:   for  $\forall k \in M$  do
15:     计算  $L_k$ 
16:     更新  $\tau_{r,s}$ 
17:   end for
18: end function
19: function main
20:   for  $\forall dege(r, s) \in E$  do
21:      $\tau_{r,s} \leftarrow \tau_0$ 
22:      $\eta_{r,s} \leftarrow \frac{1}{c_{r,s}}$ 
23:   end for
24:   while 结束条件不满足 do
25:     调用 setInitInfo 函数
26:     调用 constructRoutes 函数
27:     调用 updatePheromones 函数
28:   end while
29: end function

```

第四章 列表

4.1 有序列表

- (1) 随便说点东西
- (2) 随便说点东西
- (3) 随便说点东西

4.2 无序列表

- 随便说点东西
- 随便说点东西
- 随便说点东西

第五章 表格

以下是一个论文中常用的三线表的格式，其中 `tabularcccccc`，其中每一个 `l` 表示居左 `r` 表示居右 `c` 表示居中，其中用 `p` 表示每列的宽度，例如 `cp5em`，表示居中列宽 `5em`

表 5.1 年龄与工作, 房子有无, 信贷关系数据表

ID	年龄段	有工作	有自己的房子	信贷情况	类别 (是否给贷款)
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
...
498	老年	是	否	非常好	是
499	老年	否	否	一般	否
500	老年	否	否	非常好	否

第六章 论文引用

这是一段比较有趣的话^[1]，因为在这短短的一段文字中，居然有两个引用^[2]

论文引用可以使用 bibtex 或者使用 zotero 集体导出 GB/T 7714-2015 格式的引用

zotero 导出论文引用时，将所有引用论文添加到一个单独拿的文件夹中，右键选择“由所选条目创建引文条目”->“引文目录”，“复制到剪贴板”

参考文献

- [1] 王亚杰, 邱虹坤, 吴燕燕, 等. 计算机博弈的研究与发展 [J]. 智能系统学报, 2016, 11(6): 788-798.
- [2] 徐心和, 邓志立, 王骄, 等. 机器博弈研究面临的各种挑战 [J]. 智能系统学报, 2008(04): 288-293.

附录 A

随便写点东西

附录 B

标题

Listing 1 Kmeans.py

```
1 import numpy as np
2 import math as m
3 import random
4 import matplotlib.pyplot as plt
5 import evaluate as eva
6
7 dataPath = "./data/dataset.txt"
8
9
10 # 导入数据
11 def loadData():
12     points = np.loadtxt(dataPath, delimiter='\t')
13     return points
14
15
16 def calculateDistance(data, clu, k):
17     """
18     计算质点与数据点的距离
19     :param data: 样本点
20     :param clu: 质点集合
21     :param k: 类别个数
22     :return: 质心与样本点距离矩阵
23     """
24     dis = []
25     for i in range(len(data)):
26         dis.append([])
27         for j in range(k):
28             dis[i].append(m.sqrt((data[i, 0] - clu[j, 0])**2 + (data[i, 1] - clu[j, 1])**2))
29     return np.asarray(dis)
30
31
32 def divide(data, dis):
33     """
34     对数据点分组
35     :param data: 样本集合
36     :param dis: 质心与所有样本的距离
37     :param k: 类别个数
38     :return: 分割后样本
39     """
```

```

40     clusterRes = [0] * len(data)
41     for i in range(len(data)):
42         seq = np.argsort(dis[i])
43         clusterRes[i] = seq[0]
44
45     return np.asarray(clusterRes)
46
47
48 def getCenter(data, clusterRes, k):
49     """
50     计算质心
51     :param group: 分组后样本
52     :param k: 类别个数
53     :return: 计算得到的质心
54     """
55     clunew = []
56     for i in range(k):
57         # 计算每个组的新质心
58         idx = np.where(clusterRes == i)
59         sum = data[idx].sum(axis=0)
60         avg_sum = sum/len(data[idx])
61         clunew.append(avg_sum)
62     clunew = np.asarray(clunew)
63     return clunew[:, 0: 2]
64
65
66 def kmeans(data, clu, k):
67     """
68     迭代收敛更新质心
69     :param data: 样本集合
70     :param clu: 质心集合
71     :param k: 类别个数
72     :return: 误差, 新质心
73     """
74     clulist = calculateDistance(data, clu, k)
75     clusterRes = divide(data, clulist)
76     clunew = getCenter(data, clusterRes, k)
77     err = clunew - clu
78     return err, clunew, k, clusterRes
79
80
81 def drawPic(data, clusterRes, clusterNum):
82     """
83     结果可视化
84     :param data: 样本集

```

```
85 :param clusterRes:聚类结果
86 :param clusterNum: 类个数
87 :return:
88 """
89 nPoints = len(data)
90 scatterColors = ['black', 'blue', 'green', 'yellow', 'red', 'purple', 'orange', 'brown']
91 for i in range(clusterNum):
92     color = scatterColors[i % len(scatterColors)]
93     x1 = []; y1 = []
94     for j in range(nPoints):
95         if clusterRes[j] == i:
96             x1.append(data[j, 0])
97             y1.append(data[j, 1])
98     plt.scatter(x1, y1, c=color, alpha=1, marker='+')
99 plt.show()
100
101
102 if __name__ == '__main__':
103     k = 10 # 类别个数
104     data = loadData()
105     clu = random.sample(data[:, 0:2].tolist(), k) # 随机取质心
106     clu = np.asarray(clu)
107     err, clunew, k, clusterRes = kmeans(data, clu, k)
108     while np.any(abs(err) > 0):
109         print(clunew)
110         err, clunew, k, clusterRes = kmeans(data, clunew, k)
111
112     clulist = calculateDistance(data, clunew, k)
113     clusterResult = divide(data, clulist)
114
115     nmi, acc, purity = eva.eva(clusterResult, np.asarray(data[:, 2]))
116     print(nmi, acc, purity)
117     drawPic(data, clusterResult, k)
```


致 谢

感谢大家