

欧式距离 $L_2(x_i, x_j) = \left(\sum_{n=1}^N x_i^n - x_j^n \right)^2$

街区距离 $L_1(x_i, x_j) = \sum_{n=1}^N |x_i^n - x_j^n|$

闵氏距离 $L_m(x_i, x_j) = \sqrt[m]{\sum_{n=1}^N |x_i^n - x_j^n|^m}$

Lp距离 $L_p(x_i, x_j) = \left(\sum_{n=1}^N |x_i^n - x_j^n|^p \right)^{\frac{1}{p}}$

远近度量就要使用距离的概念

K=1就是最近邻

K>=1就是投票表决方式

最近邻分类的数学描述

对特征空间 $x \in R^N$ ，最近邻规则在 N 个训练点中构造集合 $X_k(x)$ ，如果距离 $X_k(x)$ 的区域的类标签是 c_k ，那么最近邻分类

最近邻分类算法

最近邻分类算法上构造最近邻图，距离 $\frac{1}{N} \sum_{i=1}^N (x_i - x_j)^2$ 最小，所以多数类取

最近邻分类算法上构造最近邻图，距离 $\frac{1}{N} \sum_{i=1}^N (x_i - x_j)^2$ 最小，所以多数类取

KNN不具有显示的模型

但是具体的过程如下：最近邻选取其同类别，或寻找半径为 R 的圆，寻找圆内数据点的方式产生预测点的类别，方式上有一定的差异，但是本质都是分类的经验风险最小化

特征空间中，对每个训练实例点 X_i ，距离该点比其他点更近的所有点组成一个区域，叫作单元 (cell)。每个训练实例点拥有一个单元，所有训练实例点的单元构成特征空间的一个划分。最近邻法将实例点的类 y 作为其单元中所有点的类标记 (class label)。这样，每个单元的实例点的类别是确定的，如图，灰色区域就是 "x" 类，白色区域就是 "." 类

构造KDTree相当于不断地用垂直于坐标轴的超平面将 N 维空间划分，构成一系列的 N 维超矩形区域，kd树的每个结点对应于一个 N 维超矩形区域。

通过构造KDTree检索最近邻

伪代码

```
Algorithm: Constructing kd-tree
Input:  input of type rangeobject
Output:  kd-tree kd-tree
Data:  data
Pre:  data is a list of type rangeobject
Post:  output is a kd-tree
Code:
1. if input is empty then return the empty kd-tree
2. full pivot choosing procedure, which returns best value
3. mid = (a+b)/2
4. split = the splitting dimension
5. if data is sorted by split
6. if data is sorted by split
7. if data is sorted by split
8. if data is sorted by split
9. if data is sorted by split
10. if data is sorted by split
11. if data is sorted by split
12. if data is sorted by split
13. if data is sorted by split
14. if data is sorted by split
15. if data is sorted by split
16. if data is sorted by split
17. if data is sorted by split
18. if data is sorted by split
19. if data is sorted by split
20. if data is sorted by split
21. if data is sorted by split
22. if data is sorted by split
23. if data is sorted by split
24. if data is sorted by split
25. if data is sorted by split
26. if data is sorted by split
27. if data is sorted by split
28. if data is sorted by split
29. if data is sorted by split
30. if data is sorted by split
31. if data is sorted by split
32. if data is sorted by split
33. if data is sorted by split
34. if data is sorted by split
35. if data is sorted by split
36. if data is sorted by split
37. if data is sorted by split
38. if data is sorted by split
39. if data is sorted by split
40. if data is sorted by split
41. if data is sorted by split
42. if data is sorted by split
43. if data is sorted by split
44. if data is sorted by split
45. if data is sorted by split
46. if data is sorted by split
47. if data is sorted by split
48. if data is sorted by split
49. if data is sorted by split
50. if data is sorted by split
51. if data is sorted by split
52. if data is sorted by split
53. if data is sorted by split
54. if data is sorted by split
55. if data is sorted by split
56. if data is sorted by split
57. if data is sorted by split
58. if data is sorted by split
59. if data is sorted by split
60. if data is sorted by split
61. if data is sorted by split
62. if data is sorted by split
63. if data is sorted by split
64. if data is sorted by split
65. if data is sorted by split
66. if data is sorted by split
67. if data is sorted by split
68. if data is sorted by split
69. if data is sorted by split
70. if data is sorted by split
71. if data is sorted by split
72. if data is sorted by split
73. if data is sorted by split
74. if data is sorted by split
75. if data is sorted by split
76. if data is sorted by split
77. if data is sorted by split
78. if data is sorted by split
79. if data is sorted by split
80. if data is sorted by split
81. if data is sorted by split
82. if data is sorted by split
83. if data is sorted by split
84. if data is sorted by split
85. if data is sorted by split
86. if data is sorted by split
87. if data is sorted by split
88. if data is sorted by split
89. if data is sorted by split
90. if data is sorted by split
91. if data is sorted by split
92. if data is sorted by split
93. if data is sorted by split
94. if data is sorted by split
95. if data is sorted by split
96. if data is sorted by split
97. if data is sorted by split
98. if data is sorted by split
99. if data is sorted by split
100. if data is sorted by split
```

如图所，在搜索过程中只需要搜索图中白色区域就行了，其他灰色区域的不需要进行判断，这样就降低了搜索空间

KDTree为什么能够减少搜索

在实际应用中，我们采取某一维度的中位数进行划分，这样的生成的KDTree最接近与平衡二叉树

但是注意，这样生成的二叉树是平衡的，但是搜索时的效率不一定是最优的

在列两中划分方法中，如果我们使用最宽维度划分就会造成生成的二叉树极度不平衡，但是使用最窄维度的划分数据使得二叉树的生成更平衡一些。

最窄维度划分

最宽维度划分

样本分布几乎在一个椭圆面上

样本分布对KDTree的影响

样本均匀分布

KDTree的中位数选择法

这里可以看出，在样本划分时，如果样本基本处于一个椭圆面在KDTree搜索的效率会非常低，基本会接近搜索整个状态空间但是在样本均匀分布时，这KDTree的搜索效果就会好很多

如果添加的节点是一个叶子节点，这划分很简单，在现有的KDTree基础上添加一个叶子节点就好了，但是实际像往往比这个复杂的，如果我们通过中位数的方式创建KDTree，原有的KDTree本身是平衡二叉树，新加入的节点必定会打破原有的平衡，所以在某些情况下新加入的节点的父节点找到，然后递归该节点下面的所有节点，很明显在查找父节点的过程中还是会浪费大量的时间，空间，在一般情况下的复杂度为 $O(N)$

添加新节点

删除节点

KDTree的查询

先查左子树 (实际上是划分的维度小于根节点的子树)

与根节点比较，判断是否都需要遍历另外一颗子树

查询另外一颗子树

KNN

验证

K近邻

使用数据集: sklearn datasets iris

算法主体

节点类

KDTree类

KDTree与暴力搜索的比较

从图中明显能看出，KD树的搜索效率比纯暴力搜索要好

算法结果

自己的

sklearn

运行结果相同，说明算法正确

手写识别

部分数据图片

需要识别的数字已经使用图形处理软件，处理成具有相同的色彩和大小，或构造是2维数x2维数的前向图像，后期再与文本格式存储图像不能有效地利用存储空间，但是为了方便理解，在数据存储格式仍然保存为文本格式，上图为可视化之后的格式

主要代码

验证结果

总共错了12个数据，错误率为1.268499%

部分数据确实难以辨认，但是错误率只有1.27%，表明分类效果较好