

# L2图像分类，数据驱动方法

2020年3月15日 16:52

最近邻分类器，线性分类器，两层神经网络

## 1. 问题：

- a. 语义鸿沟（图片在计算机中的数据表现为图片中各个像素点的红蓝绿像素值）
- b. 视角，
- c. 照明，
- d. 变形，（物体可能发生的形状变化）
- e. 遮挡，
- f. 类内差异（同类物体之间差异的区分）

## 2. 数据驱动方法：

### Data-Driven Approach

1. Collect a dataset of images and labels
2. Use Machine Learning to train a classifier
3. Evaluate the classifier on new images

训练函数：接收图片和标签，输出模型

预测函数：接收一个模型，对测试图片种类进行

预测

#### a. 最近邻分类 ( $K=1$ )

- i. 训练：记录所有训练数据
- ii. 预测：找到训练数据中最相似的图片，据此预测图片种类（相似度判断方法L1距离, 如下）

## Distance Metric to compare images

网易云课堂

**L1 distance:**  $d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$

iii.

	test image				training image				pixel-wise absolute value differences			
56	32	10	18		10	20	24	17	46	12	14	1
90	23	128	133	-	8	10	89	100	82	13	39	33
24	26	178	200		12	16	178	170	12	10	0	30
2	0	255	220		4	32	233	112	2	32	22	108

= add → 456

iv. 优点：训练过程简单

v. 缺点：

1) 预测过程复杂

2) 决策区域过小，准确率较低（优化：K-

最近邻算法，如下图）

K-Nearest Neighbors

Instead of copying label from nearest neighbor,  
take **majority vote** from K closest points



K = 1



K = 3



K = 5

3) 还可以针对目标点与近邻点的距离值进行加权投票

b. K-最近邻算法

i. L2距离（欧氏距离）

ii. L1与L2区别：

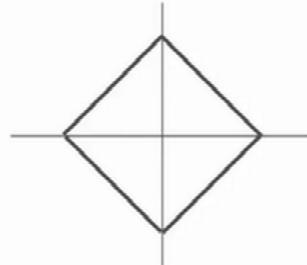
# K-Nearest Neighbors: Distance Metric



L1 (Manhattan) distance

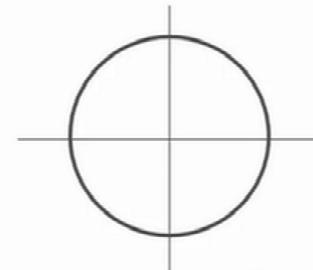
$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

? 1)



L2 (Euclidean) distance

$$d_2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2}$$



2) L1 距离会随坐标轴改变而改变 (可用于突出某个属性) , L2 不会

iii. k和距离度量等超参数需要提前设置为算法的一部分，不会从训练中得到

1) 通过尝试找到较合适的值，但不能简单考虑准确率

2) 不需要过度追求对训练集的拟合，而要让分类器能更准确的对未知数据进行分类 (反例：

## Setting Hyperparameters

网易云课堂

Idea #1: Choose hyperparameters that work best on the data

BAD: K = 1 always works perfectly on training data

Your Dataset

Idea #2: Split data into **train** and **test**, choose hyperparameters that work best on test data

BAD: No idea how algorithm will perform on new data

train

test

3)

Idea #3: Split data into **train**, **val**, and **test**; choose hyperparameters on val and evaluate on test

Better!



**Idea #4: Cross-Validation:** Split data into **folds**, try each fold as validation and average the results

4)

fold 1	fold 2	fold 3	fold 4	fold 5	test
fold 1	fold 2	fold 3	fold 4	fold 5	test
fold 1	fold 2	fold 3	fold 4	fold 5	test

Useful for small datasets, but not used too frequently in deep learning

### c. 最近邻算法总结：不常用于图像分类

- 测试时间长
- L1, L2距离用在比较图像上不太合适

k-Nearest Neighbor on images **never used.**



all image is  
public domain

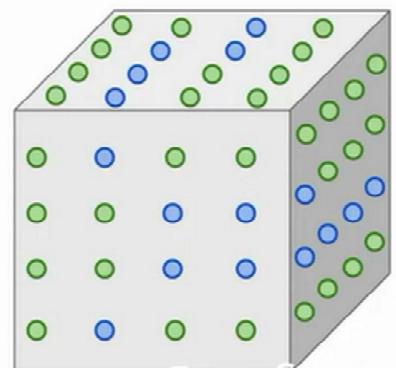
### iii. 维度灾难

k-Nearest Neighbor on images **never used.**



- Curse of dimensionality

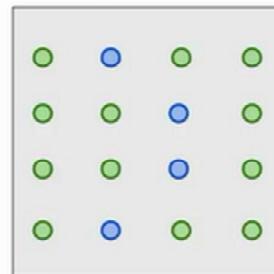
Dimensions = 3  
Points =  $4^3$



Dimensions = 1  
Points = 4



Dimensions = 2  
Points =  $4^2$



## 3. cv思考角度：

- 平面上的高维点概念
- 具体图像的观察