kNN 机器学习算法

Machine learning algorithm

C++ 语言实现

Programming language: C++

张　昊　Holger Zhang

2020/05/03

项目地址

Machine-Learning-k-Nearest-Neighbor

https://github.com/HolgerZhang/Machine-Learning-k-Nearest-Neighbor.git

gitee

Machine-Learning-k-Nearest-Neighbor

https://gitee.com/hzhang19/Machine-Learning-k-Nearest-Neighbor.git

项目介绍

(中文)

机器学习算法：kNN（k近邻学习）

这个项目为 2020 春季学期 C++ 面向过程实践项目。

作者：张昊

时间：2020/05/02 （重构）

简介

有关 kNN 和项目具体实现方法参见 `document` 目录下文件 `kNN_algorithm.pdf` 。

使用 C++ 实现了 k 近邻学习这一种最简单的机器学习算法。

要求数据集的内容格式为： `type,data1,data2,...` ，即类型在前，样本数据在后，用逗号隔开（而不是空格），一条数据占一行。在 `data` 文件夹下给出了样例数据集 `iris.data`（鸢尾花数据集），可供测试使用。

实现了评价算法（评价该数据集使用 kNN 算法的预测正确性，并给出最为适合的参数 `k`）和预测算法（根据提供的数据集，预测输入的测试用例的所属类型），采用命令行参数的方式提供数据集路径等信息。

编译运行

依赖 cmake 编译运行，cmake 的最低版本为 3.5，源代码的组织方式详见 `CMakeLists.txt` 。

```
mkdir build
cd build
cmake ..
make
./evaluate <options>
./predict <options>
```

也可以使用支持 cmake 的集成开发环境（例如 CLion，Visual Studio 2019 等）编译运行。

运行方法

1 评价算法

用于评价该数据集使用 kNN 算法的预测正确性（使用准确率 precision 表示），并给出最为适合的参数 k。

文件和与之对应参数 k 将保存到二进制文件所在文件夹下的 eval.dat 文件中（不存在将被创建）。

使用方法:

```
./evaluate file [range]
```

file： 数据集的路径，可以是绝对路径，也可以是相对路径（但绝对路径和不同的相对路径表示的同一个文件会被认为是不同的文件）。文件的每一行应符合 type,data1,data2,... ，即类型在前，样本数据在后，用逗号隔开（而不是空格）。

range： （可选的，默认为 30）测试集的绝对大小，不应该大于总的数据集的一半。

2 预测算法

根据提供的数据集，预测输入的测试用例的所属类型。

与文件对应参数 k 将从二进制文件所在文件夹下的 eval.dat 文件中获取（不存在将运行评价算法以计算）。

绝对路径和不同的相对路径表示的同一个文件会被认为是不同的文件。

使用方法:

```
./predict file
```

file： 数据集的路径，可以是绝对路径，也可以是相对路径。文件的每一行应符合 type,data1,data2,... ，即类型在前，样本数据在后，用逗号隔开（而不是空格）。

文件结构

```
.
│  CMakeLists.txt
│  CMakeSettings.json
│  LICENSE
│  README.md
│
├─data
│      iris.data
│
├─document
```

```
|      kNN_algorithm.pdf
|
└─src
    ├─evaluate
    |      eval_main.cpp
    |      eval_runner.cpp
    |      eval_runner.h
    |
    ├─predict
    |      pred_main.cpp
    |      pred_runner.cpp
    |      pred_runner.h
    |
    └─shared
           kNNHelper.cpp
           kNNHelper.h
```

Project Introduction

(English)

Machine learning algorithm: kNN (k-Nearest Neighbor)

This project is C++ procedure-oriented practice project for the spring semester of 2020.

Author: Holger Zhang

Time: 2020/05/02 (reconstruction)

Abstract

For the kNN algorithm and the implementation of the project, please refer to the file `kNN_algorithm.pdf` (in Chinese) under directory `document`.

Programming language: C++ (C++11)

Using C++, this project achieves k-Nearest Neighbor, one of the most simple machine learning algorithms.

The content of the dataset is required to format as `type,data1,data2,...` , that is, the first is **type**, and **sample data** follow the **type**. They are separated by commas (rather than white-spaces), and there is only one line when describing one piece of data. The sample dataset `iris.data` (iris dataset) is provided under the `data` folder for testing.

The evaluation algorithm (evaluate the accuracy of the prediction of the dataset using kNN algorithm, and give the most suitable parameter `k`) and **the prediction algorithm** (predict the type of test data according to the provided dataset) are implemented in this project, and the path to the dataset and other information are provided by the **command line parameters**.

Compilation run

Rely on cmake whose version must be upper than 3.5, see `cmakelists.txt` to learn the organization of the source code.

```
mkdir build
cd build
cmake ..
make
./evaluate <options>
./predict <options>
```

You can also use an integrated development environment (IDE) that supports cmake (such as CLion, Visual Studio 2019, etc.).

Run

1 The evaluation algorithm

Evaluate the accuracy (`precision`) of the prediction of the dataset using kNN algorithm, and give the most suitable parameter `k` .

File and parameter `k` will be written in the file `eval.dat` (in the path to binary files, will be created if it doesn't exist).

Usage:

```
./evaluate file [range]
```

`file` : the path to the dataset, the absolute path or the relative path (but the file in the absolute path and the different relative paths will be regarded as different files). The content of the dataset is required to format as `type,data1,data2,...` , that is, the first is **type**, and **sample data** follow the **type**. They are separated by commas (rather than white-spaces), and there is only one line when describing one piece of data.

`range` : (optional, default: 30) the absolute size of test data set, should not be more than half of the range of dataset.

2 The prediction algorithm

Predict the type of test data according to the provided dataset.

The parameter `k` of this file will be get from the file `eval.dat` (in the path to binary files. If not found, will run evaluation algorithm).

The file in the absolute path and the different relative paths will be regarded as different files.

Usage:

```
./predict file
```

`file`： the path to the dataset, the absolute path or the relative path. The content of the dataset is required to format as `type,data1,data2,...` , that is, the first is **type**, and **sample data** follow the **type**. They are separated by commas (rather than white-spaces), and there is only one line when describing one piece of data.

File structure

```
.
│  CMakeLists.txt
│  CMakeSettings.json
│  LICENSE
│  README.md
│
├─data
│      iris.data
│
├─document
│      kNN_algorithm.pdf
│
└─src
    ├─evaluate
    │      eval_main.cpp
    │      eval_runner.cpp
    │      eval_runner.h
    │
    ├─predict
    │      pred_main.cpp
    │      pred_runner.cpp
    │      pred_runner.h
    │
    └─shared
            kNNHelper.cpp
            kNNHelper.h
```

机器学习算法 kNN 实现简介

机器学习算法 kNN 实现简介

k 近邻（k-Nearest Neighbor，简称 kNN）学习是一种常用的监督学习方法。所谓"近朱者赤，近墨者黑"。在分类任务中（本项目是这样），为了判定未知样本的类别，以全部训练样本作为代表点，基于某种距离度量，计算未知样本与所有训练样本的距离，并以近邻 k 个样本的大多数类别作为决策未知样本类别的依据，即选择这 k 个样本中出现最多的类别标记作为预测结果，这种方法也被形象地称为“投票法”。

在回归任务中时使用"平均法"，即将这 k 个样本的实值输出标记的平均值作为预测结果（有待在新的版本中实现）。

与其他的机器学习方法相比，训练只是将样本数据保存起来，等到收到测试样本再进行处理。由此，它看起来没有明显的训练过程，是“懒惰学习”的著名代表。

顾名思义，k 无疑是一个十分重要的参数，通常来说，k 取奇数。当 k 取不同值时，分类结果会有显著不同。如右图，考虑待测样本（用圆形表示）的类别，当 k=3 时，测试样本被判定为三角形；当 k=5 时，测试样本被判定为矩形。而且，k 对噪声数据敏感。

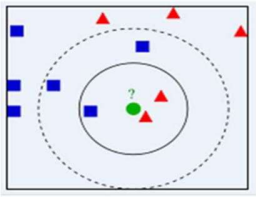

图 1 - k 的选取

由此，项目中评价算法采用了多次使用不同的参数 k 评价数据集的预测正确性，每一次计算从整个数据集中随机抽样，多次得到的结果取 precision 最大的 k 值作为结果。本项目的相应算法可由以下伪代码表示（具体实现详见代码 eval_runner.cpp ）：

```
SETTRAININGANDTESTSET(dataset, range)
1   trainset ← ‹ ›
2   testset ← ‹ ›
3   start ← RANDOM(1, 5)
4   index ← 0
5   while  index < LENGTH(dataset)
6   do  if  start%(LENGTH(dataset)/range) = 0
7           then  INSERT(testset, dataset[index])
8           else  INSERT(trainset, dataset[index])
9       start ← start+1
10      index ← index+1
11  return  trainset, testset
```

图 2 - 抽象的随机抽样算法

```
EVALUATE(dataset, range)
1   result_list ← ‹ ›
2   k ← 1
3   while   k < 20
4   do   trainset, testset ← SETTRAININGANDTESTSET(dataset, range)
5        precision ← GETPRECISION(k, trainset, testset)
6        if   precision = 1
7             then   return ‹k, precision›
8        INSERT(result_list, ‹k, precision›)
9        k ← k+2
10  INVERTEDSORTBYPRECISION(result_list)
11  return   result_list[0]
```

图 3 - 抽象的评价算法

另一方面，若采用不同的距离计算方式，则找出的"近邻"可能有显著差别，从而也会导致分类结果有显著不同。

为解决这一问题，项目实现了两种距离计算方法（有待在更新的版本中增加新的距离算法）：欧氏距离和余弦相似度。在文件 kNNHelper.h 中定义宏 _DISTANCE_TYPE 进行区分，0 表示欧氏距离（默认），1 表示余弦相似度，可以根据需要在编译前修改。

参考资料

[1]周志华.机器学习[M].清华大学出版社:北京,2016:225-226.

张昊
(Holger
Zhang)

DN：cn=张昊
(Holger Zhang),
email=hzhang19
@stu.suda.edu.c
n, c=CN