CS241《问题求解与实践》期末大作业报告

518030910150 方泓杰

Dec. 29th, 2019

注意：运行代码之前请务必认真阅读**注意事项** 一节和demo 视频的演示。

1 任务要求

根据提供的网站链接https://www.kaggle.com/datasets ，选择其中一个数据集（视频、音频、图像相关数据除外），并分别完成以下任务：

- **数据输入和存储**：择取一个数据集并为之设计合适的数据结构，以便高效地存取相关数据；
- **脏数据的发现与处理**：你的数据集中是否存在脏数据？请定义一些标准发现并处理这些脏数据；
- **数据的统计与可视化**：对数据进行统计（至少从三个维度统计数据），并使用FLTK可视化统计后的数据；
- **趋势预测或分类**：设定目标，并使用课程中介绍的线性回归技术或者自学新的技术实现目标。你如何说明所用方法的准确性？

2 数据集

本次大作业，我选择了关于PM2.5 的数据集，该数据集内部提供了2013年3月1日至2017年2月28日间，北京市12 个气象监测站的站点数据，每个数据包括时间（具体到小时）、PM2.5含量、PM10含量等的一系列空气污染物含量、以及当天的降雨量、温度、气压、风速等自然因素。

由于各监测站均处于北京市区及其周边地区，所以各监测站横向比较并没有显著差异，因此不进行监测站之间的横向比较，我们重点关注监测站监测到的PM2.5随时间、温度、压强等因素的关系。

3 数据输入

数据输入部分参见code 文件夹下的data_input 子文件夹内的data_input.cpp 文件。用到的头文件包括code 文件夹下的headfiles 子文件夹内的data_structure.hpp 头文件和tools.hpp 头文件和exceptions.hpp 头文件。

3.1 数据结构的设计

数据结构的代码实现参见headfiles 文件夹内的data_structure.hpp 头文件。

我们设计了如下数据结构用来存储数据：

数据结构 3.1 (带数据缺失的数据结构) 由于数据集中部分列存在着数据缺失的情况（以“*NA*”表示），故设计了一类模板$Data\langle T\rangle$来解决数据的统一存储问题。其由两个变量组成，分别表示是否存储有效值is_rec和存储的值dat。如果$is_rec = 0$则dat为默认值，数据缺失；如果$is_rec = 1$则dat为实际存储的值，数据未缺失。

数据结构 3.2 (一位小数的数据结构) 由于数据集中大部分数据都是一位小数所构成，而利用*double*进行存储的精度低，而一位小数乘*10*即可成为整数，因此设计了存储一位小数的数据结构*oDouble*。

数据结构 3.3 (前数据(Pre-record) 的数据结构) 所谓前数据，就是未经任何处理的数据。在数据输入一步中，初始读入的数据应该用一种合适的数据结构进行存储，这种数据结构必须能够存储*NA*（数据缺失的情况）、小数以及风向（特殊数据，用方向如*N, NE*等表示），因此利用类模板*Data*的实例化$Data\langle double\rangle$即可存储数据缺失的情况。对于风向我们用枚举类型进行存储即可。将这些数据组合在一起形成的类即为新数据结构，我们称之为存储前数据的数据结构。

数据结构 3.4 (前数据表(Pre-record List)) 顾名思义，即前数据组成的列表，主要由存储前数据*(Pre-record)*的*vector*以及一个存储站点名的*string*构成，用来存储一个站点的所有前数据。

数据结构 3.5 (真数据(Record) 的数据结构) 所谓真数据，就是经过脏数据处理后的数据。这时候我们将所有*double*类型转化为上文中的*oDouble*存储，用整型存储其他数据，用枚举类型存储风向，这些数据组合在一起的类即为新数据结构，我们称之为存储真数据的数据结构。

数据结构 3.6 (真数据表(Record List)) 顾名思义，即真数据组成的列表，主要由存储真数据*(Record)*的*vector*以及一个存储站点名的*string*构成，用来存储一个站点的所有真数据。

其他数据结构在后文中遇到我们时会继续说明。

3.2 数据输入

有了上节介绍的几种数据结构，我们先将csv 数据读入前数据表(Pre-record List) 进行存储，则已经完成了数据的输入与存储的功能，接下来为了和下一步相衔接，需要把内存中的数据重新存储在文件中。利用二进制文件容易操作的特点，将其存储在dat 扩展名的二进制文件中。二进制文件以第一个整型数n表示Pre-record的个数，下接n个Pre-record。我们设计了处理文件操作类，方便进行文件的读写。

数据结构 3.7 (文件操作数据结构(File Oper)) 文件操作数据结构由文件指针*fp*和负责二进制读、写的两个函数组成，目的是方便二进制文件操作。

将前数据表以二进制形式存入文件中，则数据读入部分任务完成。

3.3 运行方式

在code 文件夹的子文件夹data_input 下的命令行中，输入如下指令即可

```
g++ data_input.cpp -o data_input.exe -std=c++11
data_input.exe
```

然后从提示列表中选择监测站即可。

注意，需要开启C++11 开关（即如上代码所示）!

4 脏数据处理

脏数据处理部分参见code 文件夹下的data_selection 子文件夹内的data_selection.cpp 文件。用到的头文件包括code 文件夹下的headfiles 子文件夹内的data_selection.hpp 头文件和tools.hpp 头文件和exceptions.hpp 头文件。

4.1 脏数据的定义和处理

经过对于数据的观察以及对于相关资料的查阅，我们对脏数据作如下定义：

定义 4.1 (数据缺失的脏数据) 由于数据的完整性对于预测有着较大的影响，因此如果一组数据存在某些数据缺失，则这组数据为脏数据。

定义 4.2 (小数异常的脏数据) 数据文件中，大部分文件的数据都只有一位小数，极少数情况会出现多位小数，这种情况应该为监测器异常，归结为脏数据。

定义 4.3 (与定义自相矛盾的脏数据) *PM2.5* 的定义是直径小于等于*2.5* 微米的颗粒物，而*PM10* 的定义是直径小于等于*10* 微米的颗粒物，因此空气中*PM10* 的含量必然大于*PM2.5* 的含量（由于*PM2.5* 的所有颗粒物包含于*PM10* 的所有颗粒物中），所以所有*PM10* 小于*PM2.5* 的数据均为脏数据。

以上三类均属于脏数据，需要进行清除或修复。

对于数据缺失的脏数据，无法进行修复因此只能清除；对于小数异常的脏数据，无法确定是否是部分异常还是全部异常，为保险起见进行清除；与定义矛盾的脏数据显然只能进行清除。因此对于三部分脏数据均无法进行修复，只能进行清除。清楚后发现，清除的脏数据占比总数据的不到15%，因此较为合理。

前数据(Pre-record) 经过脏数据过滤之后即为真数据(Record)，即可再次将真数据存储进二进制文件以方便下一步操作。存储方式和上节相同。

4.2 运行方式

在code 文件夹的子文件夹data_selection 下的命令行中，输入如下指令即可

```
g++ data_selection.cpp -o data_selection.exe -std=c++11
data_selection.exe
```

然后从提示列表中选择监测站即可。

注意，需要开启C++11 开关（即如上代码所示）!

5 数据的可视化(FLTK)

数据的可视化部分参见code 文件夹下的data_visualization 子文件夹内的data_visualization.cbp 工程文件，需要采用CodeBlocks 打开! 用到的头文件包括code 文件夹下的headfiles 子文件夹内的data_selection.hpp 头文件和tools.hpp 头文件和exceptions.hpp 头文件，以及GUI 文件夹下的所有文件。

我主要采用了折线图、饼状图、柱状图和雷达图一共四个维度来对于数据进行可视化处理。其中折线图反映了数据的变化程度，从数据变化的角度进行分析；柱状图反映了数据的集中范围，从数据分布的角度进行分析；饼状图和雷达图主要分析数据中的“风向”的概率分布。可视化示例如下图所示：

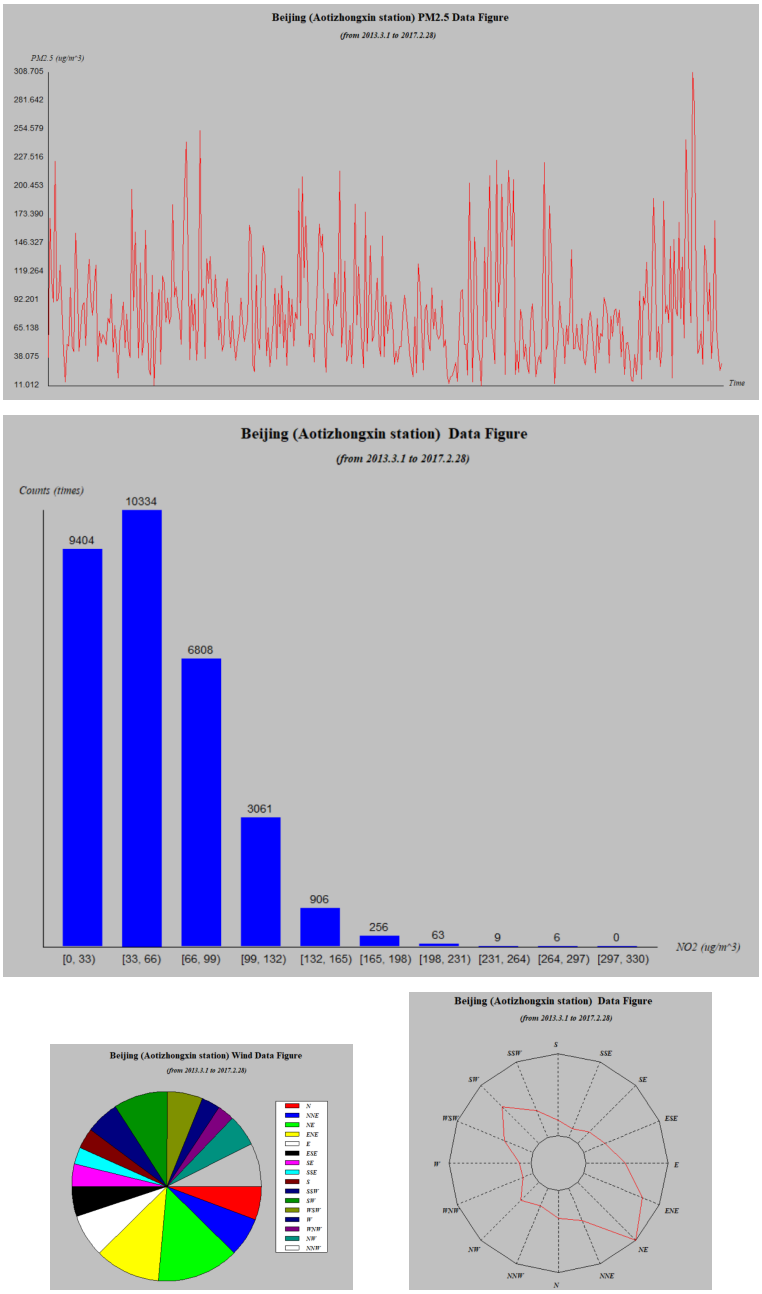

图 1: 数据可视化图形举例

我们对于每个监测站的每个数据（包括PM2.5、PM10、SO2、温度等）都提供了至少两个可视化图形（折线图柱状图或饼状图雷达图）以供选择，具体可以运行程序进一步查看。

运行方式 用CodeBlocks（需要配置好FLTK）开启cbp 工程文件编译运行即可，需要开启C++11 开关。

6 数据预测

数据处理部分参见code 文件夹下的data_prediction 子文件夹内的data_prediction_linear.cpp 文件、data_prediction_argumentssample.cpp文件、data_prediction_nn.cpp文件，分别表示线性回归预测、带多余参数的线性回归预测以及神经网络预测。用到的头文件包括code 文件夹下的headfiles 子文件夹内的data_selection.hpp 头文件和tools.hpp 头文件和exceptions.hpp 头文件和prediction.hpp 头文件，以及matrix.hpp 矩阵类的头文件和nn 子文件夹下的神经网络相关头文件。

数据预测部分，我们将所有数据按照90% : 10% 的比例随机划分成训练集和测试集。针对的目标是：**给定气象的测量值（如温度、大气压等等）、时间以及特殊气体的测量值，预测PM2.5 的检测值**。针对训练集进行训练后得出模型，再通过测试集来测试模型的可靠性和准确性。利用RMS （方均根）来评价预测的准确性，其中RMS 定义为：

$$RMS = \sqrt{\sum_{i=1}^{n}(\hat{w}_i - w_i)^2}$$

其中，$\hat{w}_i$ 表示预测数据，而w_i 表示真实数据。

6.1 线性回归预测

利用线性回归进行预测，设$A \in \mathbb{R}^{n\times m}$ 表示n 组数据，每组数据的维数为m 的已知数据，$B \in \mathbb{R}^{n\times 1}$ 表示n 组数据的PM2.5 实际值。则根据线性回归的方程解得

$$\hat{\theta} = (A^T A)^{-1} A^T B \tag{1}$$

从而我们在矩阵类（matrix类）中实现了矩阵乘法、矩阵求逆等一系列操作，即可方便地求得$\hat{\theta}$ 系数矩阵。从而对于每组测试数据$A' \in \mathbb{R}^{1\times m}$，$A'\hat{\theta} \in \mathbb{R}$ 即为预测的PM2.5 值。由于PM2.5 的数值不能为负，因此若数据为负则对其取绝对值得到预测值。

我们还对于输入数据进行了正规化，使得计算精度更准确。

以Aotizhongxin 监测站为例，利用线性回归预测得到的$RMS_1 = 28.9767$。

6.2 带多余参数的线性回归预测

观察到线性回归方程中，有一些参数对于最终的结果影响较大，因此在这个模型中，加入了一些参数的特殊形式（如指数等）来作为输入的另一些维度。其余实现方式和上一节基本相同。

以Aotizhongxin 监测站为例，利用线性回归预测得到的$RMS_1 = 28.7356$，对比上一节有些许优化，但是效果不够明显。

6.3 神经网络预测

考虑到线性回归的方程过于简单，无法拟合复杂函数，而PM2.5 的值可能和气象因素之间是复杂函数关系，因此采取神经网络来拟合复杂函数。

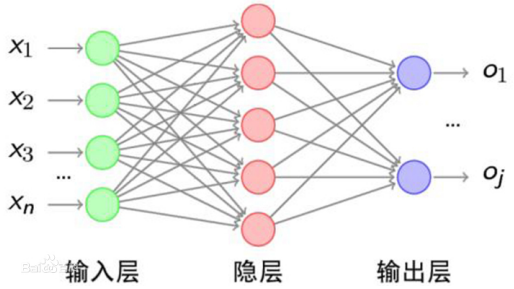

图 2: 神经网络架构

我们采用四层$\{15, 10, 5, 1\}$ 神经网络（一层15 维输入层；两层隐藏层，分别为10维和5 维；最后一层为输出层）进行拟合，激励函数选取为$tanh(x)$，且进行了相关的参数调整。

以Aotizhongxin 监测站为例，经过3×10^7 次训练之后，预测得到的$RMS_3 = 22.0192$，对比上面两种方法有了非常明显的优化。

6.4 分析和总结

从上面三节中可以看出，模型可以拟合的函数越复杂，模型的拟合程度越好。因此神经网络预测的准确性显著大于线性回归预测，而针对数据进行优化的带多余参数的线性回归预测只能对正确率有着小幅度的提升。神经网络的复杂性以及神经元中的激励函数的选择可以有效地提升预测的准确程度。利用C++高效的性能，我们在不到一分钟的时间内进行了3000万次训练，并且较大幅度的提升了预测的准确率。因此在未来相信利用C++ 进行机器学习也是一个可以探索的方向。

6.5 运行方式

在code 文件夹的子文件夹data_prediction 下的命令行中，输入如下指令，从提示列表中选取监测站后，即可采取线性回归模型进行预测。

```
g++ data_prediction_linear.cpp -o data_prediction_linear.exe -std=c++11
data_prediction_linear.exe
```

预测的结果和理论比较将在程序运行后存储在该文件夹下的对应csv 文件中。

同理，将上述指令的data_prediction_linear 替换为data_prediction_argumentssample即可采取带多余参数的线性回归模型进行预测：

针对于神经网络模型，由于其训练速度相较于线性回归模型较慢，推荐开启O2优化开关，即输入如下代码：

```
g++ data_prediction_nn.cpp -o data_prediction_nn.exe -O2 -std=c++11
data_prediction_nn.exe
```

注意，需要开启C++11 开关（即如上代码所示）!

7 注意事项

- 全部代码运行的时候，请开启C++11 开关，即-std=c++11!
- 运行神经网络代码的时候，由于是现场训练3000 万组数据，所以请打开O2 开关。训练时间不会很长，不超过一分钟。
- 请按照数据输入、脏数据处理、数据可视化和数据预测的顺序来依次执行命令行代码，务必在这四步中选取同一检测站点，否则可能会出现文件的错误!
- 请不要改变文件夹的结构以及文件的位置，以防出现路径问题!
- 请将”demo字幕.srt” 文件添加到播放器中，这样可以配合字幕观看demo 更容易理解。
- 初始时，已经有Aotizhongxin 监测站的完整运行结果（包括数据输入储存的中间文件、脏数据处理储存的中间文件和数据预测导出的csv 预测表），因此可以对于任意部分，在监测站选择时选择Aotizhongxin 来进行测试!

8 致谢

- 感谢老师在课堂上对于神经网络的初步介绍，让我领略了这个算法的魅力，以及尝试着实现这个优美的算法。
- 感谢助教对于相关问题的解答。