关联规则挖掘过程报告

数据集

本文中进行数据挖掘的数据集为 Building_Permits 数据集，数据集格式为 csv 文件。该数据集的部分属性如下图所示：

	Sl No	Column name	Description
2	1	Permit Number	Number assigned while filing
3	2	Permit Type	Type of the permit represented numerically.
4	3	Permit Type Definition	Description of the Permit type, for example new construction, alterations
5	4	Permit Creation Date	Date on which permit created, later than or same as filing date
6	5	Block	Related to address
7	6	Lot	Related to address
8	7	Street Number	Related to address
9	8	Street Number Suffix	Related to address
10	9	Street Name	Related to address
11	10	Street Name Suffix	Related to address
12	11	Unit	Unit of a building
13	12	Unit suffix	Suffix if any, for the unit
14	13	Description	Details about purpose of the permit. Example: reroofing, bathroom renovation
15	14	Current Status	Current status of the permit application.
16	15	Current Status Date	Date at which current status was entered
17	16	Filed Date	Filed date for the permit
18	17	Issued Date	Issued date for the permit
19	18	Completed Date	The date on which project was completed, applicable if Current Status = "com
20	19	First Construction Document Date	Date on which construction was documented
21	20	Structural Notification	Notification to meet some legal need, given or not
22	21	Number of Existing Stories	Number of existing stories in the building. Not applicable for certain permit ty
23	22	Number of Proposed Stories	Number of proposed stories for the construction/alteration
24	23	Voluntary Soft-Story Retrofit	Soft story to meet earth quake regulations
25	24	Fire Only Permit	Fire hazard prevention related permit
26	25	Permit Expiration Date	Expiration date related to issued permit.
27	26	Estimated Cost	Initial estimation of the cost of the project
28	27	Revised Cost	Revised estimation of the cost of the project
29	28	Existing Use	Existing use of the building
30	29	Existing Units	Existing number of units
31	30	Proposed Use	Proposed use of the building
32	31	Proposed Units	Proposed number of units
33	32	Plansets	Plan representation indicating the general design intent of the foundation..
34	33	TIDF Compliance	TIDF compliant or not, this is a new legal requirement
35	34	Existing Construction Type	Construction type, existing,as categories represented numerically
36	35	Existing Construction Type Description	Description of the above, for example, wood or other construction types
37	36	Proposed Construction Type	Construction type, proposed, as categories represented numerically
38	37	Proposed Construction Type Description	Description of the above
39	38	Site Permit	Permit for site
40	39	Supervisor District	Supervisor District to which the building location belongs to

数据预处理

为了便于进行数据挖掘，需要对数据集进行处理，转换成适合关联规则挖掘的形式。由于并不是所有的属性都有进行关联规则挖掘的价值，因此在这里选了 9 个标称属性进行关联规则挖掘。这 9 个属性分别是：

1. Permit Type
2. Street Number

3. Current Status
4. Structural Notification
5. Existing Use
6. Proposed Use
7. Existing Construction Type(Definition)
8. Proposed Construction Type(Definition)
9. Site Permit

这 9 个属性分别对应上图中的 2, 7, 14, 20, 28, 30, 34, 36, 38

选出这 9 个属性后，由于数据集中存在很多数据的缺失，我们这里首先对缺失数据进行填充，然后剔除。数据预处理代码段如下：

```python
import csv
import pickle
database = './dataset/Building_Permits.csv'
NA = ['NA', 'None', '', 'NONE', 'none', 'Na']


def preprocess():
    dt = []

    with open('./dataset/item', 'r') as fp:
        attr = [int(i) - 1 for i in fp.read().split(' ')]

    with open(database, encoding='utf-8') as fp:
        reader = csv.reader(fp)
        for _, row in enumerate(reader):
            item = []
            if _ == 0:
                name = row
            for i in attr:
                at = row[i]
                if at in NA:
                    at = '(%s) NA' % (name[i])
                else:
                    at = '(%s) %s' % (name[i], row[i])

                item.append(at)

            dt.append(item)
            # print(item)

    with open('./dataset/preprocessed.pkl', 'wb') as fp:
        pickle.dump(dt, fp)


if __name__ == '__main__':
    preprocess()
```

挖掘算法

这里用了 Apriori 算法进行数据挖掘，该算法的核心算法过程如下：

- 过单趟扫描数据库 D 计算出各个 1 项集的支持度，得到频繁 1 项集的集合。
- 连接步：为了生成，预先生成,由 2 个只有一个项不同的属于的频集做一 个（k-2）JOIN 运算得到的。
- 剪枝步：由于是的超集，所以可能有些元素不是频繁的。在 潜在 k 项集的某个子集不是中的成员是，则该潜在频繁项集不可能是频繁的可以从中移去。
- 通过 单趟扫描数据库 D，计算中各个项集的支持度，将中不满足支持度的项集去掉形成。

通过迭代循环，重复步骤 2～4，直到有某个 r 值使得为空，这时算法停止。在剪枝步中的每个元素需在交易数据库中进行验证来决定其是否加入，这里的验证过程 是算法性能的一个瓶颈。这个方法要求多次扫描可能很大的交易数据库。可能产生大量的候选集，以及可能需要重复扫描数据库，是 Apriori 算法的两大缺 点。

- 支持度

支持度(support)=(X,Y).count/T.count, (T 是事务总和，(X,Y).count 是 X、Y 同时出现的次数)

- 置信度

置信度(confidence)=(X,Y).count/X.count

- 期望置信度

期望置信度=Y.count/T.count

- 去冗余

总体来说，规则 2 是 规则 1 的衍生规则，如果规则 2 和 规则 1 有相同的提升度或者比规则 1 更低的提升度，那么规则 2 就被认为是冗余的。

- 借用 Aprior 算法的思想，可以得到两条有用的去除冗余的规则。

- 如果一个集合是频繁项集，则它的所有子集都是频繁项集。举例：假设一个集合{A,B}是频繁项集，即 A、B 同时出现在一条记录的次数大于等于最小支持度 min_support，则它的子集{A},{B}出现次数必定大于等于 min_support，即它的子集都是频繁项集。

- 如果一个集合不是频繁项集，则它的所有超集都不是频繁项集。举例：假设集合{A}不是频繁项集，即 A 出现的次数小于 min_support，则它的任何超集

如{A,B}出现的次数必定小于 min_support，因此其超集必定也不是频繁项集。

除此之外，我们主要挖掘的是特征都结果的映射，可以把 a 开头的属性和 d 开头的属性分开，因为某种程度上来说 a 开头的属性是特征属性，而 d 开头的属性是结果属性。这样，我们就可以得到一系列从特征到结果的规则，形如 X->Y。

- lift 评价

提升度是可信度与期望可信度的比值，提升度大于 1 表示正相关，小于 1 表示负相关，等于 1 表示不相关。lift 评价可以弥补置信度、支持度自身的不足，使得评价更为合理。利用这个值可以对关键规则进行排序，在一定程度上也是一个规则优化挑选的过程。

数据处理结果分析见“数据处理结果分析”