Prediction of Electricity Comsumption

唐鉴恒
15级 数学与应用数学(基地)
2017/7/22

Outline

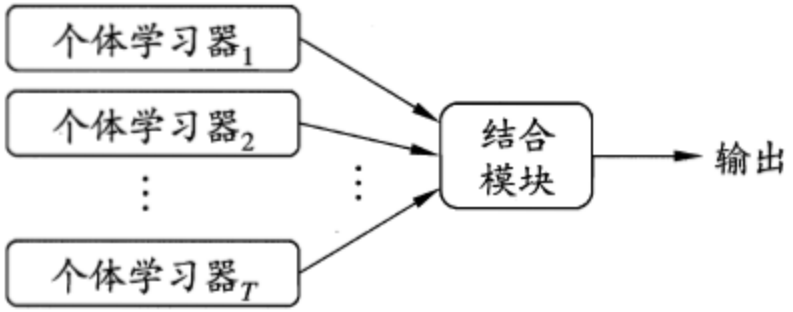
- A review of Gredient Tree Boosting
- XGBoost
- Problem analysis
- Feature engineering
- My solution
- Future work

A review of Gredient Tree Boosting (GBDT \ GBRT)

1. Ensemble learning
2. Tree learning
3. Tree ensemble model
4. Boosting
5. Gradient tree boosting

1.Ensemble learning

- Bagging (Random Forest)
- Boosting (Adaptive Boosting)
- Stacking



	测试例1	测试例2	测试例3
h_1	√	√	×
h_2	×	√	√
h_3	√	×	√
集成	√	√	√

(a) 集成提升性能

	测试例1	测试例2	测试例3
h_1	√	√	×
h_2	√	√	×
h_3	√	√	×
集成	√	√	×

(b) 集成不起作用

	测试例1	测试例2	测试例3
h_1	√	×	×
h_2	×	√	×
h_3	×	×	√
集成	×	×	×

(c) 集成起负作用

图 8.2 集成个体应“好而不同”(h_i 表示第 i 个分类器)

2.Tree Learning

$$F = \{f(\mathbf{x}) = w_{q(\mathbf{x})}\}(q : \mathbb{R}^m \rightarrow T, w \in \mathbb{R}^T)$$

- Here q represents the structure of each tree that maps an example to the corresponding leaf index.
- T is the number of leaves in the tree.

```
检测数据集中的每个子项是否属于同一分类：
    If so return 类标签；
    Else
        寻找划分数据集的最好特征
        划分数据集
        创建分支节点
            for 每个划分的子集
                调用函数createBranch并增加返回结果到分支节点中
        return 分支节点
```

3.Tree ensemble model

$$\hat{y}_i = \phi(\mathbf{x_i}) = \sum_{k=1}^{K} f_k(\mathbf{x}_i), f_k \in F$$

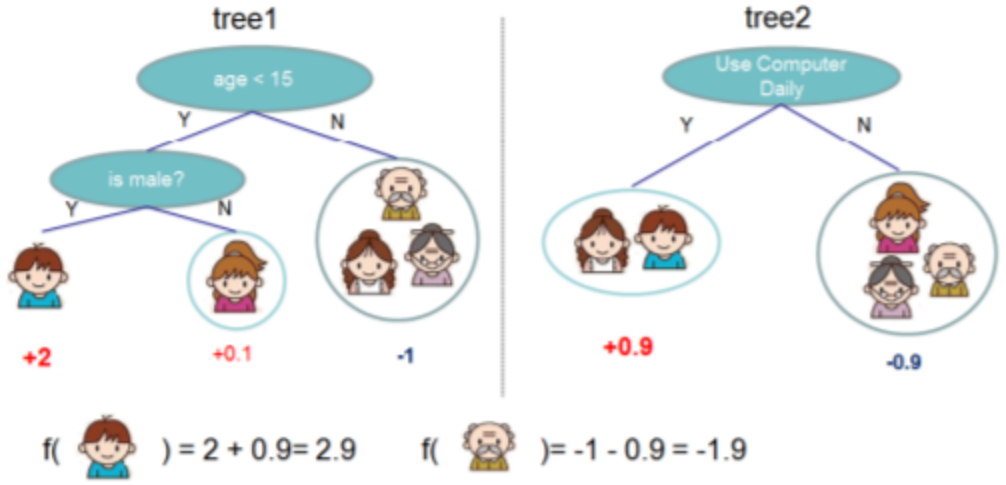


Figure 1: Tree Ensemble Model. The final prediction for a given example is the sum of predictions from each tree.

4.Boosting

- adaboost

输入: 训练集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
基学习算法 $\mathfrak{L}$;
训练轮数 T.

过程:

1: $\mathcal{D}_1(\boldsymbol{x}) = 1/m$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: $\quad h_t = \mathfrak{L}(D, \mathcal{D}_t)$;
4: $\quad \epsilon_t = P_{\boldsymbol{x} \sim \mathcal{D}_t}(h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}))$;
5: $\quad$ **if** $\epsilon_t > 0.5$ **then break**
6: $\quad \alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$;
7: $\quad \mathcal{D}_{t+1}(\boldsymbol{x}) = \frac{\mathcal{D}_t(\boldsymbol{x})}{Z_t} \times \begin{cases} \exp(-\alpha_t), & \text{if } h_t(\boldsymbol{x}) = f(\boldsymbol{x}) \\ \exp(\alpha_t), & \text{if } h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}) \end{cases}$
$= \frac{\mathcal{D}_t(\boldsymbol{x}) \exp(-\alpha_t f(\boldsymbol{x}) h_t(\boldsymbol{x}))}{Z_t}$
8: **end for**

输出: $H(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}) \right)$

图 8.3 AdaBoost算法

5.Gradient Boost

wiki

XGBoost

> [1]Among the 29 challenge winning solutions published at Kaggle's blog during 2015, 17 solutions used XGBoost. Among these solutions, eight solely used XGBoost to train the model,while most others combined XGBoost with neural nets in ensembles.

We minimiaze the following regularized objective:

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$$

$$where\ \ \Omega(f) = \gamma T + \frac{1}{2}\lambda||w||^2$$

- γT : 控制叶子节点的个数
- $\frac{1}{2}\lambda||w||^2$:控制每个叶子节点的权重

Let $\hat{y_i}^{(t)}$ be the predition of the i-th instance at the t-th iteration, we will need to add f_t to minimize the following objective.

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{(t-1)} + f_t(\mathbf{x})) + \Omega(f_t)$$

Second-order approximation can be used to quicklyoptimize the objective in the general setting.

$L^{(t)} \simeq \sum_{i=1}^{n} [l(y_i, y^{t-1}) + g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t)$
where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}), h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$

Define $I_j = \{i|q(\mathbf{x}_i) = j\}$as the instance set of leaf j. We can rewrite Eq by expanding Ω as follows

$$\tilde{L}^{(t)} = \sum_{i=1}^{n}[g_i f_t(\mathbf{x}_i) + \frac{1}{2}h_i f_t^2(x_i)] + \gamma T + \frac{1}{2}\sum_{j=1}^{t} w_j^2$$

$$= \sum_{j=1}^{T}[\sum_{i\in I_j} g_i\omega_j + \frac{1}{2}(\sum_{i\in I_j} h_i + \lambda)\omega_j^2] + \gamma T$$

For a fixed structure $q(\mathbf{x})$, we can compute the optimal weight w_j^* of leaf j by $w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}$ and calculate the corresponding optimal value by

Problem Analysis

大航杯“智造扬中”电力AI大赛

主办方提供2015年1月1日到2016年8月30号1454家企业每日用电量，要求预测2016年9月每日所有企业总用电量

- 数据经过脱敏处理
- 评价公式为:

$$\frac{1}{n}\sum_{i=1}^{n} e^{-5\left|\frac{y_i-\hat{y_i}}{y_i}\right|}$$

Attention

- 数据缺省值为1(数据遗失),小企业的数据缺省对于结果没有根本性影响，但是大企业（企业id：1416）在11月份的数据缺省对于评分有着明显的影响. 可以考虑利用线性差值等方式填充缺省值.
- 评价公式有指数衰减, 最后的得分取决于是否能精准预测.
- 前100名的成绩为75%, 第一名的成绩为86%

Analysis

- 数据量较小, 适合新手
- 数据和时间有着非常强的关系, 考虑与时间序列有关的模型
- 每个企业用电量规律不同, 但1454家企业都单独训练不太现实, 可以考虑分类别进行训练
- 政策、经济、节假日、气象、自然灾害、电力事故、企业自身因素和季节日期等均可能对结果造成影响, 有些影响无法用模型体现(自然灾害\G20峰会\企业停产), 需要人工干预

Feature engineering(1)

- Date
  - day
  - month
  - year
  - day of week
  - day of year
  - if holiday
  - holiday_num

Feature engineering(2)

- Weather
  - highest temperature
  - lowest temperature
  - if sun
- Statistics
  - last month everage
  - the electricity consumption value 30 days ago

My solution

1. 收集天气数据, 假期数据
2. 利用pandas做数据分析, 生成训练集和测试集
3. 使用xgboost实现GBDT算法, 进行超参数选择和交叉验证
4. 结果分析, 数据可视化

Future work

1. Cluster analysis

- 7天为周期的企业 / 波动大的企业 / 平稳耗电的企业 (难以找到衡量标准)
- 大型企业 / 中小型企业 (可行, 注意到前三大的企业占据了半壁江山

Future work

2. ARIMA model
   ARIMA模型全称为自回归积分滑动平均模型(Autoregressive Integrated Moving Average Model,简记ARIMA)，是由博克思(Box)和詹金斯(Jenkins)于70年代初提出一著名时间序列预测方法

- ARIMA模型对于周期性强的企业应该更有效果
- 很多获奖选手使用了ARIMA + GBDT的混合模型

3. Visualization

Future work

4. Other problems

- AMS 2013-2014 Solar Energy Prediction Contest from Kaggle

5. Tips

- use python + pandas + numpy + jupyter for data mining
- Kaggle is better for the freshman

Reference

- [1]Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016.
- [2]Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." Annals of statistics (2001): 1189-1232.
- [3]Tso, Geoffrey KF, and Kelvin KW Yau. "Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks." Energy 32.9 (2007): 1761-1768.

Reference

- Pandas官方文档
- xgboost的github主页
- xgboost官方文档
- matplotlib官方文档
- 周志华<机器学习>