9.机器学习系统设计

笔记本: 日常
创建时间: 2019/11/18 9:13 **更新时间:** 2019/11/18 9:13
作者: 296645429@qq.com

推荐的构建模型方法

Recommended approach

- Start with a simple algorithm that you can implement quickly. Implement it and test it on your cross-validation data.
- Plot learning curves to decide if more data, more features, etc. are likely to help.
- Error analysis: Manually examine the examples (in cross validation set) that your algorithm made errors on. See if you spot any systematic trend in what type of examples it is making errors on.

错误分析-交叉验证集

Error Analysis

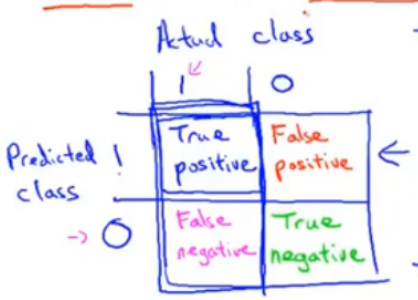
m_{CV} = 500 examples in cross validation set
Algorithm misclassifies 100 emails.
Manually examine the 100 errors, and categorize them based on:
(i) What type of email it is — pharma, replica, steal passwords, ...
(ii) What cues (features) you think would have helped the algorithm classify them correctly.

Pharma: 12
Replica/fake: 4
Steal passwords: 53
Other: 31
Deliberate misspellings: 5 (m0rgage, med1cine, etc.)
Unusual email routing: 16
Unusual (spamming) punctuation: 32

对于一些样本分类不均衡（比如测试癌症，10000例子只有10例是癌症）的度量评估方法：查准率和召回率，都是结果越高越好

Precision/Recall

$y = 1$ in presence of rare class that we want to detect



→ **Precision**
(Of all patients where we predicted $y = 1$, what fraction actually has cancer?)

$$\frac{\text{True positives}}{\#\text{predicted positive}} = \frac{\text{True positive}}{\text{True pos + False pos}}$$

→ **Recall**
(Of all patients that actually have cancer, what fraction did we correctly detect as having cancer?)

$$\frac{\text{True positives}}{\#\text{actual positives}} = \frac{\text{True positives}}{\text{True pos + False neg}}$$

$y = 0$
recall = 0

查准率与召回率的权衡方法，用F1 Score，结果越好越好

F_1 Score (F score)

How to compare precision/recall numbers?

	Precision(P)	Recall (R)	Average	F_1 Score
→ Algorithm 1	0.5	0.4	0.45	0.444
→ Algorithm 2	0.7	0.1	0.4	0.175
Algorithm 3	0.02	1.0	0.51	0.0392

Predict y=1 all the time

Average: $\frac{P+R}{2}$

F_1 Score: $2\frac{PR}{P+R}$

P=0 or R=0 ⇒ F-score = 0.
P=1 and R=1 ⇒ F-score = 1

大特征量+大训练集成就更好的模型

Large data rationale

Use a learning algorithm with many parameters (e.g. logistic regression/linear regression with many features; neural network with many hidden units). low bias algorithms.

→ $J_{train}(\theta)$ will be small.

Use a very large training set (unlikely to overfit) low variance

→ $J_{train}(\theta) \approx J_{test}(\theta)$

$J_{test}(\theta)$ will be small