COMP472: Artificial Intelligence
Machine Learning
Evaluation

- Russell & Norvig: Sections 19.4

Today

1. Introduction to ML
2. Naïve Bayes Classification
   a. Application to Spam Filtering
3. Decision Trees
4. (Evaluation

我们怎么评价machine-learning model效率

5. Unsupervised Learning)
6. Neural Networks
   a. Perceptrons
   b. Multi Layered Neural Networks

Data Sets

还是supervised learning的领域（有f(x)）

- How do you know if what you learned is correct?
- You run your classifier on a data set of unseen examples (that you did not use for training) for which you know the correct classification

指的就是test set

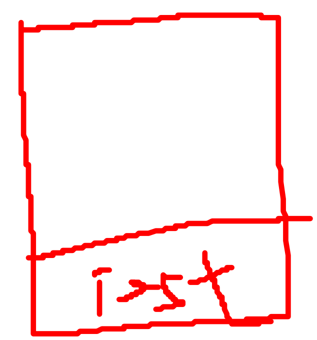
- Split data set into 3 sub-sets
  1. Actual **training** set (~80%)
  2. **Validation** set (~20%)
  3. **Test** set

~80%

~20%

training

no overlap



dataset被分为三部分，前两部分叫做training,，用来构建model,在构建完以后test set才会被投入使用将预测结果与实际结果进行比较

方法学

Standard Methodology

1. Collect a large set of examples (all with correct classifications)
2. Divide collection into training, validation and test set

build the model

Loop:

conditional probability, prior probability...

3. Apply learning algorithm to the training set to learn the parameters
4. Measure performance with the validation set, and adjust hyper-parameters* to improve performance

validation set用来评估我们通过training set建立的model,
并且通过hyper parameters进行tweak,adjust微调

5. Measure performance with the test set

34无限重复，，直到你得到了一个好model， 开始5

- DO NOT LOOK AT THE TEST SET until step 5.

Parameters:
basic values learned by the ML model. eg. ML学习的数据
- for NB: prior & conditional probabilities naive bayes:各种几率
- for DTs: features to split
- for ANNs: weights

Hyper-parameters: parameters used to set up the ML model. eg.
- for NB: value of delta for smoothing, smoothing曲线用的，1，0.5这种
- for DTs: pruning level
- for ANNs: nb of hidden layers, nb of nodes per layer...

Metrics 评判标准

- accuracy TEST SET里 我们模型准确预测的百分比
  - % of instances of the test set the algorithm correctly classifies
  - when all classes are equally important and represented

f(x),所有f(x)重要性相等

balanced,例如f(x)有80%class是狗，10%是猫，10%是老鼠，就不算represented/ballanced dataset.

- Recall, Precision & F-measure
  - when one class is more important and the others

如果没有满足equally important and balanced,我们要使用第二套标准

Accuracy

- % of instances of the test set the algorithm correctly classifies
- when all classes are equally important and represented
- problem:
  - when one class (eg. sick) is more important and the others
  - eg. when data set is unbalanced

	Target	system 1
X1	sick	ok ×
X2	sick	ok ×
X3	sick	ok ×
X4	sick	ok ×
X5	sick	ok ×
X6	ok	ok ✓
X7	ok	ok ✓
…	…	…
…	…	…
X500	ok	ok ✓
Accuracy		495/500 = 99% !

accuracy其实挺好的，但生活中没用，我们更想检查出生病的人（sick is more important），所以unbalanced不能用第一种

Recall, Precision

比例

- Recall: What proportion of the instances in the class of interest (eg. sick) are labelled correctly? 所有实际sick的instance有几个被标出来了
- Precision: What proportion of instances labeled with the class of interest (eg. sick) are actually correct? 所有标出来的sick有几个是实际sick

		Correct class	
		instance should be in class *C*	instance should <u>not</u> be in class *C*
Model prediction	instance is put in class *C*	True Positive (TP)	False Positive (FP)
	instance is <u>not</u> put in class *C*	False Negative (FN)	True Negative (TN)

$$\text{Precision} = \frac{TP}{TP+FP}$$

TP: *nb of instances that are in class C and that the model identified as class C*

TP+FP: *nb of instances that the model labelled as class C*

$$\text{Recall} = \frac{TP}{TP+FN}$$

TP: *nb of instances that are in class C and that the model identified as class C*

TP+FN: *All instances that are in class C*

Example

	Target	*system 1*	*system 2*	*system 3*
X1	sick	sick ✓	sick ✓	ok ×
X2	sick	ok ×	ok ×	sick ✓
X3	sick	ok ×	sick ✓	sick ✓
X4	sick	ok ×	sick ✓	sick ✓
X5	sick	ok ×	ok ×	sick ✓
X6	ok	ok ✓	ok ✓	sick ×
X7	ok	ok ✓	ok ✓	sick ×
..	ok	ok ✓	ok ✓	ok ✓
..	ok	ok ✓	ok ✓	ok ✓
X500	ok	ok ✓	ok ✓	ok ✓
Accuracy		496/500 = 99%	498/500 = 99.6%	497/500 = 99.4%
Precision		1/1 = 100%	3/3 = 100%	4/6 = 66.7%
Recall		1/5 = 20%	3/5 = 60%	4/5 = 80%

标出来的sick有几个是真sick

实际5个sick，标出来了1个，3个，4个

Which system is better?

2肯定比一好
但是23不好说，看我们具体需求有的时候我们觉得查错不要紧，尽量查出来，选3
有的时候我们觉得舍弃掉一些哪怕正确的是可以接受的，而精度更重要，选2

A Single Measure

- cannot take mean of P&R
  - if R = 50% P = 50% M = 50%
  - if R = 100% P = 10% ~~M = 55% (not fair)~~ 500个。50个sick，你全标sick，但实际上你这模型很差

1. take harmonic mean 调和平均数
   - which penalizes extreme values 他让极端情况的评分下降

使..不利

$$HM = \frac{2}{\frac{1}{R} + \frac{1}{P}}$$

HM is high only when both P&R are high

if R = 50% and P = 50% HM = 50%

if R = 100% and P = 10% ~~HM = 18.2%~~

2. if P and R should not have the same importance in the problem domain, take <u>weighted</u> harmonic mean 如果recall与precision有时候有一个侧重点，我们就改变系数

$$WHM = \frac{1}{\frac{1}{2}\frac{1}{R} + \frac{1}{2}\frac{1}{P}}$$ // if weight R = weight P = $\frac{1}{2}$

$$WHM = \frac{1}{\frac{1}{a}\frac{1}{R} + \frac{1}{b}\frac{1}{P}}$$ // if weight R = $\frac{1}{a}$ weight P = $\frac{1}{b}$ and $\frac{1}{a} + \frac{1}{b} = 1$

Weighted Harmonic Mean of P&R

$$WHM = \frac{1}{\frac{1}{a}\frac{1}{R} + \frac{1}{b}\frac{1}{P}}$$ // if weight R = $\frac{1}{a}$ weight P = $\frac{1}{b}$ and $\frac{1}{a} + \frac{1}{b} = 1$

1/a+1/b需要等于1

1. let $\mathrm{w_R} = \frac{\delta}{\delta+1}$ $\mathrm{w_P} = \frac{1}{\delta+1}$ // so that $\mathrm{w_R} + \mathrm{w_P} = \frac{\delta+1}{\delta+1} = 1$

$$WHM = \frac{1}{\left(\frac{\delta}{\delta+1}\right)\frac{1}{R} + \left(\frac{1}{\delta+1}\right)\frac{1}{P}} = \frac{\delta+1}{\delta\frac{1}{R} + 1\frac{1}{P}} = \frac{(\delta+1)PR}{\delta P + 1R}$$

可以换成以下形式

2. let $\delta = \beta^2$

$$WHM = \frac{(\beta^2+1)PR}{\beta^2 P + 1R}$$ // called the F-measure

可以改成这个形式，叫做F-measure

F-measure

- A weighted harmonic mean of precision and recall

$$F_\beta = \frac{(\beta^2+1)PR}{(\beta^2 P + R)}$$

F measure是一组measure，由β决定，例如F1 measure，F0.5measure...

- β represents the relative importance of recall to precision
  - when β = 1
    - F1 measure
    - precision & recall have same importance

β代表着recall相对重要性与precision相比，=1同样重要，大于1recall更重要
小于1precision更重要，

  - when β > 1
    - recall is given more weigth
    - e.g. F_2 measure, recall is considered 2x more important than precision

β等于2，说明recall两倍重要

  - when β < 1
    - precision is given more weigth
    - e.g. $F_{0.5}$ measure, precision is considered 2x more important than recall

Example

	Target	*system 1*	*system 2*	*system 3*
X1	sick	sick ✓	sick ✓	ok ×
X2	sick	ok ×	ok ×	sick ✓
X3	sick	ok ×	sick ✓	sick ✓
X4	sick	ok ×	sick ✓	sick ✓
X5	sick	ok ×	ok ×	sick ✓
X6	ok	ok ✓	ok ✓	sick ×
X7	ok	ok ✓	ok ✓	sick ×
..	ok	ok ✓	ok ✓	ok ✓
..	ok	ok ✓	ok ✓	ok ✓
X500	ok	ok ✓	ok ✓	ok ✓
Accuracy		496/500 = 99%	498/500 = 99.6%	497/500 = 99.4%
Precision		1/1 = 100%	3/3 = 100%	4/6 = 66.7%
Recall		1/5 = 20%	3/5 = 60%	4/5 = 80%
F1-measure 2PR/(P+R)	β等于1	2*100*20/ (100+20) = 33%	75%	72.9%

P, R and F for Multiclass Classification

上面的是偏重一项例如SICK的，如果我们偏重多个CLASS（鼠，猫，狗这种不同结果的F(X)），那么我们需要

- previous P, R and F are ok when 1 particular class interests us (eg. sick)
- What if several classes interest us?
- then
  - compute *per-class* P, R, F 把每个class的PRF算一遍
  - and to have a single measure for all classes: combine per-class F-measures via
    - macro F-measure, or
    - weighted-average F-measure

Per-class Precision & Per-class Recall

		Correct Class			
		Cat	Dog	Fish	Total
Predicted Class	Cat	4	6	3	13
	Dog	1	2	0	3
	Fish	1	2	6	9
	Total	6	10	9	25

FP.FN都是针对猫的

就是前面TP,FP,FN,TN一套，
红色圈的是猜了猫，实际也是猫，也就是好的TP
下划线就是猜了猫，实际上是狗或者实际上是鱼，那对于猫来说他是FP

X代表着FALSE NEGATIVE,我们猜没有猫（猜成别的了），实际上是猫，就是FN FALSE NEGATIVE

每一格代表的猜成X(行)，实际上是Y（列），横着的TOTAL代表一共猜了几次猫
竖着的TOTAL代表一共有几个猫

- precision of class Cat: 4/(4+6+3) = 30.8%
- precision of class Dog: 2/(1+2+0) = 66.7%
- precision of class Fish: 6/(1+2+6) = 66.7%

precision就是猜了几次对了几个

- recall of class Cat: 4/(4+1+1) = 66.7%
- recall of class Dog: 2/(2+6+2) = 20%
- recall of class Fish: 6/(3+0+6) = 66.7%

reccall就是有那么多你猜对了几个

Per-class F1-measure

	Precision	Recall	F1
Cat	30.8%	66.7%	42.1%
Dog	66.7%	20.0%	30.8%
Fish	66.7%	66.7%	66.7%

F1 = 2PR/(P+R)

- F1 of class Cat: (2 x .308 x .667) / (.308 + .667) = 0.421
- F1 of class Dog: (2 x .667 x .200) / (.667 + .200) = 0.308
- F1 of class Fish: (2 x .667 x .667) / (.667 + .667) = 0.667

有了PR求出F1

Macro and Weighted-Average Measures

macro precision,
macro recall,
macro F1

	Precision	Recall	F1
Cat	30.8%	66.7%	42.1%
Dog	66.7%	20.0%	30.8%
Fish	66.7%	66.7%	66.7%
average	(30.8+66.7+66.7) / 3 = 54.7%	(66.7 + 20.0 + 66.7) / 3 = 51.1%	(42.1+30.8+66.7) = 46.5%
weighted-average	(6x30.8 // 6 cat + 10x66.7 // 10 dog + 9x66.7) // 9 fish / (6+10+9) // 25 samples = 58.1%	(6x66.7 + 10x20.0 + 6x66.7) / 25 = 48.0%	(6x42.1 + 10x30.8 + 6x66.7) / 25 = 46.4%

weighted-averaged precision,
weighted-averaged recall,
weighted-averaged F1

- To combine measures into a single one, we can:
  - take simple average macro:简单的就求平均
    - --> macro precision, macro recall, macro F1
  - take weighted average 实际上我们有6只猫，10只狗，9只鱼，每个人都有一个系数6/25，10/25，9/25
    - ie. weight the average based on the nb of samples from each class
    - --> weighted averaged precision, weighted averaged recall, weighted averaged F1

Confusion Matrix 也叫contingency table

- to do an error analysis and find out where the model went wrong ?
- aka contingency table 用来分析错误，找到Model哪儿出问题了
- eg. 6 classes, 100 test instances 就是第14页那个

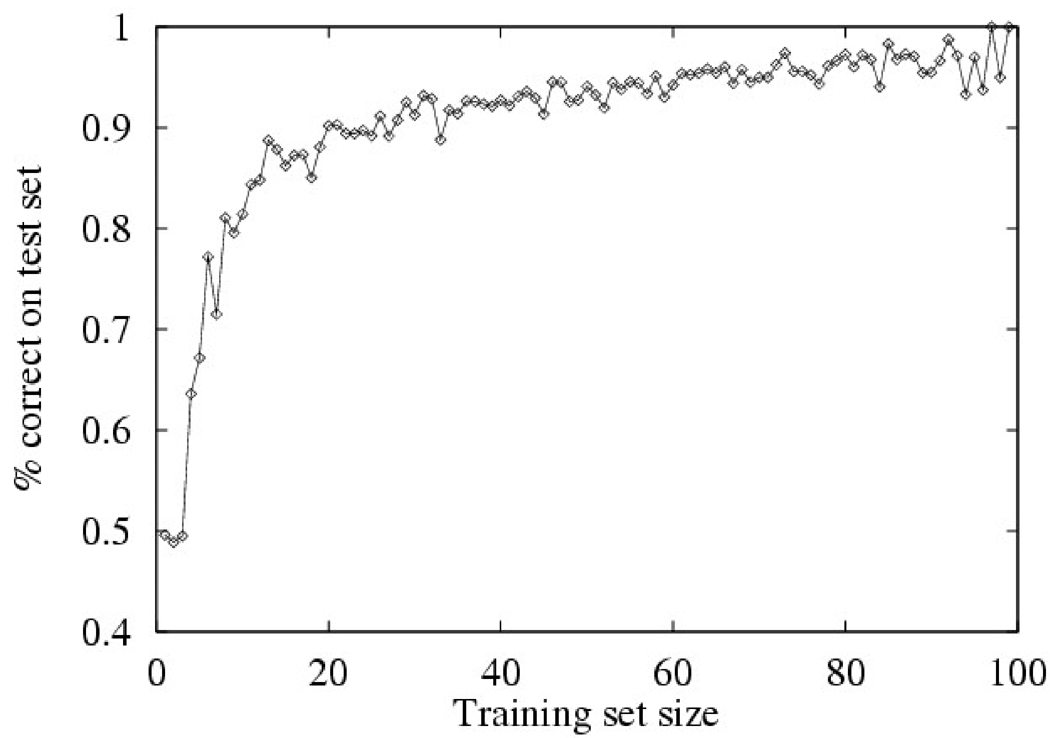
		Correct Class						
		C1	C2	C3	C4	C5	C6	Total
Predicted Class	C1	10✓	3×	0	0	3×	0	*16*
	C2	0	12✓	3×	4×	0	0	*19*
	C3	0	1×	9✓	2×	1×	2×	*15*
	C4	0	1×	3×	5✓	2×	0	*11*
	C5	0	0	3×	2×	10✓	3×	*18*
	C6	0	0	5×	0	5×	11✓	*21*
	Total	*10*	*17*	*23*	*13*	*21*	*16*	*100*

实际猜了16次C1

实际有10个C1

A Learning Curve

数据越多，the more you learn，但是到了后面加起来很微弱



理论上你可以得到一个完全精确地set，但实际上很难，因为生活中dataset很难完全正确，甚至你用同样的training set构建模型，再用同样的set作为test set，也很难达到1

- Size of training set
  - the more, the better
  - but after a while, not much improvement…

source: Mitchell (1997)

Some Words on Training

- In all types of learning… watch out for:
  - Noisy input
  - Overfitting/underfitting the training data

Noisy Input

- In all types of learning… watch out for:
  - Noisy Input:
    - Two examples have the same feature-value pairs, but different outputs

Size	Color	Shape	Output
Big	Red	Circle	+
Big	Red	Circle	-

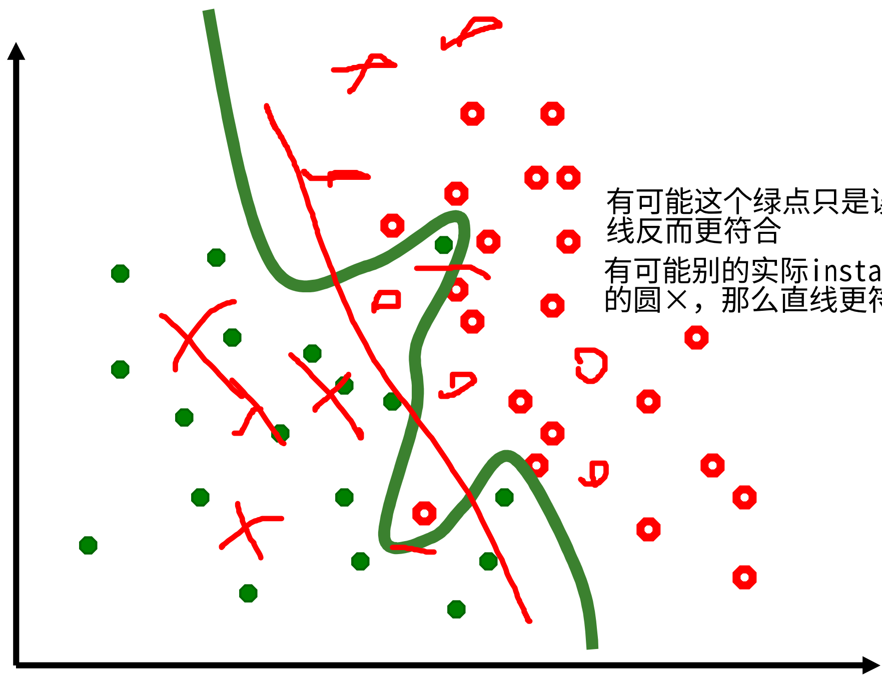
    - Some values of features are incorrect or missing (ex. errors in the data acquisition)
    - Some relevant attributes are not taken into account in the data set

可能第二个shape是circle，，也可能Output是+，或者是4TH feature没有被考虑到
这也是为什么即使你用你的training set作为test set，也可能做不到100%精确率

Overfitting

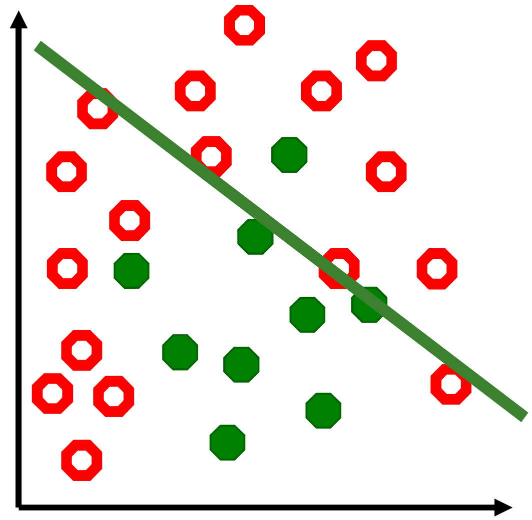
有很多实际上并不相关的feature，在这个training data里面因为巧合显得相关，我们如果面面俱到，会产生overfit，过于符合training data了

- If a large number of irrelevant features are there, we may find meaningless regularities in the data that are particular to the training data but irrelevant to the general problem.
- Complicated boundaries *overfit* the data
- they are too tuned to the particular training data at hand 过于符合//吉他里的tune调音
- They do not *generalize* well to the new data 概括
- Extreme case: "rote learning" 死记硬背
- Training error is low training error少
- Testing error is high



Underfitting

- We can also underfit data, i.e. find a decision boundary that is too simple
- Model is not expressive enough (not enough features, or not enough capacity)
- eg. There is no way to fit a linear decision boundary so that the training examples are well separated



过于简单，testing training两个都达不到

- Training error is high
- Testing error is high

Cross-validation

- K-fold cross-validation　　　　K就是折叠成几分
  - run k experiments, each time you test on 1/k of the data, and train on the rest
  - than you average the results
- ex: 10-fold cross validation
  1. Collect a large set of examples (all with correct classifications)
  2. Divide collection into two disjoint sets: training (90%) and test (10% = 1/k)
  3. Apply learning algorithm to training set
  4. Measure performance with the test set
  5. Repeat steps 2-4, with the 10 different portions
  6. Average the results of the 10 experiments

exp1:	train									test
exp2:	train								test	train
exp3:	train							test	train	
…	…									

Today

1. Introduction to ML
2. Naïve Bayes Classification
   a. Application to Spam Filtering
3. Decision Trees
4. (Evaluation
5. Unsupervised Learning)
6. Neural Networks
   a. Perceptrons
   b. Multi Layered Neural Networks

Up Next

1. Introduction to ML
2. Naïve Bayes Classification
    a. Application to Spam Filtering
3. Decision Trees
4. (Evaluation
5. Unsupervised Learning)
6. Neural Networks
    a. Perceptrons
    b. Multi Layered Neural Networks