<머신러닝> Date. No.

Chapter 02. 모델 평가 및 선택

2.1 경험 오차 및 과적합

m개 샘플, a개 샘플의 오분류 ($m > a$)

E (error rate) $= \frac{a}{m}$, Accuracy (정확도) $= 1 - \frac{a}{m}$

- Training error / Empirical error
- * Generalization error

"새로운 샘플 데이터를 대상으로 좋은 성능을 발휘하는 학습기" → 우리의 목표!

↳ 이를 위해 (Train) 데이터에서 (최적) 의 잠재적인 "보편규칙"을 찾아야함!

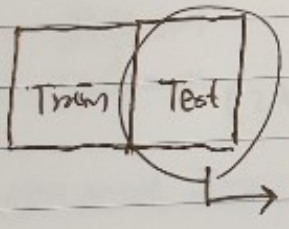
과적합: 학습을 과도하게 잘하면 → 일반화 성능↓ "Overfitting"

또는 그렇지 못하면 → 학습이 제대로 X "Underfitting"

Then, 현실 프로젝트에 직면하여, ① 어떤 학습 알고리즘을 사용?

② 어떠한 파라미터로 선택?

→ Model Selection



→ 학습기의 일반화 오차에 대해 평가!! * train과 같으면 X

Test error ≈ Generalization error.

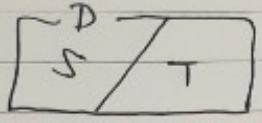
Date. 2.2 평가방법

Assume that,

$$D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$$

Data set D를 어떻게 적절히 훈련셋 S와 테스트셋 T로 나눌 것인가?

① Hold-out



간단히 생각해 봅시다!!

* 데이터의 분포가 같게 나눠야 함! → 데이터 편중으로 인한 편향 결과를 얻을 수 있음.

↓

예) 층화 추출법 (Stratified Sampling) 등 이용.

↳ 층화 추출에도 문제가 존재한다고 함.

클래스 불균형, 총 샘플 수 이슈 등 고려해야 할 문제들에서 층화하는 것이 어렵다고 함!

* 일반적으로 나눠진 데이터에 어떻게 추출하냐에 따라 (train 및 평가 결과가 달라짐.

+ 비율의 이슈로 존재! (일반적으로 2/3 ~ 4/5)

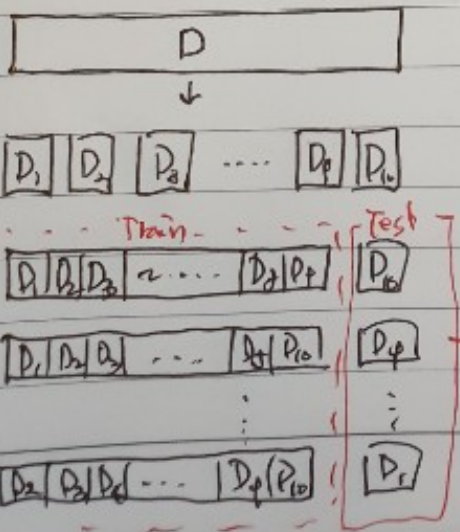
② Cross Validation

$$D = \bigcup_{i=1}^{k} D_i, \quad D_i \cap D_j = \emptyset \ (i \neq j)$$

↳ 최대한 데이터의 분포가 나뉘지게!

층화추출!!

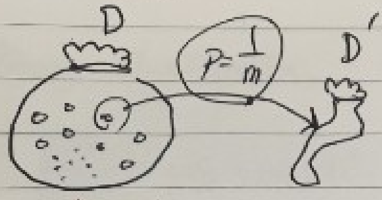
k-fold cross validation. (일반적으로 K=10)



+ LOOCV (Leave-One-Out CV)

k=m when |D|=m.

③ Bootstrapping



m개 sample

임의로 sample 1개를 복사!

(뽑힌 sample이 다시 추출될 수 있음!)

× m번 반복!

* 여러번 뽑힌 sample도 있고
 한번도 뽑히지 않는 sample도 존재할 것!

$$\lim_{m \to \infty} \left(1 - \frac{1}{m}\right)^m = \frac{1}{e} \approx 0.368$$

∴ 데이터가 충분히 클 때, D중 36.8% 의 샘플은 D'로 들어가지 못한다.
 ($m \to \infty$)

63.2 : 36.8 → Out-of-Bag
train test

중복 추출되어
$|D'| = m$.

언제 쓰기에 좋은가? ① 데이터 세트가 비교적 작을 때
② 훈련 / 테스트 셋으로 분류하기 힘들 때
+ 초기에 다양한 S를 구성할 수 있기 때문에 (∵ 확률적)
앙상블 기법에 적용하기 좋음!!

But, 초기 데이터 분포와 다를 가능성 多.

데이터가 충분할 경우엔 Hold-out / CV 를 더 선호.

Date. No.

④ 파라미터 튜닝과 최종 모델

- 알고리즘 파라미터, Hyper Parameter
- 모델 파라미터

↳ 튜닝을 위해 Validation Set을 사용하기도 한다!

2.3 모델 성능측정!

Performance Measure.

$$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$$

f : learner. Compare $f(x)$, y

〈Mean Squared Error〉 → 주로 Regression! (예시중요)

$$E(f; D) = \frac{1}{m} \sum_{i=1}^{m} (f(x_i) - y_i)^2$$

Generally

$$E(f; D) = \int_{x \sim D} (f(x) - y)^2 p(x)\, dx$$

〈오차율, 정확도〉

$$E(f; D) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}(f(x_i) \neq y_i)$$

$$acc(f; D) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}(f(x_i) = y_i) = 1 - E(f; D)$$

Generally

$$E(f; D) = \int_{x \sim D} \mathbb{I}(f(x) \neq y)\, p(x)\, dx$$

$$acc(f; D) = \int_{x \sim D} \mathbb{I}(f(x) = y)\, p(x)\, dx = 1 - E(f; D)$$

Date No.

< Recall, Precision, F1-score >

In Binary-Classification,

Predicted → Confusion Matrix 헷갈리지 말자 !!

		예측	
		양성	음성
실제	양성	TP	FN
	음성	FP	TN

$$\text{precision} = \frac{TP}{TP + FP}$$: ~~예측은 실제에서 정확히 맞는 것~~

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

Trade-off 관계 !!

⇒ 때문에, precision과 recall을 종합적으로 고려할 척도 필요.

① BEP (Break-even point)

$$P = R$$

② F1-score

$$F_1 = \frac{2PR}{P+R} = \frac{2 \times TP}{m + TP - TN}$$

$$\frac{1}{F_1} = \frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right)$$

③ General F_1

$$F_\beta = \frac{(1+\beta^2) \times P \times R}{(\beta^2 \times P) + R} \qquad \left(\frac{1}{F_\beta} = \frac{1}{1+\beta^2}\left(\frac{1}{P} + \frac{\beta^2}{R}\right)\right)$$

$\beta > 1$ ⇒ R (재현율)에 가중!

$\beta < 1$ ⇒ P (정밀도)에 가중!

Date. No.

Q) 다중의 혼동 행렬 Confusion Matrix 를 통해, 다중분류의 성능?

① 모든 Confusion Matrix 에 대해 P, R 계산 후 평균값 계산

$$\text{macro-P} = \frac{1}{n}\sum_{i=1}^{n} P_i = \underset{i}{\text{mean}}(P_i)$$

$$\text{macro-R} = \frac{1}{n}\sum_{i=1}^{n} R_i = \underset{i}{\text{mean}}(R_i)$$

$$\text{macro-F}_1 = \frac{2 \cdot \text{macro-P} * \text{macro-R}}{\text{macro-P} + \text{macro-R}}$$

② 각 Confusion matrix의 원소값 (TP, TN, FP, FN) 에 평균을 계산

$$\text{micro-P} = \frac{\overline{TP}}{\overline{TP} + \overline{FP}}$$

$$\text{micro-R} = \frac{\overline{TP}}{\overline{TP} + \overline{FN}}$$

* TP, TN, FP, FN 헷갈리지 말자 !!

뭘 만들고 싶은가? / 안풀린다고, 못해도 안된다고 좌절하지 말기!!

Date. No.

< ROC, AUC >

Cut point

ROC (Receiver Operating Characteristics)

→ 2차 대전 레이더 신호 분석기술, 심리학, 의학분야 에서도 널리 활용.

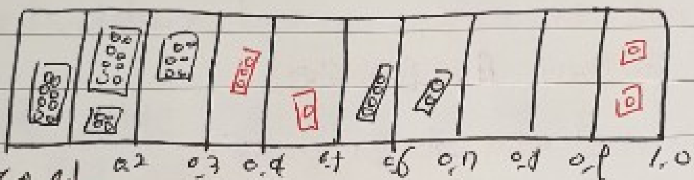
$$TPR\ (\text{True-Positive rate}) = \frac{TP}{TP + FN}$$

$$FPR\ (\text{False-Positive rate}) = \frac{FP}{TN + FP}$$

		Predict 양성	음성
Actual	양성	TP	FN
	음성	FP	TN

출처: 김태영 블로그

홀수 : P / 짝수 : N

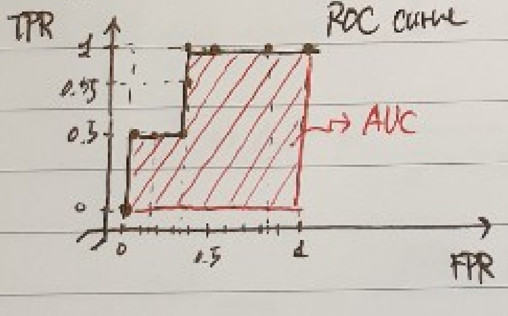


모두 홀수불량이라 예측!

모두 짝수불량이라 예측!

홀수불량 임계값	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
TP 맞은 홀수	4	4	4	4	3	2	2	2	2	2	0
TN 〃 짝수	0	1	3	4	4	4	5	6	6	6	6
FN 틀린 홀수	0	0	0	0	1	2	2	2	2	2	4
FP 〃 짝수	6	5	3	2	2	2	1	0	0	0	0
TPR	1.0	1.0	1.0	1.0	0.75	0.5	0.5	0.5	0.5	0.5	0.0
FPR	1.0	0.84	0.5	0.34	0.34	0.34	0.16	0.0	0.0	0.0	0.0

ROC curve

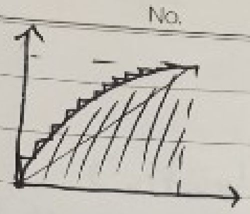


※ 참고 !!

- 민감도 (Sensitivity) = TPR
- 특이도 (Specificity) = 1 - FPR = TNR (True Negative Rate)

Date. No.

하면된다!



$$AUC = \sum_{i=1}^{m-1}(x_{i+1}-x_i)\cdot\frac{(y_i+y_{i+1})}{2}$$

$$\fallingdotseq \sum_{i=1}^{m-1}(x_{i+1}-x_i)\,y_{i+1}$$

(y_i, y_{i+1}가 달라지는 순간... 그 있어가3)

↓

FPR이 변할 때 TPR도 변하는 경우가 있는가?

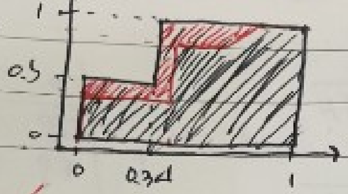
↓

있을 수 있지!

threshold 가운데 0.1 둘다 들어 있으면

⌐ 가 아닌 ⟋ 로 변할 수 있다!

ex)

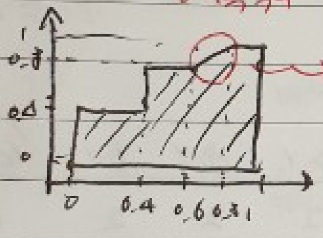


$0.34\times0.3 + 0.68\times1 = 0.83$

앞의 예제에서 0.1 ~ 0.2 의

정수 음성 하나를 흑수로 바꾸면,

예측력이 낮아졌으니 AUC 감소가 당연!



⟶ 이 관 때문에 위와 같이 수식 사용!

- AUC는 "샘플 예측의 배열 순서 품질"을 고려.

m^+의 양성값, m^-의 음성값, D^+, D^-; 양/음성값의 집합

$$\ell_{rank} = \frac{1}{m^+m^-}\sum_{x^+\in D^+}\sum_{x^-\in D^-}\left(\mathbb{I}\left(f(x^+)<f(x^-)\right)+\frac{1}{2}\mathbb{I}\left(f(x^+)=f(x^-)\right)\right)$$

순서손실 (ℓ_{rank}) / ~~중복~~ 번갈화의 ($\frac{1}{2}$)

↓

ROC 곡선 위의 면적, 거짓 양성률!

$$AUC = 1 - \ell_{rank}$$

Date No.

< 비용민감 오차율과 비용곡선 >

In real, '서로 다른 종류의 오차가 가져오는 결과가 다르다는 현실'.

↓

이러한 서로 다른 종류의 cost에 대한 균형을 맞추기 위해

비균등 비용 unequal cost 사용.

- Binary classification

cost matrix (비용 행렬) $cost_{ij}$ = 실제 class i를 예측 class j로 분류할 때의 비용

Real \ predict	0 type	1 type
0 type	0	$cost_{01}$
1 type	$cost_{10}$	0

✓ 앞서 정의한 오차율은 '오차 횟수'로 직접 계산되고

오차가 입은 영향력에 대해 고려 X

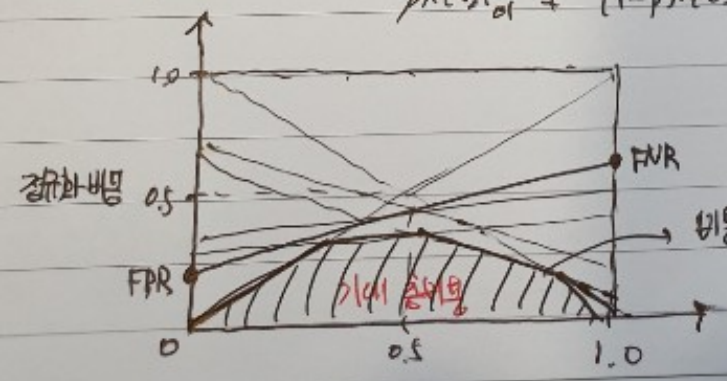
★ 비용민감 cost-sensitive 오차율

$$E(f; D; cost) = \frac{1}{m}\left(\sum_{x_i \in D^+} \mathbb{I}(f(x_i) \neq y_i) \times cost_{01} + \sum_{x_i \in D^-} \mathbb{I}(f(x_i) \neq y_i) \times cost_{10} \right)$$

★ 비용곡선 cost curve (ROC 곡선 대체!)

$$P(+)cost = \frac{p \times cost_{01}}{p \times cost_{01} + (1-p) \times cost_{10}} \quad (\text{가로축})$$

$$cost_{norm} = \frac{FNR \times p \times cost_{01} + FPR \times (1-p) \times cost_{10}}{p \times cost_{01} + (1-p) \times cost_{10}} \quad \text{where } FNR = 1 - TPR$$



→ 학습 알고리즘 일반화 성능을 해석할 수 있는 중요한 도구.

Date.
No.

2.5 Bias and Variance

편향-분산 분해 bias-variance decomposition.

$\bar{f}(x) = E_D[f(x;D)]$; 학습 알고리즘의 기대 예측.
(f(x;D): 예측값)

$var(x) = E_D[(f(x;D) - \bar{f}(x))^2]$ 데이터셋 바뀔 때의 학습 성능 변화 측정

$\varepsilon^2 = E_D[(y_D - y)^2]$ 기대 일반화 오차의 lower bound.

$bias^2(x) = (\bar{f}(x) - y)^2$ 학습 알고리즘의 기대 예측과 실제 결과의 차이.

Assume that $E_D[y_D - y] = 0$.

$$E(f;D) = E_D[(f(x;D) - y_D)^2]$$

일반화 기대 오차 분해

$$= E_D[(f(x;D) - \bar{f}(x) + \bar{f}(x) - y_D)^2]$$

(∵ 기대값은)

$$= E_D[(f(x;D) - \bar{f}(x))^2] + E_D[(\bar{f}(x) - y_D)^2]$$

$$+ E_D[2(f(x;D) - \bar{f}(x))(\bar{f}(x) - y_D)]$$

(∵ $\bar{f}(x) = E_D[f(x;D)]$ 로부터 $f(x;D)$ 상관이 없어짐) → 0

$$= E_D[(f(x;D) - \bar{f}(x))^2] + E_D[(\bar{f}(x) + y - y - y_D)^2]$$

$$= E_D[(f(x;D) - \bar{f}(x))^2] + E_D[(\bar{f}(x) - y)^2]$$

(∵ By Assumption)

$$+ E_D[(y - y_D)^2] + E_D[2(\bar{f}(x) - y)(y - y_D)]$$

$$= var(x) + bias^2(x) + \varepsilon^2$$

bias → bias-variance dilemma! (trade-off)