**3. ROC space**

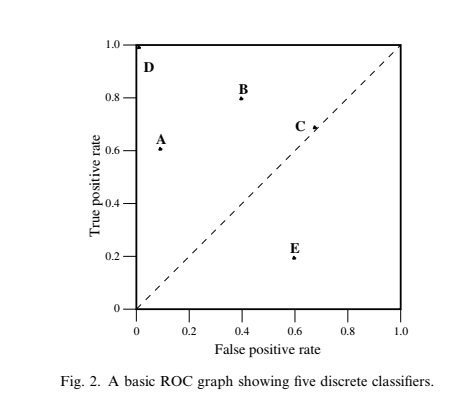
대각선 y = x는 클래스를 무작위로 추측하는 전략을 나타냅니다. 예를 들어 분류기가 양성 클래스의 절반을 무작위로 추측하면 양성의 절반과 음성의 절반이 정확할 것으로 예상할 수 있습니다. 이것은 ROC 공간에서 포인트 (0.5,0.5)를 산출합니다.

시간의 90%에서 포지티브 클래스를 추측하면 포지티브의 90%가 정확할 것으로 예상할 수 있지만 잘못된 포지티브 비율도 90%로 증가하여 ROC 공간에서 (0.9,0.9)를 산출합니다. 따라서 랜덤 분류기는 양성 클래스를 추측하는 빈도를 기반으로 대각선에서 앞뒤로 "슬라이딩"하는 ROC 포인트를 생성합니다.

이 대각선에서 위쪽 삼각형 영역으로 이동하기 위해 분류기는 데이터의 일부 정보를 이용해야 합니다. 그림 2에서 Cs 성능은 사실상 무작위입니다. (0.7,0.7)에서 C는 시간의 70%에서 양성 클래스를 추측한다고 말할 수 있습니다.

오른쪽 아래 삼각형에 나타나는 분류기는 임의 추측보다 성능이 떨어집니다. 따라서 이 삼각형은 일반적으로 ROC 그래프에서 비어 있습니다. 분류기를 부정하는 경우(즉, 모든 인스턴스에서 분류 결정을 반대로 하면) 참 양성 분류는 거짓 음성 오류가 되고 거짓 양성은 참 음성이 됩니다. 따라서 오른쪽 아래 삼각형의 점을 생성하는 분류기는 왼쪽 위 삼각형의 점을 생성하기 위해 무효화될 수 있습니다. 그림 2에서 E는 무작위보다 훨씬 나쁜 성능을 보이며 실제로 B의 부정입니다. 대각선에 있는 분류기는 클래스에 대한 정보가 없다고 말할 수 있습니다. 대각선 아래의 분류기는 유용한 정보를 가지고 있다고 할 수 있지만 정보를 잘못 적용하고 있습니다(Flach and Wu, 2003).

분류기 성능이 무작위보다 약간 더 나은 것처럼 보이는 ROC 그래프가 주어지면 ''이 분류기 성능이 진정으로 유의미한가요 아니면 우연히 무작위보다 더 나은가요?''라고 묻는 것이 당연합니다. 이에 대한 결정적인 테스트는 없습니다. 그러나 Forman(2002)은 ROC 곡선을 사용하여 이 문제를 해결하는 방법론을 보여주었습니다.



**4. Curves in ROC space**

결정 트리 또는 규칙 세트와 같은 많은 분류기는 클래스 결정(즉, 각 인스턴스에서 Y 또는 N)만 생성하도록 설계되었습니다.

이러한 이산 분류기를 테스트 세트에 적용하면 단일 혼동 행렬이 생성되며, 이는 차례로 하나의 ROC 포인트에 해당합니다. 따라서 이산 분류기는 ROC 공간에서 단일 지점만 생성합니다.

Naive Bayes 분류기 또는 신경망과 같은 일부 분류기는 인스턴스가 클래스의 구성원인 정도를 나타내는 숫자 값인 인스턴스 확률 또는 점수를 자연스럽게 산출합니다. 이러한 값은 엄격한 확률일 수 있으며, 이 경우 표준 확률 정리를 따릅니다. 또는 일반적이고 보정되지 않은 점수일 수 있습니다. 이 경우 보유하는 유일한 속성은 점수가 높을수록 확률이 높다는 것입니다. 출력이 적절한 확률이 아닐 수도 있다는 사실에도 불구하고 우리는 둘 다 확률적 분류기라고 부를 것입니다.

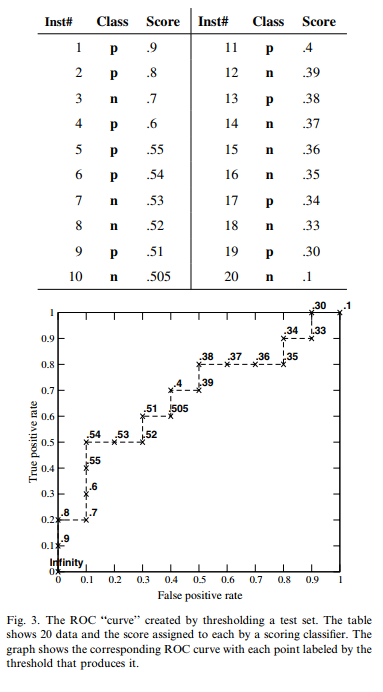
이러한 순위 또는 점수 분류기는 임계값과 함께 사용하여 이산(이진) 분류기를 생성할 수 있습니다. 분류기 출력이 임계값보다 높으면 분류기는 Y를 생성하고 그렇지 않으면 N을 생성합니다. 각 임계값은 ROC 공간에서 다른 포인트를 생성합니다. . 개념적으로 임계값을 1에서 +1로 변경하고 곡선을 추적하는 것을 상상할 수 있습니다.

ROC 공간을 통해 계산적으로 이것은 ROC 곡선을 생성하는 데 좋지 않은 방법이며 다음 섹션에서는 보다 효율적이고 신중한 방법을 설명합니다.

그림 3은 20개 인스턴스의 테스트 세트에 대한 ROC '곡선'의 예를 보여줍니다. 10개의 긍정 및 10개의 부정 인스턴스가 그래프 옆의 표에 표시됩니다. 유한한 인스턴스 집합에서 생성된 모든 ROC 곡선은 실제로 인스턴스 수가 무한대에 가까워지면 실제 곡선에 접근하는 계단 함수입니다. 그림 3의 단계 함수는 각 점 유도를 이해할 수 있도록 설정된 매우 작은 인스턴스에서 가져온 것입니다. 그림 3의 표에서 인스턴스는 점수별로 정렬되고 ROC 그래프의 각 포인트는 이를 생성하는 점수 임계값으로 레이블이 지정됩니다. 임계값이 +1이면 포인트(0,0)가 생성됩니다. 임계값을 0.9로 낮추면 첫 번째 양성 인스턴스가 양성으로 분류되어 (0,0.1)이 생성됩니다. 임계값이 더 감소함에 따라 곡선은 위로 올라가 오른쪽으로 올라가 임계값이 0.1인 (1,1)에서 끝납니다. 이 임계값을 낮추는 것은 그래프의 '보수적''에서 '자유주의적'' 영역으로 이동하는 것과 같습니다.

테스트 세트는 매우 작지만 분류기에 대해 몇 가지 잠정적인 관찰을 할 수 있습니다. 그래프의 보수적인 영역에서 더 잘 수행되는 것으로 보입니다. (0.1,0.5)의 ROC 포인트는 가장 높은 정확도(70%)를 생성합니다. 이것은 분류기가 가능한 부정을 식별하는 것보다 가능성이 있는 긍정을 식별하는 데 더 우수하다고 말하는 것과 같습니다. 또한 분류기 최고의 정확도는 균형 분포에서 기대할 수 있는 P0.5가 아니라 임계값 P0.54에서 발생합니다.

다음 섹션에서는 이 현상에 대해 설명합니다.



4.1. Relative versus absolute scores 상대 점수 대 절대 점수

ROC 그래프의 중요한 점은 분류기가 좋은 상대 인스턴스 점수를 생성하는 능력을 측정한다는 것입니다. 분류기는 정확하고 보정된 확률 추정치를 생성할 필요가 없습니다. 긍정적인 경우와 부정적인 경우를 구별하는 역할을 하는 상대적으로 정확한 점수만 생성하면 됩니다.

Naive Bayes 분류기에서 가져온 그림 4의 간단한 인스턴스 점수를 고려하십시오. 가상 클래스(득점 > 0.5이면 Y, 그렇지 않으면 N)를 실제 클래스와 비교하면 분류기가 인스턴스 7과 8을 잘못 가져와 80%의 정확도를 제공한다는 것을 알 수 있습니다. 그러나 그림의 왼쪽에 있는 ROC 곡선을 고려하십시오. 곡선은 (0, 0)에서 (0,1)까지 수직으로 상승한 다음 수평으로 (1, 1)까지 상승합니다.

이것은 이 테스트 세트에 대한 완벽한 분류 성능을 나타냅니다. 왜 불일치가 있습니까?

설명은 각각이 무엇을 측정하는지에 있습니다. ROC 곡선은 분류기가 부정적인 인스턴스에 상대적인 긍정적 인스턴스의 순위를 매기는 능력을 보여주며, 이 능력에서 실제로 완벽합니다. 정확도 메트릭은 임계값(점수 > 0.5)을 부과하고 점수와 관련하여 결과 분류를 측정합니다. 정확도 측정은 점수가 적절한 확률인 경우 적절하지만 그렇지 않습니다. 이것을 말하는 또 다른 방법은 실제 확률처럼 점수가 제대로 보정되지 않았다는 것입니다.

ROC 공간에서 0.5 임계값을 부과하면 그림 4에서 동그라미로 표시된 '정확도 지점'으로 지정된 성능이 나타납니다. 이 작동 지점은 차선책입니다. 훈련 세트를 사용하여 p(p) = 6/10 = 0.6에 대한 사전을 추정하고 이를 임계값으로 사용할 수 있지만 여전히 차선의 성능(90% 정확도)을 생성합니다.

이 현상을 제거하는 한 가지 방법은 분류기 점수를 보정하는 것입니다. 이를 위한 몇 가지 방법이 있습니다(Zadrozny and Elkan, 2001). 또 다른 접근 방식은 상대적인 성능을 기반으로 작동 지점을 선택하는 ROC 방법을 사용하는 것이며 이를 수행하는 방법도 있습니다(Provost and Fawcett, 1998, 2001). 이러한 후자의 방법은 섹션 6에서 간략하게 논의됩니다.

상대 스코어링의 결과는 분류자 스코어가 모델 클래스 간에 비교되어서는 안 된다는 것입니다. 한 모델 클래스는 [0,1] 범위의 점수를 생성하도록 설계되고 다른 모델 클래스는 [1,+1] 또는 [1,100]의 점수를 생성합니다.

공통 임계값에서 모델 성능을 비교하는 것은 의미가 없습니다.

4.2. 클래스 스큐

ROC 곡선은 매력적인 속성을 가지고 있습니다. 클래스 분포의 변화에 둔감합니다. 테스트 세트에서 양성 대 음성 인스턴스의 비율이 변경되면 ROC 곡선은 변경되지 않습니다. 왜 그런지 알아보려면 그림 1의 정오분류표를 살펴보십시오.

클래스 분포(양수 대 음수 인스턴스의 비율)는 왼쪽(+) 열과 오른쪽( ) 열의 관계입니다. 두 열의 값을 모두 사용하는 성능 메트릭은 본질적으로 클래스 편향에 민감합니다. 정확도, 정밀도, 리프트 및 F 점수와 같은 측정항목은 혼동 행렬의 두 열 값을 사용합니다. 클래스 분포가 변경되면 기본 분류기 성능이 변경되지 않더라도 이러한 측정값도 변경됩니다. ROC 그래프는 tp rate 및 fp rate를 기반으로 하며 각 차원은 엄격한 기둥 비율이므로 클래스 분포에 의존하지 마십시오.

일부 연구자들에게는 큰 클래스 왜곡과 클래스 분포의 큰 변화가 인위적이고 비현실적으로 보일 수 있습니다.

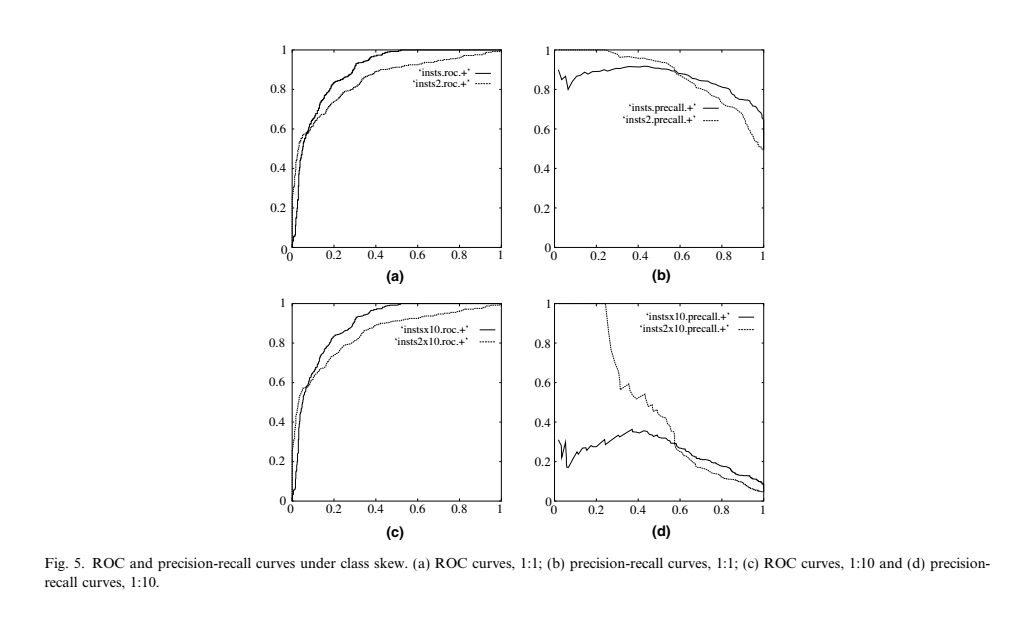
그러나 101과 102의 클래스 스큐는 실제 영역에서 매우 일반적이며 일부 영역에서는 최대 106의 스큐가 관찰되었습니다(Clearwater and Stern, 1991;

Fawcett 및 Provost, 1996; Kubat et al., 1998; 사이타와 네리, 1998). 클래스 분포의 실질적인 변화도 비현실적이지 않습니다. 예를 들어, 의학적 의사결정에서 전염병은 질병의 발병을 유발할 수 있습니다.

시간이 지남에 따라 증가합니다. 사기 탐지에서 사기의 비율은 매달 그리고 장소에 따라 크게 다릅니다(Fawcett and Provost, 1997). 제조 관행의 변경으로 인해 제조 라인에서 생산되는 불량품의 비율이 증가하거나 감소할 수 있습니다.

이러한 각각의 예에서 클래스의 보급은 클래스의 기본 특성, 즉 대상 개념을 변경하지 않고 크게 변경될 수 있습니다.

검색(분류) 성능을 평가하기 위한 정보 검색에서는 정밀도와 회상이 일반적입니다(Lewis, 1990, 1991). Precision-recall 그래프는 정적 문서 세트를 가정할 수 있는 곳에서 일반적으로 사용됩니다. 그러나 웹 페이지 검색과 같은 동적 환경에서도 사용됩니다. 여기서 쿼리와 관련이 없는 페이지의 수(N)는 P보다 수십 배 더 많고 웹 페이지가 생성됨에 따라 시간이 지남에 따라 꾸준히 증가할 것입니다.



클래스 스큐의 효과를 보기 위해 ROC 곡선과 정밀 재현율 곡선을 사용하여 평가된 두 개의 분류기를 보여주는 그림 5의 곡선을 고려하십시오. 그림 5a와 b에서 테스트 세트는 균형 잡힌 1:1 클래스 분포를 가지고 있습니다. 그래프 5c와 d는 동일한 도메인에 있는 동일한 두 분류자를 보여주지만 부정적인 인스턴스의 수는 10배 증가했습니다. 분류자와 기본 개념은 변경되지 않았습니다. 클래스 분포만 다릅니다. 그림 5a와 c의 ROC 그래프는 동일한 반면 그림 5b와 d의 정밀 재현율 그래프는 상당히 다릅니다. 어떤 경우 분류기의 성능이 더 우수하다는 결론은 분포가 이동함에 따라 바뀔 수 있습니다.

4.3. 점수 분류기 만들기

많은 분류기 모델은 개별적입니다. 각 테스트 인스턴스에서 클래스 레이블만 생성하도록 설계되었습니다.

그러나 우리는 종종 단일 점 대신 분류기에서 전체 ROC 곡선을 생성하기를 원합니다. 이를 위해 클래스 레이블이 아닌 분류기에서 점수를 생성하려고 합니다. 이러한 점수를 생성하는 방법에는 여러 가지가 있습니다.

많은 이산 분류기 모델은 그들이 보관하는 인스턴스 통계에서 '내부를 살펴봄'으로써 점수 분류기로 쉽게 변환될 수 있습니다. 예를 들어, 의사 결정 트리는 노드의 인스턴스 비율에서 리프 노드의 클래스 레이블을 결정합니다. 클래스 결정은 단순히 가장 널리 퍼진 클래스입니다. 이러한 학급 비율은 점수로 사용될 수 있습니다(Provost and Domingos, 2001).

규칙 학습자는 규칙 신뢰도에 대해 유사한 통계를 유지하며 인스턴스와 일치하는 규칙의 신뢰도를 점수로 사용할 수 있습니다(Fawcett, 2001).

분류기가 클래스 레이블만 생성하더라도 이들의 집계를 사용하여 점수를 생성할 수 있습니다.

MetaCost(Domingos, 1999)는 배깅을 사용하여 개별 분류기의 앙상블을 생성하고 각 분류기는 투표를 생성합니다. 투표 세트를 사용하여 점수를 생성할 수 있습니다.

마지막으로 채점과 투표의 일부 조합을 사용할 수 있습니다. 예를 들어, 규칙은 기본 확률 추정치를 제공할 수 있으며, 이는 가중 투표에 사용될 수 있습니다.

(포셋, 2001)

5. Efficient generation of ROC curves

5. ROC 곡선의 효율적인 생성

주어진 테스트 세트에서 우리는 종종 ROC 곡선을 효율적으로 생성하기를 원합니다. 임계값 분류의 단조성을 활용할 수 있습니다. 주어진 임계값과 관련하여 양성으로 분류된 인스턴스는 모든 낮은 임계값에 대해서도 양성으로 분류됩니다. 따라서 테스트 인스턴스를 f 점수만큼 감소시켜 정렬하고 목록 아래로 이동하여 한 번에 하나의 인스턴스를 처리하고 진행하면서 TP와 FP를 업데이트할 수 있습니다. 이러한 방식으로 선형 스캔에서 ROC 그래프를 생성할 수 있습니다.

알고리즘은 알고리즘 1에 나와 있습니다. TP와 FP는 모두 0에서 시작합니다. 각 양수 인스턴스에 대해 TP를 증가시키고 모든 음수 인스턴스에 대해 FP를 증가시킵니다. ROC 포인트의 스택 R을 유지하고 각 인스턴스가 처리된 후 새 포인트를 R로 푸시합니다. 최종 출력은 ROC 곡선의 점을 포함하는 스택 R입니다.

n을 테스트 세트의 포인트 수라고 하자. 이 알고리즘은 O(nlogn) 정렬 후 목록 아래로 O(n) 스캔이 필요하므로 O(nlogn) 총 복잡성이 발생합니다.

진술 7-10은 ​​약간의 설명이 필요합니다. 이는 동일하게 점수가 매겨진 인스턴스의 시퀀스를 올바르게 처리하기 위해 필요합니다. 그림 6에 표시된 ROC 곡선을 고려하십시오.

4개의 부정과 6개의 긍정의 인스턴스 시퀀스가 ​​있고 모두 f에 의해 동일하게 점수가 매겨진 테스트 세트가 있다고 가정합니다. 알고리즘 1의 라인 1에 있는 정렬은 f 점수가 동일하기 때문에 이러한 인스턴스에 특정 순서를 부과하지 않습니다. ROC 곡선을 생성하면 어떻게 됩니까? 극단적인 경우에는 모든 긍정적인 결과가 시퀀스의 시작 부분에서 끝나고 우리는 그림 6과 같이 '낙관적인' 상위 L 세그먼트를 생성합니다.

반대 극단에서 모든 음수는 시퀀스의 시작 부분에서 끝나고 다음과 같이 "비관적"인 낮은 L을 얻습니다.

그림 6. 인스턴스의 혼합 순서는 이러한 두 극단에 의해 형성된 직사각형 내에서 다른 단계 세그먼트 세트를 제공합니다. 그러나 ROC 곡선은 다른 정보가 없는 분류기의 예상 성능을 나타내야 하며, 이는 비관적 세그먼트와 낙관적 세그먼트의 평균입니다. 이 평균은 직사각형의 대각선이며 ROC 곡선 알고리즘에서 생성할 수 있습니다.

f 값이 동일한 모든 인스턴스가 처리될 때까지 ROC 포인트를 방출하지 않습니다. 이것이 fprev 변수와 7행의 if 문이 달성하는 것입니다.

동일하게 점수가 매겨진 사례는 비정상적으로 보일 수 있습니다.

그러나 일부 분류기 모델에서는 일반적입니다. 예를 들어, 의사결정 트리의 노드에서 인스턴스 수를 사용하여 인스턴스에 점수를 매긴다면, 크고 엔트로피가 높은 리프 노드는 두 클래스 모두에서 동일하게 점수가 매겨진 많은 인스턴스를 생성할 수 있습니다. 이러한 인스턴스가 평균화되지 않은 경우 결과 ROC 곡선은 테스트 세트 순서에 민감하며 다른 순서는 매우 잘못된 곡선을 생성할 수 있습니다.

이것은 섹션 7에서 논의된 ROC 곡선 아래의 면적을 계산할 때 특히 중요할 수 있습니다. n개의 긍정 및 m개의 부정을 설명하는 리프 노드를 포함하는 의사 결정 트리를 고려하십시오. 이 리프 노드로 분류된 모든 인스턴스에는 동일한 점수가 할당됩니다. 그림 6의 직사각형은 nm 크기입니다.

PN 및 이러한 인스턴스가 평균화되지 않은 경우 이 하나의 잎은 nm 2PN만큼 높은 ROC 곡선 영역의 오류를 설명할 수 있습니다.