



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2011-0001466
(43) 공개일자 2011년01월06일

(51) Int. Cl.

G06F 17/00 (2006.01) G06F 17/30 (2006.01)
G06F 19/00 (2011.01)

(21) 출원번호 10-2009-0059012

(22) 출원일자 2009년06월30일

심사청구일자 2009년06월30일

(71) 출원인

경희대학교 산학협력단

경기도 용인시 기흥구 서천동 1 경희대학교 국제 캠퍼스내

(72) 발명자

이영구

경기 수원시 영통구 영통동 1024-14 경희유니빌 706호

이승룡

경기 성남시 분당구 구미동 250 삼환빌라 102동 102호

(뒷면에 계속)

(74) 대리인

김삼수

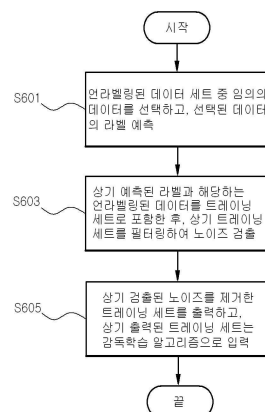
전체 청구항 수 : 총 5 항

(54) 언라벨링된 데이터를 이용한 미스라벨링된 트레이닝 데이터를 식별하는 방법

(57) 요약

본 발명은 감독학습 알고리즘들을 위하여 미스라벨링된 트레이닝 인스턴스들을 식별하고 제거하기 위한 방법을 나타낸다. 이 방법은 미스라벨링된 인스턴스들을 검출하기 위해 언라벨링된 인스턴스들을 이용하는 것이다. 이는 오직 라벨링된 트레이닝 인스턴스들에 의존하는 종래의 방법들과는 반대되는 것이다. 본 발명은 간단하고, 오직 요구되는 최소한의 수정만으로 현존하는 많은 노이즈 검출 방법들에 적용될 수 있다. 본 발명의 이점을 평가하기 위하여, 두 가지의 일반적인 노이즈 검출 방법들인 다수 필터링(Majority Filtering:MF)과 일치 필터링(Consensus Filtering:CF)를 선택하였다. MFAUD(Majority Filtering with the Aid of Unlabeled Data) 및 CFAUD(Consensus Filtering with the Aid of Unlabeled Data)는 언라벨링된 데이터를 이용한 MF 및 CF의 변형을 나타낸다. 본 발명은 분류기의 예측 정확성을 더욱 향상시키는 효과가 있다.

대표도 - 도6



(72) 발명자

후안 동하이

경기도 용인시 기흥구 서천동 1번지 경희대학교 전
자정보과 351

한용구

경기도 수원시 영통구 영통동 1013-14 B04호

특허청구의 범위

청구항 1

언라벨링된(unlabeled) 데이터 세트 중 임의의 언라벨링된 데이터를 선택하고, 상기 선택된 언라벨링된 데이터에 대한 라벨을 예측하는 제1단계;

상기 예측된 라벨을 갖는 언라벨링된(unlabeled) 데이터를 트레이닝 데이터 세트에 포함시킨 후, 필터를 이용하여 상기 트레이닝 데이터 세트로부터 미스라벨링된 트레이닝 데이터를 식별하는 제2단계

를 포함하는 언라벨링된 데이터를 이용한 미스라벨링된 트레이닝 데이터를 식별하는 방법.

청구항 2

제 1항에 있어서, 상기 제1단계에서,

상기 언라벨링된 데이터에 대한 라벨 예측은 앙상블 학습(ensemble learning)과 코-트레이닝(co-training)이 결합된 앙상블-기반 코-트레이닝(En-co-training)에 의해 실행되는 것을 특징으로 하는 언라벨링된 데이터를 이용한 미스라벨링된 트레이닝 데이터를 식별하는 방법.

청구항 3

제 1항에 있어서, 상기 제1단계는

상기 언라벨링된 데이터 세트내의 데이터 수 만큼 반복하여 실행되는 것을 특징으로 하는 언라벨링된 데이터를 이용한 미스라벨링된 트레이닝 데이터를 식별하는 방법.

청구항 4

제 1항에 있어서, 상기 제2단계 이후,

상기 식별된 미스라벨링된 트레이닝 데이터를 제거한 트레이닝 데이터 세트를 출력하는 단계를 더 포함하는 언라벨링된 데이터를 이용한 미스라벨링된 트레이닝 데이터를 식별하는 방법.

청구항 5

제 4항에 있어서,

상기 출력된 트레이닝 데이터 세트는 감독학습(supervised learning) 알고리즘으로 입력되어 상기 출력된 트레이닝 데이터 세트 상에서 분류 훈련이 이루어지는 것을 특징으로 하는 언라벨링된 데이터를 이용한 미스라벨링된 트레이닝 데이터를 식별하는 방법.

명세서

발명의 상세한 설명

기술 분야

[0001] 본 발명은 감독학습(supervised learning) 알고리즘을 위한 미스라벨링된 트레이닝 인스턴스(instance)들을 식별하고 제거하기 위한 방법에 관한 것이다.

배경 기술

[0002] 귀납적 학습 알고리즘의 목적은 트레이닝 인스턴스들에 대해서 구성되는 좋은 일반화 모델을 구성하는 것이다. 일반적으로 일반화 모델의 질을 결정하는 두 가지의 주요인은 트레이닝 데이터의 질과 트레이닝 데이터에 의해 선택된 학습 알고리즘의 바이어스들의 적절함이다.

[0003] 학습 알고리즘이 주어지면, 일반화 모델의 질은 주로 트레이닝 데이터의 질에 의존한다. 트레이닝 데이터가 항상 일반화 모델의 질을 떨어뜨리는 경향이 있는 노이즈(noise)들을 포함하는 것을 고려하면, 효과적인 노이즈

핸들링(handling)은 귀납적 학습에서 가장 중요한 문제 중 하나이다. 잡음이 섞인(noisy) 트레이닝 인스턴스들의 부정적인 측면을 최소화하기 위하여, 사람들은 주로 두 가지 접근법들(노이즈 내성(noise tolerance) 및 노이즈 제거(noise elimination)) 중 하나를 사용한다. 노이즈 내성은 일반적으로 노이즈에 둔감한 로버스트(robust) 알고리즘들을 설계하여 노이즈 삭제 없이 잡음이 섞인 인스턴스들의 부정적인 영향을 제어하려고 하는 것이다. 상기 카테고리의 전형적인 방법들은 규칙절단(rule truncation)[1] 및 트리 전지(tree pruning)[2]를 포함한다.

[0004] 이에 반하여, 노이즈 제거는 학습 알고리즘을 적용하기 전에 잡음이 섞인 인스턴스들을 식별하고 제거함으로써 트레이닝 데이터의 질을 향상시키려고 하는 것이다. 본 카테고리의 하나의 전형적인 방법은 복합 분류기들을 사용하고, 노이즈로 미스라벨링된 트레이닝 사례를 치료하는 것이다. 노이즈 내성이 노이즈 제거보다 덜 효과적이라는 것은 [3]에 의해 논의되었었다. 본 발명에서 우리는 노이즈 제거에 초점을 둔다.

[0005] 잡음이 섞인 트레이닝 인스턴스들은 주로 두 가지 타입을 포함한다. 즉, 속성(attribute) 노이즈와 클래스(class) 노이즈이다. 속성 노이즈들은 인스턴스들의 속성 값에 삽입된 에러들이다. 클래스 노이즈는 미스라벨링에 의해 야기되므로, 미스라벨링된 노이즈라 불린다.

[0006] 우리의 포커스는 클래스 노이즈들(미스라벨링된 인스턴스들)을 식별 및 제거하고, 그로 인해 분류기의 예측 정확성을 증가시키는 것이다.

[0007] 지금까지는 효과적인 학습을 위해 미스라벨링된 인스턴스들을 제거하는데 많은 연구 노력들이 경주되어왔다. Guyon[5]는 인스턴스들의 전형성(typicality)을 측정하기 위해 정보 표준을 사용하는 접근법을 제공했고, 전형적이지 않은 인스턴스들은 그들이 미스라벨링된 인스턴스들인지 또는 반대인지를 결정하기 위해 인간 전문가에게 주어진다.

[0008] Gamberger[6]의 노이즈 검출 알고리즘은 잡음이 섞인 예들(examples)의 제거가 트레이닝 세트의 CLCH(Complexity of the Least Complex correct Hypothesis)값을 감소시킨다는 관찰에 기초하고 있다. 그들은 그들의 노이즈 제거 알고리즘을 포화(saturation) 필터라 부르고, 그것은 트레이닝 세트가 포화되었는지 아닌지를 테스트하기 위해 CLCH 값을 사용하기 때문이다. Brodley 및 Friedl[7,8]은 노이즈 알고리즘을 필터링 동작처럼 간단하게 했는데, 잡음이 섞인 트레이닝 데이터로부터 학습된 복합 분류기들은 노이즈를 식별하기 위해 사용되고, 노이즈는 복합 분류기들에 의해 부정확하게 분류된 인스턴스들로 간주된다. 두 가지의 주요 필터링 방법들로 그들은 다수 필터링(Majority Filtering:MF)과 일치 필터링(Consensus Filtering:CF)을 제안했다. 또한, 가장 근접한 이웃(nearest neighbor) 분류기들을 위해 특별하게 제안된 몇몇의 노이즈 검출 방법이 있다. Wilson[9]은 인스턴스들을 선택하기 위해 3개의 가장 근접한 이웃 분류기(3-NN)를 사용했고, 이는 다시 1-NN을 형성하는데 사용된다. Aha, Kibler 및 Albert[10]은 인스턴스 기반 분류기 내에서 분류 정확도에 대한 그들의 기여의 기록들에 기초하여 인스턴스들을 필터링하는 것은 그 결과로 만들어진 분류기의 정확도를 향상시키는 것임을 증명하였다.

[0009] 상기 방법들의 노이즈 식별 프로세스는 다음 식에 의해 표현될 수 있다.

[0010]
$$R(t) = f(t, T)$$

[0011] 여기에서, $f()$ 는 인스턴스들의 전형성[5], CHCL의 축소[6], 오분류(misclassification)의 수[7,8] 및 분류 정확도의 향상[10] 등과 같은 특정한 노이즈 측정에 의존하는 노이즈 식별 함수를 나타낸다. t 는 트레이닝 세트 T 내의 트레이닝 인스턴스를 나타낸다. $R(t)$ 는 두 가지의 값 즉, 0(제거)과 1(보유)로 구성되는 인스턴스 t 에 대한 식별 및 제거 결과를 나타낸다. 상기 방법들에 대하여, 어떤 종류의 식별 함수가 사용되는 간에, 그 식별 함수와 함께 항상 두 개의 파라미터들이 있다. 트레이닝 세트 T 와 평가될 인스턴스 t 이다. 바꿔 말하면, 식별 함수가 주어지면, 트레이닝 인스턴스에 대한 식별 결과는 오직 트레이닝 세트와 이 인스턴스 자신에 기초한다. 상기 방법은 노이즈 식별에 대한 언라벨링된 데이터의 유용성과 잠재적인 활용성을 고려하지 않는다.

발명의 내용

해결 하고자하는 과제

[0012] 본 발명은 분류기들의 분류 정확도를 증가시키기 위해 트레이닝 데이터에서 노이즈를 식별하고 삭제하기 위한 방법을 제공하는 데 그 목적이 있다.

과제 해결수단

- [0013] 본 발명에 따른 언라벨링된 데이터를 이용한 미스라벨링된 트레이닝 데이터를 식별하는 방법은 언라벨링된(unlabeled) 데이터 세트 중 임의의 언라벨링된 데이터를 선택하고, 상기 선택된 언라벨링된 데이터에 대한 라벨을 예측하는 제1단계와 상기 예측된 라벨을 갖는 언라벨링된(unlabeled) 데이터를 트레이닝 데이터 세트에 포함시킨 후, 필터를 이용하여 상기 트레이닝 데이터 세트로부터 미스라벨링된 트레이닝 데이터를 식별하는 제2단계를 포함한다.
- [0014] 상기 제1단계에서, 상기 언라벨링된 데이터에 대한 라벨 예측은 앙상블 학습(ensemble learning)과 코-트레이닝(co-training)이 결합된 앙상블-기반 코-트레이닝(En-co-training)에 의해 실행되며, 상기 제1단계는 상기 언라벨링된 데이터 세트내의 데이터 수 만큼 반복하여 실행되는 것이 바람직하다.
- [0015] 상기 제2단계 이후, 상기 식별된 미스라벨링된 트레이닝 데이터를 제거한 트레이닝 데이터 세트를 출력하는 단계를 더 포함하며, 상기 출력된 트레이닝 데이터 세트는 감독학습(supervised learning) 알고리즘으로 입력되어 상기 출력된 트레이닝 데이터 세트 상에서 분류 훈련이 이루어지는 것이 바람직하다.

효과

- [0016] 본 발명은 언라벨링된 데이터의 라벨을 예측하고, 언라벨링된 데이터와 그의 예측된 라벨을 이용하여 트레이닝 데이터 내의 미스라벨링된 데이터를 식별하여 제거함으로써, 분류기의 예측 정확성을 향상시킬 수 있는 효과가 있다.

발명의 실시를 위한 구체적인 내용

- [0017] 본 발명에 따른 접근법의 새로움은 트레이닝 데이터 내의 노이즈 검출을 돕기 위해 언라벨링된 데이터를 사용하는 것이다. 이것은 오직 트레이닝 세트에만 의존하는 종래 방법들과는 다르다. 언라벨링된 데이터를 I 라고 나타내면, 본 발명에 따른 접근법은 다음과 같이 표현될 수 있다.
- [0018]
$$R(t) = f(t, T, U)$$
- [0019] 본 발명의 이점은 종종 현존하는 방법들에서 최소한의 수정만으로 많은 어플리케이션들에서 얻기에 편리한 언라벨링된 데이터를 사용한다는 점을 포함한다. 본 발명의 또 다른 이점은 폭 넓은 적용성이다. 본 발명은 상기 언급된 것들을 포함하는 많은 다양한 노이즈 검출 방법과 함께 사용될 수 있다.
- [0020] 본 발명에 따른 접근법의 이점을 평가하기 위하여, 두 가지의 일반적인 방법들, 다수 필터링(MF)과 일치 필터링(CF)이 선택된다. 본 발명에 따른 접근법에 기초하여, MF(및 CF)의 변형이 제안되고, 이는 언라벨링된 데이터를 이용한 MF(및 CF) 필터링을 의미하는 MFAUD(및 CFAUD)라 불린다. MF, CF와 제안된 MFAUD, CFAUD의 비교를 위해 잡음이 섞인 데이터 세트들 중 한 세트에 적용한다. 경험적 연구에 따르면 MFAUD와 CFAUD가 10%에서 40%까지의 서로 다른 노이즈 레벨에서 MF 및 CF의 성능을 두드러지게 향상시킬 수 있다. 게다가, 상기 향상은 노이즈 레벨에 정비례한다.
- [0021] 종래에도 미스라벨링된 트레이닝 인스턴스들을 식별하고 제거하기 위한 많은 방법들이 있다. 본 발명에서는 Brodley의 다수 필터링(Majority Filtering:MF)과 일치 필터링(Consensus Filtering:CF)를 고려한다.
- [0022] MF와 CF의 일반적인 개념은 다음과 같다. 이들은 일련의 기본 레벨 분류기들을 구성하고 미스라벨링된 인스턴스들을 식별하는데 그들의 분류를 사용하는 것에 의해 미스라벨링된 인스턴스들을 검출하기 위해 복합 분류기(ensemble classifier)를 사용한다. 일반적인 접근법은 만일 m 개의 기본 레벨 분류기들 중 x 개의 분류기가 인스턴스를 정확하게 분류하지 못하면, 해당 인스턴스를 미스라벨링으로 태그하는 것이다.
- [0023] MF는 만일 m 개의 기본 레벨 분류기들 중 반 이상이 인스턴스들을 부정확하게 분류하면 해당 인스턴스들을 미스라벨링으로 태그한다. CF는 트레이닝 데이터로부터 미스라벨링된 인스턴스가 제거되도록 하기 위해 모든 기본

레벨 분류기들이 트레이닝 라벨에 의해 주어진 클래스로 인스턴스를 분류하는 것이 실패해야함을 요구한다.

- [0024] MF 및 CF에서 복합 분류기들을 사용하는 이유는, 두 가지 조건 즉, (1) 개별 분류기에 의한 정확한 분류의 확률이 0.5보다 클 것, (2) 기본 레벨 분류기들의 예측상의 어려움이 독립적일 것을 만족한다면 복합 분류기가 데이터 세트에 대해서 각각의 기본 레벨 분류기보다 더 나은 성능을 갖기 때문이다.
- [0025] 도 1에서 보는 바와 같이, MF는 트레이닝 세트 E의 n개의 같은 크기의 디스조인트 서브셋들(단계 1)과 검출된 잡음있는 예들(examples)의 공 출력세트 A(단계 2)로 시작된다. 메인 루프(단계 3-16)는 각각의 트레이닝 서브셋 E_i 에 대하여 반복된다. 단계 4에서는, 서브셋 E_i 가 E_i 에 있는 예들을 제외한 E로부터의 모든 예들(examples)을 포함하여 형성되고, 이는 가설(분류기) H_i 를 유도하는 임의의 귀납적인 학습 알고리즘의 입력으로 사용된다(단계 6). 가설들의 대다수가 정확한 분류를 제공하지 않는 E_i 로부터의 예들은 잠재적으로 잡음섞인 예들로서 A에 추가된다(단계 14).
- [0026] CF의 알고리즘은 도 2에서 보여진다. 그것은 MF와는 오직 단계 14만 다르다. CF에서는, E_i 내의 예들(examples)은 오직 모든 가설들이 그것을 부정확하게 분류할때만 잡음이 섞인 예(example)로 간주된다. MF와 비교하면, CF는 노이즈 식별을 위한 보다 엄격한 조건으로 인해 좀더 보수적이며, 보다 적은 인스턴스들이 트레이닝 세트로부터 삭제될 것이다. CF의 결점은 배드 데이터(bad data)를 보유하는데 따르는 추가되는 리스크이다.
- [0027] MF 및 CF 둘 다 n-크로스 확인(n-cross-validation)을 통해 잡음이 섞인 인스턴스들을 검출하기 위하여 복합 분류기를 사용한다. 크로스 i에서 서브셋 i가 추출되고 체크된다. 다른 서브셋들의 결합은 학습 알고리즘에 기초하여 일련의 분류기들을 구성하기 위한 트레이닝 세트로서 사용되고, 노이즈들을 검출하기 위해 서브셋 i 내의 인스턴스들을 분류한다. 이러한 분류기들의 신뢰성은 중대하고, 노이즈 검출 실행은 이러한 분류기들의 분류 정확성이 증가될 때 향상될 것으로 기대된다. 본 발명은 이러한 분류기들의 분류 정확도를 증가시키기 위해 언라벨링된 데이터를 활용하는 것이다.
- [0028] MF 및 CF에서 학습 알고리즘이 감독되는 것처럼, 언라벨링된 데이터를 활용하기 위한 첫 번째 단계는 도 6에 도시된 바와 같이, 언라벨링된 데이터에 대한 라벨들을 예측하는 것이다(S601). 그 다음 두 번째 단계는 분류기들의 예측 능력을 증가시키기 위해 예측된 라벨들을 갖는 언라벨링된 데이터를 사용하는 것이다. 그것에 의하여 노이즈 검출 실행을 향상시킬 것이다. 즉, 도 6에 도시한 바와 같이, 예측된 라벨을 갖는 언라벨링된 데이터를 트레이닝 데이터 세트에 포함시킨 후, 상기 트레이닝 데이터 세트를 필터링하여 노이즈를 검출하고(S603), 검출된 노이즈를 제거한 트레이닝 데이터 세트를 출력한다(S605). 이후, 상기 출력된 트레이닝 데이터 세트는 감독 학습 알고리즘으로 입력된다. 상기 단계는 섹션 a 및 b에 각각 나타내었다.
- [0029] a. 단계 1: 언라벨링된 데이터에 대한 라벨 예측
- [0030] 기계 학습 영역에서, 부분감독 분류(semi-supervised classification)는 이 단계에 대한 해법을 제공할 수 있다. 전형적인 분류기들은 훈련을 위해 오직 레이블된 label 레이블(labeled) 데이터만을 사용한다. 선택적으로, 부분감독 분류는 더 나은 분류기들로 훈련하기 위해 레이블된 label 레이블 데이터와 함께 언라벨링된 데이터를 사용한다. 부분감독 분류는 레이블된 label 레이블 데이터로부터 획득한 가설들을 수정 또는 재순위화하기 위하여 언라벨링된 데이터를 사용한다.
- [0031] 부분감독 분류는 언라벨링된 데이터를 다양한 방법으로 활용할 수 있다. 그러나 우리는 오직 언라벨링된 데이터에 대한 라벨을 명백하게 예측하여 트레이닝 세트가 확장되는 방법에만 흥미가 있다.
- [0032] 일반적으로, 두 개의 간단하고 일반적인 방법에 따른 부분감독 방법들이 있다. 그것은 셀프-트레이닝(self-training)과 코-트레이닝(co-training)이다. 셀프-트레이닝[11]에서는, 처음에 기본 학습기가 레이블된 label 레이블 세트에서 트레이닝된 후, 언라벨링된 세트에서 가장 확신있는 몇몇의 인스턴스들을 반복적으로 라벨로 선택한다. 후에, 이러한 셀프-라벨링된 인스턴스들을 갖는 라벨링된 트레이닝 세트로 확대한다. 코-트레이닝[12]은 특징들은 두 개의 세트로 분리될 수 있을 것, 각 서브셋은 좋은 분류기를 트레이닝하기에 충분한 것, 상기 두 개의 세트는 클래스가 주어지면 조건적으로 독립적일 것을 요구한다. 처음에 두 개의 개별적인 분류기는 두 개의 서브-특징(sub-feature) 세트 상의 라벨링된 데이터로 각각 훈련된다. 그리고 나서, 각 분류기는 언라벨링된 데이터를 분류하고, 그들이 가장 확신하는 선택된 언라벨링된 인스턴스들(예측된 라벨들을 갖는)로 다른

분류기들을 돕는다. 각 분류기는 다른 분류기들에 의해 주어진 부가적인 트레이닝 인스턴스들로 재훈련되고, 상기 순서를 반복한다.

[0033] 그러나, 셀프-트레이닝과 코-트레이닝 둘 다 약간의 문제점들을 보인다. 셀프-트레이닝에 대해서는, 한편으로는, 분류기가 언라벨링된 데이터의 '신뢰'를 평가하기 위한 어떤 수단을 요구한다. 그러나, k-nearest neighbors와 같은 많은 분류기들은 이러한 수단을 용이하게 줄 수 없다. 한편으로는, 비록 분류기가 신뢰를 측정할 수 있을지라도, 언라벨링된 데이터에 대한 그 자신만의 예측은 코-트레이닝의 두 개의 분류기들만큼 정확하게 되지 않을 수 있다. 코-트레이닝의 경우에는, 오직 두 개의 충분하고 독립적인 뷰에 의해 나타내어질 수 있는 데이터 세트에 대해서만 동작하므로 보편성이 없다.

[0034] 본 발명은 앙상블 학습기와 코-트레이닝의 결합을 통한 앙상블-기반 코-트레이닝(En-co-training) 방법을 제안한다.

[0035] 도 3에 도시한 것처럼, 복합 분류기(두 개 이상)가 셀프-트레이닝의 단일 분류기 및 코-트레이닝의 두 개의 분류기들 대신 사용된다. 앙상블-기반 코-트레이닝은 셀프-트레이닝 및 코-트레이닝의 한계를 극복한다. 이들 분류기들의 앙상블 보팅(ensemble voting)으로 인하여 신뢰도를 명시적으로 나타내는 값들은 요구되지 않는다. 게다가, 코-트레이닝 내의 두 분류기들의 다양성은 여러 세트의 특징(feature)들을 사용함으로써 획득된다. 이 프로세스는 특징들에 대한 두 가지의 관점을 요구한다. 그러나, 앙상블-기반 코-트레이닝에서, 분류기의 다양성은 다른 알고리즘을 사용함으로써 획득된다. 예를 들어, 세 가지의 다양한 알고리즘(k-nearest neighbor, naive Bayes and decision tree)이 우리의 실험에서 사용된다.

[0036] 앙상블-기반 코-트레이닝의 시작에서는 U 로부터 작은 수의 예들(examples)인 U' 이 임의로 선택되고, 이후 각 반복의 끝에서는, 보다 많은 예들(examples)이 U' 으로 채워진다. 관련 논문 [12]는 이러한 종류의 세팅이 언라벨링된 데이터 전체인 U 를 직접적으로 다루는 것보다 낫다는 것을 보인다.

[0037] *b. 언라벨링된 데이터와 그들의 예측된 라벨들을 활용하는 것.*

[0038] 도 3에 도시된 바와 같이, Tu 가 단계 1의 출력을 나타낸다고 하면, Tu 는 앙상블-기반 코-트레이닝에 의해 선택된 언라벨링된 데이터와 그들의 예측된 라벨들을 포함한다. 이 부분에서, 트레이닝 데이터 T 에서의 노이즈 검출을 돕기 위해 Tu 를 사용하는 MF와 CF의 변형들이 고안된다.

[0039] 도 4 및 도 5에 도시한 바와 같이, Tu 는 요구되는 MF 및 CF에 대하여 오직 제한적인 변형들만을 갖는 본 발명에 따른 MFAUD 및 CFAUD에 의해 활용된다.

[0040] 그러나, 언라벨링된 데이터를 사용하는 것은 항상 위험하다는 것을 명심해야한다. 또한, 언라벨링된 데이터의 사용이 트레이닝 세트에서 노이즈 검출 실행을 확실히 향상시킬 수 있음을 당연하게 생각하지 말아야한다. 이유는 다음과 같다. 즉, 비록 본 발명에 따른 앙상블-기반 코-트레이닝을 포함하는 부분감독 방법이 몇몇의 선택된 언라벨링된 데이터에 대한 라벨들을 예측할 수 있다 하더라도, 상기 예측된 라벨은 일반적으로 에러들을 갖는다. 정확하게 예측된 라벨들을 갖는 언라벨링된 인스턴스들은 보통 노이즈 검출 실행을 향상시키지만, 예측으로부터 잡음이 섞인 라벨들은 분류기의 예측 정확도를 떨어뜨릴 수 있고, 트레이닝 세트에서의 노이즈 검출의 성능 저하를 가져오게 된다.

[0041] · 경험적 연구

[0042] 경험적 연구의 주요 목표는 트레이닝 데이터 내에서의 노이즈 검출에 대한 언라벨링된 데이터의 이점을 평가하는 것이다.

[0043] - 실험 환경 : 현재의 MF, CF 및 본 발명의 MFAUD, CFAUD는 Machine Learning Database Repository [13]으로부터의 벤치마크 데이터 세트(benchmark data sets)에서 테스트된다. 이러한 데이터 세트들의 정보는 < 표 1 >과 같다.

Data set	Attribute	Size	Class	Class distribution
iris	4	150	3	50/50/50
voting	16	435	2	267/168
heart2	13	294	2	188/106
horse	15	368	2	232/136
sonar	60	208	2	111/97
wine	13	178	3	59/71/48
breast	9	1000	2	700/300
yeast	8	1484	10	463/429/244/163/51/44/37/30/20/5
australian	14	690	2	383/307
bupa	6	345	2	145/200
diabetes	8	768	2	500/268
echo	7	131	2	88/43
german	24	1000	2	700/300
glass	9	214	6	70/76/17/13/9/30
magic	10	2000	2	1000/1000
credit	15	690	2	307/383
spect	44	267	2	212/55
wdbc	31	569	2	357/212
ecoli	7	336	8	143/77/52/35/20/5/2/2
ionosphere	34	351	2	225/126

< 표 1 >

이러한 데이터 세트들은 다양한 도메인들에서 다른 실-세계(real-world) 어플리케이션으로부터 수집된다. 각 데이터 세트는 트레이닝 세트와 테스트 세트로 나뉜다. 노이즈 검출 방법이 트레이닝 세트에 대해서 동작되어 필터링 된 트레이닝 세트를 출력한다. 후에, 테스트 세트는 다양한 필터링 된 트레이닝 세트들 상에서 훈련된 분류기에 의해 분류된다. 분류 에러율은 분류기에 대한 각 노이즈 검출 방법의 성능을 평가하기 위한 척도이다. 여기에서, 분류 에러율은

분류 에러율 = 테스트 인스턴스들에 대한 부정확한 분류들의 수 / 테스트 인스턴스들의 수

두 가지의 노이즈 검출 방법들이 같은 학습 알고리즘을 갖는 같은 데이터 세트에 적용될 때, 낮은 분류 에러율은 노이즈 검출 성능이 더 우수함을 나타낸다. 분류 에러율을 얻기 위하여, 각 데이터 세트 D 는 다음과 같이 처리된다.

단계1 : D 는 두 개의 부분 즉, 레이블된 label 레이블 세트 L 과 언라벨링된 세트 U 로 임의로 분할된다.

단계2 : 10분 교차 훈련법(ten-fold cross-validation)으로부터 얻은 10개의 시도(ten trials)는 각 노이즈 검출 방법의 성능을 평가하기 위해 사용된다. 각 시도에서, L 의 90%가 첫 번째로 선택되고, 그것은 T_{temp} 로 표시된다.

그러나, 노이즈 제거 방법의 성능은 잡음이 섞인 데이터 세트 상에서 평가될 필요가 있다. 이러한 목적으로, 우리는 몇 개의 인스턴스들을 임의로 선택함으로써, T_{temp} 내에 몇몇의 노이즈를 인위적으로 생성한 후, 그들의 라벨을 부정확하게 바꾼다. 선택된 인스턴스들의 수는, 생성된 노이즈들의 수이고, 정의된 잡음비에 기초하며, T_{temp} 내의 데이터에 대한 잡음이 섞인 데이터의 비율이다. T 를 T_{temp} 에 노이즈들이 더해진 후의 데이터를 나타낸다고 하자. T 는 트레이닝 세트로 사용되고, 그것은 MF, CF, MFAUD, CFAUD에 의해 반복적으로 처리될 것이다. L 의 잔존하는 10%는 T 의 각 필터링된 세트상에서 훈련되는 알고리즘에 의해 분류될 테스트 세트로 사용된다.

단계3 : 다른 노이즈 필터링 방법들을 갖는 각 알고리즘의 평균 분류 에러율은 10개의 시도(ten trials)의 에러율을 평균 내는 것에 의해 얻어진다.

- [0053] 단계4 : 데이터 세트의 분할이 이 평균 분류 에러율에 영향을 끼칠 수 있음을 생각하여, 우리는 5회 분할을 실행하여 5개의 분류 에러율을 얻는다(단계1-3을 5회 실행)
- [0054] 단계5 : 마지막으로 다른 노이즈 필터링 방법들을 갖는 각 알고리즘의 보고된 에러율은 이 5개의 값들의 값을 더 평균내는 것이다.
- [0055] 이 실험에서, 네 가지의 노이즈 검출 방법들(MF, CF, MFAUD 및 CFAUD)은 다음과 같은 동일한 구성을 따른다. n , 이것은 서브셋의 수이고, 5로 세트된다. y , 이것은 학습 알고리즘들의 수이고, 3으로 세트된다. $A1$, $A2$ 및 $A3$ 은 k-nearest neighbor(kNN), naive Bayes 및 decision tree로 불리는 세 가지의 학습 알고리즘을 나타낸다.
- [0056] MFAUD 및 CFAUD에서의 추가적인 파라미터들은 앙상블-기반 코-트레이닝에 대한 것이고, 다음과 같이 구성된다. k , 이것은 반복의 수이고, 4로 세트된다. u , 이것은 처음에 선택되는 언라벨링된 인스턴스들의 수이고, 트레이닝 데이터의 수와 같다.
- [0057] 노이즈 검출 방법들에서의 파라미터들에 더하여, 상기 실험 순서 내에 실험에 영향을 끼칠 수 있는 두 가지의 주요한 파라미터가 있다. 첫 번째 파라미터는 데이터 분할(data partitioning)을 결정하고(상기 실험 순서에서 단계 1), 그것은 전체 데이터에 대한 레이블된 label 레이블 데이터의 비율로서, 레이블된 label 레이블 비율이라고 불린다. 상기 실험에서, 레이블된 label 레이블 비율은 0.5로 세트된다. 두 번째이고 가장 중요한 파라미터는 트레이닝 세트 T 에서의 노이즈 레벨을 결정하고, 그것은 잡음비이다(상기 실험 순서에서 단계 2). 실제 어플리케이션에서 얻어지는 데이터 세트가 다른 노이즈 레벨들을 가질 수 있다는 것을 고려하여, 우리는 실험들을 포괄적으로 만들기 위해 잡음비가 변화해가면서 몇몇의 실험을 실행하였다.
- [0058] 상기 실험들은 두 부분을 포함한다.
- [0059] 실험 A : 잡음비는 10%, 20%, 30%, 40%, 50% 및 60%를 포함한다. 여기에서의 실험은 MF와 MFAUD간의 잡음비 변화에 따른 비교를 나타낸다. 목표는 언라벨링된 데이터가 MF의 실행을 향상시킬 수 있는지 없는지를 테스트하는 것이다.
- [0060] 실험 B : 잡음비는 10%, 20%, 30%, 40%, 50% 및 60%를 포함한다. 여기에서의 실험은 CF와 CFAUD간의 잡음비 변화에 따른 비교를 나타낸다. 목표는 언라벨링된 데이터가 CF의 실행을 향상시킬 수 있는지 없는지를 테스트하는 것이다.
- [0061] 경험적 평가의 결과는 MFAUD와 CFAUD가 MF와 CF의 분류 실행을 두드러지게 향상시켰음을 증명했다. 게다가, 본 발명의 실시 예는 잡음비가 증가하는 만큼 MF 에 대한 MFAUD의 성능개선 및 CF에 대한 CFAUD의 성능개선이 증가됨을 보여준다.
- [0062] · 참조문헌(references)
- [0063] [1] J. Mingers, An empirical comparison of pruning methods for decision tree induction, in: Machine Learning, 4 (2), (1989) 227-243.
- [0064] [2] J.R. Quinlan, C4.5: programs for machine learning, Morgan Kaufmann
- [0065] Publishers Inc., San Francisco, CA, USA, 1993.
- [0066] [3] D. Gamberger, N. Lavrac, S. Dzeroski, Noise detection and eliminaton in data preprocessing: Experiments in medical domains, in: Applied Artificial Intelligence, 14 (2), (2000) 205-223.
- [0067] [4] J.R. Quinlan, Induction of decision trees, in: Machine Learning, 1 (1), (1986) 81-106.
- [0068] [5] I. Guyon, N. Matic, V. Vapnik, Discovering information patterns and data cleaning, in: Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, (1996) 181-203.

- [0069] [6] D. Gamberger, N. Lavrac, C. Groselj, Experiments with noise filtering in a medical domain, in: Proceedings of 16th International Conference on Machine Learning, (1999) 143-151.
- [0070] [7] C.E. Brodley, M.A. Friedl, Identifying and eliminating mislabeled training instances, in: Proceedings of 13th National Conference on Artificial Intelligence, (1996) 799-805.
- [0071] [8] C.E. Brodley, M.A. Friedl, Identifying mislabeled training data, in: Journal of Artificial Intelligence Research, 11, (1999) 131-167.
- [0072] [9] D. Wilson, Asymptotic properties of nearest neighbor rules using edited data, in: IEEE Transactions on System, Man and Cybernetics, 2, (1972) 408-421.
- [0073] [10] D. Aha, D. Kibler, M. Albert, Instance-based learning algorithms, in: Machine Learning, 6 (1), (1991) 37-66.
- [0074] [11] E. Riloff, J. Wiebe, T. Wilson, Learning subjective nouns using extraction pattern bootstrapping, in: Proceedings of the 7th Conference on Natural Language Learning, (2003) 25-32.
- [0075] [12] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of 11th Annual Conference on Computational Learning Theory, (1998) 92-100.
- [0076] [13] UCI KDD Archive, <<http://kdd.ics.uci.edu>>.
- [0077] [14] D. Angluin, P. Laird, Learning from noisy examples, in: Machine Learning 2 (2), (1988) 343-370.

도면의 간단한 설명

- [0078] 도 1은 종래의 다수 필터링(Majority Filtering:MF) 알고리즘을 나타내는 도면이다.
- [0079] 도 2는 종래의 일치 필터링(Consensus Filtering:CF) 알고리즘을 나타내는 도면이다.
- [0080] 도 3은 본 발명의 실시 예에 따른 앙상블-기반 코-트레이닝(En-co-training) 알고리즘을 나타내는 도면이다.
- [0081] 도 4는 본 발명의 실시 예에 따른 언라벨링된 데이터를 이용한 다수 필터링(Majority Filtering with the Aid of Unlabeled Data:MFAUD) 알고리즘을 나타내는 도면이다.
- [0082] 도 5는 본 발명의 실시 예에 따른 언라벨링된 데이터를 이용한 다수 필터링(Consensus Filtering with the Aid of Unlabeled Data:CFAUD) 알고리즘을 나타내는 도면이다.
- [0083] 도 6은 본 발명의 실시 예에 따른 언라벨링된 데이터를 이용한 미스라벨링된 트레이닝 데이터를 식별하는 방법을 나타내는 순서도이다.

도면

도면1

Algorithm 1: MajorityFiltering (MF)

Input: E (training set)

Parameter: n (number of subsets), y (number of learning algorithms)
 $A_1, A_2, \dots, A_y$ (y kinds of learning algorithms)

Output: A (detected noisy subset of E)

```

(1) form  $n$  disjoint almost equally sized subsets of  $E$ , where  $\cup_i E_i = E$ 
(2)  $A \leftarrow \emptyset$ 
(3) for  $i=1, \dots, n$  do
(4)   form  $E_i \leftarrow E \setminus E_i$ 
(5)   for  $j=1, \dots, y$  do
(6)     induce  $H_j$  based on examples in  $E_i$  and  $A_j$ 
(7)   end for
(8)   for every  $e \in E_i$  do
(9)      $ErrorCounter \leftarrow 0$ 
(10)    for  $j=1, \dots, y$  do
(11)      if  $H_j$  incorrectly classifies  $e$ 
(12)        then  $ErrorCounter \leftarrow ErrorCounter + 1$ 
(13)      end for
(14)      if  $ErrorCounter > \frac{y}{2}$ , then  $A \leftarrow A \cup \{e\}$ 
(15)    end for
(16)  end for

```

도면2

Algorithm 2: ConsensusFiltering (CF)

Input: E (training set)

Parameter: n (number of subsets), y (number of learning algorithms)
 $A_1, A_2, \dots, A_y$ (y kinds of learning algorithms)

Output: A (detected noisy subset of E)

```

(1) form  $n$  disjoint almost equally sized subsets of  $E$ , where  $\cup_i E_i = E$ 
(2)  $A \leftarrow \emptyset$ 
(3) for  $i=1, \dots, n$  do
(4)   form  $E_i \leftarrow E \setminus E_i$ 
(5)   for  $j=1, \dots, y$  do
(6)     induce  $H_j$  based on examples in  $E_i$  and  $A_j$ 
(7)   end for
(8)   for every  $e \in E_i$  do
(9)      $ErrorCounter \leftarrow 0$ 
(10)    for  $j=1, \dots, y$  do
(11)      if  $H_j$  incorrectly classifies  $e$ 
(12)        then  $ErrorCounter \leftarrow ErrorCounter + 1$ 
(13)      end for
(14)      if  $ErrorCounter = y$ , then  $A \leftarrow A \cup \{e\}$ 
(15)    end for
(16)  end for

```

도면3

Algorithm 3: Ensemble-based Co-training (En-co-training)
Input: E (training set), U (unlabeled set)
Parameter: k (number of iterations), y (number of learning algorithms)
 u (number of initially selected unlabeled instances)
 $A_1, A_2, \dots, A_y$ (y kinds of learning algorithms)
Output: T_U (selected unlabeled instances from U with predicted labels)

```

(1) create  $U'$  by choosing u instances at random from U
(2)  $T_U \leftarrow \emptyset$ 
(3) for i=1,...,k do
(4)    $U \leftarrow U \setminus U', \text{ num}_{before} \leftarrow |T_U|$ 
(5)   for j=1,...,y do
(6)     induce  $H_j$  based on instances in L and algorithm  $A_j$ 
(7)   end for
(8)   for every  $t \in U'$  do
(9)     for j=1,...,y do
(10)       $pl_j(t) \leftarrow H_j(t)$  // predicted label of  $H_j$  on t
(11)    end for
(12)    if  $pl_1(t) = pl_2(t) = \dots, pl_y(t)$ 
(13)      then  $T_U \leftarrow t \cup pl_1(t), U' \leftarrow U' \setminus t$ 
(14)    end for
(15)    $L \leftarrow L \cup T_U$ 
(16)    $\text{num}_{after} \leftarrow |T_U|, \Delta num \leftarrow \text{num}_{after} - \text{num}_{before}$ 
(17)   if  $|U| \geq \Delta num$ 
(18)     then randomly choose  $\Delta num$  instances from U to replenish  $U'$ 
(19)   if  $0 < |U| < \Delta num$ 
(20)     then choose all instances of U to replenish  $U'$ 
(21)   if  $|U| = 0$ 
(22)     then exit;
(23) end for

```

도면4

Algorithm 4: Majority Filtering with the Aid of Unlabeled Data (MFAUD)

Input: E (training set), U (unlabeled set)

Parameter: n (number of subsets), y (number of learning algorithms)

k (refer to Alg. of En-co-training)

u (refer to Alg. of En-co-training)

$A_1, A_2, \dots, A_y$ (y kinds of learning algorithms)

Output: A (detected noisy subset of E)

```

(1) form  $n$  disjoint almost equally sized subsets of  $E$ , where  $\cup_i E_i = E$ 
(2)  $A \leftarrow \emptyset$ 
(3) for  $i=1, \dots, n$  do
(4)   form  $E_i \leftarrow E \setminus E_i$ 
(5)    $T_U = \text{En-co-training}(E_i, U, k, u, A_1, A_2, \dots, A_y)$ 
(6)    $E_i \leftarrow E_i \cup T_U$ 
(7)   for  $j=1, \dots, y$  do
(8)     induce  $H_j$  based on examples in  $E_i$  and  $A_j$ 
(9)   end for
(10)  for every  $e \in E_i$  do
(11)     $ErrorCounter \leftarrow 0$ 
(12)    for  $j=1, \dots, y$  do
(13)      if  $H_j$  incorrectly classifies  $e$ 
(14)        then  $ErrorCounter \leftarrow ErrorCounter + 1$ 
(15)    end for
(16)    if  $ErrorCounter > \frac{y}{2}$ , then  $A \leftarrow A \cup \{e\}$ 
(17)  end for
(18) end for

```

도면5

Algorithm 5: Consensus Filtering with the Aid of Unlabeled Data (CFAUD)

Input: E (training set), U (unlabeled set)

Parameter: n (number of subsets), y (number of learning algorithms)

k (refer to Alg. of En-co-training)

u (refer to Alg. of En-co-training)

$A_1, A_2, \dots, A_y$ (y kinds of learning algorithms)

Output: A (detected noisy subset of E)

```

(1) form  $n$  disjoint almost equally sized subsets of  $E$ , where  $\cup_i E_i = E$ 
(2)  $A \leftarrow \emptyset$ 
(3) for  $i=1, \dots, n$  do
(4)   form  $E_t \leftarrow E \setminus E_i$ 
(5)    $T_U = \text{En-co-training}(E_t, U, k, u, A_1, A_2, \dots, A_y)$ 
(6)    $E_t \leftarrow E_t \cup T_U$ 
(7)   for  $j=1, \dots, y$  do
(8)     induce  $H_j$  based on examples in  $E_t$  and  $A_j$ 
(9)   end for
(10)  for every  $e \in E_i$  do
(11)     $ErrorCounter \leftarrow 0$ 
(12)    for  $j=1, \dots, y$  do
(13)      if  $H_j$  incorrectly classifies  $e$ 
(14)        then  $ErrorCounter \leftarrow ErrorCounter + 1$ 
(15)      end for
(16)      if  $ErrorCounter = y$ , then  $A \leftarrow A \cup \{e\}$ 
(17)    end for
(18)  end for

```

도면6

