응용통계연구 (2010)
23(1), 197–206

대용량 자료와 순차적 자료를 위한 부스팅 알고리즘

윤영주[1]

[1]Department of Statistics, University of Georgia

(2009년 12월 접수, 2010년 1월 채택)

요 약

본 논문에서는 대용량 자료 혹은 시간에 따라 순차적으로 들어오는 자료의 분류를 위한 부스팅(boosting) 알고리즘을 제안한다. 대용량 자료나 순차적 자료의 경우 분석시 모든 훈련 자료(training data)들을 한번에 이용하기 어려우므로 보통의 부스팅 알고리즘은 적절하지 못하다. 이러한 상황을 극복하기 위해 AdaBoost와 Arc-x4와 같은 부스팅 알고리즘을 수정하여 제안한다. 모의 실험과 실제 자료 분석을 통해 대용량 자료나 순차적 자료에 제안된 알고리즘이 잘 적용됨을 보였다.

주요용어: 개념 변화, 대용량 자료, 부스팅, 순차적 자료, 앙상블 방법.

1. 서론

지난 30년간 의사결정나무(decision tree), 신경망(neural network), 서포트 벡터 머신(support vector machine)과 같은 분류 방법에 대한 수 많은 머신 러닝(machine learning) 방법들이 개발되어 왔다. 이러한 방법들은 실제 자료의 문제에서 성공적으로 적용되어 왔다. 그러나 이러한 방법들은 한 번에 모든 자료를 이용하기 때문에 대용량 자료나 순차적 자료에는 부적절하다.

많은 경우 매일마다 수많은 자료가 수집되고 있다. 소매업 체인이 실시하는 특별한 마케팅 켐페인의 성공 여부를 예측하거나 온라인 업계에서 소비자들의 구매 패턴을 분석하여 상품을 추천하는 경우에 필요한 자료는 매일 수집되어 분석을 하여야 한다. 또한 자료를 한 번에 읽지 못하는 경우에도 모든 이용 가능한 정보를 사용하여 분류를 하여야 한다. 게다가 위에서 언급한 예들은, 예측 변수들에 반영되지 않는 조건들이 사전 인지도 없이 변하여 개념이 서서히 또는 급격하게 변하게 된다(예를 들면 경제 환경의 변화에 따른 구매 패턴의 변화). 따라서 이러한 환경의 변화에 빠르게 조정할 수 있는 알고리즘이 필요하게 된다 (Street와 Kim, 2001).

Wang 등 (2003)은 환경 변화를 반영하는데는 이를 반영할 수 있는 단일 분류자(single classifier)보다는 정확도나 효율성, 편이성의 측면에서 앙상블(ensemble) 방법이 좀 더 좋은 방법이라 주장했다. Kuncheva (2004)는 러닝 시간이 주요 목적이 아니고 정확도가 중요하다면 앙상블 방법이 자연스러운 해결책이 될 것이라 주장했다. 그 중 배깅(bagging)의 개념을 이용한 방법인 SEA(Streaming Ensemble Algorithm, Street와 Kim (2001))와 Wang 등 (2003)의 방법이 가장 간단하면서도 널리 알려진 방법이다. SEA는 앙상블을 구성할 때 균일한 가중치를 사용한 반면에 Wang 등 (2003)의 방법은 가중치를 부여하는 방법을 사용하였다. Yeon 등 (2005)에서 보다시피 이 두 가지 방법이 정확도의 측면에서는 크게 차이를 보이지 않았으므로 본 논문에서는 SEA만을 소개하고 제안된 방법과 비교할 예정이다.

[1]Postdoctoral Fellow, Department of Statistics, University of Georgia, 101 Cedar Street Athens, GA, USA 30602. E-mail: yoonyj74@uga.edu

부스팅(boosting)은 예측의 정확도를 향상시키기 위한 앙상블 방법 가운데 가장 좋은 방법 중 하나이다. C4.5 (Quinlan, 1993)이나 CART (Breiman 등, 1984)를 이용한 AdaBoost (Freund와 Schapire, 1997)와 Arc-x4 (Breiman, 1998)가 널리 연구되어 왔다. 부스팅 알고리즘은 분류자를 생성하고 생성된 분류자들을 결합하여 예측하는 앙상블 방법이다. 부스팅의 개념은 이전 분류자에 의해 오분류된 개체에 더 집중하여(혹은 높은 가중치를 주어) 순차적으로 분류자를 생성하고 그렇게 생성된 "약한(weak)" 분류자들을 결합하여 "강한(strong)" 분류자를 만들어 내는데 있다. 개별 분류자들을 앙상블로 결합할 때 AdaBoost는 각 개별 분류자에 서로 다른 가중치를 부여하여 결합하고 Arc-x4는 동일한 가중치를 부여하여 결합한다.

본 논문에서는 부스팅 방법의 개념을 기초로 대용량 자료나 순차적 자료를 위한 새로운 앙상블 알고리즘을 제안하고자 한다. 대부분의 정적인 상황(static situations)에서는 부스팅이 배깅보다는 좋은 성능을 보이고 있다 (Breiman, 1998). 따라서 정적인 상황에서처럼 대용량이나 순차적 자료 상황에서도 배깅의 개념을 이용한 방법보다 부스팅에 기초한 앙상블 방법이 좋은 성능을 가지리라 예상할 수 있다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 AdaBoost와 Arc-x4 알고리즘에 대해 간략하게 소개하며 또한 배깅의 개념을 이용한 SEA에 대해 살펴본다. 부스팅의 개념을 이용한 새로운 앙상블 방법은 3장에서 제안한다. 4장에서는 제안된 방법과 기존의 방법의 비교를 위해 모의 실험과 실제 자료 분석을 실시하고 그 결과를 비교한다. 제안된 방법의 성능은 기존의 방법보다 우수하며 상황의 변화(혹은 개념의 변화)에도 더 빠르게 잘 조절됨을 알 수 있었다. 5장에서는 결과에 대한 요약 및 결론을 서술한다.

2. 부스팅과 배깅 개념을 이용한 SEA

2.1. 부스팅

부스팅은 가장 성능이 우수한 학습(learning) 알고리즘 중 하나로 알려져 있다. 여러 부스팅 알고리즘 가운데 가장 많이 알려진 부스팅 알고리즘으로는 Freund와 Schapire (1997)가 제안한 AdaBoost와 Brieman (1998)이 제안한 Arc-x4가 있다.

본 논문에서는 그룹이 2개인 경우만을 고려하도록 하겠다. 훈련 자료 집합을 $\{(\mathrm{x}_1, y_1), \ldots, (\mathrm{x}_n, y_n)\}$이라 하자. 여기서 x는 예측변수들로 이루어진 벡터이며 y는 -1과 1을 갖는 그룹 변수이다. 약한 분류 알고리즘(예를 들면 의사결정나무)에 수정된 가중치가 부여된 주어진 훈련 자료 집합에 반복적으로 적용하여 약한(weak) 분류자 $h_t(\mathrm{x})$, $t = 1, 2, \ldots, T$를 만든다.

다음 장의 편의를 위해 Rudin 등 (2004)이 서술한 AdaBoost 알고리즘을 소개한다. $d_t \in R^n$를 t번째 반복에서의 가중치를 나타내는 열벡터(column vector)라 하고 N을 약한 분류자에 의해 만들어질 수 있는 총 분류자의 수라 하자. 두 개의 그룹을 갖는 문제이므로 N은 유한(기껏해야 2^n개)이지만 매우 클 수는 있다. 가능한 모든 분류자를 $h_1, \ldots, h_N$이라 표시하기로 한다. $M_{ij} = y_i h_j(\mathrm{x}_i)$, 즉 i번째 개체가 분류자 h_j에 의해 옳게 분류되면 $M_{ij} = 1$ 그렇지 않으면 $M_{ij} = -1$이 되는 행렬 M을 고려한다. 최종적으로 결합된 분류자에서 분류자 h_j에 대한 계수를 λ_j라 하면 최종적으로 결합된 분류자는 $H(\mathrm{x}) = \sum \lambda_j / ||\lambda||_1 h_j(\mathrm{x})$ (단 $||\lambda||_j = \sum \lambda_j$)가 될 것이다. 그림 2.1에서는 AdaBoost 알고리즘을 자세히 기술해 놓았다. Arc-x4는 AdaBoost 알고리즘에서 (a)와 (d)부분을 수정하면 된다. (a)에서는 가중치 부분인 $d_{t,i}$를 $d_{t,i} = [1 + \{\sum_j (1 - M_{ij})/2\, \lambda_j^{(t)}\}^4] / \sum_k [1 + \{\sum_j (1 - M_{kj})/2\, \lambda_j^{(t)}\}^4]$로 수정하고 (d) 부분은 $\alpha_t = 1$(즉 동일 가중치)로 바꾸면 된다.

2.2. Streaming Ensemble Algorithm

Street와 Kim (2001)은 Streaming Ensemble Algorithm(SEA)라고 불리는 순차적 데이터나 대용량 데

1. 입력: Matrix M, 반복수 $t_{\max}$
2. 초기화: $\lambda_j^{(1)} = 0$ for $j = 1, 2, \ldots, N$ (즉 $\lambda^{(1)}$은 0이 N개인 열벡터임.)
3. $t = 1, 2, \ldots, t_{\max}$ 반복
   (a) $d_{t,i} = e^{(-M\lambda^{(t)})_i} / \sum_k e^{(-M\lambda^{(t)})_k}$, for $i = 1, \ldots, n$.
   (b) $j_t = \arg\max_j (d_t^T M)_j$.
   (c) $r_t = (d_t^T M)_{j_t}$, $\epsilon_t = (1 - r_t)/2$.
   (d) $\alpha_t = 1/2 \ln\{(1 - \epsilon_t)/\epsilon_t\}$.
   (e) $\lambda^{(t+1)} = \lambda^{(t)} + \alpha_t e_{j_t}$, 단 e_{j_t}는 j_t번째만 1이고 나머지는 0인 단위벡터임.
4. 결과 : $H(\mathrm{x}) = \text{sign}\left(\sum_j \frac{\lambda_j^{(t_{\max}+1)}}{||\lambda^{(t_{\max}+1)}||_1} h_j(\mathrm{x}) \right)$.

그림 **2.1.** AdaBoost 알고리즘

이터를 위한 배깅(bagging) 형태의 알고리즘을 제안하였다. 이 알고리즘을 요약하면 다음과 같다. 개별 분류자는 각 자료 배치(batch)나 자료의 한 부분에서 만들고 순차적으로 읽어 들인다. 각 분류자는 정해진 숫자만큼 결합하여 앙상블(ensemble)을 만든다. 결합한 분류자의 수가 미리 정한 수가 되면 새로 들어오는 분류자는 앙상블의 성능을 향상시키는 정도를 기초로 한 퀄리티(quality) 조건을 만족시킬 때만 결합한다. 이 경우 기존의 분류자 중 하나는 앙상블 크기를 유지하기 위해 앙상블에서 제거되어야 한다. 성능 추정은 새로 들어오는 데이터를 이용해서 새로 만들어진 개별 분류자와 기존의 앙상블에 속한 개별 분류자의 퀄리티를 계산하여 이루어진다. 이 알고리즘과 사용된 퀄리티 측도에 대한 자세한 내용은 Street와 Kim (2001)을 참조한다.

3. 부스팅 개념을 이용한 순차적 앙상블 방법

대부분의 정적 자료 상황에서는, 부스팅의 성능이 배깅의 성능보다 좋은 것으로 알려져 있다 (Brieman, 1998). 따라서 순차적 자료 상황에서도 부스팅에 기초한 앙상블 방법이 배깅에 기초한 방법보다 우수한 성능을 가질 수 있을 거라 예상할 수 있다. 이번 장에서는 순차적 자료에 이용할 수 있는 부스팅 알고리즘에 기초한 새로운 앙상블 방법을 소개하도록 한다. 대용량 자료의 경우 자료를 T개로 분할하여 적절하게 순서를 정하여 마치 순차적으로 자료가 들어오는 것처럼 생각하여 알고리즘을 적용할 수 있으므로 생략하도록 한다.

t번째 훈련 자료 집합을 $\{(\mathrm{x}_{t,1}, y_{t,1}), \ldots, (\mathrm{x}_{t,n_t}, y_{t,n_t})\}$이라 하자. 여기서 $\mathrm{x}_{t,j}$는 예측 변수로 이루어진 벡터이며 $y_{t,j}$는 그룹을 나타내는 변수로 1 또는 -1의 값을 갖는다. 이 알고리즘에서, 순차적으로 t시점의 개별 분류자 h_t를 만드는데 필요한 t시점 자료에 대한 가중치는, $t-1$시점까지 만들어진 개별 분류자 $h_1, \ldots, h_{t-1}$를 부스팅 알고리즘에서의 개별 분류자처럼 간주하여 이 분류자들을 부스팅 알고리즘에 적용한 후 가중치를 결정하고, 이를 이용하여 가중 오류(weighted error)를 최소로 하는 h_t를 만들어 낸다. 정적인 상황의 부스팅 알고리즘과의 차이점은, 매시점에서 부스팅 앙상블을 새롭게 만들어야 하므로 부스팅 앙상블을 만들어 낼 때 기존에 생성된 분류자들에 대한 계수가 달라진다는 것이다. 즉 각 시점에서의 부스팅 앙상블이 기존에 만들어진 개별 분류자들을 토대로 새롭게 만들어 진다는 점에서 차이가 있다. 이는 시간이 흘러감에 따라 올 수 있는 변화에도 대응할 수 있다는 점에서 장점을 갖는

T: 순차적으로 들어온 데이터 배치의 수, Mval: 부스팅 회수

$t = 1, \ldots, T$반복

1. $t = 1$이면 $h_1 = \arg\max \sum y_{1,i} h(\mathrm{x}_1, i)$.
2. $t \neq 1$이면
   (a) $\beta_t^{(1)} = 0$: $t-1$ 차원 벡터.
   (b) $m_i = \left(y_{t,i} h_1(\mathrm{x}_{t,i}), \ldots, y_{t,i} h_{t-1}(\mathrm{x}_{t,i})\right)'$
   (c) $M = (m_1, m_2, \ldots, m_{n_t})$, $m^* = \min(\mathrm{Mval}, t-1)$
   (d) $j = 1, \ldots, m^*$ 반복
      (1) $d_{j,i} = e^{(-M\beta_t^{(j)})_i} / \sum_k e^{(-M\beta_t^{(j)})_k}$
      $\left(\text{Arc-x4의 경우 } d_{j,i} = \left[1 + \left\{\sum_l (1 - M_{il})/2\ \beta_{t,l}^{(j)}\right\}^4\right] \Big/ \sum_k \left[1 + \left\{\sum_l (1 - M_{kl})/2\ \beta_{t,l}^{(j)}\right\}^4\right]\right.$
      $\left.\text{단 } \beta_t^{(j)} = \left(\beta_{t,1}^{(j)}, \ldots, \beta_{t,t-1}^{(j)}\right)', M = (M_{ij})_{n_t \times t-1}\right)$
      (2) $k_j = \arg\max_k (d_j' M)_k$ 단 $d_j = (d_{j,1}, d_{j,2}, \ldots, d_{j,n_t})'$
      (3) $r_j = (d_j' M)_{k_j}$, $\epsilon_j = (1 - r_j)/2$
      (4) $\alpha_j = 1/2 \ln\{(1 - \epsilon_j)/\epsilon_j\}$ (Arc-x4의 경우 $\alpha_j = 1$)
      (5) $\beta_t^{(j+1)} = \beta_t^{(j)} + \alpha_j e_{k_j}$ 단 e_{k_j}는 k_j번째만 1이고 나머지는 0인 $t-1$차원 단위벡터임.
   (e) $d_{t,i} = e^{\left(-M\beta_t^{(m^*+1)}\right)_i} / \sum_k e^{\left(-M\beta_t^{(m^*+1)}\right)_k}$
   $\left(\text{Arc-x4의 경우 } d_{j,i} = \left[1 + \left\{\sum_l (1 - M_{il})/2\ \beta_{t,l}^{(m^*+1)}\right\}^4\right] \Big/ \sum_k \left[1 + \left\{\sum_l (1 - M_{kl})/2\ \beta_{t,l}^{(m^*+1)}\right\}^4\right]\right)$
   (f) $h_t = \arg\max \sum d_{t,i} y_{t,i} h(\mathrm{x}_{t,i})$, ϵ_t : h_t의 가중 오류
   (g) $\beta = \dfrac{1}{2} \ln\left(\dfrac{1 - \epsilon_t}{\epsilon_t}\right)$ (Arc-x4의 경우 $\beta = 1$)
   (h) $\lambda_t = \left(\beta_t^{(m^*+1)'}, \beta\right)'$
   (i) $H_t(\mathrm{x}) = \mathrm{sign}\left(\dfrac{h(\mathrm{x})' \lambda_t}{||\lambda_t||_1}\right)$, 단 $h(\mathrm{x}) = (h_1(\mathrm{x}), \ldots, h_t(\mathrm{x}))'$

그림 **3.1.** 순차적 데이터를 위한 부스팅 알고리즘

다고 볼 수 있다. 다만 개별 분류자를 기억하고 있어야 하므로 개별 분류자가 간단할수록 장점이 커지는데 부스팅 알고리즘은 간단한 분류자로도 좋은 성능을 보이는 것으로 알려져 있어 (Hastie 등, 2001, Chapter 10) 이러한 문제를 해결할 수 있다. 자세한 알고리즘은 그림 3.1에 나와 있다. 그림 3.1에서 2번 항의 (a)–(d)는 이전 시점까지 얻어진 개별 분류자들을 이용해 가중치를 만들어 가는 과정이며 이렇게 해서 만들어진 가중치를 이용해 새로운 분류자를 만드는 과정이 (e)와 (f)이며 (g)–(i)는 t시점에서 부스팅 앙상블을 만드는 과정이다. 고정된 t시점에서만 고려하면 기존 $t-1$개의 분류자와 현재 시점에서 새로 만들어진 분류자를 이용하는 부스팅 알고리즘과 거의 유사하다. SEA 방법은 배깅과 마찬가지로 기존에 만들어진 개별 분류자에 동일한 가중치를 주어 앙상블을 형성하므로 개념의 변화가 일어나도 개념의 변화전에 생성된 개별 분류자의 영향이 한동안 남아있게 된다. 하지만 제안된 부스팅 방법은 기

존의 분류자들이 마찬가지로 이용되지만 앙상블 형성 때 이용되는 계수(혹은 가중치)를 달리하여 그 영향이 빨리 극복될 수 있을 것이다. 게다가 SEA 방법은 일단 앙상블에서 제거된 개별 분류자를 다시 사용할 수 없지만 부스팅 방법은 현재 시점에 적합한 개별 분류자를 이미 만들어진 분류자 가운데 선택하므로 기존의 개별 분류자가 제거되어 생기는 약점을 극복할 수 있다.

4. 모의 실험과 실제 자료 분석

4.1. 모의 실험

배깅 타입의 방법 즉 SEA와의 비교를 위해 다음과 같은 모의 실험을 실시하였다.

4.1.1. 모의 실험 자료 비교를 위해 다음과 같은 자료를 사용하였다.

- Sphere 자료: 샘플 (x, y)는 3차원의 서로 독립인 $\mathrm{x} = (x_1, x_2, x_3)$을 갖는다. 단 $x_i \in [0,1]$, $i = 1,2,3$이다. 기하학적으로 보면 샘플은 3차원 입방체(cube)에 위치하고 있다. 실제 그룹의 경계는 다음과 같이 정의되는 구면이다:

$$B(x) = \sum_{i=1}^{3}(x_i - c_i)^2 - r^2 = 0,$$

  여기서 $c = (c_1, c_2, c_3)$는 구의 중심이며, r은 반지름이다. $B(x) \leq 0$이면 $y = 1$이고 $B(x) > 0$이면 $y = -1$이다. 이 자료는 예측 변수들이 모두 연속형이고 그룹 경계가 비선형이기 때문에 학습(learning)이 쉽지 않다.

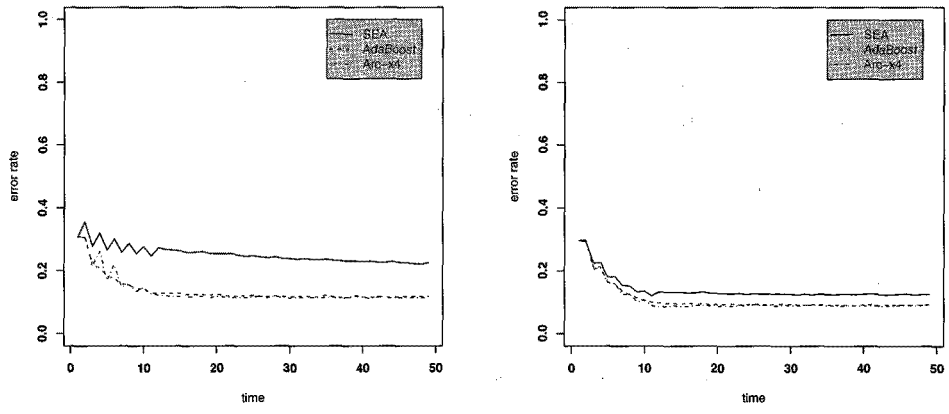
- Twonorm 자료: 이 자료는 예측 변수가 20차원이며 2개의 그룹을 갖는 데이터이다. 각 그룹은 단위 공분산 행렬을 갖는 다변량 정규분포에서 생성된다. 그룹 1은 모평균이 $(a, a, \ldots, a)$이며 그룹 2는 모평균이 $(-a, -a, \ldots, -a)$이다.

Sphere 자료의 경우, 개념의 변화가 없는 경우(no concept drift)는 구의 중심 c를 변화시키지 않으며 개념의 변화가 있으면 구의 중심을 각 차원별로 $\pm\delta$만큼 변화시킨다. 예를 들면 현재 시점(block)에서 구의 중심이 $c = (0.40, 0.60, 0.50)$, $\delta = 0.05$이고 각 차원 이동 부호가 $(+, -, -)$이라면 다음 시점(block)에서 구의 중심은 $c = (0.45, 0.55, 0.45)$가 된다. 본 모의 실험에서는 시작 시점의 구의 중심을 $c = (0.5, 0.5, 0.5)$에서 시작하였고 구의 반지름 $r = 0.5$, 개념의 변화가 있을 경우 $\delta = 0.2$ 없을 경우, $\delta = 0$으로 하였다.

Twonorm 자료의 경우, 개념 변화가 없으면 각 그룹의 평균이 고정되며 개념의 변화가 있는 경우에는 평균의 부호가 r%가 변화하도록 하였다. 본 모의 실험에서는 $a = 2/\sqrt{20}$, $r = 40$을 사용하였다.

각 시점에서의 훈련 자료 집합의 크기는 500이다. 각 시점에서 생성된 앙상블의 오분류율을 측정하기 위해 2,000개의 검증 자료 집합(test data set)을 사용하였다. 50번째 시점까지 자료를 생성하였고 개념의 변화(concept drift)가 있는 경우 20번째 시점에서 변화가 생기도록 하였다.

4.1.2. 대용량 자료 분류를 위한 모의 실험 SEA 방법이나 제안된 부스팅 방법은 3장에서 언급했듯이 대용량 자료에 이용할 수 있다. 대용량 자료를 상대적으로 작은 숫자의 자료로 나누어서 나누어진 각각의 자료에서 분류자들을 만들고 이를 이용한 앙상블 방법을 적용하면 가능하다. 이런 방법을 사용한다면 전체 대용량 자료를 이용하는 일반적인 배깅이나 부스팅 방법보다 더 빠르면서도 메모리가 적게 드는 결과를 얻을 수 있을 것이다. 비교를 위해 4.1.1의 두 모의 실험 자료를 이용하였다. 훈련 자료 집합의



(a) Sphere 자료 (b) Twonorm 자료

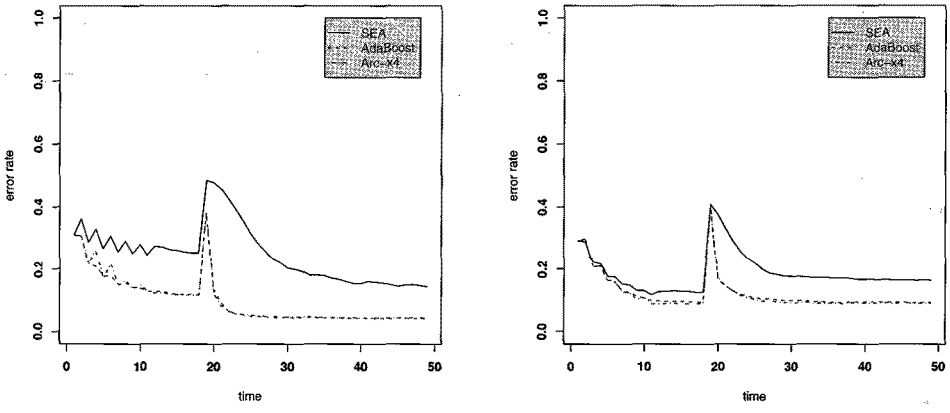
그림 4.1. 개념 변화가 없는 경우의 오분류율

크기는 25,000이고 이 자료를 임의로 50개로 나누었다. 2,000개의 검증 자료 집합을 이용하여 각 방법의 오분류율(misclassification rate)을 계산하였다.

4.1.3. 개별 분류자와 비교 앙상블 방법 본 모의 실험에서는 개별 분류자를 가지치기(pruning)가 없는 CART를 이용하여 배깅 타입 방법인 SEA와 2개의 부스팅 타입 앙상블 방법을 비교하였다. 각 시점에서 간단한 분류자를 이용해야만 위 세가지 앙상블 방법이 더욱 효과적이므로 뎁스(depth)가 2인 의사결정나무를 생성하였다. 각 시점에서의 평균 오분류율을 계산하기 위해 50번 반복 시행하였으며 각 시점에서 앙상블의 크기는 10으로 고정하였다.

4.1.4. 모의 실험 결과 그림 4.1은 개념의 변화가 없는 경우의 결과이다. Sphere 자료의 경우 두 개의 부스팅 방법이 SEA 방법보다 좋은 결과를 보인다. 일반적으로 배깅은 분산만을 줄이는 것으로 알려져 있다. 만약 각 개별 분류자가 큰 편의(bias)를 갖고 있다면 배깅 앙상블도 편의를 가지게 된다. 정적인 상황과 마찬가지로 순차적 데이터 상황에서도 배깅 타입의 방법은 각 시점의 개별 분류자가 상대적 큰 편의를 가지고 있다면 부스팅 타입 방법보다는 좋지 않은 결과를 나타내는 것으로 보인다. 본 모의 실험에서 각 의사결정나무의 뎁스가 2이므로 Sphere 자료에 대해서는 편의를 갖고 있으므로 배깅 타입은 좋은 선택이 아니라 볼 수 있다. Twonorm 자료의 경우 3가지 방법 모두 비슷한 결과를 보였지만 부스팅 타입의 방법의 정확도가 약간 큰 것을 볼 수 있다. 이는 정적인 상황에서의 Twonorm 자료의 경우와 비슷한 결과이다 (Breiman, 1998).

개념의 변화가 있는 경우의 결과는 그림 4.2에서 볼 수 있다. 개념의 변화가 생긴 후에 오분류율을 보면 SEA의 경우 6–7번의 시점이 지나야 회복이 되는 반면, 두 부스팅 방법은 2–3번의 시점이 지나면 곧바로 회복이 됨을 알 수 있다. 이를 통해 SEA는 부스팅 방법에 비해 개념 변화로 인한 충격에서 상대적으로 천천히 회복됨을 볼 수 있다. 이는 SEA의 특성에 기인하는데 개념의 변화가 생기더라도 개념이 변화되기 전에 만들어진 많은 개별 분류자가 앙상블에 포함되어 있기 때문이다. 이러한 분류자는 개념이 변화된 후에는 변화된 상황에 적절하지 않으므로 제거되거나 가중치의 변화가 있어야 하지만 SEA 앙상블 방법은 빠른 시간 안에 그렇게 할 수 없다. 그러나, 부스팅 방법은 가중치의 변화와 적합한 분류자의 선택을 통해 이를 극복할 수 있는 것으로 보인다. 이 결과를 통해 개념의 변화가 있는 경우 부스팅 타입



(a) Sphere 자료 (b) Twonorm 자료

그림 4.2. 개념 변화가 있는 경우의 오분류율

표 4.1. 대용량 자료 모의 실험 결과

방법	Sphere 자료		Twonorm 자료	
	Test error 평균	Test error 표준오차	Test error 평균	Test error 표준오차
Bagging	0.231	0.002	0.117	0.002
AdaBoost	0.079	0.001	0.090	0.001
Arc-x4	0.091	0.001	0.089	0.001

이 배깅 타입의 앙상블 방법보다는 훨씬 더 적합한 것을 알 수 있다.

대용량 자료의 경우, 표 4.1에 결과를 나타내었다. Sphere 자료의 경우 두 개의 부스팅 타입의 앙상블 방법이 SEA 방법보다는 좋은 결과를 얻었는데 이는 앞에서 언급했던 것처럼 의사결정나무가 가진 편의 효과가 그대로 SEA 앙상블에도 전해진 것으로 보인다. Twonorm 자료에서는 배깅 방법도 좋은 결과를 보였지만 제안된 방법의 오분류율이 SEA보다는 약간 작게 나타났다. 이 결과는 순차적 자료 경우와 같이 대용량 자료의 분류의 경우에도 부스팅 타입의 앙상블이 배깅 타입보다는 좋은 결과를 보일 수 있음을 보여준다.

4.2. 실제 자료 분석

다음의 두 데이터는 대용량 자료의 분류를 위한 제안된 방법의 효과를 측정하기 위해 사용하였다.

- Adult: 이 데이터는 Kohavi (1996)가 여러 가지 분류 방법을 비교하기 위해 사용한 미국 Census Bureau의 자료이다. 나이와 교육 수준, 직업, 성별 등 14가지 인구통계학적 특성 등을 기초로 연 50,000달러 이상 혹은 이하의 수입을 내는지를 예측하는 문제이다. 이 자료에는 50,000달러 이상의 소득이 23.93%를 차지하고 있으며 총 48,842명의 자료가 있다. 훈련 자료(training data)는 32,561개이며 나머지는 검증 자료로 사용한다.
- Anonymous Web browsing: 이 자료는 Microsoft 웹 싸이트를 방문한 32,117명의 Web browsing 특성을 기록한 자료이다. 여기서는 사용자가 방문한 web 페이지를 기초로 하여 "Free downloads" 페이지를 방문하는지를 예측하고자 한다. 이 자료에는 "Free downloads"페이지를 방문한 유저가 10,835명(33.1%)이며, 294개의 이항 예측변수가 있다. 검증 자료의 크기는 5,000이다.

표 4.2. 실제 자료 분석 결과

자료	방법	블록 크기: 500		블록 크기: 1000	
		오분류율 평균	표준오차	오분류율 평균	표준오차
Adult	Bagging	0.158	0.001	0.161	0.001
	AdaBoost	0.165	0.002	0.150	0.001
	Arc-x4	0.156	0.002	0.149	0.001
Anonymous Web Browsing	Bagging	0.289	0.001	0.290	0.001
	AdaBoost	0.277	0.002	0.270	0.002
	Arc-x4	0.280	0.003	0.272	0.002

이 두 자료는 UCI machine learning repository (Asuncion과 Newman, 2007)에서 이용할 수 있다. 자료를 임의로 나누어야 하기 때문에 10번 반복 시행하여 오분류율의 평균을 비교하였다. 또한 각 블록 크기가 500과 1,000 정도가 되도록 하였다. 개별 분류자로는 뎁스가 2인 의사결정나무를 사용하였다. 표 4.2에서는 이 두 자료에 대한 결과를 볼 수 있다. 대부분의 경우 부스팅 방법이 배깅 방법보다는 약간 좋은 결과를 보였다. 그러나 모의 실험에서와 마찬가지로 AdaBoost와 Arc-x4에는 큰 차이를 보이지 않았다. 모든 자료를 이용한 부스팅 방법(사용된 의사결정나무의 뎁스는 2이며 반복 회수는 30회(30회 이후의 오분류율은 변함이 없었음)로 하였음)을 적합시키면 검증 자료의 오분류은 각각 0.141(Adult 자료), 0.275(Anonymous Web browsing 자료)로 Adult 자료의 경우는 순차적 알고리즘보다 약간 좋았다. 하지만 Anonymous Web browsing 자료의 경우 순차적 부스팅 방법이 오히려 평균적으로 더 좋은 결과를 보이는 경우도 있었다.

5. 결론

대용량 자료나 순차적 자료를 기존의 방법으로 분석하는 것은 쉽지 않다. 이러한 자료들을 분석하는데 앙상블 방법은 좋은 해결책이 될 수 있음이 알려져 있다. SEA 방법은 이를 해결하기 위한 앙상블 방법 중 구현이 쉽고 효과적인 방법 중 하나이다. 그러나 SEA는 배깅이 갖고 있는 약점인 편의 문제를 그대로 갖고 있다. 또한 개념의 변화(concept drift)에 상대적으로 느리게 적응하는 약점도 있다. 이런 약점은, 정적 상황에서 배깅이 갖고 있는 약점들을 보완할 수 있는 것으로 알려져 있는 부스팅 방법의 개념을 대용량 자료나 순차적 자료 상황에도 적용시켜면 해결할 수 있음을 모의 실험과 실제 자료 분석을 통해 보였다.

본 연구에서는 앙상블의 크기, 즉 앙상블에 포함되는 분류자의 수를 고정시켰지만 데이터에 적합한 크기를 자동적으로 선택하는 것에 대해 연구해 볼만한 가치가 있다. 이를 위해서 Regularization 방법을 도입하는 것도 고려해 볼 수 있을 것이다. 또한 여러 다양한 앙상블 방법을 이용하여 기존의 방법과 비교하는 것도 필요하다고 본다. 이론적 혹은 근사적으로 자료나 독립변수의 개수와 적합하는데 걸리는 시간이나 사용된 저장 공간과의 관계를 구하는 것도 차후 연구 주제 중 하나이다.

참고문헌

Asuncion, A. and Newman, D. J. (2007). UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science.

Breiman, L. (1998). Arcing classifiers (with discussion), *Annals of Statistics*, **26**, 801–849.

Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984), *Classification and Regression Trees*, Chapman & Hall, New York.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of online learning and application to boosting, *Journal of Computer and System Science*, **55**, 119–139.

Hastie, T., Tibshirani, R. and Friedman, J. (2001). *The Elements of Statistical Learning*, Springer-Verlag, New York.

Kohavi, R. (1996). Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid, *Proceedings of the second International Conference on Knowledge Discovery and Data Mining*, 202–207.

Kuncheva, L. I. (2004). Classification ensemble for changing environments, *Proceedings of 5th International Workshop on Multiple Classifier Systems*, 1–15.

Quinlan, J. R. (1993). *C4.5: Prigrams for Machine Learning*, Morgan Kaufmann, San Maeto, CA.

Rudin, C., Daubechies, I. and Schapire, R. E. (2004). The dynamics of AdaBoost: cyclic behavior and convergence of margins, *Journal of Machine Learning Research*, **5**, 1557–1595.

Street, W. N. and Kim, Y. S. (2001). A streaming ensemble algorithm (SEA) for large scale classification, *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 377–382.

Wang, H., Fan, W., Yu, P. S. and Han, J. (2003). Mining concept drifting data streams using ensemble classifiers, *Proceedings of then 9th ACM SIGKDD International Conference on Knowledge discovery and Data Mining*, 226–235.

Yeon, K., Choi, H., Yoon, Y. J. and Song, M. S. (2005). Model based ensemble learning for tracking concept drift, *Proceedings of 55th Session of the International Statistical Institute.*

Boosting Algorithms for Large-Scale Data and Data Batch Stream

Young Joo Yoon[1]

[1]Department of Statistics, University of Georgia

(Received December 2009; accepted January 2010)

Abstract

In this paper, we propose boosting algorithms when data are very large or coming in batches sequentially over time. In this situation, ordinary boosting algorithm may be inappropriate because it requires the availability of all of the training set at once. To apply to large scale data or data batch stream, we modify the AdaBoost and Arc-x4. These algorithms have good results for both large scale data and data batch stream with or without concept drift on simulated data and real data sets.

Keywords: AdaBoost, Arc-x4, concept drift, data stream, ensemble method, large scale data.

[1]Postdoctoral Fellow, Department of Statistics, University of Georgia, 101 Cedar Street Athens, GA, USA 30602. E-mail: yoonyj74@uga.edu