

# Dimension Reduction Methods on High Dimensional Streaming Data with Concept Drift

Cheong Hee Park<sup>†</sup>

## ABSTRACT

While dimension reduction methods on high dimensional data have been widely studied, research on dimension reduction methods for high dimensional streaming data with concept drift is limited. In this paper, we review incremental dimension reduction methods and propose a method to apply dimension reduction efficiently in order to improve classification performance on high dimensional streaming data with concept drift.

**Keywords :** High Dimensional Streaming Data, Incremental Dimension Reduction Method, Adaptive Classifier, Concept Drift

## 개념 변동 고차원 스트리밍 데이터에 대한 차원 감소 방법

박 정 희<sup>†</sup>

### 요 약

고차원데이터에 대한 차원 감소 기법들은 많이 연구되어져 온 반면, 개념 변동을 가진 고차원 스트리밍 데이터에서 적용할 수 있는 차원 감소 기법에 대한 연구는 제한적이다. 이 논문에서는 스트리밍 데이터에서 적용할 수 있는 점층적 차원 감소 기법들을 살펴보고, 개념 변동 고차원 스트리밍 데이터에 대해 분류 성능을 향상시킬 수 있도록 차원 감소를 효과적으로 적용하는 방법을 제안한다.

**키워드 :** 고차원 스트리밍 데이터, 점층적 차원 감소 기법, 적응적 분류기, 개념 변동

### 1. 서 론

대부분의 데이터마이닝 알고리즘들은 데이터가 수집되고 저장되어진 후 이것을 한꺼번에 메모리에 올려 분석하는 배치 모드 방식을 기본으로 개발되어져 왔다. 그러나 점점 더 다양한 형태의 데이터 생성과 저장 방식이 가능해짐에 따라 이에 맞는 데이터 분석 방법의 개발도 꾸준히 이루어지고 있다. 시간이 흐름에 따라 끊임없이 생성되는 연속된 형태의 스트리밍 데이터(streaming data)는 시간에 따라 데이터 패턴이 변화되는 가능성을 내포하고 있다. 통신회사에서 통신내역을 기록하는 데이터나 전력회사에서의 전력생산 및 전력소비와 관련된 데이터, 각종 센서 데이터 등을 예로 들 수 있다.

개념 변동이란 시간이 흐름에 따라 데이터 분포가 변하거나 관심있는 개념이 달라지는 것 등을 모두 포함한다[1]. 개념 변동 스트리밍 데이터 분석에서는 데이터 발생 분포에서의 변화나 유지의 관심이 변하는 경향을 탐지함으로써 분류모델이나 군집모델의 새로운 학습을 유도하는 것이 중요하다. 분류 기법을 스트리밍 데이터에 적용시키는 방법은 크게 세 가지로 구분할 수 있다. 첫 번째는 기본 분류 알고리즘을 새로운 데이터 샘플이 들어올 때 마다 또는 새로운 데이터 샘플들의 부분집합(데이터 청크)이 형성될 때마다 업데이트하는 것이다[2]. 두 번째는 최근의 데이터 청크(chunk)에 대해 모델링된 분류기를 과거의 데이터 청크마다 학습된 분류기들과 함께 분류기 앙상블을 구성하여 예측에 적용하는 것이다[3, 4]. 세 번째는 데이터 발생 분포에서의 변화나 유지의 관심이 변하는 경향을 탐지하는 변화탐지기법과 연동하여 변화 탐지가 될 때마다 새로운 분류기 모델링을 시작한다[5].

개념 변화 발생 또는 미발생의 이분법적인 판단은 자칫 분류기 성능을 크게 떨어뜨리는 오류를 범하기 쉽다. 대신에 새로운 데이터 샘플이나 데이터 청크가 들어왔을 때 개

※ 이 연구는 2015년도 충남대학교 자체연구비에 의해 지원되었음.

† 정 희 원 : 충남대학교 컴퓨터공학과 교수

Manuscript Received : January 8, 2016

First Revision : March 21, 2016

Second Revision : March 24, 2016

Third Revision : April 5, 2016

Accepted : April 6, 2016

\* Corresponding Author : Cheong Hee Park(cheonghee@cnu.ac.kr)

념 변화 발생 정도를 감지하여 업데이트 과정에서 최근 데이터에 대한 비중을 과거 데이터에 대한 비중보다 높게 함으로써 변화 발생에 적응 가능한 분류기를 구성할 수 있다[6]. 논문[7]에서는 변화 발생 예측 척도를 사용하여 최신 데이터 샘플의 비중을 자동적으로 조절하는 적응적(adaptive)이면서 점층적(incremental)인 분류 모델 업데이트 방법을 개발하고 이를 두 개의 선형 분류기에 적용하였다. 개념 변화가 발생했을 때 빠르게 분류기가 적응하도록 업데이트함으로써 분류 성능을 유지할 수 있음을 보였다. 그러나 고차원 스트리밍 데이터에서는 고차원 행렬연산에서 요구되는 높은 시간과 공간복잡도로 인하여 차원 감소 등의 데이터 전처리 과정이 필요하게 된다.

스트리밍 데이터 분석에서의 시간적, 공간적 제약 조건은 배치모드에서 높은 성능을 보이는 차원감소기법들의 적용에 한계를 두게 한다. 고차원데이터에 대한 성능이 우수한 차원 감소 기법들은 많이 연구되어져 온 반면, 고차원 스트리밍 데이터에서 적용할 수 있는 차원 감소 기법은 incremental PCA[8] 또는 이를 이용한 차원 감소법[9, 10] 등으로 제한적이다. 더구나 개념 변동 가능성이 있는 고차원 스트리밍 데이터에서 차원감소 기법을 어떻게 적용시켜야 분류성능 향상을 가져올 수 있는지에 대한 연구는 미비하다. 다큐먼트, 이메일, 이미지나 동영상 등 고차원 속성을 지닌 스트리밍 데이터의 발생이 보편적임을 고려할 때, 효과적으로 차원 감소 기법을 적용하여 개념 변동 고차원 스트리밍 데이터의 분석 성능을 향상시킬 수 있는 방법에 대한 연구가 필요하다.

본 논문에서는 개념 변동 고차원 스트리밍 데이터에서 분류기법을 효과적으로 적용할 수 있는 방법으로서 점층적 차원 감소와 적응적 분류기를 결합시키는 방법을 제안하고, 다양한 상황에서의 실험을 통해 효과를 분석한다. 제안하는 방법은 시간적 공간적 제약이 있는 스트리밍 모드에서 점층적 차원 감소 방법으로 고차원 데이터를 저차원으로 변환하고, 변환된 저차원 공간에서 개념 변동 발생 정도를 계산하여 그 정도에 따라 적응적으로 분류기를 업데이트함으로써 개념 변동에 능동적으로 대처할 수 있게 된다. 또한, 실험을 통해 개념 변동 발생으로 인한 갑작스런 성능 악화를 보다 신속하게 회복할 수 있는 차원 감소 방법이 무엇인지, 감소되는 차원에 따른 성능 변화는 어떠한지, 적응적 분류기의 사용이 분류 성능에 미치는 영향 등 실제적인 사용을 위한 가이드를 제공한다.

본 논문의 구성은 다음과 같다. 2절에서는 스트리밍 데이터에 대한 차원 감소 기법들을 살펴보고, [7]에서 제안된 적응적 선형 분류 기법에 대해 간단히 요약한다. 3절에서는 개념 변동 고차원 스트리밍 데이터에 대해 점층적 차원 감소 기법과 적응적 분류 기법을 결합하여 적용하는 방법을 제안한다. 4절에서는 고차원 스트리밍 데이터를 이용하여 개념 변동 발생 상황에서 제안된 방법에 의한 분류 성능을 분석한다.

## 2. 관련 연구

### 2.1 스트리밍 데이터에 대한 점층적 차원감소

차원감소를 스트리밍 데이터에 적용하기 위해 많이 사용되는 방법은 점층적 주성분분석 기법이다. CCIPCA (Candid covariance-free incremental PCA)[8]는 공분산 행렬의 계산을 피하면서 점층적으로 주성분(principal components)들을 업데이트하는 차원 감소 방법이다. 데이터 샘플  $x_1, \dots, x_{n-1}$ 을 이용한 첫 번째 주성분 벡터를  $v_1(n-1)$ 이라고 할 때,  $x_n$ 을 이용한  $v_1(n-1)$ 의 업데이트는 Equation (1)을 사용하여  $i=1$ ,  $z=x_n$ 으로 하여 계산된다. Equation (1)에서 매개변수  $l$ 은 새로운 샘플에 더 많은 가중치를 주기 위한 것으로 2에서 4까지의 범위의 값이 추천된다.

$$v_i(n) = \frac{n-1-l}{n}v_i(n-1) + \frac{1+l}{n}z z^T \frac{v_i(n-1)}{\|v_i(n-1)\|} \quad (1)$$

두 번째 주성분 벡터  $v_2(n)$ 은 벡터  $z=x_n - x_n^T \frac{v_1(n)}{\|v_1(n)\|} \frac{v_1(n)}{\|v_1(n)\|}$ 에 대해 Equation (1)로부터 구해진다. 나머지 주성분 벡터는 같은 방법을 되풀이하여 차례대로 계산할 수 있다. CCIPCA[8]을 Table 1에 요약하였다.

Table 1. Algorithm for CCIPCA [8]

CCIPCA : compute the first $k$ dominant eigenvectors $v_1(n), \dots, v_k(n)$ from data streams $x_1, x_2, \dots$ ,
for $n = 1, 2, \dots$ , do the following steps,
$u_1(n) = x_n$
for $i = 1, \dots, \min(k, n)$ do,
if $i = n$
initialize the $i$ th eigenvector as $v_i(n) = u_i(n)$
else
$v_i(n) = \frac{n-1-l}{n}v_i(n-1) + \frac{1+l}{n}u_i(n)u_i(n)^T \frac{v_i(n-1)}{\ v_i(n-1)\ }$
$u_{i+1}(n) = u_i(n) - u_i(n)^T \frac{v_i(n)}{\ v_i(n)\ } \frac{v_i(n)}{\ v_i(n)\ }$
end if
end for
end for

주성분분석(PCA)가 데이터 샘플들의 클래스 라벨을 고려하지 않은 차원 감소 방법인 반면, 같은 클래스 내의 데이터들의 분산은 작게 하고 다른 클래스 간 분산은 크게 하는 선형 판별 분석(LDA)에 의한 차원 감소는 일반적으로 PCA보다 나은 분류 성능을 보이는 경우가 많다. 스트리밍 데이터에 적용할 수 있는 LDA 기반 점층적 차원감소 방법들

[9-12] 중에서 Incremental partial least squares(IPLS) 방법은 CCIPCA를 전처리로 이용하여 분산 행렬의 근사 행렬을 구한 후 PLS를 점층적으로 업데이트하는 방법으로 좋은 분류 성능을 나타낸다고 보고되었다[10]. PLS는 데이터 샘플들의 특징값 벡터와 클래스 라벨 벡터와의 공분산을 최대로 하는 상관성이 없는(uncorrelated) 선형 변환을 구하는 차원 감소 방법이다. IPLS의 알고리즘을 Table 2에 요약한다. 자세한 설명은 논문 [10]을 참조할 수 있다.

Table 2. Algorithm of IPLS [10]

<p>compute <math>v_1(n), \dots, v_K(n)</math> from data streams <math>(x_1, y_1), (x_2, y_2), \dots</math> where <math>y_i</math> is a class label vector of <math>x_i</math>.</p> <p>L : the number of PCA components</p>
<p><math>\bar{x}_1 = x_1, \bar{y}_1 = y_1</math>  <math>v_1(1) = 0, u_1(1) = x_1</math>                  for <math>n = 2, 3, \dots</math>, do the following steps  <math>\bar{x}_n = \frac{n-1}{n}\bar{x}_{n-1} + \frac{1}{n}x_n</math>  <math>\bar{y}_n = \frac{n-1}{n}\bar{y}_{n-1} + \frac{1}{n}y_n</math>  <math>x_n^1 = x_n - \bar{x}_n</math>  <math>\delta(n) = \bar{x}_n - \bar{x}_{n-1}</math>  <math>v_1(n) = v_1(n-1) - (n-1)\bar{y}_{n-1}\delta(n) + y_n x_n^1</math>                  for <math>i = 1, \dots, \min(n, L)</math>                  if <math>i = n</math>  <math>u_i(n) = x_n^i</math>                  else  <math>u_i(n) = \frac{n-1-l}{n}u_i(n-1) + \frac{1+l}{n}x_n^i x_n^{i^T} \frac{u_i(n-1)}{\ u_i(n-1)\ }</math>  <math>x_n^{i+1} = x_n^i - x_n^{i^T} \frac{u_i(n)}{\ u_i(n)\ } \frac{u_i(n)}{\ u_i(n)\ }</math>                  end if                  end for  <math>w = v_1(n)</math>                  for <math>i = 2, \dots, K</math>  <math>w = \frac{1}{n} \sum_{j=1}^L \frac{1}{\ u_j(n)\ } u_j(n) u_j(n)^T w</math>  <math>v_i(n) = w</math>                  for <math>j = 1, \dots, i-1</math>  <math>v_i(n) = v_i(n) - v_i(n)^T \frac{v_j(n)}{\ v_j(n)\ } \frac{v_j(n)}{\ v_j(n)\ }</math>                  end for                  end for                  Output normalized <math>v_1(n), \dots, v_K(n)</math>                  end for</p>

2.2 개념변동 스트리밍 데이터에 대한 분류기의 적응적 학습

스트리밍 데이터에서 새로운 데이터 샘플의 클래스 예측 후에 즉각적으로 실제 클래스라벨이 알려진다는 가정 하에 일반적으로 사용되는 변화탐지방법은 분류예측 오류율을 이용하는 것이다[13]. 그러나 분류 오류율은 분류기에 의한 예측 결과에 대해 0(예측이 맞음) 또는 1(예측이 틀림)이라는 이분법적인 정보만을 이용하게 된다. 반면에 분류기에 의해 데이터 샘플  $x$ 가 각 클래스  $i$ 로 예측되는 것에 대한 확률 또는 신뢰도를  $C_i(x)$ 라고 한다면  $C(x) = [C_1(x), C_2(x), \dots, C_r(x)]$ 는 분류기의 행동패턴에 대한 풍부한 정보를 줄 수 있다. 클래스  $i$ 에 속하는 데이터 샘플의 클래스 정보를 나타내는 벡터  $y \in R^{1 \times r}$ 를  $i$ -th 성분이 1이고 나머지는 0인 벡터로 나타낼 때, 랜덤 변수  $X$ 를  $X(x) = \|C(x) - y\|_2^2$ 라고 정의함으로써 분류기의 성능 패턴을 묘사할 수 있다.

데이터 스트림  $x_1, x_2, \dots$ 에 대해  $x_{n+1}$ 이 이제 막 도착한 데이터 샘플이라고 하자. 랜덤변수  $X$ 의 분포를 정규분포라고 가정하고, 평균  $\mu$ 와 분산  $\sigma^2$ 을 가장 최근에 도착한  $t$ 개의 데이터 샘플들에 의한 시퀀스  $X(x_{n-t+1}), \dots, X(x_n)$ 으로부터 구한다.  $Z$ 를 표준정규분포라고 할 때 데이터 샘플  $x_{n+1}$ 에 의해 탐지되는 개념 변화 정도는 Equation (2)의 p-value에 의해 나타낼 수 있다.

$$p\text{-value} = P(Z > \frac{X(x_{n+1}) - \mu}{\sigma}) \quad (2)$$

p-value가 작을수록  $x_{n+1}$ 에 의한 개념변동 암시 정도는 커지게 된다. 따라서 p-value가 작을수록 과거 데이터의 비중을 줄이는 적응적 학습이 필요하다. p-value로부터 과거 데이터의 비중을 나타내는 학습률  $\beta$ 로의 매핑을 위해 Fig. 1에 나타낸 함수를 이용하였다.

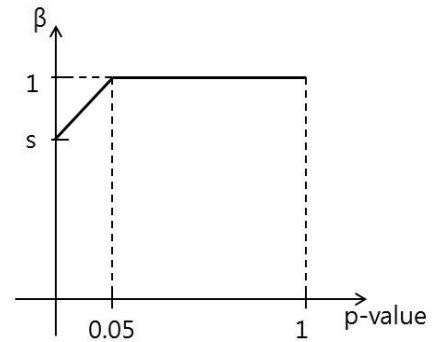


Fig. 1. A function from p-value to a learning rate

분류기의 적응적 학습을 위한 일반적인 방법을 Table 3에 요약하였다. Table 4는 Minimum Squared Error Solution (MSE)[14]를 Table 3의 적응적 학습 모델에 적용하기 위해

필요한 부분을 설명하고, Table 5는 Linear Discriminant Analysis(LDA)[6]를 Table 3의 적응적 학습 모델에 적용하기 위한 요소들을 나타낸다.

Table 3. Adaptive incremental learning algorithm [7]

Input : $F = \{X(x_{n-t+1}), \dots, X(x_n)\}$ : records by the recently received data samples $x_{n-t+1}, \dots, x_n$ $x_{n+1}, x_{n+2}, x_{n+3}, \dots$ : incoming data stream
Let $k = n + 1$ . While (incoming data sample $x_k$ is available) Compute $X(x_k) = \ C(x_k) - y_k\ _2^2$ Let $\mu$ and $\sigma$ be the mean and standard deviation of $F$ Compute $p\text{-value} = P(Z > \frac{X(x_k) - \mu}{\sigma})$ Compute $\beta$ using the function given in Fig.1 Update the classifier by the updating formula $f \leftarrow \text{Update}(\beta, f, x_k)$ Remove the oldest element and add $X(x_k)$ to $F$ . $k = k + 1$ . end of while

Table 4. ADMSE (adaptive MSE)

a classifier : $g(x) = W_n^T \hat{x}$ , where $\hat{x} = [1 \ x^T]^T$ , $M_n = A_n^T A_n$ and $W_n = M_n^{-1} A_n^T Y_n$ a confidence score vector : $C(x) = W_n^T \hat{x}$ an updating formula of a classifier : $M_{n+1}^{-1} = \frac{1}{\beta} M_n^{-1} - \frac{\frac{1}{\beta} M_n^{-1} \hat{x}_{n+1}}{I + \hat{x}_{n+1}^T \frac{1}{\beta} M_n^{-1} \hat{x}_{n+1}} \hat{x}_{n+1}^T \frac{1}{\beta} M_n^{-1}$ $W_{n+1} = W_n + M_{n+1}^{-1} \hat{x}_{n+1} (y_{n+1} - \hat{x}_{n+1}^T W_n)$
--

Table 5. ADLDA (adaptive LDA)

a classifier: $\{g_i(x) = \mu_i^T \Sigma^{-1} x - \frac{1}{2} \mu_i^T \Sigma^{-1} \mu_i + \ln P(i)\}_i$ where $\Sigma$ : within-class scatter matrix $\mu_i$ and $P(i)$ : mean and prior probability of class $i$ a confidence score vector : $C(x) = [\tilde{C}_i(x) / \sum_j \tilde{C}_j(x)]_{1 \leq j \leq r}$ where $\tilde{C}_i(x) \leftarrow P(i) \exp(-\frac{1}{2}(x - \mu_i)^T \Sigma^{-1} (x - \mu_i))$ an updating formula of a classifier : $\mu_i \leftarrow (\beta n_i \mu_i + x_{n+1}) / (\beta n_i + 1)$ $P(i) \leftarrow (\beta n_i + 1) / (\beta n + 1)$ $P(j) \leftarrow (\beta n_j) / (\beta n + 1), \quad j \neq i$ $\Sigma^{-1} \leftarrow \frac{\beta n + 1}{\beta n} (\Sigma^{-1} - \frac{\Sigma^{-1} z z^T \Sigma^{-1}}{\beta n + z^T \Sigma^{-1} z})$ $n_i \leftarrow \beta n_i + 1, \quad n_j \leftarrow \beta n_j \text{ if } j \neq i, \quad n \leftarrow \beta n + 1$
--

### 3. 개념 변동 스트리밍 데이터에서의 차원 감소 방법의 적용

CCIPCA나 IPLS와 같은 점층적(incremental) 차원 감소 기법에서는 현재 모델을 새로 유입된 데이터 샘플을 이용해 업데이트함으로써 과거의 모든 데이터를 저장할 필요가 없게 된다. 그러나 개념 변동의 발생 가능성을 고려하지 않고 모델 업데이트를 수행하기 때문에 개념 변동 발생 시 차원 감소에 의한 전처리 과정이 오히려 이후의 분류 성능을 악화시킬 수도 있다.

반면에 분류 모델 업데이트 과정에서 개념 변동 발생 가능성 정도에 따라 현재 분류 모델의 가중치를 다르게 적용해 업데이트하는 적응적(adaptive) 학습은 고차원 스트리밍 데이터에서의 적용이 용이하지 않다. 적응적 분류기 ADMSE (Table 4)와 ADLDA (Table 5)도 차원 크기 정방향렬의 역행렬 계산이 필요하기 때문에 고차원 데이터에서 적용이 불가능하다.

본 논문에서는 개념 변동 고차원 스트리밍 데이터에서 적응적 분류 모델의 효과적인 적용을 위해 점층적 차원 감소를 전처리과정으로 수행하는 것을 제안한다. 특히 다음의 사항들에 대해 개념 변동이 발생할 때와 발생하지 않을 때를 비교해서 분석한다.

- 점층적 차원 감소 방법과 적응적 분류 모델 학습을 결합했을 때와 각각을 단독으로 사용할 때의 분류 성능 변화
- PCA와 LDA를 기반으로 하는 점층적 차원감소 방법들의 사용에 대한 적응적 분류기의 성능 비교

Fig. 2는 개념 변동 스트리밍 데이터에서 점층적 차원감소 방법과 적응적 분류기의 적용을 위한 프레임워크를 보여준다.  $x_1, \dots, x_n, \dots$ 을 입력되는 데이터 스트림이라고 하고,  $P_n : R^p \rightarrow R^q$ 을 데이터 샘플들의 스트림  $x_1, \dots, x_n$ 을 처리한 결과 얻어지는 차원감소를 위한 변환행렬이라고 하자, 변환되는 데이터 샘플은  $z_i = P_i^T x_i$ 에 의해서 얻어진다. 변환된 데이터 스트림  $z_1, \dots, z_n, \dots$ 에 대해 적응적 학습에 의한 분류기 업데이트를 수행한다.

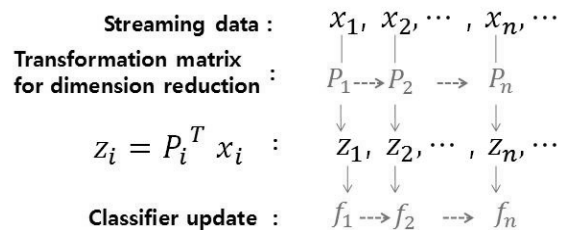


Fig. 2. The proposed framework for incremental dimension reduction and adaptive classifier

Table 6은 점층적 차원 감소를 수행 한 후에 적응적 분류 모델을 적용하는 알고리즘을 보여준다. 4절의 성능 비교를 위한 실험에서는  $\text{Update\_transformation\_matrix}()$  방법으로 PCA 기반 점층적 차원 감소 방법인 CCIPCA

(Table 1)과 LDA 기반 점층적 차원 감소 방법 IPLS (Table 2)를 사용한다.  $f \leftarrow Update(\beta, f, z_k)$  부분에서 사용되는 적응적 분류 기법으로는 ADMSE (Table 4)와 ADLDA (Table 5)를 사용한다.

Table 6. The application of adaptive learning after incremental dimension reduction

Input : $F = \{X(z_{n-t+1}), \dots, X(z_n)\}$ : records from recently received data samples $x_{n-t+1}, \dots, x_n$ and the transformation $z_i = P_i^T x_i$ $x_{n+1}, x_{n+2}, x_{n+3}, \dots$ : incoming data stream
Let $k = n + 1$ . While (incoming data sample $x_k$ is available) Update the transformation matrix by $P \leftarrow Update\_transformation\_matrix(P, x_k)$ $z_k = P^T x_k$ Compute $X(z_k) = \ C(z_k) - y_k\ _2^2$ Let $\mu$ and $\sigma$ be the mean and standard deviation of $F$ Compute $p\text{-value} = P(Z > \frac{X(z_k) - \mu}{\sigma})$ Compute $\beta$ using the function given in Fig.1 Update the classifier by the updating formula $f \leftarrow Update(\beta, f, z_k)$ Remove the oldest element and add $X(z_k)$ to $F$ . $k = k + 1$ . end of while

#### 4. 실험 결과

스트리밍 데이터에서의 개념 변동 설정을 위해 어느 시점에 유저의 관심 타겟이 변하게 되는 상황을 가정하였다[15, 16]. 초기에 한 개의 클래스를 포지티브 클래스로 두고 나머지는 모두 네거티브 클래스로 설정한다. 개념 변동의 설정을 위해 어느 정도의 시간이 지난 후에 네거티브 클래스에 속한 한 클래스를 포지티브 클래스로 전환하고 나머지는 모두 네거티브 클래스에 속하게 한다. 비교 방법들의 성능을 테스트하기 위해, 텍스트 데이터 20-newsgroup과 Reuters-21578, 숫자 이미지 데이터 MNIST를 이용하였다. 각 데이터에서 80%는 스트리밍 모드로 입력되는 학습 데이터로 사용하고 나머지 20%는 테스트 데이터로 성능 측정에 이용하였고 이러한 분할은 랜덤하게 10번 반복하여 평균값을 계산하였다.

20-newsgroup 데이터는 20개의 뉴스그룹으로부터 약 20000만개의 기사를 포함하고 있다[17]. 20news-bydate 버전을 stopwords removal, stemming, tf-idf transformation, unit norm을 가지도록 전처리한 후, 44,713개의 단어를 가진 18,774개의 다큐먼트 데이터를 구성한다. 20개의 뉴스그룹 클래스들은 클래스 7(misc.forsale)을 제외한 19개의 클래스들이 5개의 토픽 카테고리로 나누어져 있다. 초기에는 다섯 개의 카테고리 중에서 computer 카테고리(클래스 2, 3, 4, 5, 6)을 포지티브 타겟으로 설정하고 나머지는 네거티브 타겟

으로 둔다. 어느 시간이 경과 후 포지티브 타겟은 나머지 네 개의 카테고리 중의 하나로 바뀌게 된다. 2,500개의 데이터 샘플들로 구성되는 각 구간마다 포지티브 타겟은 computer, recreation(클래스 8, 9, 10, 11), science(클래스 12, 13, 14, 15), talk(클래스 17, 18, 19), religion(클래스 1, 16, 20)으로 차례대로 설정되었다.

텍스트 데이터 Reuters-21578[18]에서 135개의 TOPICS 카테고리에 속하는 다큐먼트들을 사용하였다. 20-newsgroup 데이터와 유사한 전처리 후에 15,484개의 용어로 구성되는 9805개의 다큐먼트를 가지게 된다. 135개의 카테고리 중에서 가장 많은 다큐먼트들을 가지는 1, 36, 46, 74번째 카테고리들이 차례로 1,500개 길이의 각 구간마다 포지티브 타겟으로 설정되었다. MNIST 데이터는 0부터 9까지 숫자를 손으로 쓴 70,000개의 숫자이미지들로 구성된 데이터이다[19]. 숫자이미지는 사이즈 28\*28의 그레이 레벨 이미지로써 각 이미지는 784차원의 벡터로 저장된다. 이 데이터에서는 3,000개 길이의 각 구간마다 숫자 1, 2, 3, 4, 5의 클래스를 차례대로 포지티브 타겟으로서 설정하였다.

학습데이터의 샘플들이 하나씩 들어올 때마다, 차원감소 행렬 업데이트와 분류기 업데이트가 일어나고, 테스트 집합을 이용하여 정확도(precision)와 재현률(recall)을 측정하였다. 정확도는 포지티브 클래스로 예측되는 데이터 샘플들이 실제로 포지티브 클래스에 속해 있는 비율이고, 재현률은 포지티브 클래스에 속하는 데이터 샘플들이 포지티브로 예측되는 비율이다. 정확도와 재현률은  $F1\text{-값} = 2 * \text{정확도} * \text{재현률} / (\text{정확도} + \text{재현률})$ 에 의해 통합된다. 두 개의 차원 감소 방법 CCIPCA, IPLS와 두 개의 적응적 분류 모델 ADMSE, ADLDA를 조합하여 구성된 4개의 방법 CCIPCA+ADMSE, IPLS+ADMSE, CCIPCA+ADLDA, IPLS+ADLDA에 대한 평균 F1-값을 비교하였다. 모든 실험데이터에서 CCIPCA와 IPLS에 의해 변형되는 데이터 차원을 30으로 설정하여 진행하였다. Fig. 3에서 20-newsgroup, Reuters-21578, MNIST 데이터에 대한 실험결과를 보여준다. 세 개의 데이터셋을 이용한 실험들에서 공통적으로 나타나는 현상은 개념 변동 발생 전인 첫 번째 구간에서는 차원감소를 위해 IPLS를 사용하는 것이 성능이 높지만, 개념 변동이 일어난 후에는 CCIPCA를 사용하는 것이 IPLS를 사용하는 것보다 더 효과적이라는 것이다. 이는 IPLS에서 컨셉 변화를 반영하지 않으면서 클래스 라벨 정보를 사용하는 것이 오히려 나쁜 영향을 줄 수 있다는 것을 의미한다.

다음 실험에서는 점층적 차원감소와 적응적 분류모델 학습을 결합하여 사용했을 때와 각각 단독으로 사용했을 때를 비교하였다. 먼저 점층적 차원 감소를 수행한 후에 개념 변동을 고려하지 않고 모델을 업데이트하는 분류 방법인 MSE[14]와 LDA[6]를 적용하여 성능을 측정하여 Fig. 4에 나타내었다. Fig. 4의 각 데이터에 대한 실험결과 그래프를 Fig. 3의 대응하는 그래프와 비교하면, 적응적 분류모델 학습이 개념 변동 발생 후 짧은 시간 안에 분류 성능을 회복

시킴을 알 수 있다. Fig. 5에서는 차원 감소를 하지 않고도 적응적 분류기를 적용할 수 있는 비교적 작은 데이터 차원을 가진 MNIST 데이터를 이용하여, 적응적 분류기 ADMSE와 ADLDA만 사용했을 경우의 성능을 CCIPCA+ADMSE와 CCIPCA+ADLDA의 성능과 비교하고 있다. 분류기만 사용했을 때보다 CCIPCA를 적용하여 차원 감소를 하는 것이 분류기의 성능을 향상시킴을 알 수 있다.

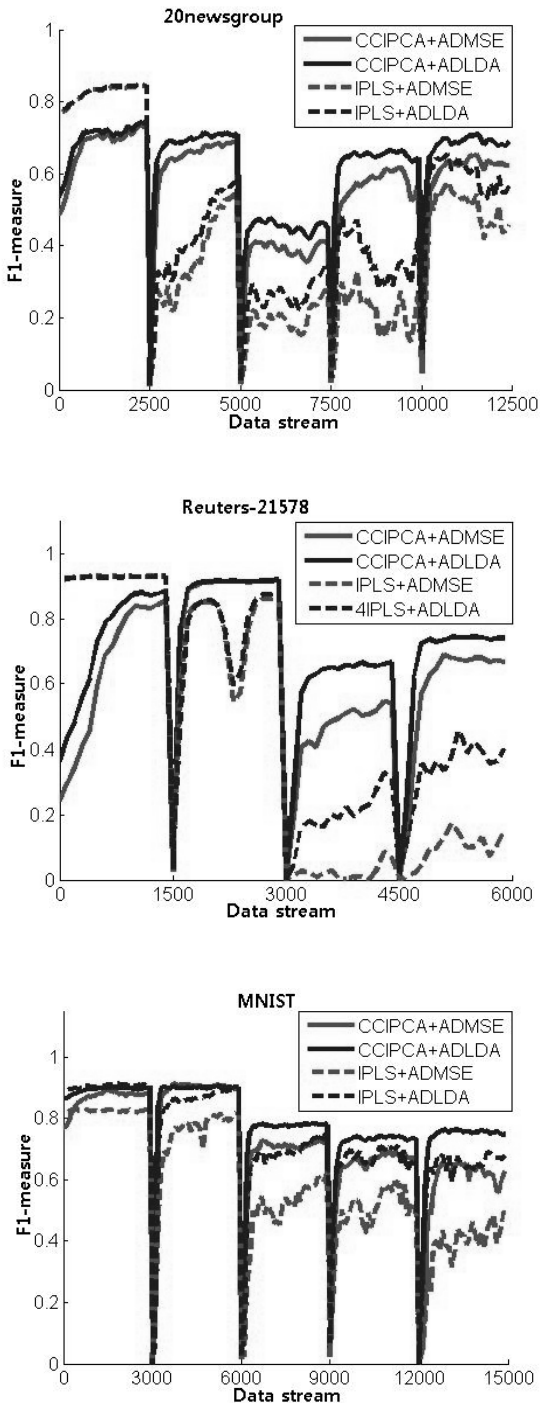


Fig. 3. Performance comparison in the combination of incremental dimension reduction and adaptive classifier learning

마지막으로 감소되는 차원의 크기가 분류 성능에 미치는 영향을 분석하였다. Fig. 6은 CCIPCA를 적용하여 감소되는 차원의 크기를 다르게 하면서 ADMSE를 사용한 결과를 보여준다. 차원을 10으로 크게 줄였을 때는 정확도가 많이 감소하나, 차원을 60에서 120 사이로 줄였을 때는 성능에 큰 차이가 없음을 알 수 있다.

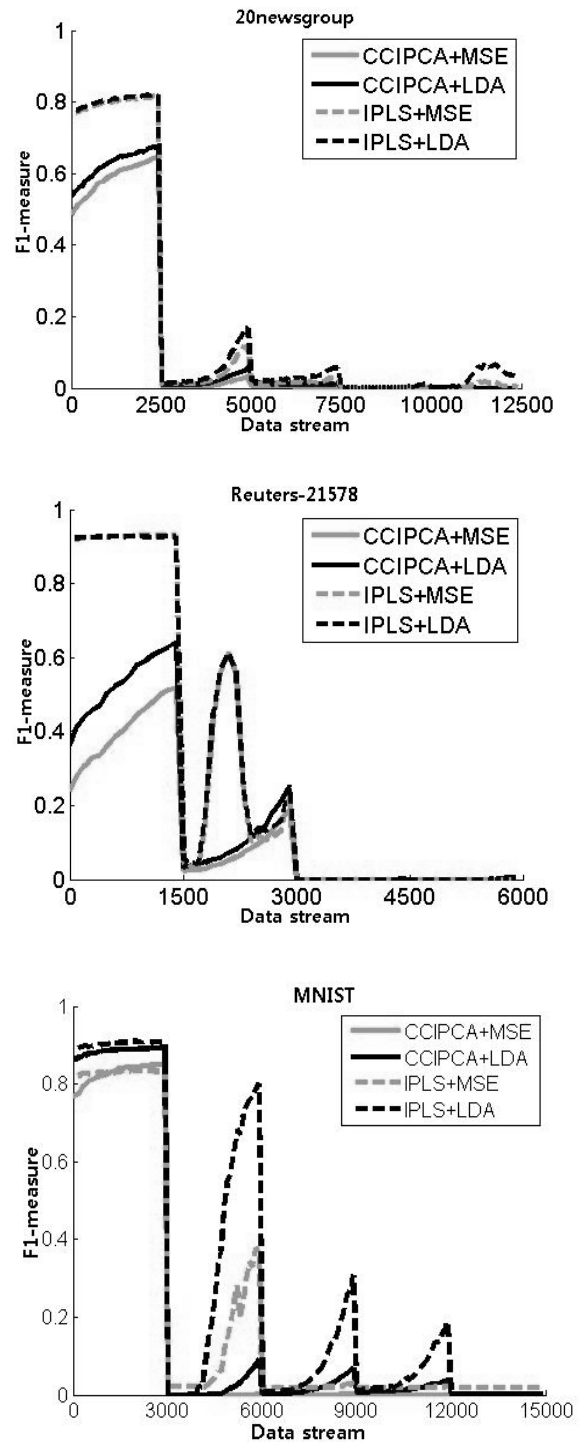


Fig. 4. Performance comparison in cases of not using adaptive classifier learning after incremental dimension reduction

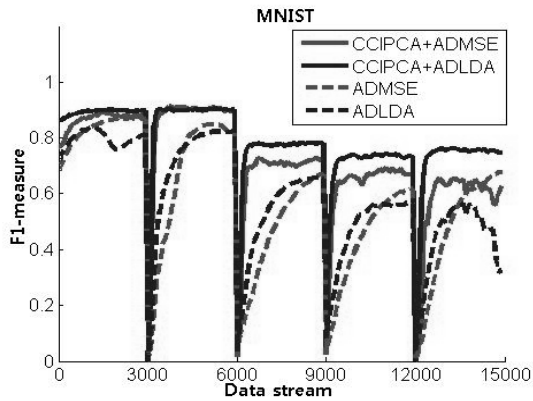


Fig. 5. Effects of dimension reduction on classification

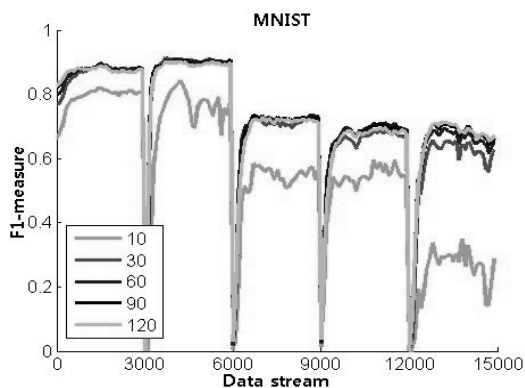
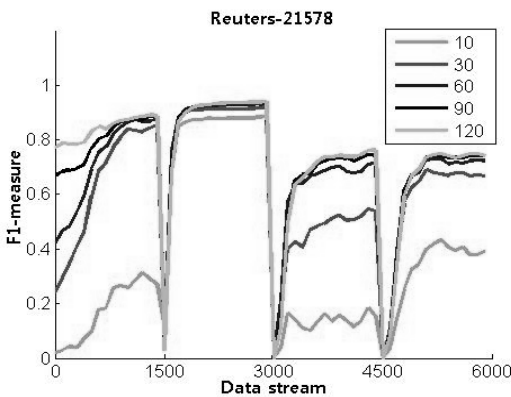
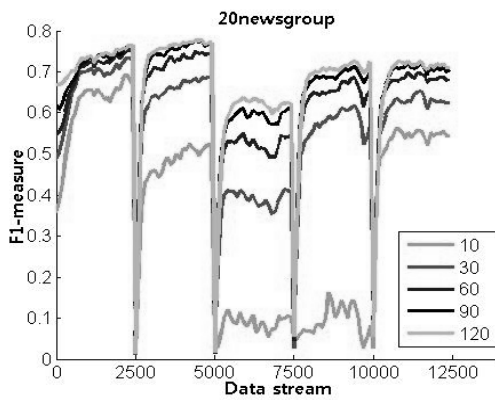


Fig. 6. Effects of reduced dimension size

## 5. 결론

본 논문에서는 개념 변동 가능성을 가진 고차원 스트리밍 데이터에 대해 분류성능을 향상시킬 수 있도록 차원 감소를 효과적으로 적용하는 방법을 제안하였다. 새로운 데이터 샘플이 유입될 때마다 차원 감소 변환 행렬을 업데이트하는 점층적 차원 감소 방법을 적용하여 데이터 스트림을 저차원으로 변환시킨 후 적응적 분류 모델 학습을 수행함으로써 고차원 스트리밍 데이터에서 개념 변동 발생에 효과적으로 대처할 수 있다.

고차원 텍스트 데이터와 이미지 데이터를 이용한 실험 결과를 다음과 같이 요약할 수 있다.

- 개념 변동이 없는 경우에는 LDA 기반 차원 감소 방법을 사용하는 것이 성능이 높았지만, 개념 변동이 일어난 후에는 오히려 PCA 기반 차원 감소가 더 효과적이었다. 개념 변동 상황을 반영하지 않으면서 클래스 라벨 정보를 사용하는 것이 오히려 나쁜 영향을 줄 수 있다는 것을 의미한다.
- 점층적 차원 감소 방법과 적응적 분류기의 결합은 각각을 단독으로 사용할 때에 비해 더 좋은 성능을 보여주었다.

특히 개념 변동에 대한 대처를 분류 모델 학습 단계로 미루지 않고 차원 감소 적용 단계에서 할 수 있는 방법을 개발한다면, 다양한 점층적 분류기법들의 적용이 용이하게 될 것이다.

## References

- [1] J. Gama, I. Zliobaite, A. Bifet, M. Pechennizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, Vol.46, No.4, pp.1-37, 2014.
- [2] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 2000.
- [3] H. Wang, W. Fan, P. Yu, and J. Han, "Mining concept-drifting data streams using ensemble classifiers," in *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 2003.
- [4] J. Z. Kolter and M. A. Malloof, "Dynamic weighted majority: an ensemble method for drifting concepts," *Journal of Machine Learning Research*, Vol.8, pp.2755-2790, 2007.
- [5] S. Ho and H. Wechsler, "A martingale framework for detecting changes in data streams by testing exchange ability," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.32, No.12, pp.2113-2127, 2010.
- [6] L. I. Kuncheva and C. O. Plumpton, "Adaptive learning rate for online linear discriminant classifiers," *LNCS 5342*, pp. 510-519, 2008.
- [7] C. H. Park, "An adaptive incremental classifier for data stream with concept drift," submitted for publication.

- [8] Juyang Weng, Yilu Zhang, and Wey-Shiuan Hwang, "Candid covariance-free incremental principal component analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.25, No.8, pp.1034-1040, 2003.
- [9] J. Yan, B. Zhang, S. Yan, N. Liu, Q. Yang, Q. Cheng, H. Li, Z. Chen, and W. Ma, "A scalable supervised algorithm for dimensionality reduction on streaming data," *Information Sciences*, Vol.17, No.6, pp.2042-2065, 2006.
- [10] X. Zeng and G. Li, "Incremental partial least squares analysis of big streaming data," *Pattern Recognition*, Vol.47, pp. 3726-3735, 2014.
- [11] J. Yan, B. Zhang, S. Yan, Q. Yang, and H. Li, "IMMC: Incremental Maximum Margin Criterion," in *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 2004.
- [12] Y. Ghassabeh and H. Moghaddam, "Adaptive linear discriminant analysis for online feature extraction," *Machine Vision and Applications*, Vol.24, pp.777-794, 2013.
- [13] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Proceedings of SBIA Brazilian Symposium on Artificial Intelligence*, 2004.
- [14] R. Duda, P. Hart, and D. Stork, "Pattern Classification," New York: Wiley-Interscience, 2001.
- [15] C. Lanquillon, "Information filtering in changing domains," in *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, 1999.
- [16] P. Lindstrom, B. Namee, and S. Delany, "Drift detection using uncertainty distribution divergence," *Evolving Systems*, Vol. 4, pp.13-25, 2013.
- [17] <http://people.csail.mit.edu/jrennie/20Newsgroups>.
- [18] UCI machine learning repository.
- [19] <http://yann.lecun.com/exdb/mnist>.



### 박정희

e-mail : cheonghee@cnu.ac.kr  
1998년 연세대학교 수학과(박사)  
2004년 University of Minnesota, Computer  
Science & Engineering(박사)  
2005년~현재 충남대학교 컴퓨터공학과  
교수  
관심분야 : Data Mining, 패턴인식