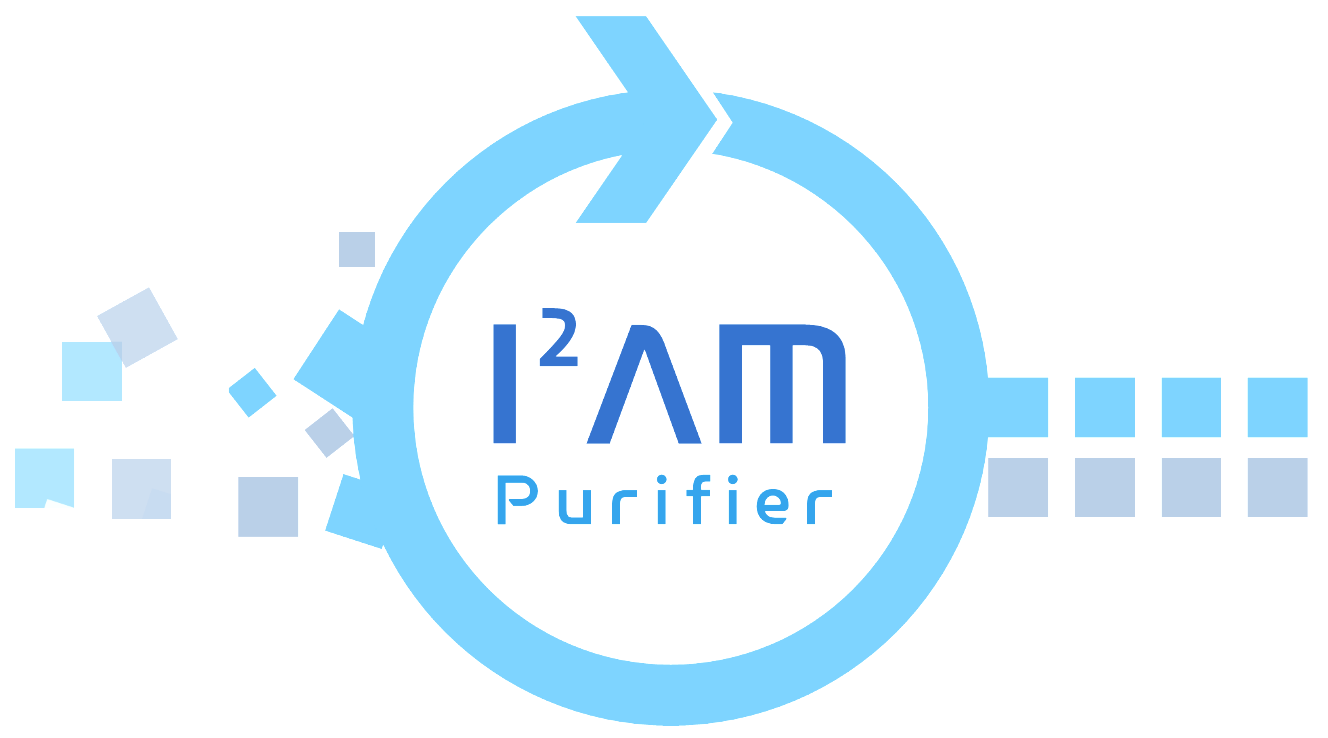
****

2016. 11.

연세대학교(원주캠퍼스), 강원대학교

데이터스트림 지능형 샘플링/필터링을 위한 Concept Drift 기술 동향 분석

문서 정보

|  |  |
| --- | --- |
| **버 전** | 0.4 |
| **작성일** | 2016-11-20 |
| **상 태** | 🞎 완료 🞎 진행 중 🗹 초안 |
| **작성자** | 이주원, 이형석, 김만수, 김대원, 임효상 (연세대학교 원주캠퍼스) |
| **검토자** | 임효상 (연세대학교 원주캠퍼스) |
| **승인자** |  |

**목 차**

[1. Concept Drift 개요 1](#_Toc467179743)

[1.1. 데이터 스트림 특성 및 스트림 데이터 분석의 한계 1](#_Toc467179744)

[1.2. concept drift 기본 개념[1] 1](#_Toc467179745)

[1.3. concept drift를 왜 사용해야 하는가 1](#_Toc467179746)

[2. Concept Drift 정의 2](#_Toc467179747)

[2.1. Informal한 정의[1,2] 2](#_Toc467179748)

[2.2. Formal한 정의[2] 2](#_Toc467179749)

[2.3. Concept drift 종류[1] 2](#_Toc467179750)

[3. 관련연구 동향 3](#_Toc467179751)

[3.1. Concept Drift를 measure 3](#_Toc467179752)

[3.1.1. Wilcoxon Test[3] 3](#_Toc467179753)

[3.1.2. Jensen-Shannon Divergence(JSD) 4](#_Toc467179754)

[3.1.3. Kolmogorov-Smirnov(K-S) Test[3] 5](#_Toc467179755)

[3.2. Detect하는 algorithm[4] 5](#_Toc467179756)

[3.2.1. Data Management 6](#_Toc467179757)

[3.2.2. Detection Methods 7](#_Toc467179758)

[3.2.3. Adaptation Methods 8](#_Toc467179759)

[4. Discussion 9](#_Toc467179760)

[5. 결론 10](#_Toc467179761)

[6. Reference 10](#_Toc467179762)

# Concept Drift 개요

## 데이터 스트림 특성 및 스트림 데이터 분석의 한계

데이터 스트림이란 시간이 경과함에 따라 지속적으로 생성되는 데이터 샘플들의 시퀀스를 말한다. 이러한 스트림 데이터를 분석하는 데 있어, 다음과 같은 문제들이 있다. 먼저 모든 생성되는 데이터 샘플들을 저장하여 작업하는 것은 불가능하다. 다음으로 동일한 데이터 청크를 여러 번 반복적으로 접근하여 작업하는 것도 바람직하지 않다. 마지막으로 시간이 지남에 따라 데이터 분포가 변할 수도 있고, 유저의 관심 분야가 달라지기도 하는 개념변화(Concept drift)가 발생할 수 있다.

## concept drift 기본 개념[1]

Concept drift란 지금까지 들어오던 것과 다른 개념(Concept)의 데이터가 지속적으로 들어옴으로써, 기존에 만들어진 classification model의 정확도가 저하되는 현상을 말한다. 여기서 개념은 데이터의 평균, 분포 등의 통계/스케치 정보로 표현될 수 있다.

## concept drift를 왜 사용해야 하는가

본 과제에서는 실시간으로 빠르게 입력되는 데이터에 대한 처리(sampling/filtering)을 위해 지능형 엔진을 개발한다. 그러므로 현재 입력되는 데이터 스트림의 상태를 classification model로 모델링하고, 이에 입각하여 sampling/filtering을 정확히 수행하고자 한다. 그러나 concept drift가 발생하면 sampling/filtering 정확도에 영향을 주게 되므로, 이에 적응하여 classification model을 업데이트 하는 것이 필요하다.

# Concept Drift 정의

## Informal한 정의[1,2]

Concept drift란 지금까지 들어오던 것과 다른 개념(Concept)의 데이터가 지속적으로 들어옴으로써, 기존에 만들어진 classification model의 정확도가 저하되는 현상을 말한다. 여기서 개념은 데이터의 평균, 분포 등의 통계/스케치 정보로 표현될 수 있다.

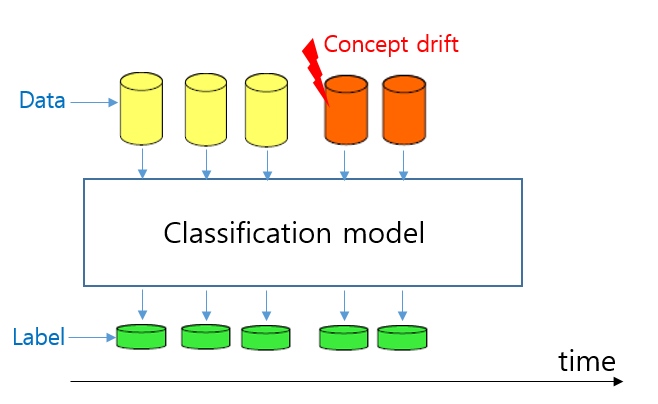


Figure Concept Drift

## Formal한 정의[2]

독립적인 확률 분포 로부터 생성된 데이터 들의 시퀀스 가 존재할 때, 이면 concept drift가 발생한 것이고, i+1 시점을 concept drift가 발생한 시점이라 할 수 있다.

## Concept drift 종류[1]

1. Sudden drift: 어느 한 시점에서 concept가 갑자기 변하는 경우를 말한다.
2. Gradual drift: 일정 기간동안 두 가지 concept이 반복해서 나오다가 시간이 지남에 따라 자주 나오는 concept으로 변하는 경우를 말한다.
3. Incremental drift: concept가 긴 시간동안 서서히 변경되는 경우를 말한다.
4. Reoccurring context: 이전에 활동했던 concept이 시간이 지남에 따라 다시 등장한 것을 의미한다.

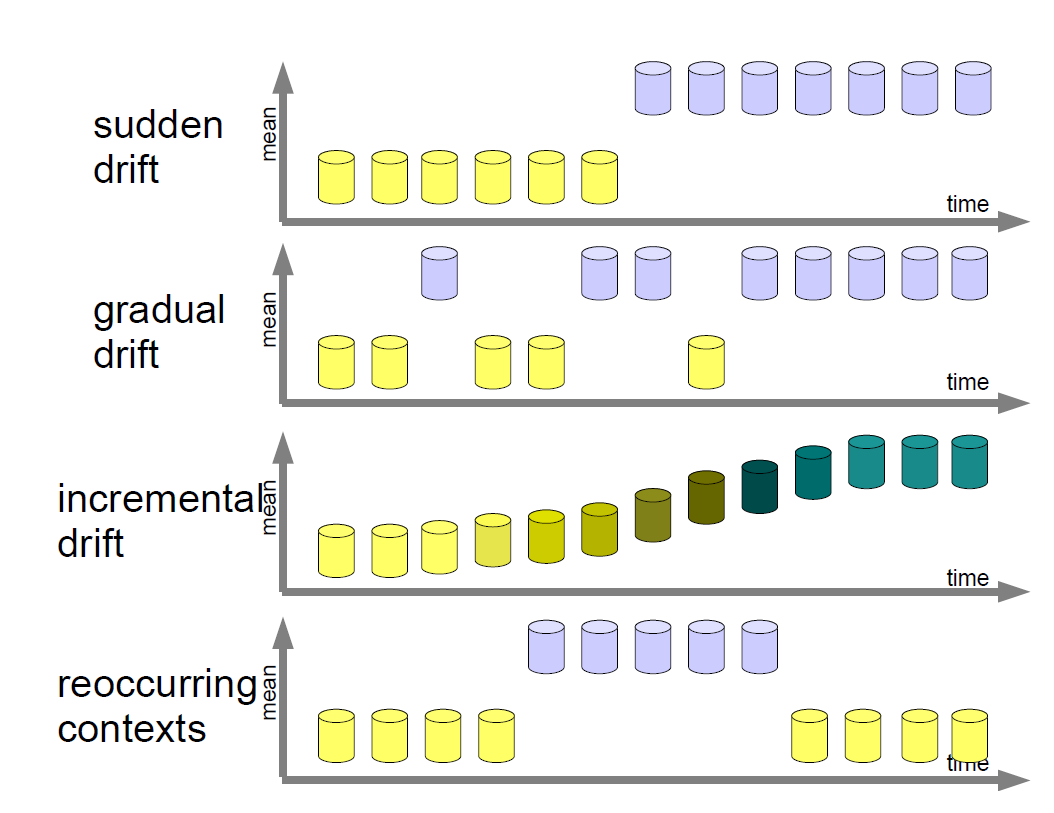


Figure Type of Concept Drift

# 관련연구 동향

## Concept Drift를 measure

concept drift는 독립적인 확률 분포 로부터 생성된 데이터 들의 시퀀스 가 존재할 때, 이면 concept drift가 발생한 것이고, i+1 시점을 concept가 변경된 시점이다. 즉, 분포의 변화를 concept drift라 할 수 있다. 분포의 변화를 탐지, 혹은 두 분포가 유사한지 아닌지를 판단하는 방법은 다음과 같다.

### Wilcoxon Test [3]

비모수 통계(non-parametric statistics) 방법 중 하나로 같은 모집단(population)에서 추출된 표본(sample)의 전후를 비교를 하는 방법이다. 여기서 비모수 통계란 모집단(population)의 형태에 관계없이 주어진 데이터에서 직접 확률을 계산하여 통계학적 검정을 하는 분석법을 말한다. 그리고 비모수 통계 방법은 표본이 정규분포를 따르지 않고, 표본의 크기가 작을 때 사용되는 방법이다.

방법은 다음과 같다. 먼저 추출된 표본의 전(), 후()의 각 자료 차이()를 구한다. 그리고 각 자료의 차이 가 ‘-‘부호인 rank들의 합()과, 각 자료의 차이가 가 ‘+’부호인 rank들의 합()을 구한다. 여기서 rank는 의 절대값 오름차순으로 순위를 나타낸 것을 의미한다. 결과적으로 rank들의 합인 와 의 차이가 적을수록 추출된 표본이 비슷하다고 판단하게 된다. 이때 절대값 합을 쓰지 않고 rank 합을 쓰는 이유는 특정 의 절대값이 너무 커서 또는 값에 너무 큰 영향을 주는 것을 방지하기 위함이다.

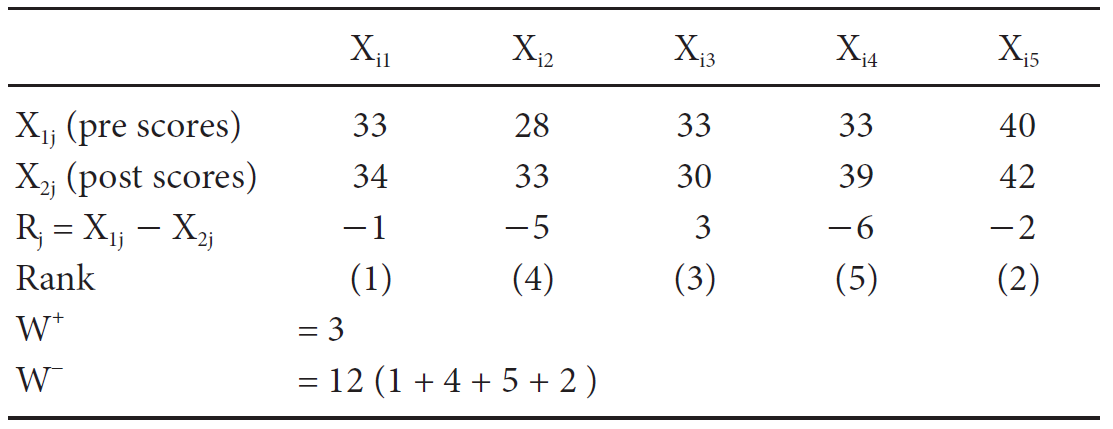


Figure Wilcoxon Test

.

### Jensen-Shannon Divergence(JSD)

Divergence란 확률분포간의 차이를 측정하는 방법을 말한다. Jensen-Shannon Divergence(JSD)는 Kullback-Leibler Divergence(KLD)를 응용한 방법으로, KLD에서 발생하는 문제점을 보완한 방법이다. KLD란 두 확률분포 간의 차이를 측정하는 가장 대표적인 방법으로, 아래의 수식과 같이 표현된다.

이때, 값은 에 대한 의 비대칭 값을 의미하게 된다. 즉, 두 확률분포의 계산 순서에 따라서 결과가 달라지는 문제가 발생하게 된다. 이와 같은 문제점을 해결하기 위해 JSD가 제안되었고, 수식은 다음과 같다.

JSD의 방법은 다음과 같다. 먼저 두 확률분포 의 개별 지점마다 두 확률분포의 값을 평균한 값으로 이루어진 평균 분포()를 구한다. 그리고 평균 분포와 각각의 두 확률분포 사이의 KLD, , 를 구한다. 마지막으로 이 두 값의 평균이 JSD가 된다.

### Kolmogorov-Smirnov(K-S) Test [3]

KSD는 비모수 통계 방법 중 하나로, 독립된 두 표본(sample)의 누적 분포 함수(CDF)가 동일한지 알아봄으로써 두 표본이 동일한 분포를 가졌는지, 혹은 동일 모집단으로부터 추출되었는지 검정하는 방법이다. 방법은 다음과 같다.

1. 두 표본 X, Y를 통틀어 가장 큰 값 max와 가장 작은 값 min을 구한다.
2. 구한 max와 min의 사이를 일정한 크기의 구간(interval)으로 나눈다.
3. 두 표본의 각 구간에서의 누적 확률 , 를 구한다.
4. 두 표본의 각 구간에서의 누적 확률 간의 편차()를 구한다.
5. 구한 편차들 중 가장 큰 값(Max difference)을 찾는다.
6. Max difference를 임계 값과 비교하여 두 표본의 동질성 여부를 판단한다.

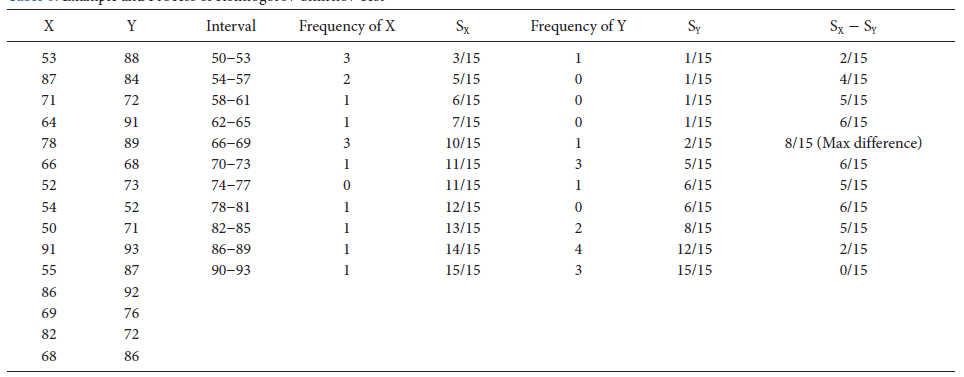


Figure Kolmogorov-Smirnov(K-S) Test

## Detect하는 algorithm[4]

Concept drift란 수집된 data의 concept이 시간이 지남에 따라 shift되는 것을 의미한다. 보통 과거에 training에 쓰였던 data는 현재 상태와는 무관한 결과값을 나타낸다. 그렇기 때문에 오래된 concept이 다른 training data는 지워야 한다. 이 때, concept drift detection 방법을 이용하여 불필요한 training data를 찾을 수 있다. concept drift detection algorithm은 다음과 같은 방법들이 제안되었다.

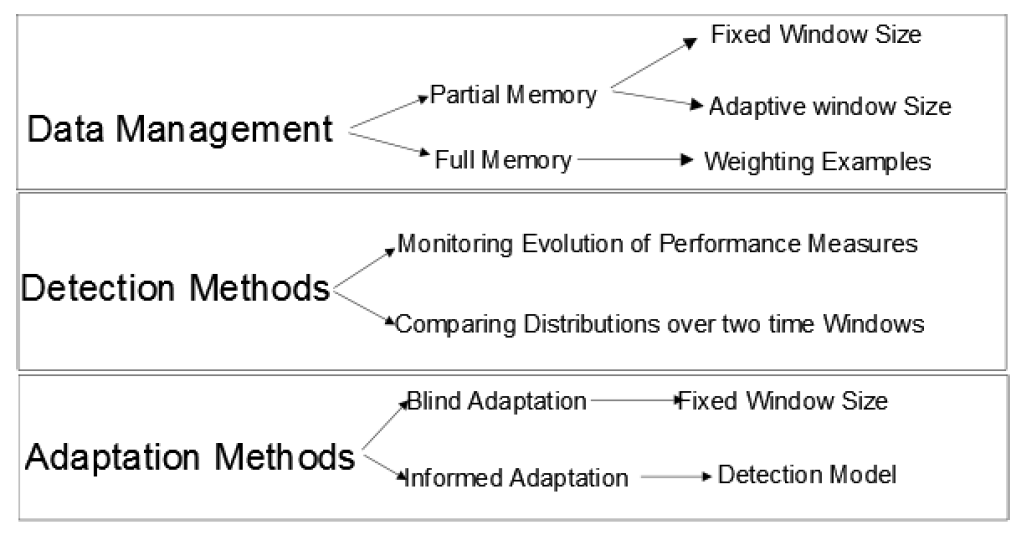


Figure Type of Detection Algorithm

### Data Management

일반적으로 decision model의 일관된 성능을 유지하기 위해 data management 방법을 사용한다. 먼저 Full memory 방법은 모든 example들을 memory에 저장하는 방법으로, example들은 각각의 age에 따라 가중치가 다르게 주어진다. 이때 오래된 example일수록 가중치가 낮아진다. 그리고 가중치를 linear 혹은 exponential하게 주는 방법으로 나뉘어진다. 다음으로 partial memory 방법은 mops recent examples만을 memory에 저장하는 방법으로, 윈도우 기반(FIFO 구조)으로 저장된다. 즉, 각 time step마다 learner는 window에 포함된 example만을 이용한다. 그러나 가장 큰 문제는 window의 적절한 size를 정하는 것이다. 만약 작은 window size를 이용하게 되면 최신 분포를 정확하게 반영할 수 있고 change에 대해 빠르게 적응할 수 있지만, concept이 안정된 example이 들어오고 있는 상황에서는 performance가 떨어지게 된다. 반면 큰 window size를 이용할 경우, concept이 안정된 example이 들어오고 있는 상황에서 performance가 좋지만, concept change에 대한 적응은 빠르게 반영할 수 없다. 그리고 윈도우는 두 가지 방법으로 구성할 수 있다. 먼저 하나가 들어오면 하나가 제거되는 형태의 fixed sized window가 있고, 다음으로 window size가 가변적으로 변경되는 adaptive size window가 있다.

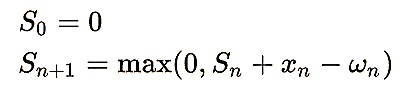
### Detection Methods

Detection method는 decision model의 일관된 성능을 유지하기 위해 change를 detect하는 방법을 사용한다. 즉, change point 혹은 change가 발생한 window를 제시하고, 현재 상황에 맞는 데이터를 이용하여 model을 구축한다. 또한 change를 detection하는데 중요한 점은 noise와 change를 구분할 수 있어야 한다는 것이다. 즉 change detection 알고리즘은 noise에 대해선 robust해야 하고, change에 대해선 sensitive해야 한다.

Detection method는 크게 두가지 방법이 존재한다. 먼저 decision model의 성능을 계속해서 살펴보며 성능이 안 좋아지면 drift가 일어났다고 판단하는 방법인 “monitoring the evolution of performance indicators”가 있다. 해당 알고리즘으로는 CUSUM, DDM, EDDM 등이 있다. 다음으로 두 개의 window내에 example 분포를 비교하여 차이가 나면 drift가 일어났다 판단하는 “monitoring distributions on two different time-windows” 방법이 있다. 그 예로는 window를 사용한 concept drift 탐지 알고리즘이 있다.

#### The Cumulative Sum Algorithm(CUSUM) [4]

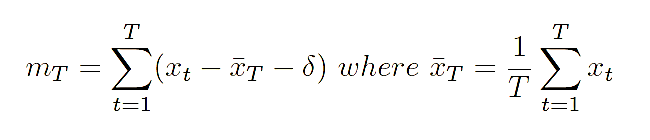
이 방법은 클래식한 change detection 알고리즘으로, example의 mean이 0과 significantly different할 때 alarm이 발생하게 된다. 이를 수식으로 나타내면 다음과 같다.



위 수식에서 은 현재 들어온 값을 의미하고, 은 likelihood function, 예측 값을 의미한다. 여기서 값이 임계 값 를 넘기면 alarm을 주고 이 된다. 이때 이 작다는 것은 빠르게 탐지는 가능하지만, false alarms이 증가함을 의미한다. 그러나 이 수식은 positive direction의 change들만 detect가 가능하다. 그렇기 때문에 negative direction의 change를 detect 하기 위해서는 max가 아닌 min을 사용해야 한다. 그리고 CUSUM algorithm은 memoryless하다는 특징이 있다.

#### Page-Hinkley(PH) Test

PH test는 신호 처리에서 사용되는 change detection 방법으로, 가우시안 신호의 평균에서 abrupt한 change를 찾아내는데 사용된다. 사용되는 수식은 다음과 같다.







여기서 는 time T까지의 입력 값 평균을 나타내고, 는 의 최소값을 나타낸다. 그리고 는 최소값과 현재 입력 값의 차이를 나타낸다. 최종적으로 가 임계 값보다 크면 drift가 일어났다고 판단한다. 이때 임계 값로 비교적 큰 값을 사용한다면, false alarm은 줄지만 change를 못 찾는 경우가 늘어나게 된다. 아래 그림은 PH test가 change의 velocity를 반영함을 보여준다. 아래 그림은 좌측부터 learning algorithm의 online error rates, accumulated online errors, result of PH test를 나타낸다.

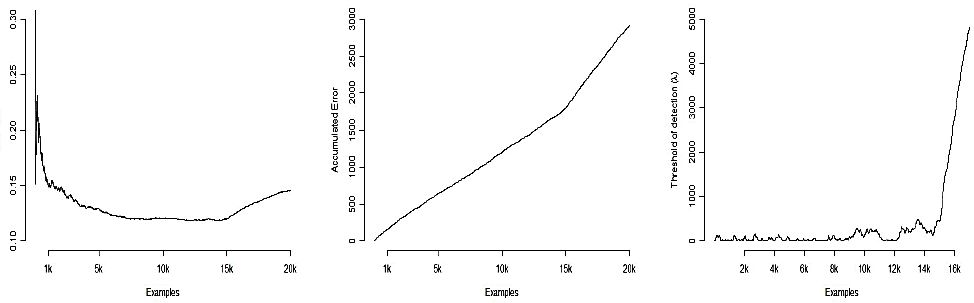


Figure PH Test

#### Drift Detection Method(DDM) [5], Early Drift Detection Method(EDDM) [6]

Drift Detection Method(DDM)은 오류율을 concept drift 발상의 시그널로 이용하는 방법이다. 오류율은 스트리밍 데이터에서 개념 변화 탐지를 위해 많이 사용되는 척도로, 모델(classification)의 데이터 샘플에 대한 예측이 맞을 경우 0, 그렇지 않은 경우 0으로 설정되는 이진 값으로 나타내어 베르누이 분포의 확률 변수로 표현한다. 이러한 오류율을 concept drift 발생의 시그널로 이용하게 된다.

DDM은 다음과 같은 순서로 수행된다. 먼저 모델을 통해 나온 training dataset 내에 각 data의 label 오류를 측정한다. 그리고 측정된 오류 수를 이용하여 이항분포를 만든다. 마지막으로 들어오는 training dataset에 의해서 만들어진 이항분포가 일정 수준으로 변화를 보인다면 concept drift가 발생했음을 알린다.

Early Drift Detection Method(EDDM)은 기존의 DDM을 보완한 방법이다. DDM에서는 오류의 횟수를 이용하여 concept drift 탐색을 수행하였지만, EDDM은 오류의 횟수가 아닌 오류들 사이의 거리를 이용하였다. EDDM은 오류들 사이의 거리를 이용함으로써 DDM보다 gradual drift에 대한 탐지 성능을 향상 시켰다.

#### Window를 사용한 concept drift 탐지 알고리즘[2]

데이터스트림 처리에 있어, 모든 history를 memory에 저장하는 것은 비현실적이다. 그렇기에 두 개의 윈도우를 사용하여 변화를 탐지하는 알고리즘을 제안하였다. 사용되는 윈도우는 다음과 같다. 먼저 reference window는 기존 데이터에 대한 윈도우이고, current window는 새로 들어오는 데이터를 따라 슬라이딩 하는 윈도우를 의미한다. 그리고 reference window는 concept drift가 발생 된 경우 current window의 위치로 업데이트하여 새로운 concept drift 탐지를 수행하게 된다. 또한 각 데이터 분포 사이의 거리를 측정하는 새로운 방법을 제안하여, concept drift 탐지의 정확도를 보장하면서 분석적, 수치적 성능을 제공하였다. 두 분포에 대한 차이를 계산하는 수식은 다음과 같다.

이 수식은 확률론적 관점에서 total variation을 사용하여 두 분포 사이의 차이를 구하는 방법으로, 두 분포 사이에서의 전체적인 값의 차이에 대한 상한(최대값)이 어느 정도 되는지 계산하는 방법이다. 이때, 는 collection of measurable sets를 의미한다. 그러나 이 수식은 전체적인 분포, 즉 많은 샘플이 필요하기 때문에 window를 이용하여 적은 수의 샘플로부터 분포의 변화를 측정하는 방법이 필요하므로 다음과 같은 수식으로 다시 표현하였다.

이 수식은 통계학적인 측면에서 제한된 샘플이 전체적인 분포를 대표할 수 있도록 보장한다. 그렇기 때문에 제한된 샘플을 이용하여 전체 분포의 차이를 측정할 수 있다. 이때 는 finite domain subset을 나타내고, S(A)는 를 나타내며 이는 subset 에서 measurable subset 가 차지하는 비중을 나타낸다. 그리고 제안한 방법은 기존 방법보다 모든 상황에서 더 좋다고 할 수는 없지만, 특정 상황에서는 가장 우수하다는 것을 실험을 통해 보였다.

### Adaptation Methods

Decision model이 adaptation되는 두가지 방법은 다음과 같다. 먼저 blind method는 concept drift와 상관없이 일정한 시간 간격으로 model을 adapt하는 방법으로, weighted examples and fixed size time window 등이 있다. 다음으로 informed method는 concept drift가 일어난 경우에만 model을 adapt한다.

# Discussion

본 과제에서는 실시간으로 빠르게 입력되는 데이터에 대한 처리(sampling/filtering)을 위해 지능형 엔진을 개발한다. 이를 그림으로 나타내면 다음과 같다.

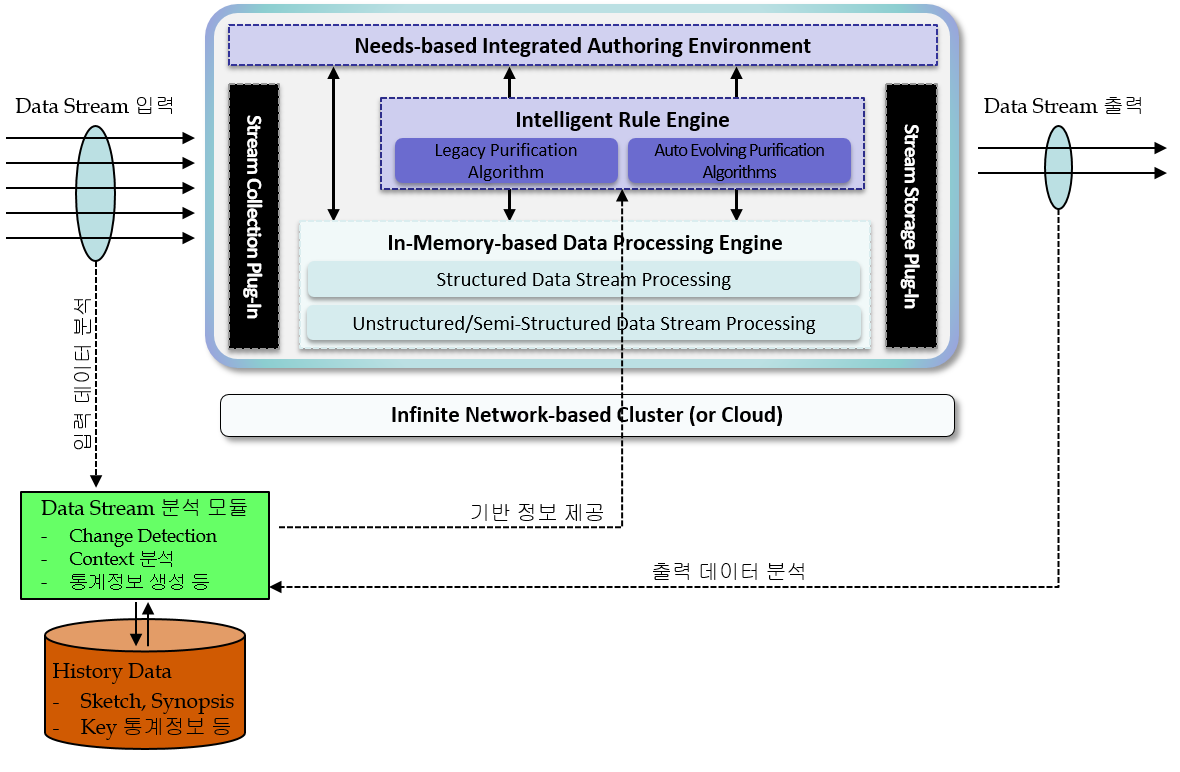


Figure 지능형 엔진 개요

본 과제는 그림과 같이 현재 입력되는 데이터 스트림의 상태를 분석하고, 이에 입각하여 sampling/filtering과 같은 데이터 처리를 지능적으로 상황에 맞게 수행하고자 한다. 이때 data stream을 분석하는 방법으로 change detection, context 분석, 통계정보 생성 등을 이용한다. 또한 과거 데이터에 대한 간략화 한 정보를 저장해 놓고 이 또한 데이터 스트림 분석에 이용할 수 있게 구성한다. 이렇게 분석한 기반 정보를 지은형 엔진에 제공함으로써 지능적으로 상황에 맞는 데이터 처리 방법을 사용할 수 있다. 마지막으로 출력된 데이터 스트림을 다시 분석하여, 이 또한 데이터 스트림 분석에 이용한다.

이와 같이 본 과제에서 입력되는 데이터의 상태를 판단 할 수 있는 concept drift 개념 및 detection method는 큰 비중을 차지함을 알 수 있다.

# 결론

본 과제에서는 실시간으로 빠르게 입력되는 데이터에 대한 처리(sampling/filtering)을 위해 지능형 엔진을 개발한다. 지능형 엔진 개발에 있어 입력되는 데이터의 상태를 분석 및 판단 할 수 있는 concept drift 개념은 중요한 비중을 차지함을 알 수 있다.

# Reference

1. Indr˙e Zliobait˙e, “Learning under Concept Drift: an Overview”, Technical report, Vilnius University, Faculty of Mathematics and Informatics, 2009.
2. Kifer, Daniel, Shai Ben-David, and Johannes Gehrke. "Detecting change in data streams."  *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 2004.
3. Nahm, Francis Sahngun. "Nonparametric statistical tests for the continuous data: the basic concept and the practical use." Korean journal of anesthesiology 69.1 (2016): 8-14
4. Chapman and Hall/CRC; 1 edition (May 25, 2010), Knowledge Discovery from Data Streams (Chapter 3)
5. Gama, Joao, et al. "Learning with drift detection." Advances in artificial intelligence–SBIA 2004. Springer Berlin Heidelberg, 2004. 286-295.
6. Baena-Garcıa, Manuel, et al. "Early drift detection method." Fourth international workshop on knowledge discovery from data streams. Vol. 6. 2006.