슬라이딩 윈도우 기반 다변량 스트림 데이타 분류 기법

(A Sliding Window-based Multivariate Stream Data Classification)

서 성 보 † 강 재 우 †† 남 광 우 ††† 류 근 호 ††††
(Sungbo Seo) (Jaewoo Kang) (Kwang Woo Nam) (Keun Ho Ryu)

요 약 분산 센서 네트워크에서 대용량 스트림 데이타를 제한된 네트워크, 전력, 프로세서를 이용하여 모든 센서 데이타를 전송하고 분석하는 것은 어렵고 바람직하지 않다. 그러므로 연속적으로 입력되는 데이타를 사전에 분류하여 특성에 따라 선택적으로 데이타를 처리하는 데이타 분류 기법이 요구된다. 이 논문에서는 다차원 센서에서 주기적으로 수집되는 스트림 데이타를 슬라이딩 윈도우 단위로 데이타를 분류하는 기법을 제안한다. 제안된 기법은 전처리 단계와 분류단계로 구성된다. 전처리 단계는 다변량 스트림 데이타를 포함한 각 슬라이딩 윈도우 입력에 대해 데이타의 변화 특성에 따라 문자 기호를 이용하여 다양한 이산적 문자열 데이타 집합으로 변환한다. 분류단계는 각 윈도우마다 생성된 이산적 문자열 데이타를 분류하기 위해 표준 문서 분류 알고리즘을 이용하였다. 실험을 위해 우리는 Supervised 학습(베이지안 분류기, SVM)과 Unsupervised 학습(Jaccard, TFIDF, Jaro, Jaro Winkler) 알고리즘을 비교하고 평가하였다. 실험결과 SVM과 TFIDF 기법이 우수한 결과를 보였으며, 특히 속성간의 상관 정도와 인접한 각 문자 기호를 연결한 n-gram방식을 함께 고려하였을 때 높은 정확도를 보였다.

키워드 : 스트림 데이타, 데이타 분류, 센서 데이타 처리

Abstract In distributed wireless sensor network, it is difficult to transmit and analyze the entire stream data depending on limited networks, power and processor. Therefore it is suitable to use alternative stream data processing after classifying the continuous stream data. We propose a classification framework for continuous multivariate stream data. The proposed approach works in two steps. In the preprocessing step, it takes input as a sliding window of multivariate stream data and discretizes the data in the window into a string of symbols that characterize the signal changes. In the classification step, it uses a standard text classification algorithm to classify the discretized data in the window. We evaluated both supervised and unsupervised classification algorithms. For supervised, we tested Bayesian classifier and SVM, and for unsupervised, we tested Jaccard, TFIDF, Jaro and JaroWinkler. In our experiments, SVM and TFIDF outperformed other classification methods. In particular, we observed that classification accuracy is improved when the correlation of attributes is also considered along with the n-gram tokens of symbols.

Key words : Stream Data, Data Classification, Sensor Data Processing

· 본 연구는 정보통신부 대학 IT연구센터 육성·지원사업의 연구 결과로 수행되었습니다.

† 학생회원 : 충북대학교 전자계산학과
sbseo@dblab.chungbuk.ac.kr
†† 정 회 원 : 미국 노스캐롤라이나 주립대 전자계산학과 교수
kang@csc.ncsu.edu
††† 정 회 원 : 군산대학교 컴퓨터정보과학과 교수
kwnam@kunsan.ac.kr
†††† 종신회원 : 충북대학교 전자계산학과 교수
khryu@dblab.chungbuk.ac.kr

논문접수 : 2005년 10월 10일
심사완료 : 2006년 1월 26일

1. 서 론

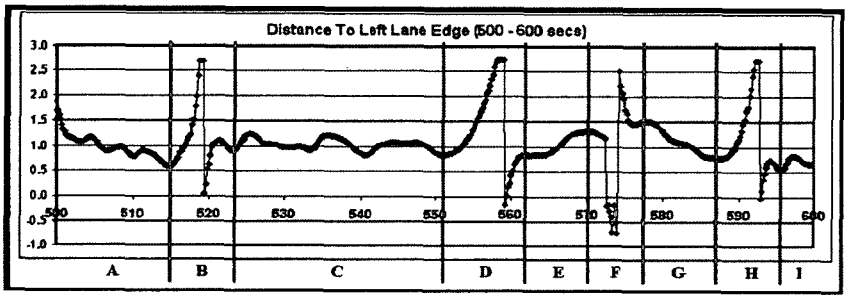
최근 센서, 무선기기와 네트워크의 발달로 인해 다차원 센서를 가진 센서 노드에서 수집된 스트림 데이타가 센서 네트워크의 상위 노드에 전송되어 저장되거나 분석된다. 일반적으로 센서 네트워크의 노드는 제한된 통신 대역폭, 저전력, 소형 프로세스의 특성이 있으므로 모든 데이타를 전송하거나 완전한 데이타 분석에 많은 비용이 소모된다[1,2].

센서 네트워크의 응용분야는 크게 환경 모니터링

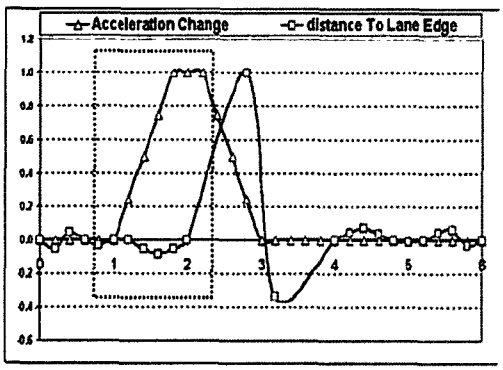
(environmental monitoring), 객체 추적(object tracking), 객체 감시(object guarding)으로 분류된다[1-3]. 환경 모니터링은 홍수 감시, 홈 애플리케이션, 동식물/생태 감시 분야이며, 저주파 데이타 특성을 가지며 장기적인 데이타 분석이 가능하다. 객체 추적은 이동 객체 추적, 군사 분야, 물류 분야이며, 고주파 데이타 특성과 다차원 객체 속성을 가진다. 객체 감시는 응급의료 관리, 침입탐지와 지진 탐사 분야이며, 비정상적인 변화와 같은 이상치의 특성과 실시간 처리의 특성이 있다. 이와 같이 다양한 센서 네트워크 응용분야는 데이타 처리를 위해 다양한 요구사항과 다양한 데이타 타입을 가진다. 예를 들어 모니터링 애플리케이션에서 입력되는 데이타가 정상이라면 사용자는 원래 데이타의 단지 5%만 받길 원하지만 비정상적인(침입탐지, 화재 등) 패턴이 인식되면 원래 데이타를 모두 받아 분석하길 원할 것이다. 따라서 데이타 중심의 센서 응용에서는 시간에 따라 변화하는 스트림 데이타를 선택적으로 처리하기 위해 데이타 특성에 기반한 데이타 분류기법이 적용되어야 한다.

그림 1(a)는 [4]에서 차선의 왼쪽에서 차량의 중심까지의 거리를 측정한 데이타이다. 이 그림에서 B, D, F, H 세그먼트는 차량이 현재 운행 중인 차선에서 다른 차선으로 이동한 상태를 나타낸다. 이와 같이 일정한 윈도우 크기마다 스트림 데이타의 입력 값에 변화를 관찰하여 데이타 특성 분류가 가능하다. 그림 1(b)에서 차선을 변경하는 센서의 값이 변화할 때 가속기 센서 값이 함께 증가하는 상관 관계를 가지고 변화하는 특성이 있다. 실 세계의 센서 노드가 다양한 센서를 포함할 때 그림 1(c)와 같이 일정한 주기로 데이타를 처리한다면 기존의 튜플(tuple) 기반 데이타 분류 기법에 윈도우 단위 데이타를 어떻게 적용할 것인가?

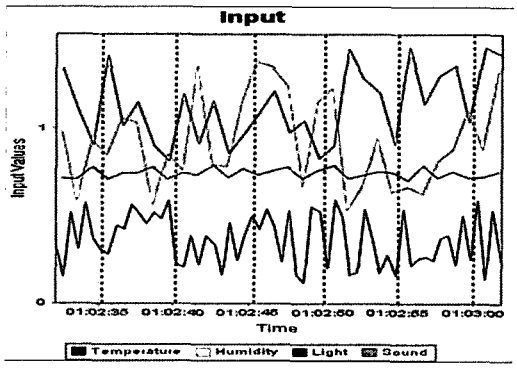
전통적인 데이타 분류 기법 알고리즘인 의사결정트리, 베이지안 분류, 신경망 등은 학습 데이타의 형태와 클래스가 튜플 기반이다[5]. 또한 대용량 데이타베이스에서 반복적인 데이타 접근을 통해 지식을 추출한다. 하지만 센서 네트워크에서 데이타의 전송 비용은 데이타 처리를 위해 CPU 프로세싱에 소모되는 에너지량 보다 수십 배 이상 크기 때문에 매번 데이타를 전송하는 것 보다 일정한 주기로 데이타를 전송하는 방식을 가진다[3,6]. 따라서 다변량 스트림 데이타는 끊임없이 입력되는 데이타의 도착시간을 기준으로 연속적인 윈도우 단위로 분할된다. 그러므로 스트림 데이타는 튜플 기반이 아닌 슬라이딩 윈도우 기반으로 데이타를 분류해야 하며, 빠르게 변화하는 데이타에 대해 제한된 메모리와 프로세스 환경에서 원 패스로 처리되어야 하는 제약성이 있다.



(a) 차량의 운행에 대한 스트림 세그먼트 데이타[4]



(b) 속성간의 연관성[4]



(c) 윈도우 기반 다변량 스트림 데이타

그림 1 윈도우 기반 다변량 스트림 데이타 예

그림 1(c)와 같이 연속적인 다변량 스트림 데이타 특성에 따라 선택적으로 데이타를 처리하기 위해, 이 논문에서는 하나의 윈도우를 하나의 문서로 간주하고 스트림 데이타를 문자열 집합으로 변환하여 표준 문서 분류 기법을 적용하여 다변량 스트림 데이타를 분류한다. 제안된 기법은 슬라이딩 윈도우 단위의 전처리 과정과 분류 단계로 크게 구성된다. 전처리 과정에서는 입력되는 스트림 데이타를 일정한 크기로 분할하고 데이타의 증감변화 정도를 고려하여 지정된 문자 기호로 할당한다. 변환된 이산적 문자열을 계층형 부분 선형 표현(HPL: Hierarchical Piecewise Linear representation)[7]과 n-gram[8]을 이용하여 하나의 슬라이딩 윈도우에 다양한 문자열 집합을 생성한다. 분류단계에서는 하나의 슬라이딩 윈도우를 하나의 문서로 간주하여 생성된 문자열의 집합을 기존의 표준 문서 분류 알고리즘에 적용하여 분류하였다.

분류 모델 실험을 위해 우리는 Supervised 학습(베이지안 분류기, SVM)과 Unsupervised 학습(Jaccard, TFIDF, Jaro, Jaro Winkler) 알고리즘을 비교하고 평가하였다. 이 알고리즘은 모두 문자열 집합에 대해 쉽게 데이타 분류가 가능한 장점이 있다. 실험결과 최적의 분리 경계면을 이용한 SVM알고리즘과 두 문자열의 각 토근을 기반으로 비교한 TFIDF 알고리즘이 우수한 결과를 보였다. 특히 속성간의 상관관계(correlation)와 인접한 각 문자 기호를 n-gram 방식으로 변환하여 함께 고려하였을 때 우수한 정확도를 보였다.

이 논문의 구성은 다음과 같다. 먼저 제2장에서 관련 연구를 통해 기존의 연구와 이 논문에서 제안한 기법과의 차별성을 제시한다. 제3장에서는 다변량 스트림 데이타 분류를 위한 전처리 과정과 분류 기법을 설명한다. 제4장에서는 시계열 데이타와 로봇 센서에서 수집되어 분류된 데이타를 이용하여 제안한 분류 기법의 정확도를 실험하고 평가하며, 마지막으로 제5장에서 연구결과를 요약한다.

2. 관련연구

이 논문에서 제안하는 다변량 스트림 데이타는 다차원 시계열 데이타와 유사하다. 다차원 시계열 데이타에 대한 분류 기법의 연구는 [6,9-12]에 의해 수행되어왔다. 이 연구들의 주요 기법을 보면 순차 데이타에 대해 확률을 이용하여 상태(state)와 전이(transition) 구조의 HMM(Hidden Markov Models)과 입력과 출력 사이에 하나 이상의 숨겨진 계층 구조로 각 입력 값의 연속된 가중치 정보를 이용하는 RNN(Recurrent Neural Networks) 기법이 있다. 이 두 기법은 복잡한 입력 매개변수 값을 가지며 긴 순차적인 데이타에 대해 부정확한 결과를 가진다[10]. Kedous는 [10]에서 DTW(Dynamic Time Warping)을 제안하였으며 이는 음성언어 인식 분야에 입력값과 템플릿 사이에 최소 거리 사상 방법을 이용하여 상대적으로 간단하게 분류 기법이 적용 가능하다.

최근 스트림 데이타의 분류기법으로 [6]에 의해 Weighted Ensemble Classifier 방법과 [11]에 의해 SAX (Symbolic Aggregate approXimation) 기법이 제안되었다. Weighted Ensemble Classifier 는 연속적으로 입력되는 스트림 데이타를 고정된 윈도우 크기로 분할하고 각각의 윈도우를 이용하여 트레이닝 데이타를 생성한다. 각 트레이닝 데이타에서 베이스 분류기를 구성하는 단계에서 다중 의사결정 트리를 이용하며 베이스 분류기를 이용하여 예측을 한다. 이 기법은 ensemble classifier와 가중치 스키마를 이용하여 에러를 줄였지만 테스트 단계에서 확률적 근사치 값과 통계적 기법에 의존하여 최적의 유사 클래스를 선택하는 과정의 복잡도가 큰 단점이 있다. 따라서 실시간 수집되는 대용량 스트림 데이타를 효과적으로 처리하는데 적합하지 않다. SAX 기법은 시계열 데이타에 대해 PAA(Piecewise Aggregate Approximation) 기법을 이용하여 이산적인 문자 기호(a, b, c 등)로 변환한 후 유사성 측정 단계에서 각 기호들의 유클리디안 거리를 비교한다. 이 기법은 순서를 고려한 두 시계열 데이타의 유사성(similarity) 검사에 매우 유용하지만, 트레이닝 데이타와 테스트 데이타의 윈도우 크기가 다른 다차원 속성의 시계열 데이타의 유사성 검사에 정확도가 작아지며 다양한 기호를 이용할 경우 변환 단계와 수행단계에서 복잡도가 증가하는 단점이 있다. 예를 들면 [11]에서, 데이타의 증감 변화에 따라 기호를 a부터 n 범위에서 변환한다고 가정할 때, 각 시계열 데이타를 다양한 문자로 변환하여 분류하는 기법은 정확도는 증가하지만, 실시간, 원패스로 처리해야 하는 스트림 데이타 특성상 변환 비용이 큰 단점이 있다. 또한 하나의 시계열 속성 데이타를 "baabccbc", 다른 속성을 "baacb"라 가정할 때 데이타의 길이가 상이하여 단순히 순서만 고려하여 유사도 측정 기법으로 데이타를 분류할 때 정확도가 낮아진다.

이 논문에서는 윈도우 크기에 따른 정확도 감소와 복잡한 분류 알고리즘을 개선하기 위해 다변량 스트림 데이타의 각 슬라이딩 윈도우를 하나의 문서와 같이 고려하여 변환한 다음에 표준 문서 분류기법을 적용하였다. 제안된 기법의 기본 아이디어는 하나의 문서(슬라이딩 윈도우) 내에서 자주 발생하는 단어(증감 기호)들과 연관된 단어(유사한 변화를 갖는 속성)가 함께 포함된 문서를 유사한 문서로 간주하는 것이다. 아울러, 제안하는 기법은 단순히 데이타의 변화 정도만을 고려하여 문자

열로 변환 경우, 주기가 다르면서 유사한 변화를 가지는 시계열 데이타가 동일하게 분류되는 단점을 개선하기 위해, 각 속성간에 연관관계(correlation) 정보를 문자열 집합에 추가하여 분류 정확도를 높였다.

3. 다변량 스트림 데이타 분류

이 장에서는 센서 네트워크의 구조와 데이타 모델을 소개하고 슬라이딩 윈도우 단위로 분류된 다변량 스트림 데이타 분류를 위해 전처리 과정과 분류 단계로 구분하여 설명한다. 전처리 과정에서는 스트림 데이타를 문서 분류 기법에 적용하기 위해 연속된 시계열 데이타를 이산적인 문자기호의 집합으로 변환하고 문자 기호의 집합을 이용하여 문자열을 생성하는 과정을 기술한다. 분류 단계에서는 Supervised 학습 기법으로 베이지안 모델과 SVM 기법을 Unsupervised 학습 기법으로는 문자열 기반 거리 기법을 적용하는 과정을 기술한다.

3.1 시스템 구조와 데이타 모델

센서 네트워크의 일반적인 구조는 분산/계층형 모델이며 그림 2와 같이 센서 노드, 싱크 노드, 베이스 스테이션과 서버 노드로 구성된다[1,2]. 대용량 데이타를 무선으로 제한된 통신 대역폭을 이용하여 상위 노드로 근사적 전송과 축소 저장 기법이 일반적이다. 일반적으로 데이타의 처리는 데이타를 최초 수집한 노드와 가장 근접한 곳에서 처리하는 것이 비용이 작으며 물리적, 논리적 관점에서 시간에 따른 데이타 사이에 유사성이 크므로 분석에 용이하다[3]. 예를 들어 동식물 생태 모니터링 응용에서 물리적인 지역성을 고려할 때, 특정한 지역에서 일정한 시간 동안 수집되는 동식물의 생태 변화와 온도, 빛, 습도 등의 센서 정보는 거의 유사하다. 따라서 일정한 주기마다 하위 노드에서 상위 노드로 전송할 데이타를 요약, 집계하여 처리하는 것이 배터리 절약과 전송하는 통신 비용을 줄일 수 있다.

그림 2에서 센서 네트워크의 단말 노드는 센서 노드(S)로 구성되며, 각 센서(x)는 온도, 습도와 같은 값(v)을 가진다. 만약 n개의 센서 노드가 m개의 센서를 가질 때 $S_n = \{(x_1, v_1),\cdots,(x_m, v_m)\}$이다. 각 센서가 동일한 기능을 하고 일정한 주기로 센서에서 센서 노드로 데이타를 푸쉬 방식으로 전송한다. 따라서 센서 노드는 큐(queue) 형태의 메모리 버퍼와 소형의 프로세스를 가지며 고정 또는 이동 특성을 가진다. 또한 배터리 기반의 전력에 의존하며, 무선 통신이므로 주기적으로 싱크 노드에 데이타 전송시 데이타 손실이 있을 수 있다. 싱크 노드(SN)가 N개의 센서 노드로 구성되면 $SN=\{S_1, \cdots,S_n\}$이고, 메인 메모리 기반의 센서 DBMS을 포함한다. 싱크 노드는 센서에서 수집된 다차원 속성의 데이타를 요약하여 일정기간 저장하거나 질의의 요청에 따라 데이타를 전송한다. 센서 네트워크의 제약사항을 고려할 때 데이타 수집단계에서 모든 노드에서 정확한 데이타를 수집하는 것이 불가능하며, 실시간 처리해야 하기 때문에 데이타 요약이나 근사적 데이타 처리를 한다[13].

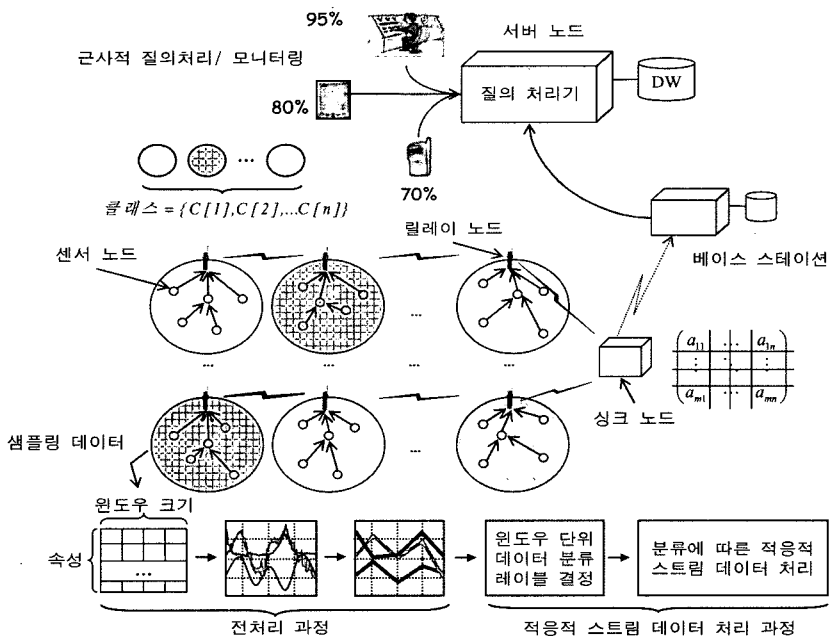


그림 2 무선 센서 네트워크에서 실시간 데이타 분석 및 처리 과정

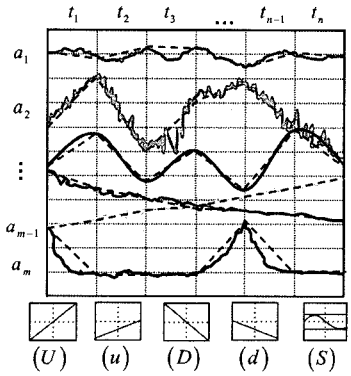
3.2 전처리 과정

전처리 단계는 윈도우 기반의 다차원 속성의 데이타 신호를 데이타의 변화 정도에 따라 이산적인 기호로 변환하고 이 기호를 이용하여 문자열의 집합을 생성하는 과정이다. 이 단계는 기존의 튜플 기반 데이타 분류 알고리즘에 윈도우 기반 데이타를 적용하기 위한 것이다. 그림 3(a)는 다차원 속성($a_1 \sim a_m$)이 일정한 시간 윈도우($t_1 \sim t_n$) 사이에서 서로 다른 주기와 파형으로 수집되는 예제이다. 각 신호 변화는 정규화 과정을 통해 변화 정도에 따라 5개의 기호(*U, u, D, d, S*)로 정의하였다. 이 기호는 단위 시간 간격($t_k \sim t_{k-1}$)에서 수집된 값이 크게 증가할 때 *U*, 증가의 정도가 적을 때 *u*, 크게 감소일 때 *D*, 작게 감소할 때 *d*, 변화의 정도가 크기 않을 때 *S* 기호로 한다. 따라서 각 속성은 데이타 변환 기호를 이용하여 문자열이 생성된다. 하지만 속성마다 하나의 문자열은 데이타의 변환을 다양하게 표현하기 힘들다.

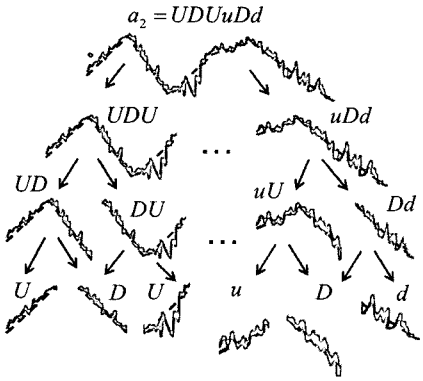
이 논문에서는 하나의 속성에 대해 변환된 문자열을 다양하게 표현하기 위해 그림 3(b)의 HPL[7]를 적용하였으며, 이 기법은 정보검색의 n-gram 방식[8]과 유사하다. 예를 들어 하나의 문자열 "센서데이타"의 경우 2-gram은 "{센서, 서데, 데이, 이타}"가 된다. 그러므로 온도 센서의 변화가 "*UDUuDd*" 라 가정할 때 1-gram은 {*U, u, D, d*}, 2-gram은 {*UD, DU, Uu, uD, Dd*}, 3-gram은 {*UDU, DUu, UuD, uDd*}, 4-gram은 {*UDUu, DUuD, UuDd*}, 5-gram은 {*UDUuD, DUuDd*}로 구성된다.

그림 3(c)는 하나의 속성을 단순히 데이타의 증감의 변화 정도에 따라 5개의 기호로 표시할 때 문제점을 보여준다. 그림과 같이 시간의 변화에 따라 빛과 온도 센서에서 수집되는 데이타 흐름은 유사하지만 소리 센서 값은 두 속성과 반대의 흐름을 보이고 있다. 하지만 그림 3(b)와 같은 단지 n-gram 방식을 이용하면 소리, 빛과 온도의 변화가 유사한 기호로 나열되게 된다. 따라서 시간에 따라 각 속성의 변화 정도와 각 속성간에 연관관계도 함께 고려해야만 정확한 시계열 데이타 분석이 가능하다. 예를 들어 그림 1(b)에서 차량에서 수집된 스트림 데이타에서 차선을 변경할 때의 측정된 값과 가속기 센서의 값이 상관관계를 가지고 변환하는 것이 데이타 특성을 파악하는데 중요한 정보이다.

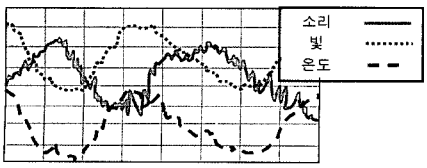
그림 3(d)는 n-gram과 내부 속성간의 상관관계에서 얻어진 정보를 단어로 표시한 예이다. 속성간에 상관성은 각 속성의 쌍에 대해 상관관계가 높은 속성의 쌍 만을 단어로 변환("a_1a_2")하여 추가하였다. 실험 과정에서 만약 온도와 소리 센서의 상관계수가 0.6 이상이 되면 두 센서의 속성 값을 단어로 변환하여 추가하였다. 이



(a) 윈도우 기반 다차원 데이타와 변환 기호



(b) HPL (Hierarchical Piecewise Linear representation)



(c) 다차원 속성간의 상관관계

1-gram	*U, u, D, d, S*
2-gram	*UU, Uu, UD, ... ,DS, dS, SS*
3-gram	*UUU, UuD , ... , UuD,DSD*
4-gram	*UUUU, UuDd, ..., DuUS*
5-gram	*UUUUU, UuDdS, ..., DuudS*
Correlation	a_1a_2 , a_1a_3 , ... , $a_{m-1}a_m$

(d) n-gram과 속성간 상관관계 집합

그림 3 윈도우 기반 다변량 데이타의 전처리 과정

논문에서 HPL과 n-gram을 이용하여 하나의 긴 문자열을 부분 문자열의 집합으로 생성함으로써 다양한 타입의 스트림 데이타를 쉽게 변환이 가능하다. 또한 순서가 고려된 부분 패턴에 대해 패턴 매칭 알고리즘을 이용하여 쉽고 빠르게 분석이 가능한 장점이 있다[14].

그림 4(a)는 행과 열로 구성된 다변량 스트림 데이타를 기호 문자열로 변환한 예이다. 그림 4(c)는 각 문자열에 대해 HPL 기법을 이용하여 1-gram에서 5-gram까지 부분 문자열을 생성한 예이다. 그림 4(d)는 부분 문자열의 집합을 빈도수를 기준으로 가중치를 부여하여 하나의 윈도우에 대해 벡터를 구성한 예이다. 이 논문에서는 각 부분 문자열의 빈도수에 따라 5개의 그룹(1.0, 0.8, 0.6, 0.4, 0.2, 0.0)으로 분류한다. 그림 4(b)는 윈도우 단위로 구축된 가중치 벡터에 대해 데이타베이스를 구축한 예이다. 각 행은 3905개와 속성의 쌍으로 구성된 가중치 벡터이고 각 행에 대해 상관관계 계수를 구해 유사도 측정에 활용한다.

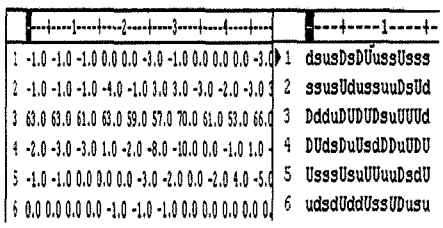
알고리즘 1은 다변량 스트림 데이타의 전처리 과정에 대한 알고리즘이다. 입력 데이타는 행과 열로 구성된 행렬 구조이며, 출력 데이타는 각 윈도우 마다 생성된 단어 리스트이다. 1 단계에서 각 행 벡터에 대한 정규화와 단위 문자를 생성한 후, 2단계에서 n-gram 방식과 속성간의 연관성이 높은 속성에 대해 단어를 생성한다. 알고리즘 1에서 생성된 WordVector[]는 5개의 기호(U, u, D, d, S)에 의해 생성된 부분 문자열의 조합으로 생성된 벡터이다. 벡터의 길이는 5개의 기호에 의해 생성될 수 있는 3905개와 속성의 쌍으로 구성된 문자열 벡터이다.

3.3 윈도우 기반 데이타 분류 기법

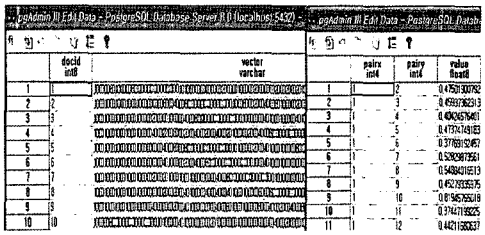
이 장에서는 3.2절에서 전처리 과정을 거친 스트림 데이타에 대해 표준 문서 분류기법을 적용하는 과정을 설명한다. 적용된 분류 기법은 Supervised 학습으로 베이지안 분류와 SVM을 이용하였으며 Unsupervised 학습 기법으로 Jaro, Jaro-Winkler, Jaccard, TFIDF와 같은 문자열 기반 거리 함수를 이용하였다.

3.3.1 Supervised 분류 기법

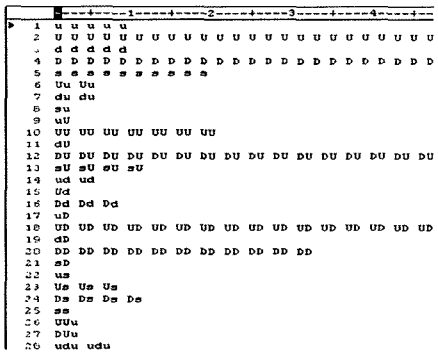
베이지안 분류(Bayesian Classification): 베이지안 분류는 베이지안 이론으로 확률에 기반하여 가장 가능성 있는 클래스를 선택하는 것으로 데이타를 분류한다[5,15]. 기본적인 가정은 주어진 하나의 클래스의 속성값이 다른 속성값들과 독립이다. 이 기법은 트레이닝 데이타로부터 계산된 이전 확률을 사용하여 최대 사후 확률 클래스를 선택하는 기법이다. 예를 들어 트레이닝 단계에서 트레이닝 데이타로부터 $P(uU \mid class = \text{"}intrusion\text{"})$과 $P(a_1a_2 \mid class = \text{"}normal\text{"})$과 같은 이전의 확률 분포를 가진다면, 테스트 단계에서 클래스 레이블이 지정되지 않은 윈도우 기반 테스트 데이타에 대해 수식 (1)에 따라 각 클래스 C_i에 대해 사후 확률을 평가한다. 테



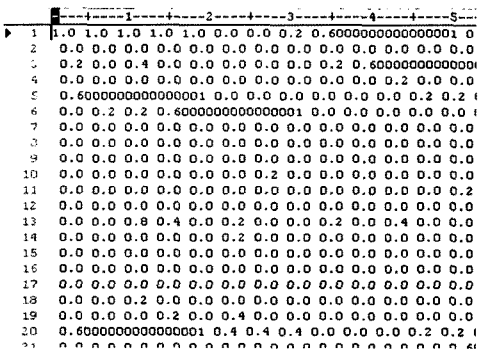
(a) 다변량 데이타의 기호 문자열 변환



(b) 각 부분 문자열 가중치 벡터와 연관관계



(c) n-gram을 이용한 문자열 생성



(d) 빈발도 기준의 가중치 벡터

그림 4 데이타 전처리 과정과 가중치 벡터 구성 단계

알고리즘 1 단위 윈도우에 대한 문자열 생성 단계

```
Begin
  입력: 행렬 구조로 구성된다. 각 행은 속성(x)을 의미하며 열은 시간 단위마다 수집된 값(v)을 의미한다. S_n
       ={(x_1, v_1),...,(x_m, v_m)}일 때 윈도우와 속성의 크기는 m 이다.
  출력: 다차원 속성의 변화와 속성간의 연관성을 고려한 단어 리스트

Step 1: CreateSymbol()
  Normalization(Inputdata[]);
  //각 행 벡터에 대한 정규화(Normalization)를 통한 각 기호의 증감 범위 설정
  For i = 1 to m Do { // m = 속성의 크기
    MakeSymbol(Inputdata[i]);
    //입력 데이타인 행렬의 각 행마다 증감의 정도에 따라 문자열 생성
    //단위 시간 변화(t_k - t_k-1)가 사전에 정의된 기호(U, u, D, d, S) 범위에 따라 할당함
  } //End of For

Step 2: MakeWordList()
  For i = 1 to m Do {// m = 윈도우 크기
    WordVector[index] = MakeHPL(); //HPL 기법을 이용하여 부분 문자열 생성
    //각 행의 문자열 마다 1-gram에서 5-gram까지 단어 생성
  } //End of For
  For j = 1 to n Do { //n = 속성의 개수
    WordVector[index++] = MakeCorrelation(j, j+1);
    //속성간의 상관관계(correlation)을 구하기 위해 j와 j+1번째 두 행 벡터에 대해 피어슨 상관 함수를 이용하
    여 상관 계수(coefficient)를 구함. 만약 상관계수 값이 0.6 이상이면 a_j a_j+1 단어 생성 추가
  } //End of For
  Return WordVector[];
End
```

알고리즘 2 베이지안 분류 알고리즘

```
Begin
  입력: 각 데이타 샘플은 n-차원의 feature 벡터이다.
       만약 트레이닝 데이타가 알고리즘 1을 수행하여 3개의 속성이 문자열로 변환되고 윈도우 마다
       Normal과 Abnormal로 분류가 되었다면 다음과 같은 쌍을 가진다. 각 속성은 공란으로 구분된다.
       예: {UDUDUDdUDU uUDdsuUdDu sUUDDsssUU, Normal
            uUDdsuUdDu uUDdsuUdDu uUDdsuUdDu, Abnormal}
  출력: 이전 확률 값을 기반으로 최대 사후 확률 클래스를 반환
Step 1: MakePriorPro() //다차원 해쉬 테이블을 이용한 이전 확률 생성
       While (ReadLine() != Null) {
         Splite (Word_list, separator) //separator = ","
         MakeWordlist () // 알고리즘 1의 2단계 참조
         MakeMultipleHash (ClassName, WordSymbol, Probability)
         // 클래스와 단어에 대해 이전 확률을 해쉬 테이블에 구성
         // HashUpdate()와 HashKeyReturn()함수 이용
       } // end of While
Step 2: FindMaxPosterior() // 최대 사후 확률 클래스 선택
       For i = 0 to Class.length Do{
         ProVal = ProductOfPosteriorProb(C_i); // 각 클래스에 대한 이전 확률 값의 곱
         MaxClass = FindMaxClass(); // 최대 사후 확률 클래스 선택
       } //end of For
       Return MaxClass;
End
```

스트 데이타는 $P(C_i \mid X)$가 최대가 되는 클래스 C_i를 할당하게 된다.

$$P(C_i \mid X) = \frac{P(X \mid C_i)P(C_i)}{P(X)} \quad (1)$$

이때 $P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)$ 이며,

$$1 \le i, j \le m,\ i \ne j$$

베이지안 분류 기법의 주요 수행단계는 알고리즘 2와 같다. 이 알고리즘은 다차원 해쉬 테이블을 이용하여 이전 확률을 생성하는 단계와 최대 사후 확률 클래스를 생성하는 단계로 구성된다.

SVM(Support Vector Machine): SVM은 통계학적인 이론에 근거하며 패턴 인식에서 문서 분류 응용까지 널리 사용된다. SVM은 다변량 데이타에 대해서 잘 동작하고 기본적으로 두 개의 클래스에 대한 분류기이며, 일반적으로 다차원 클래스 분류기(one vs. the rest)로 확장이 가능하다[15,16]. 이 논문의 모델에서 SVM의 입력으로 Feature 벡터에 그림 3(d)의 n-gram 리스트와 각 속성간의 연관 쌍으로 구성된다. Feature 벡터의 값은 슬라이딩 윈도우에 나타난 n-gram 리스트의 빈도수를 고려하여 가중치를 부여한 값과 상관 함수의 상관계수 값을 이용하였다.

예를 들어 하나의 슬라이딩 윈도우(W)에 W_i={uD=2, UUd=10, UDDD=5, a_ia_{i+1}=0.8)일 때 Feature 벡터는 {uD, UUd, UDDD, a_ia_{i+1}}이고 빈도수를 이용한 가중치(예: 0.2, 0.6, 0.8, 1.0)와 상관계수 값을 고려할 때 Feature 벡터의 값은 {0.2, 0.8, 0.6, 0.8}로 표현된다. 주어진 트레이닝 데이타의 각 Feature에 대해 SVM은 최대 여분(margin)이 많게 생성되는 hyper-plane을 생성한다.

테스트 단계에서는 트레이닝 단계에서 생성된 hyper-plane에 의해 분할된 다차원 공간에 매핑하여 새로운 데이타를 분류한다. 실험에서 우리는 RBF(Radial Basis Function) Kernel, $K(x_i, y_i) = e^{-\gamma\|x_i - y_i\|^2}, (\gamma > 0)$ 을 사용했다. 트레이닝 과정 동안 노이즈 데이타 처리와 과대적합(overfitting) 문제에 대해 성능을 향상시키기 위해 소프트 여분(soft margin)에 대한 에러를 허용하였다. 또한 Two-Norm 소프트 여분 값을 0.1로 설정하였다.

3.3.2 Unsupervised 분류 기법

문자열 기반 거리(String-based Distance): Unsupervised 분류 기법은 사전에 어떠한 트레이닝도 요구되지 않으며 클래스도 정의되어 있지 않다. 문자열 기반 거리 기법은 두 문자열 사이의 유사성을 측정하기 위해 두 문자열의 거리를 이용한다. 예를 들어 두 개의 문자열 UuDDSS와 USDDSS의 거리는 *Dist*(UuDDSS, USDDSS)로 계산한다. 거리 측정에 사용되는 많은 기법 들 중에 이 논문에서는 토근(token) 기반 문자열 기법(Jaccard, TFIDF)과 편집거리(Edit-distance) 기반(Jaro, Jaro-Winkler)을 이용한다. 이 두 기법은 일반적인 문자열 매칭 문제에 좋은 결과를 보인다[17,18].

이 두 기법에 대한 상세한 설명은 [14,17]을 참조하며, 표 1의 용어에 따라 4개의 측정 기준 식 (2)에서 (5)까지 제시한다.

$$\text{Jaccard}(x, y) = \frac{|T_x \cap T_y|}{|T_x \cup T_y|} \quad (2)$$

$$\text{TFIDF}(x, y) = \sum_{w \in T_x \cap T_y} V(w, T_x) \times V(w, T_y),\ \text{이때}$$

$$V(w, T_x) = log\left(TF_{w,T_y} + 1\right) \times \frac{log\left(IDF_w\right)}{\sqrt{\sum_{w'}\left(log\left(TF_{w,T_y} + 1\right) \times log\left(IDF_w\right)\right)}}$$

($V(w, T_y)$)에 대한 대칭) 이때 TF_{w,T_x} 는 T_x 안에 w의 빈도수, IDF_w 는 역 문서 빈도수 (3)

$$\text{Jaro}(x, y) = \frac{1}{3} \times \left(\frac{|CC_{x,y}|}{C_x} + \frac{|CC_{y,x}|}{C_y} + \frac{|CC_{x,y}| - X_{CC_{x,y}, CC_{y,x}}}{2|CC_{x,y}|} \right) \quad (4)$$

$$\text{Jaro-Winkler}(x, y) = Jaro(x, y) + \frac{max(|L|, 4)}{10} \times (1 - Jaro(x, y)),$$

이때 L 은 x와 y의 가장 긴 공통 접두사 (5)

벡터 기반 코사인 거리(Vector-based Cosine Distance): 이 기법은 문자열 사이의 거리가 아닌 하나의 슬라이딩 윈도우에서 발생하는 n-gram 문자열과 상관 속성을 하나의 벡터로 생성한 후 비교하는 것이다. 이 논문에서는 각 윈도우 단위의 n-gram 문자열에 대해 발생 빈도에 따라 6 그룹{0.0, 0.2, 0.4, 0.6, 0.8, 1.0}으로 가중치를 고려하며, 속성간에 피어슨 상관계수가 0.6 이상일 때 벡터에 추가한다. 예를 들어 단어와 속성의 집합(T)이 T = {*Uu*, *DDS*, *UDu*, a_1a_2, a_2a_3}일 때

표 1 문자열 기반 측정에 따른 용어들

기호	설명	기호	설명
x, y	각 센서 속성들에 대한 n-gram 리스트와 상관관계	$CC_{x,y}$	x, y 문자열에 공통된 모든 문자열들
C_x	x 문자열의 모든 문자들	T_x	x 에 대한 모든 n-gram과 상관관계 용어들
$X_{x,y}$	하나의 문자열과 다른 문자열에 대응되는 문자의 순서가 다른 문자의 개수		

W_1= <*Uu*, *Udu*, a_1a_2 > = <0.8, 0, 0.6, 0.7, 0>이고 W_2= <*Uu*, *DDS*, a_1a_2 > = <0.8, 0.6, 0, 0.7, 0>라면 두 윈도우 W_1과 W_2의 거리는 $Cos\theta = \frac{W_2 \cdot W_2}{\| W_1 \| \| W_2 \|}$ 이다[15]. 위 예제에서 *Uu*의 가중치는 0.8, *DDS*는 0.6, *UDu*는 0.7이며, a_1a_2의 상관계수는 0.7이라 가정했다.

4. 실험 및 평가

우리는 그림 5와 같이 실험을 위해 [19]에서 제공된 두 개의 다차원 시계열 데이타 셋을 이용하였다. 이 데이타는 다차원 속성이며, 사전에 데이타 제공자에 의해 데이타에 대한 특성과 클래스 레이블이 이미 정해진 데이타이다. 따라서 그림 5의 데이타는 이 논문에서 제안한 분류 기법에 대한 실험 및 평가에 적합하다.

4.1 실험 데이타

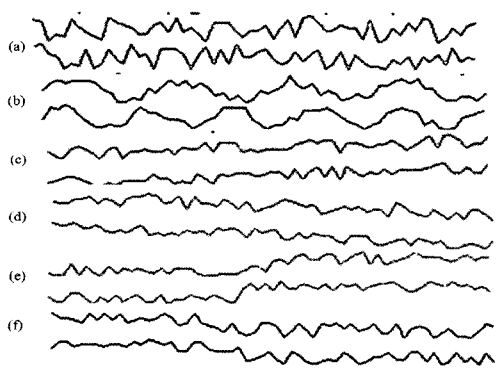
그림 5(a)는 SCCTS로 Alcock와 Manolopoulos의 프로세스에 의해 인위적으로 생성된 시계열 차트의 600개 예제를 포함한다. 이 데이타는 6개(Normal(a), Cyclic(b), Increasing Trend(c), Decreasing Trend(d), Upward Shift(e), Downward Shift(f))의 상이한 클래스로 구성된다. 그림 5(b)는 로봇이 객체를 한 장소에서 다른 장소로 이동할 때 Force와 Torque 센서에서 측정된 시계열 데이타이다. 이 데이타는 로봇에 대한 행위의 실패가 감지된 시점에서 일정한 시간간격 동안 15개의 Force/Torque 샘플을 분류하였다. 로봇 행위 데이타는 크게 5개로 구성되며 각각은 LP_1에서 LP_5까지 다른 학습 문제를 정의한다[19]. 예를 들어 로봇이 정상, 충돌, 물건 유실 등과 같이 행위의 특성에 따라 센서에서 연속적으로 수집된 데이타 값이 다르다. 그림 5(b)의 "front collision & part lost"는 로봇이 정면에 충돌이 일어난 후 가지고 이동하던 물건을 유실했을 때의 시간에 따른 변화 예이다.

데이타 분류를 위한 트레이닝 단계에서 우리는 6개의 다른 클래스와 로봇의 행위(Normal, Collision, Obstruction 등) 클래스를 이용하였다. 3장에서 제시한 분류 기법의 정확도를 측정하기 위해 *k-fold cross-validation* 기법[5,15]을 적용하였다. 이 기법은 입력 데이타(S)를 동일한 크기의 k 개 부분집합($S=\{S_1, S_2,\cdots, S_k\}$)으로 임의로 분할한다. 만약 S_i가 테스트 데이타라면 나머지 부분 집합은 분류기를 트레이닝 하는데 사용된다. 그러므로 트레이닝과 테스팅 단계가 k 번 수행된다. 분류기의 정확도는 총 시도한 개수에서 k 번 반복을 통해 올바르게 분류된 클래스의 전체 수를 나눈 것으로 하였다.

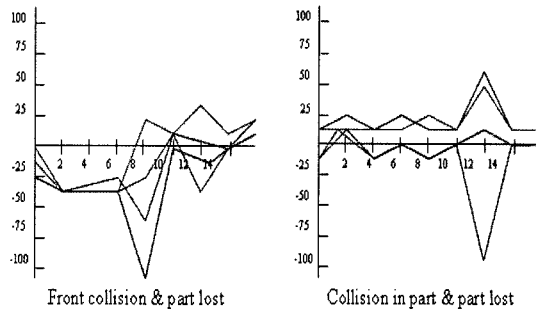
4.2 실험 결과 및 평가

실험 결과는 그림 6에서 보여준다. 그림 6(a)는 3절에서 제시한 표준 문서 분류 기법을 이용하여 정확도를 측정한 결과이다. 이 결과는 그림 3(d)와 같이 n-gram의 토큰을 1-gram만 적용하여 생성하였을 때부터 5-gram까지 모두 생성하여 함께 고려했을 때의 정확도를 비교하였다. 예를 들어 3-gram의 경우 1-gram부터 3-gram에서 생성된 모든 문자열을 포함한다. 실험 결과 단순히 데이타의 증감만을 고려하여 문자로 생성한 것보다 HPL와 n-gram을 이용하여 다양하게 분석한 것이 우수한 결과를 보였다.

하지만 단순히 시간에 따른 데이타의 증감 변화만을 고려하는 것은 순차적으로 의미에서 반대의 흐름을 가지는 파형을 구분하지 못하는 단점이 있다. 따라서 우리는 실험에서 각 속성간의 연관성을 분석하여 연관성이 강한 속성 정보를 데이타의 증감 변화와 함께 고려하여 분류 정확도를 높였다. 그림 6(b)는 n-gram 토큰과 상관관계를 함께 고려하였을 때의 정확성을 비교하였다. 실험 결과 n-gram의 토큰을 고려할수록 정확도가 증가하였다. 이는 긴 토큰에 대해 패턴의 시간적 지역성을



(a) SCCTS(Synthetic Control Chart Time Series)



(b) 로봇 행위에 따라 센서에서 수집된 데이타

그림 5 다차원 시계열 데이타 셋

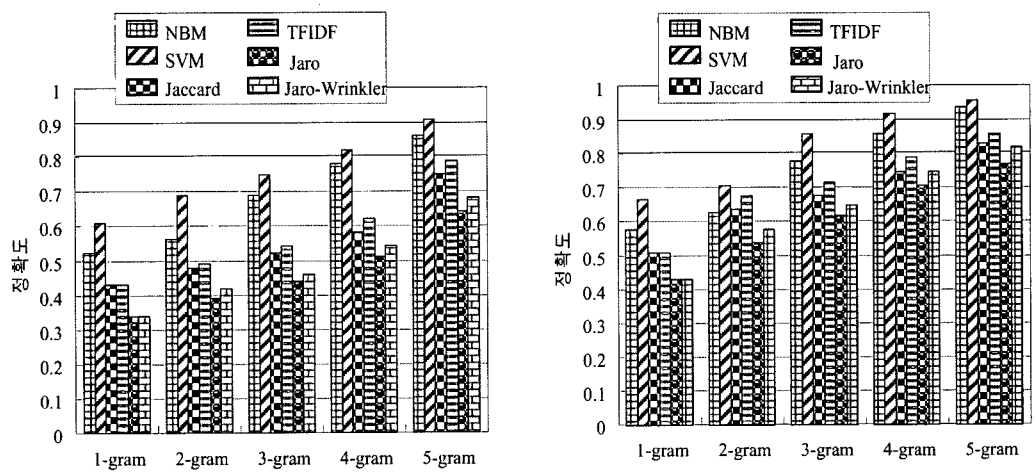
잘 표현하기 때문이다. 정확도는 n-gram과 연관 속성을 함께 고려하는 것이 일반적으로 높으며 그림 6과 같이 3과 4-gram에서 확연히 정확도가 증가함을 알 수 있다.

이 실험 결과에 대한 실 세계 예로, 그림 1(a)는 차량이 차선을 변경할 때(세그먼트= B, D, F, H) 스트림 데이타 변화가 모두 입력 값이 증가 후 감소(기호 문자열 ="UUuDDU")하는 모양을 가진다. 그리고 차선을 변경할 때 스피드가 증가하며 가속기 센서의 값이 함께 증가하는 그림 1(b)와 같은 형태를 가지므로, 스피드와 가속기의 상관관계 계수가 높은 속성도 함께 고려해야 한다. 따라서 차량 센서에서 수집되는 스트림 데이타와 기타 센서에서 수집되는 데이타를 제시한 기법에 적용하면 차량의 흐름과 상태를 정확히 분석할 수 있다.

Supervised와 Unsupervised 학습 기법을 비교해 보면 Supervised 학습기법이 정확도가 높다. SVM은 테스트 방법들 중에서 가장 우수한 결과를 보였다. Unsupervised 학습 기법에서는 토큰 기반의 문자열 거리 척도 기법이 편집거리 기반의 척도 보다 정확도가 높다. 이것은 스트림 데이타의 경우 순서가 매우 중요하기 때문에 단순히 동일한 문자 기호가 존재한다는 이유로 유사하다고 보기 힘들기 때문이다. 실험은 자바 언어를 이용하였으며 [20,21]에서 제공한 표준 분류 라이브러리와 패키지를 확장하여 적용하였다.

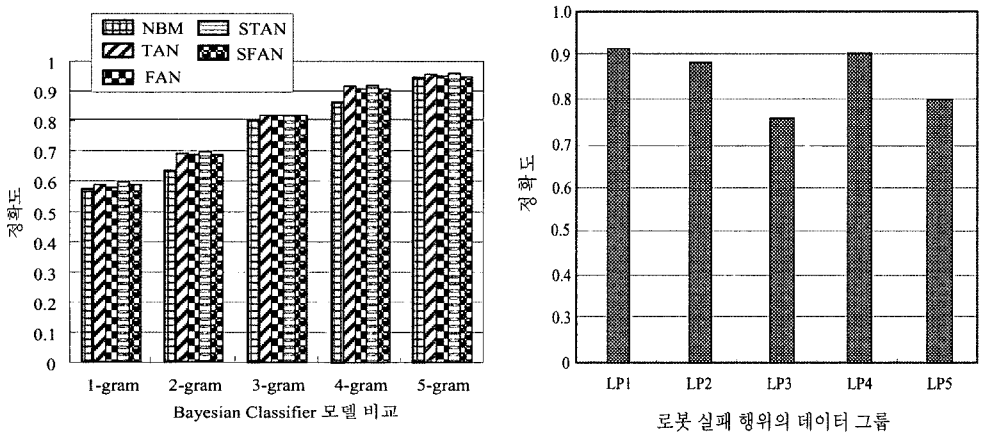
그림 6에서 실험한 베이지안 분류기는 주어진 윈도우 클래스 내부에 하나의 속성 값이 다른 속성의 값과 상호 독립을 가정하며 이를 '클래스 조건 독립'이라 한다. 하지만 WSN 환경에서 수집되는 다양한 속성의 센서 값이 모두 독립적이지 않기 때문에, 이 논문에서는 다변량 스트림 데이타의 속성간에 의존 관계를 고려한 데이타 분류 실험을 위해 그림 7(a)와 같이 베이지안 분류기의 확장된 기법을 실험하였다. 실험에 사용된 기법은 TAN(Tree Augmented Naïve Bayes), FAN(Forest Augmented Naïve Bayes), STAN(Selective Tree Augmented Naïve Bayes), SFAN(Selective Tree Augmented Naïve Bayes)이다[22,23]. 로봇 실패 행위 데이타[19]를 이용한 실험 결과 TAN과 STAN이 다른 기법에 비해 우수한 결과를 보였다. 이 실험 결과는 WSN에서 윈도우 단위로 수집되는 데이타 속성간에 의존 관계가 존재할 경우 의존도를 고려한 분류 기법을 적용해야 한다는 것을 보였다.

그림 7(b)는 트레이닝 데이타에 대해 사전에 클래스 레이블이 지정된 데이타를 이용하였다[19]. 우리는 동일한 데이타에 대해 클래스 레이블이 없다고 간주하고 k-means 클러스터링 기법을 이용하여 기존 데이타와 비교하는 실험을 하였다. 이 실험에서 k 값은 LP_1에서 LP_5내의 상이한 클래스의 개수로 하였다. 실험 방법은 3장의 전처리 과정을 이용하여 단위 윈도우마다 문자열과 속성의 쌍을 이용하여 하나의 가중치 벡터(예: <0.8,0.2,1.0,…,0.0> ,…, <0.2,0.2,1.0,…,0.0>)를 생성하였다. 이때 문자열의 빈도수에 따른 상대적 중요도에 따라 6개의 그룹 <1.0, 0.8, 0.6, 0.4, 0.2, 0.0>으로 구분하였으며, 속성의 쌍은 0.6이상 상관 계수를 갖는 속성을 값을 고려하였다. 3.2절의 전처리 과정을 통해 클러스터링 기법을 적용한 결과 Collision과 Obstruction 클래스 계열은 매우 유사한 특성이 있었다. 따라서 그림 7(b)와 같이 Normal 클래스가 많이 포함된 LP_1에서 높은 정확도를 보인 반면에 Collision과 Obstruction이 많이 포함



(a) n-gram 문자열의 단계적 적용 범위 (b) n-gram 문자열과 correlation을 단계적 적용 범위

그림 6 정확도 비교 (n-gram 문자열과 correlation의 적용 범위)



(a) 확장된 베이지안 분류 모델 (b) 클러스터링 기법의 정확도

그림 7 확장된 분류 기법과 클러스터링 기법의 정확도

된 LP_3에서는 정확도가 낮게 나타났다.

5. 결 론

이 논문은 연속된 스트림 데이타에 대한 확장 가능한 다차원 속성의 스트림 데이타 분류 기법을 제안하였다. 분류 과정에서 우리는 n-gram과 HPL 기법을 적용하여 연속된 스트림 데이타를 이산적 기호 표현으로 변환하였다. 분류 정확도를 높이기 위해 순서화된 문자의 증감 변화율 이외에 스트림 데이타의 속성간 연관계수를 함께 고려하였다.

실험을 위해 Supervised 분류 기법으로 베이지안 분류와 SVM 기법을 이용하였으며, Unsupervised 학습에는 Jaccard, TFIDF, Jaro와 Jaro Winkler 기법을 적용하였다. 실험 결과 SVM과 TFIDF가 각각 우수한 결과를 보였다. 또한 정확도 측면에서 단순히 n-gram을 적용한 것보다 속성간에 연관성을 함께 고려하였을 때 우수한 결과를 보였다.

제안된 슬라이딩 윈도우 기반의 다변량 스트림 데이타 분류 기법은 다차원 센서에서 연속적으로 데이타가 수집되는 무선 센서 네트워크의 미들웨어 시스템에 적용 가능하다. 특히 실시간으로 데이타를 분석하는 대기 및 해양 기후 모니터링, 이동객체 추적, 동식물 생태 모니터링과 환자 생체 신호 인식, 재난 및 응급 관리 분야 등에 적용 될 수 있다.

참 고 문 헌

[1] A. Mainwaring, and J. Polastre, et al., "Wireless Sensor Networks for Habitat Monitoring," In ACM Int. Workshop on WSNA, pp.88-97, 2002.

[2] B. Xu and O. Wolfson., "Time-Series Prediction with Applications to Traffic and Moving Objects Databases," In ACM Int. Workshop on MobiDE, pp.56-60, 2003.

[3] R. C. Oliver, and K. Smettem, et al., "Field Testing a Wireless Sensor Network for Reactive Environmental Monitoring," In Proc. of ISSNIP Conf., pp.7-12, 2004.

[4] M. Galan, H. Liu, and K. Torkkola, "Intelligent Instance Selection of Data Streams for Smart Sensor Applications," SPIE Defense & Security Symposium, Intelligent Computing, pp.108-119, 2005.

[5] J. Han and M. Kamber., 'Data Mining Concepts and Techniques,' Morgan Kaufmann Publishers, 2000.

[6] H. Wang, W. Fan, P. S. Yu, and J. Han., "Mining Concept-Drifting Data Streams Using Ensemble Classifiers," In Proc. of ACM SIGKDD Conf., pp. 226-235, 2003.

[7] Xianping Ge., "Pattern Matching in Financial Time Series Data," In Final Project Report for ICS 278 UC Irvine, 1998.

[8] M. Nagao and S. Mori, "A new method of N-gram statistics for large number of n and automatic extraction of words and phrases from large text data of Japanese," Int. Conf. on Computational Linguistics, pp.611-615, 1994.

[9] C. C. Aggrawal, J. Han, and P. S. Yu., "On Demand Classification of Data Streams," In Proc. of ACM SIGKDD Conf., pp.503-508, 2004.

[10] M. W. Kadous and C. Sammut., "Classification of multivariate time series and structured data using constructive induction," Machine Learning Journal, Vol. 58, pp.176-216, 2005.

[11] J. Lin, E. Keogh, S. Lonardi, and B. Chiu., "A Symbolic Representation of Time Series with Implications for Streaming Algorithms," In ACM SIGMOD Workshop on DMKD, pp.2-11, 2003.

[12] P. Geurts., "Pattern Extraction for Time Series Classification," In Proc. of PKDD, pp.115-127, 2001.
[13] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy., "Mining Data Streams: A Review," ACM SIGMOD Record Vol. 34, No. 2, pp.18-26, 2005.
[14] W. W.Cohen, P. Ravikumar, and S. Fienberg., "A Comparison of String Distance Metrics for Naming-matching tasks," In Proc. of IIWEB, pp.73-78, 2003.
[15] P.N. Tan, M. Steinbach, and V.Kumar., 'Introduction to data Mining,' Pearson Addison Wesley, 2005.
[16] N. Cristianini and J. Shawe-Taylor., 'An Introduction to Support Vector Machines,' Cambridge University Press, 2000.
[17] B.W. On, D.W. Lee, J. W. Kang, and P. Mitra, "Comparative Study of Name Disambiguation Problem using a Scalable Blocking-based Framework," In ACM/IEEE JCDL, pp.344-353, 2005.
[18] R. Agrawal, G. Psaila, E. L. Wimmers, and Mohamed Zait., "Querying Shapes of Histories," Proc. of VLBD Conf., pp.502-514, 1995.
[19] S. Hettich and S. D. Bay, 'The UCI KDD Archive [http://kdd.ics.uci.edu] (Robot Execution Failure, Synthetic Control Chart Time Series),' Irvine, CA: Univ. of California, Dept. of Information and Computer Science, 1999.
[20] A Library for Support Vector Machines., http://www.csie.ntu.edu.tw/~cjlin/libsvm
[21] SecondString (Jave-based Package of Approximate String-Matching)., http://secondstring. sourceforge.net.
[22] J. Chen and R. Greiner, "Comparing Bayesian Network Classifiers," In Proc. of UAI-99, pp.101-108, Jul., 1999.
[23] Java Bayesian Network Classifier Toolkit, "http://jbnc.sourforge.net," 2005.

서 성 보
1999년 서원대학교 전산학과 학사. 2001년 충북대학교 전산학과 석사. 2001년~2003년 한국전자통신연구원 우정기술연구센터 연구원. 2005년 미국 North Carolina State University 방문 연구원 2006년 충북대학교 전산학 박사. 관심분야는 시공간 데이타베이스, 스트림 데이타마이닝, USN 미들웨어 시스템

강 재 우
1994년 고려대학교 전산학과 학사. 1996년 University of Colorado at Boulder 전산학 석사. 1996년~1997년 AT&T Bell Labs / AT&T Labs Research 연구원. 2000년~2001년 WISEngine Inc. CTO & Founder. 2003년 University of Wisconsin-Madison 전산학 박사. 2003년~현재 North Carolina State University 전산학과 조교수. 관심분야는 Database systems, High-dimensional data management, Data mining, Bioinformatics.

남 광 우
1995년 충북대학교 전자계산학과 학사 1997년 충북대학교 전자계산학과 석사 2001년 충북대학교 전자계산학과 박사 2001년~2004년 한국전자통신연구원 LBS 연구팀 선임연구원. 2004년~현재 군산대학교 컴퓨터정보과학과 전임강사 관심분야는 LBS/텔레매틱스, 시공간 및 메인 메모리 데이타베이스, 데이타마이닝

류 근 호
정보과학회논문지 : 데이타베이스
제 33 권 제 1 호 참조