스트림 데이타 예측을 위한 슬라이딩 윈도우 기반 점진적 회귀분석

(Incremental Regression based on a Sliding Window for Stream Data Prediction)

김 성 현 † 김 룡 †† 류 근 호 †††
(Sung Hyun Kim) (Long Jin) (Keun Ho Ryu)

요 약 최근 센서 네트워크의 발달로 실세계의 많은 데이타가 시간 속성을 갖고 실시간으로 수집되고 있다. 기존의 시계열 데이타 예측 기법은 모델 갱신 없이 예측을 수행하였다. 그러나 스트림 데이타는 매우 빠르게 수집이 되고 시간이 지남에 따라 데이타의 특성이 변경될 수 있으므로 기존의 시계열 예측 기법을 적용하는 것은 적절하지 않다. 따라서 이 논문에서는 슬라이딩 윈도우와 점진적인 회귀분석을 이용한 스트림 데이타 예측 기법을 제안한다. 이 기법은 스트림 데이타를 다중 회귀 모델에 입력하기 위해 차원 분열을 통해 여러 개의 속성으로 분열(Fractal)하고, 변화되는 데이타의 분포를 반영하기 위해 슬라이딩 윈도우 기법을 사용하여 점진적으로 회귀 모델을 갱신한다. 또한 고정 크기 큐를 이용하여 최근의 데이타로만 모델을 유지한다. 이전 데이타의 유지 없이 최소 정보를 갖는 행렬을 통해 모델을 갱신하므로 낮은 공간 복잡도를 갖고 점진적으로 모델을 갱신함으로써 에러율의 증가를 방지한다. 제안된 기법의 타당성은 RME(Relative Mean Error)와 RMSE(Root Mean Square Error)를 이용하여 측정하였고, 실험 결과 다른 기법에 비해 우수하였다.

키워드 : 스트림 데이타, 데이타 예측, 점진적, 회귀분석

Abstract Time series of conventional prediction techniques uses the model which is generated from the training step. This model is applied to new input data without any change. If this model is applied directly to stream data, the rate of prediction accuracy will be decreased. This paper proposes an stream data prediction technique using sliding window and regression. This technique considers the characteristic of time series which may be changed over time. It is composed of two steps. The first step executes a fractional process for applying input data to the regression model. The second step updates the model by using its information as new data. Additionally, the model is maintained by only recent data in a queue. This approach has the following two advantages. It maintains the minimum information of the model by using a matrix, so space complexity is reduced. Moreover, it prevents the increment of error rate by updating the model over time. Accuracy rate of the proposed method is measured by RME(Relative Mean Error) and RMSE(Root Mean Square Error). The results of stream data prediction experiment are performed by the proposed technique IMQR(Incremental Multiple Quadratic Regression) is more efficient than those of MLR(Multiple Linear Regression) and SVR(Support Vector Regression).

Key words : Stream Data, Data Prediction, Incremental, Regression

· 이 논문은 한국전자통신 연구원(USN 사업단) 및 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(지방연구중심대학 육성사업/충북BIT연구중심대학육성사업단)

† 정 회 원 : SPSS 연구원
hyun@dblab.chungbuk.ac.kr
†† 정 회 원 : 한국전자통신연구원 연구원
kimlyong@dblab.chungbuk.ac.kr
††† 정 회 원 : 충북대학교 전기전자 컴퓨터공학부 교수
khryu@dblab.chungbuk.ac.kr

논문접수 : 2006년 11월 21일
심사완료 : 2007년 7월 4일

정보과학회논문지 : 데이타베이스 제34권 제6호(2007.12)

1. 서 론

최근 네트워크와 센서 기술의 발달로 시간과 공간의 제약 없이 실제 환경에서 데이타를 실시간으로 수집하고 분석하여 의사결정에 반영할 수 있게 되었다. 센서 네트워크에서 수집이 되는 데이타는 빠르고 연속적인 특징을 가진 스트림 데이타이다. 스트림 데이타가 수집되는 센서 네트워크는 객체 감시(object guarding), 환경 감시(environment monitoring), 객체 추적(object tracking) 등의 응용 분야가 있으며 현재 많은 연구가 수행되고 있다[1-4].

센서 네트워크에서 수집이 되는 스트림 데이타는 몇 가지 특징을 가지고 있다. 스트림 데이타는 시간에 따라 연속적이고 복잡하여 한시적인 접근만 가능하고 제한된 메모리를 사용하여 동적으로 변화하기 때문에 지속적인 데이타 처리 모델이 요구된다[5,6]. 또한 시간에 따른 순서를 가지기 때문에 랜덤으로 접근이 불가능하다[5,7]. 스트림 데이타는 시간이 지남에 따라 데이타의 큰 변화나 자주 발생하지 않는 패턴 등에 의해 분포가 변화될 수 있고 실세계에서는 주로 비선형적인 분포를 갖는다. 또한 스트림 데이타는 순서화된 특성을 갖고 있기 때문에 시계열 데이타(time series)로 간주할 수 있다. 시계열 데이타 예측은 과거 데이타를 통해 미래를 예측하여 유용한 정보를 얻는 것이다. 이와 마찬가지로 스트림 데이타에 대한 예측을 수행하여 의사결정에 유용한 정보를 추출할 수 있다.

시계열 데이타 예측 기법을 스트림 데이타에 적용하게 되면 몇 가지 문제점이 발생한다. 첫째, 기존의 방법은 일정한 크기의 과거 데이타를 통해 미래를 예측하거나 모델을 만든 후에 미래를 예측한다. 스트림 데이타는 짧은 시간에 매우 많은 데이타가 연속적으로 수집이 되기 때문에 과거 데이타를 저장할 수 없는 제약사항을 갖는다. 둘째, 스트림 데이타는 시간이 지남에 따라 데이타의 특성이 변화될 수 있기 때문에 모델을 만든 후에 예측을 수행하면 점진적으로 에러율이 높아질 것이다. 셋째, 모델을 갱신하는 기법을 스트림 데이타에 적용하면 일반적으로 데이타 입력 비용이 계산 비용보다 낮으므로 모든 데이타를 모델 갱신에 적용할 수 없고 적절한 갱신 주기를 판단하기도 어렵다.

이 논문에서는 투플 기반 슬라이딩 윈도우 기법과 점진적인 다중 회귀분석을 사용한 스트림 데이타 예측 기법을 제안한다. 제안하는 스트림 데이타 예측 기법은 기존의 회귀분석을 이용하여 점진적으로 모델을 갱신하는 방법이다. 먼저 일차원 속성의 데이타를 회귀 모델에 적용하기 위해 차원 분열 기법을 사용하여 여러 개의 속성으로 분열하고 모델 갱신을 위해 입력되는 스트림 데이타 중 변화량이 큰 데이타를 탐색한다. 변화량이 큰 데이타를 탐색하기 위해 투플 기반 슬라이딩 윈도우 기법을 사용한다. 일정 크기의 윈도우를 설정하고 이전 시점의 윈도우 평균과 현재 시점의 윈도우 평균 차이를 통해 데이타를 탐색한다. 또한 최신 데이타로만 모델을 유지하기 위해 일정 크기의 큐를 사용한다. 슬라이딩 윈도우에 의해 탐색된 데이타는 큐에 저장이 된다. 큐에 새로 입력되는 데이타는 모델에 적용되고 가장 오래된 데이타는 큐와 모델에서 제거가 된다. 실세계에서는 비선형적인 형태의 데이타가 많이 수집이 되므로 선형 회귀분석 보다 다중 이항 회귀분석을 적용하였다. 그러나 다중 선형 회귀분석을 이용하여 선형적인 형태의 데이타에도 적용 가능하다. 제안하는 기법은 변화량이 큰 최근의 데이타를 이용해 모델을 갱신함으로써 에러율의 증가를 막아준다. 또한 모델을 갱신할 때 기존의 모델 정보와 입력되고 제거되는 데이타의 계산만을 수행함으로써 계산 비용을 절약한다. 모델 정보는 고정 크기의 행렬을 이용하여 유지되므로 데이타의 크기에 상관없이 공간 비용도 절약된다.

점진적인 모델 갱신 기법을 이용한 스트림 데이타 예측 기법의 타당성을 검토하기 위해 비선형적인 데이타와 선형적인 데이타를 통해 실험을 실시한다. 비선형적인 데이타에 일반적인 시계열 예측 기법인 이중 지수 평활법(Double Exponential Smoothing)과 차원 분열을 통한 Support Vector Regression(SVR), 다중 선형 회귀분석(Multiple Linear Regression), 다중 이항 회귀분석(Multiple Quadratic Regression), 제안하는 점진적인 다중 이항 회귀분석(Incremental Multiple Quadratic Regression)을 적용하였다. 선형적인 데이타에 SVR, 다중 선형 회귀분석, 제안하는 점진적인 다중 선형 회귀분석(Incremental Multiple Linear Regression)을 적용하였다.

이 논문의 구성은 다음과 같다. 먼저 제2장에서 관련 연구를 통해 기존의 연구와 이 논문에서 제안한 기법과의 차별성을 제시한다. 제3장에서는 제안하는 기법의 기초가 되는 다중 회귀분석에 대해 설명한다. 제4장에서는 스트림 데이타를 예측하기 위해 슬라이딩 윈도우 기법과 점진적인 모델 갱신 기법을 설명한다. 제5장에서는 예제 데이타를 이용하여 제안한 스트림 데이타 예측 기법의 정확도를 실험하고 평가하며, 마지막으로 제6장에서 연구결과를 요약한다.

2. 관련연구

이 장에서는 스트림 데이타 예측과 관련된 기존의 시계열 데이타 예측 기법을 살펴보고, 시간에 따라 데이타의 특성을 고려한 점진적인 모델링 기법을 살펴본다.

2.1 시계열 데이타 예측 기법

시계열이란 한 사건 또는 여러 사건에 대해 시간의 흐름에 따라 일정한 간격으로 이들을 관측하여 기록한 자료를 말한다[8]. 스트림 데이타 역시 시계열 데이타와 같이 시간의 흐름에 따라 일정한 간격으로 수집되는 데이타이다. 시계열 데이타의 예로 매일 변동하는 종합주가지수, 특정 소비재의 월별 판매량, 연도별 농작물의 생산량 등이 있다. 이러한 시계열은 어떠한 경제현상이나 자연현상에 관한 시간적 변화를 나타내는 역사적 계열이므로 어느 한 시점에서 관측된 시계열 자료는 그 이전까지의 자료들에 주로 의존하게 된다. 따라서 시계열 분석을 통한 예측에서는 관측된 과거의 자료들을 분석하여 법칙성을 발견해서 이를 모형화하여 추정하고, 이 추정된 모형을 사용하여 미래에 관측될 값들을 예측하게 된다.

기존의 시계열 예측 기법으로는 단순 이동 평균법, 지수 평활법, ARIMA 모형 등이 있다[8]. 단순 이동 평균법은 가장 최근의 m-기간 동안의 자료들 평균을 다음 시점의 예측 값으로 추정한다. 이 방법은 시간의 경과에 따라 평균의 변화가 크지 않을 경우에 적용 가능하다. 지수 평활법은 최근의 자료들에 대해 더 많은 가중치를 부여하는 방법이다. 따라서 빠르게 수집이 되는 데이타에는 적용하기 어렵다. ARIMA 모형은 추세가 있는 경우 이를 제거하여 정상적 데이타로 변환한 후 AR, MA, ARMA 모형 중에서 가장 적합한 모형을 선택할 수 있게 해준다. 그러나 ARIMA 모형은 새로운 데이타가 투입될 때 모형의 모수를 쉽게 변경할 수 있는 방법이 없고 만족스러운 모형을 개발하기 위해서는 많은 비용이 소모되는 단점이 있다. 시계열 데이타 예측은 [9-12]에 의해 연구되었다. [9]는 n개의 입력을 받고 1개의 출력을 내보내는 신경망(neural network) 기법이다. 신경망은 마켓 예측, 기상이나 네트워크 트래픽 예측 등과 같이 시계열 예측에 널리 사용된다. 그러나 신경망에서 목적함수를 최적화하는 것은 매우 어렵고 주어진 함수를 근사화하기 위해 매우 많은 은닉마디(hidden units)가 필요할 수 있다. 또한 회귀분석에 비해 입력 속성들이 출력 속성에 어떤 영향을 주는지 해석하기가 어려운 단점도 있다. [10]은 웨이블릿(wavelet) 기법과 비모수 회귀분석을 이용하여 짧은 시간에 수집되는 불규칙한 시계열 데이타를 예측한다. [11]은 Kalman Smoother를 이용하여 시계열 예측을 수행하고 [12]은 연관 규칙을 이용하여 시계열 데이타를 예측한다. 이 방법은 시계열 데이타에서 반복되는 패턴을 추출하여 예측을 수행한다.

기존의 시계열 예측 기법은 긴 주기를 갖고 즉각적인 계산이 필요 없는 데이타에 적합하다. 따라서 빠르게 데이타가 수집되고 시간에 따라 데이타의 특성이 변화되며 비선형적인 스트림 데이타에는 적합하지 않다.

2.2 점진적인 모델링 기법

시간 속성을 갖는 시계열 데이타에 대한 점진적인 모델링은 [13,14]에 의해 연구되었다. [13]은 시간에 따라 데이타의 특성이 변화하는 시계열 데이타를 위해 MUSCLES(MUlti-SequenCe LEast Squares)와 Selective MUSCLES를 제안하였다. 이 기법은 점진적인 알고리즘과 적은 비용의 공간 및 입출력 연산을 통해 무한히 긴 시퀀스를 다룰 수 있다. 그러나 이 기법은 결측 값이나 지연되는 값의 예측에만 사용되고, Chaotic 데이타와 같이 비선형적인 데이타의 예측은 어렵다. [14]는 순차적인 베이지안 진화 연산을 이용하여 예측을 수행한다. 이전 단계의 모델을 통해 예측을 수행하고 새로운 데이타가 주어지면 현재의 모델을 평가하여 더 좋은 모델을 생성하도록 하는 것이다. 점진적인 모델링은 사례 기반 학습(instance-based learning)과 유사하다고 할 수 있다. 사례 기반 학습은 [15-18]에 의해 조사되고 연구되었다. 사례 기반 예측은 [15]에 의해 소개되었고 예측을 위해 로컬 선형 회귀분석 형태의 기법을 사용하였다. 로컬 선형 회귀분석은 [16]에 의해 세부적으로 조사되었다. 이것은 데이타베이스의 크기가 증가되는 것은 제한하지 못한다. [17]은 메모리 기반의 LWR(Locally Weighted Learning)과 이전 데이타를 기억할 필요가 없는 점진적인 LWR을 논의 하였다. [18]은 제어 작업과 같이 연속적인 상태를 유지해야 하는 응용을 위해 데이타를 빠르고 안전하게 학습하는 HEDGER 알고리즘을 제안하였다. HEDGER 알고리즘은 LWR에 기반한 사례 기반 알고리즘이다.

모델의 재계산을 위해 이전의 데이타를 유지해야 할 경우 스트림 데이타의 제약 사항에 위배되고 실시간으로 입력되는 스트림 데이타에 대해 모델 갱신의 최적 주기를 판단하기 어려운 단점이 있다. 따라서 기존의 연구를 빠르게 입력되고 분포가 변화될 가능성이 있는 스트림 데이타에 적용하는 것은 적합하지 않다.

3. 다중 회귀분석

여러 분야에서 사용되고 있는 다차원 속성 데이타 예측을 위해서는 속성 x를 통해 속성 y를 예측할 수 있는 모델이 필요하다. 이 장에서는 연속형 데이타의 일반적인 예측 모델인 회귀분석에 대해 설명한다.

3.1 다중 선형 회귀분석

회귀분석은 주어진 자료를 통해 속성간의 함수 관계를 파악하여, 이 함수를 이용하여 입력 속성 값에 대응되는 출력 속성 값을 예측하는 분석 방법이다[19]. 입력 속성과

출력 속성간의 선형관계에 관한 분석을 선형 회귀분석이라 하고 입력 속성이 2개 이상일 때를 다중 회귀분석(MLR)이라고 한다. 따라서 입력 속성이 k개인 MLR (Multiple Linear Regression)은 식 (1)과 같이 정의된다.

$$y = b_0 + b_1 x_1 + \cdots + b_k x_k + \varepsilon \quad (1)$$

식 (1)과 같은 함수식에 의해 입력되는 속성 x의 값들을 통해서 미지의 값 y를 예측할 수 있다. 함수식의 계수인 b값은 오차를 최소로 하는 최소제곱법[19]에 의해 계산될 수 있고 식 (2)에 의해 수행 된다. 식 (2)의 (a)에 의해 속성 x와 y를 행렬로 변환하고, 식 (2)의 (b)에 의해 행렬 X의 전치행렬과 역행렬을 통해서 b값의 집합인 행렬 B가 계산된다.

(a) 속성 x와 속성 y의 행렬

$$X = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{k1} \\ 1 & x_{12} & x_{22} & \cdots & x_{k2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1n} & x_{2n} & \cdots & x_{kn} \end{bmatrix} \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

(b) b값은 전치행렬과 역행렬 계산에 의해 유도 (2)

$$B = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_k \end{bmatrix} = (X^T \times X)^{-1} \times X^T \times y$$

3.2 다중 다항 회귀분석

입력 속성 x의 변화에 따라서 출력 속성 y의 변화가 직선적인 관계를 가질 때 선형 회귀분석을 사용하였다. 그러나 실세계의 많은 데이타들은 선형적인 관계보다는 곡선이나 불규칙한 관계의 데이타 분포를 갖는다. 입력 속성과 출력 속성의 곡선적인 관계에 관한 분석을 다항 회귀분석이라고 한다. 특히 이차항에 대한 분석을 다중 이항 회귀분석(MQR)이라 하고 입력 속성이 2개인 MQR(Multiple Quadratic Regression)은 식 (3)과 같이 정의된다.

$$y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_1^2 + b_4 x_2^2 + b_5 x_1 x_2 + \varepsilon \quad (3)$$

MQR의 b값은 식 (4)의 (a)와 같이 각 속성을 이차 형태로 변환한 후에 MLR의 b값 계산법인 식 (2)의 절차에 의해서 계산된다.

(a) 각 속성을 이차 형태로 변환

$$x_1^2 = x_1 \times x_1 \quad x_2^2 = x_2 \times x_2 \quad x_{12} = x_1 \times x_2$$

(b) 변환된 값으로 행렬 X 생성

$$X = \begin{bmatrix} 1 & (x_1)_1 & (x_2)_1 & (x_1^2)_1 & (x_2^2)_1 & (x_{12})_1 \\ 1 & (x_1)_2 & (x_2)_2 & (x_1^2)_2 & (x_2^2)_2 & (x_{12})_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & (x_1)_n & (x_2)_n & (x_1^2)_n & (x_2^2)_n & (x_{12})_n \end{bmatrix} \quad (4)$$

또한 삼차항에 대한 분석을 다중 삼항 회귀분석이라 하고 입력 속성이 2개인 MCR(Multiple Cubic Regression)은 식 (5)와 같이 정의된다.

$$y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_1^3 + b_4 x_1^2 + b_5 x_1^2 x_2 + b_6 x_2^2 + b_7 x_1 x_2^2 + b_8 x_1 x_2 + b_9 x_2^3 + \varepsilon \quad (5)$$

MCR의 b값은 식 (6)의 (a)와 같이 각 속성을 삼차 형태로 변환한 후에 MLR의 b값 계산법인 식 (2)의 절차에 의해서 계산된다.

(a) 각 속성을 삼차 형태로 변환

$$x_1^3 = x_1 \times x_1 \times x_1 \quad x_1^2 = x_1 \times x_1$$

$$x_2^3 = x_2 \times x_2 \times x_2 \quad x_2^2 = x_2 \times x_2$$

$$x_{12^2} = x_1 \times x_2 \times x_2 \quad x_{1^2 2} = x_1 \times x_1 \times x_2 \quad x_{12} = x_1 \times x_2$$

(b) 변환된 값으로 행렬 X 생성

$$X = \begin{bmatrix} 1 & (x_1)_1 & (x_2)_1 & (x_1^3)_1 & (x_1^2)_1 & (x_{1^2 2})_1 & (x_2^2)_1 & (x_{12^2})_1 & (x_{12})_1 & (x_2^3)_1 \\ 1 & (x_1)_2 & (x_2)_2 & (x_1^3)_2 & (x_1^2)_2 & (x_{1^2 2})_2 & (x_2^2)_2 & (x_{12^2})_2 & (x_{12})_2 & (x_2^3)_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & (x_1)_n & (x_2)_n & (x_1^3)_n & (x_1^2)_n & (x_{1^2 2})_n & (x_2^2)_n & (x_{12^2})_n & (x_{12})_n & (x_2^3)_n \end{bmatrix} \quad (6)$$

4. 점진적 모델 갱신 기법

이 장에서는 3장에서 설명한 일반적인 다중 이항 회귀분석을 변형하여 스트림 데이타를 예측하기 위한 기법을 설명한다. 일차원 속성을 먼저 차원 분열하고 슬라이딩 윈도우 기법을 통해 갱신에 사용될 데이타를 탐색한 후 행렬을 이용하여 효율적으로 모델을 갱신한다.

4.1 차원 분열

일반적인 다중 회귀분석은 여러 개의 입력 속성을 통해 출력 속성을 예측하는 기법이다. 따라서 모델을 생성하기 위해서 입력 속성들과 출력 속성의 값이 필요하고 새로운 데이타를 예측하기 위해서 새로 입력되는 입력 속성들의 값이 필요하다. 이 논문에서 설명하는 스트림 데이타는 하나의 속성만을 갖는 일차원 속성이기 때문에 회귀분석에 적용하기 위해 적절한 변환이 필요하다. 따라서 차원 분열 기법을 사용하여 하나의 속성을 여러 개의 속성으로 분열한다. 분열하는 원리는 시간을 갖는 스트림 데이타이기 때문에 각 시점을 하나의 속성으로 간주하는 것이다.

그림 1은 입력 스트림 x를 시간 t를 통해 3개의 속성으로 분열한 것을 보여준다. 현재 시점(n)과 1시점 이전

입력 스트림 x : <n, n-1, ..., 2, 1, 0>

(n-1), 2시점 이전(n-2) 값이 하나의 투플이 되어 현재 시점의 분열된 값을 표현한다. 스트림 데이타는 시계열 데이타 예측과 같이 과거 데이타를 통해 미래를 예측하기 때문에 과거 데이타 x_n, x_{n-1}, x_{n-2} 등을 통해 미래 값

t	x_n	x_{n-1}	x_{n-2}
n	n	n-1	n-2
n-1	n-1	n-2	n-3
n-2	n-2	n-3	n-4
n-3	n-3	n-4	n-5
n-4	n-4	n-5	n-6
⋮	⋮	⋮	⋮

그림 1 차원 분열의 예

인 x_{n+1}을 예측한다. 따라서 입력 속성과 출력 속성간의 함수 관계가 아닌 과거 데이타와 미래 데이타의 함수 관계를 찾는 것이다. 식 (7)은 식 (3)의 다중 이항 회귀 모델을 차원 분열된 데이타를 적용하기 위해 변형한 것이다.

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_1^2 + b_4x_2^2 + b_5x_1x_2 + \varepsilon$$

$$\downarrow$$

$$y \rightarrow x_{n+1},\ x_1 \rightarrow x_n,\ x_2 \rightarrow x_{n-1}$$

$$\downarrow$$

$$x_{n+1} = b_0 + b_1x_n + b_2x_{n-1} + b_3x_n^2 + b_4x_{n-1}^2 + b_5x_nx_{n-1} + \varepsilon \quad (7)$$

분열을 할 차원의 수 L은 사전에 선택된다. 선택 방법은 L을 1부터 최대 값까지 반복하면서 기준에 만족하는 L을 선택한다. 선택 기준은 각 차원에 의해 변환된 데이타를 IMQR에 의해 학습시킬 때 최대 에러율이다. 따라서 에러율이 기준 값 이하인 L이 선택된다.

4.2 슬라이딩 윈도우 기법

스트림 데이타는 시간이 지남에 따라 데이타 분포가 변화될 가능성이 있기 때문에 점진적으로 모델을 갱신하여 예측 정확도를 유지해야 한다. 그러나 빠르게 수집이 되는 스트림 데이타 전체를 모델 갱신에 적용하면 데이타 입력 시간에 비해 처리 시간이 더 많이 소모될 수 있기 때문에 오버헤드가 발생할 수 있다. 따라서 투플 기반 슬라이딩 윈도우를 사용한 회귀 모델 갱신 기법을 제안한다.

슬라이딩 윈도우 기법은 일정한 크기의 윈도우를 유지하면서 현재 시점에 존재하는 데이타들의 평균을 계산하여 데이타를 모델 갱신에 사용할지 판단한다. 만약 바로 이전 시점의 윈도우 평균과 현재 시점의 윈도우 평균의 차가 기준치(ε)를 초과할 경우 현재 시점 윈도우에 마지막으로 입력된 데이타를 모델 갱신에 사용한다. 그림 2는 크기가 5인 윈도우에 스트림 데이타의 평균이 유지되는 것을 보여준다.

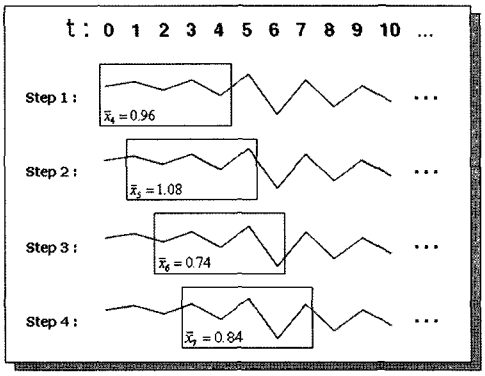


그림 2 크기가 5인 슬라이딩 윈도우

그림 2를 보면 스트림 데이타가 0시점부터 계속 입력이 되고 있다. Step 1은 현재 시점이 4가 되고 0시점부터 4시점까지의 데이타에 대해 평균 $\overline{x_4}$ (0.96)를 계산한다. Step 2는 현재 시점인 5시점의 데이타가 입력되고 0시점의 데이타가 제거되면서 1시점부터 5시점까지의 평균 $\overline{x_5}$ (1.08)를 계산한다. Step 3의 6시점에서 입력된 데이타에 의한 $\overline{x_6}$ (0.74)의 평균과 이전 윈도우의 평균 $\overline{x_5}$ (1.08)의 차이가 0.34로 ε (0.3) 보다 크기 때문에 6시점의 데이타는 모델 갱신에 적용이 된다. 반복적으로 현재 시점의 데이타가 입력되면 윈도우 크기만큼의 이전 시점 데이타를 제거하면서 회귀 모델 갱신에 사용될 데이타를 결정한다. 따라서 현재 시점의 윈도우 평균 $\overline{x_i}$와 이전 시점의 윈도우 평균 $\overline{x}_{i-1}$의 차이가 임의로 설정된 기준치 ε 보다 클 경우($|\overline{x_i} - \overline{x}_{i-1}| > \varepsilon$) 현재 시점에 입력된 i 번째 데이타는 모델 갱신에 적용된다.

윈도우 크기가 k라면 평균은 이전 k시점까지의 데이타 분포를 나타내준다. 만약 윈도우에 입력된 데이타와 제거된 데이타의 값이 비슷하다면 전체적인 분포에는 영향을 주지 않는다. 그러나 두 데이타 값이 상반된 경우라면 전체 평균에 큰 영향을 준다. 따라서 새로 들어온 데이타에 의해 이전 시점의 평균보다 변화량이 크다면 그 데이타는 회귀 모델에 반영해야 올바른 회귀 모델이 유지될 수 있다.

모델 갱신을 위해 선택된 데이타는 일정 크기의 큐에 저장이 된다. 큐에 저장이 된 데이타만 이용하여 모델을 갱신하는 것이다. 모델 갱신을 위해 새로운 데이타가 탐색이 되어 큐에 저장이 되면 새로운 데이타는 모델에 적용이 되고 큐의 마지막 데이타는 큐와 모델에서 제거가 된다.

4.3 점진적 회귀 모델 갱신

기존의 회귀분석에서는 초기 한 번의 학습으로 모델

을 생성하였다. 스트림 데이타는 시간이 지남에 따라 데이타의 특성이 변경될 수 있기 때문에 기존의 회귀분석을 그대로 적용하게 되면 시간이 지날수록 예측 정확도가 낮아질 것이다. 또한 많은 양의 데이타가 빠르게 입력되기 때문에 모델은 최근의 데이타로만 유지될 필요가 있다.

이 논문에서는 입력되는 스트림 데이타를 반영하여 회귀 모델을 갱신하는 기법인 점진적인 다중 이항 회귀분석(IMQR)을 제안한다. 만약 입력되는 스트림 데이타가 선형적인 경우에는 같은 원리로 점진적인 다중 선형 회귀분석(IMLR)을 사용할 수 있다. 이 기법은 입력된 데이타를 슬라이딩 윈도우 기법을 통해 모델 갱신 적용 여부를 판단하고 선택된 데이타로 모델을 갱신하는 기법이다. 데이타 분포의 변화가 생겼을 때 모델을 갱신하기 때문에 높은 정확도가 나올 것이라 예상된다. 제안하는 IMQR 기법은 적은 비용으로 식 (3)을 갱신하여 함수식을 최신으로 유지하는 기법이다. 함수식의 갱신을 위해 이전의 모든 데이타를 유지할 필요 없이 최소 정보만을 유지하면 된다. 행렬 X의 전치행렬과 행렬 X의 곱인 $X^{\mathrm{T}}X$와 행렬 X의 전치행렬과 행렬 y의 곱인 $X^{\mathrm{T}}y$만 유지하면 된다. 따라서 항상 같은 크기의 최소 정보만을 유지하므로 공간 복잡도를 줄일 수 있는 장점이 있다. 식 (8)의 (a)는 입력 속성이 2개인 초기 데이타를 행렬 X, 행렬 X의 전치행렬 X^{T}로, (b)는 행렬 y로 나타내고, 식 (9)는 행렬 X^{T}와 X의 곱, 행렬 X^{T} 와 y의 곱의 결과를 보여준다. 입력 속성이 2개일 때는 입력되는 데이타의 개수와는 상관없이 항상 6행 6열의 $X^{\mathrm{T}}X$와 6행 1열의 $X^{\mathrm{T}}y$만 유지하면 된다.

(a) 행렬 X^{T}, X

$$X^T = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ (x_1)_1 & (x_1)_2 & \cdots & (x_1)_n \\ (x_2)_1 & (x_2)_2 & \cdots & (x_2)_n \\ (x_1^2)_1 & (x_1^2)_2 & \cdots & (x_1^2)_n \\ (x_2^2)_1 & (x_2^2)_2 & \cdots & (x_2^2)_n \\ (x_{12})_1 & (x_{12})_2 & \cdots & (x_{12})_n \end{bmatrix}$$

$$X = \begin{bmatrix} 1 & (x_1)_1 & (x_2)_1 & (x_1^2)_1 & (x_2^2)_1 & (x_{12})_1 \\ 1 & (x_1)_2 & (x_2)_2 & (x_1^2)_2 & (x_2^2)_2 & (x_{12})_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & (x_1)_n & (x_2)_n & (x_1^2)_n & (x_2^2)_n & (x_{12})_n \end{bmatrix}$$

(b) 행렬 y

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (8)$$

• $X^{\mathrm{T}} \times X$, $X^{\mathrm{T}} \times y$

$$X^T X = \begin{bmatrix} z_{11} & z_{12} & z_{13} & z_{14} & z_{15} & z_{16} \\ z_{21} & z_{22} & z_{23} & z_{24} & z_{25} & z_{26} \\ z_{31} & z_{32} & z_{33} & z_{34} & z_{35} & z_{36} \\ z_{41} & z_{42} & z_{43} & z_{44} & z_{45} & z_{46} \\ z_{51} & z_{52} & z_{53} & z_{54} & z_{55} & z_{56} \\ z_{61} & z_{62} & z_{63} & z_{64} & z_{65} & z_{66} \end{bmatrix}$$

$$X^T y = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{bmatrix} \quad (9)$$

제안하는 기법은 식 (9)와 같은 고정 크기의 행렬에 큐에 입력된 데이타를 적용하고 큐의 마지막 데이타를 삭제하는 원리이다. 그림 3은 큐를 이용하여 행렬에 데이타가 적용되고 삭제되는 예제를 보여준다. 큐의 크기는 11이고 모델은 총 10개의 데이타로 유지가 된다. 13이라는 새로운 데이타가 입력되면 모델에 적용을 하고, 마지막 데이타인 11을 모델에서 제거한다. 따라서 모델은 항상 변화량이 큰 최근의 10개 데이타로만 유지가 되는 것이다.

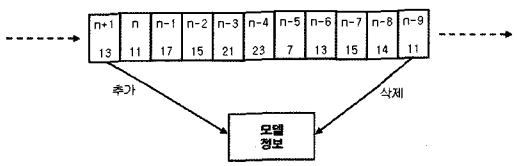


그림 3 크기가 11인 큐의 예

식 10은 큐에 입력된 데이타를 모델에 적용하는 것을 보여주고 식 11은 큐의 마지막 데이타가 모델에서 제거되는 것을 보여준다.

• 입력된 데이타를 $X^{\mathrm{T}}X$에 적용

$$x_{n+1} = \{x_1, x_2\} \rightarrow x_{n+1} = \{1, x_1, x_2, x_1^2, x_2^2, x_{12}\}$$

$$X^T X = \begin{bmatrix} z_{11}+1\times 1 & z_{12}+1\times x_1 & \cdots & z_{16}+1\times x_{12} \\ z_{21}+x_1\times 1 & z_{22}+x_1\times x_1 & \cdots & z_{26}+x_1\times x_{12} \\ \vdots & \vdots & \vdots & \vdots \\ z_{61}+x_{12}\times 1 & z_{62}+x_{12}\times x_1 & \cdots & z_{66}+x_{12}\times x_{12} \end{bmatrix}$$

• 입력된 데이타를 $X^{\mathrm{T}}y$에 적용

$$x_{n+1} = \{x_1, x_2\} \rightarrow x_{n+1} = \{1, x_1, x_2, x_1^2, x_2^2, x_{12}\},\ y_{n+1} = y$$

$$X^T y = \begin{bmatrix} c_1+1\times y \\ c_2+x_1\times y \\ c_3+x_2\times y \\ c_4+x_1^2\times y \\ c_5+x_2^2\times y \\ c_6+x_{12}\times y \end{bmatrix} \quad (10)$$

• 큐의 마지막 데이타를 X^TX에서 제거

$$x_{n-9}=\{x_1,x_2\} \rightarrow x_{n-9}=\{1,x_1,x_2,x_1^2,x_2^2,x_{12}\}$$

$$X^TX=\begin{bmatrix} z_{11}-1\times1 & z_{12}-1\times x_1 & \cdots & z_{16}-1\times x_{12} \\ z_{21}-x_1\times1 & z_{22}-x_1\times x_1 & \cdots & z_{26}-x_1\times x_{12} \\ \vdots & \vdots & \vdots & \vdots \\ z_{61}-x_{12}\times1 & z_{62}-x_{12}\times x_1 & \cdots & z_{66}-x_{12}\times x_{12} \end{bmatrix}$$

• 큐의 마지막 데이타를 X^Ty에서 제거

$$x_{n-9}=\{x_1,x_2\} \rightarrow x_{n-9}=\{1,x_1,x_2,x_1^2,x_2^2,x_{12}\},\ y_{n-9}=y$$

$$X^Ty=\begin{bmatrix} c_1-1\times y \\ c_2-x_1\times y \\ c_3-x_2\times y \\ c_4-x_1^2\times y \\ c_5-x_2^2\times y \\ c_6-x_{12}\times y \end{bmatrix} \quad (11)$$

회귀 모델 갱신에 적용할 새로운 데이타가 입력되면 입력된 데이타와 큐의 마지막 데이타를 먼저 이차항으로 변환하고 행렬에 유지되고 있는 기존의 정보와의 계산을 통해 회귀 모델의 계수인 b값 계산을 위한 정보를 갱신한다. 따라서 X^TX 행렬과 X^Ty 행렬의 갱신 공식은 식 12와 같다.

$$X^TX=\sum_{i=1}^{k}\sum_{j=1}^{k}((X^TX[i,j]+(x[i]\times x[j])-(x_{n-q}[i]\times x_{n-q}[j]))$$

$$X^Ty=\sum_{i=1}^{k}((X^Ty[i]+(x[i]\times y)-(x_{n-q}[i])\times y_{n-q})) \quad (12)$$

b값의 집합인 행렬 B를 구하기 위한 역행렬 계산은 행렬이 대칭이고 양의 정수로 되어 있을 때 편리한 Cholesky LU 분해법[20]을 사용하였다. 행렬 X^TX는 행렬 X의 전치행렬과 행렬 X의 곱이기 때문에 결과는 대칭이고 양의 정수로 구성된다. 간단한 역행렬 계산 절차는 식 (13)과 같다.

$$X^T\times X=M: \text{대칭 양정치 행렬}$$

$$M=L\times L^T : \text{L은 하삼각 행렬}$$

$$(L\times L^T)\times x=b,\ L^T\times x=y$$

$$L\times y=b \text{로 } y \text{ 계산}$$

$$L^T\times x=y \text{로 역행렬인 } x \text{ 계산} \quad (13)$$

알고리즘 1은 새로운 스트림 데이타가 입력되면 데이타 값이 기존의 행렬 X^TX, X^Ty와 함께 계산되어 새로운 b값의 집합인 행렬 B가 계산되는 절차를 보여주는 알고리즘이다.

알고리즘 1. 점진적인 다중 이항 회귀분석(IMQR) 알고리즘

Begin

Input : 스트림 데이타 (xadd[], yadd, xout[], yout)
Output : 회귀 모델의 계수(b)
Step 1: 행렬 X^TX에 입력 데이타 xadd[]와 삭제 데이타 xout[]을 적용

```
For i = 1 to r Do  // r = X^TX의 행 길이
  For j = 1 to c DO  // c = X^TX의 열 길이
    matrix[i][j] += xadd[i] * xadd[j];  matrix[i][j]
    = xout[i] * xout[j];
  End of inner For
End of outer For
```

Step 2 : 행렬 X^Ty에 입력 데이타 xadd[], yadd와 삭제 데이타 xout[], yout을 적용

```
For i = 1 to r Do  // r = X^Ty의 행 길이
  ymatrix[i] += xadd[i] * yadd;  ymatrix[i] = xout[i] *
  yout;
End of For
```

Step 3 : X^TX의 역행렬 계산(Cholesky LU 분해법)

```
inverse[][] = LU(matrix[][]);
```

Step 4 : 모델식의 계수 b계산

```
For i = 1 to r Do  // r = inverse의 행 길이
  For j = 1 to c DO  // c = inverse의 열 길이
    beta[i][j] += inverse[i][j] * ymatrix[j];
  End of inner For
End of outer For
```

End

5. 실험 및 평가

5.1 실험 방법

제안한 기법의 구현환경은 CPU 2.00GHz의 인텔 펜티엄4와 512MB 램을 사용하였다. 플랫폼은 Windows XP Professional을 사용하였고 프로그래밍 구현환경은 JDK 1.5를 사용하였다. 스트림 데이타 예측 기법을 평가하기 위해 비선형적인 데이타와 선형적인 데이타 2개를 사용하였다.

첫 번째로 점진적인 다중 이항 회귀분석(IMQR)의 비교 실험을 위해 [21]에서 사용된 불규칙한 특성을 갖는 1385개 샘플의 Mackey-Glass Time Series[22] 데이타를 사용하였다. 실험 데이타를 이중 지수 평활법(DES), Support Vector Regression(SVR), 다중 선형 회귀분석(MLR), 다중 이항 회귀분석(MQR)과 제안한 기법인 IMQR에 적용하여 에러율을 측정하였고 IMQR 기법의 슬라이딩 윈도우 크기는 10으로, ε(기준값)은 0.04로, 큐의 크기는 100으로, 차원 분열의 수는 3으로 설정하였다. 초기 모델을 위한 학습 데이타는 100개의 샘플을 사용하였고, 검증 데이타는 1285개의 샘플을 사용하였다. SVR 실험을 위해서 mySVM[23]을 사용하였다.

두 번째로 선형적인 태풍 데이타[24]를 이용하였다. 태풍 데이타는 49개의 샘플을 갖는 2005년 4호 태풍 NESAT이다. 각 샘플은 초기 발생 시점으로부터 6시간 간격으로 측정되었고 위도, 경도, 중심기압을 예측함으로써 태풍의 이동 경로를 예측한다. 태풍 데이타의 위도, 경도, 중심기압을 MLR, SVR과 제안한 기법인 점진적

인 다중 선형 회귀분석(IMLR)에 적용하여 에러율을 측정하였다. 초기 모델을 위한 학습 데이타는 10개의 샘플을 사용하였고, 검증 데이타는 39개의 샘플을 사용하였다. 큐의 크기는 10으로, 차원 분열의 수는 3으로 설정하였다. 수집 주기가 길고 선형적인 데이타의 실험을 위해 모든 데이타를 사용하여 모델 갱신을 실시하였다.

두 실험의 에러율 측정은 식 (14)와 같이 RME와 RMSE를 사용하였다. 식 (14)에서 y_i는 실제 값이고, y^*_i는 예측 값이다. 따라서 예측 값과 실제 값 사이의 오차를 계산하는 것이다.

$$RME = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - y^*_i}{y_i}\right| \quad RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y^*_i)^2} \qquad (14)$$

5.2 실험 결과

불규칙한 특성을 갖는 Mackey-Glass Time Series 데이타 실험 결과 표 1과 그림 4에서와 같이 제안한 기법인 IMQR 기법이 다른 기법에 우수하였다. 1시점 예측 결과를 보면 DES를 제외한 기법들에 비해 RME가 평균 3.1%, RMSE가 0.041 정도 우수하였다.

표 1의 1시점 예측 실험 결과를 보면 DES의 RME는 18.01%로 다른 기법에 비해 높게 나타났다. DES는 최근의 과거 데이타에 더 많은 가중치를 부여하여 미래 값을 예측을 하기 때문에 짧은 시간에 많이 수집되는 데이타에는 적합하지 않다. SVR과 MLR은 각각 8.7%와 9.56%의 상대 에러율을 보였다. 실험 데이타는 선형적인 관계가 아닌 실세계를 반영한 불규칙한 데이타를 사용했기 때문에 선형적인 관계를 고려한 SVR과 MLR 기법이 MQR과 IMQR 기법보다 에러율이 높게 나타남을 알 수 있다.

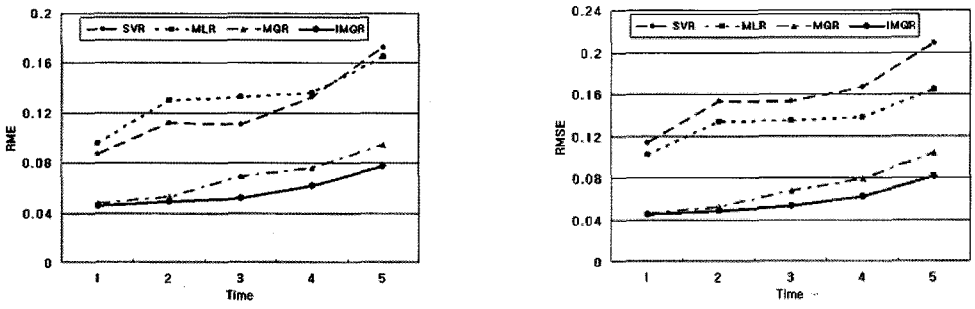
그림 4의 스트림 데이타 예측 RME 결과 그래프를 보면 MQR과 IMQR 기법의 RME는 1시점 예측일 때는 비슷하지만 예측 시점이 멀어질수록 IMQR 기법이 우수함을 알 수 있다. MQR 기법은 한 번의 학습으로 생성된 모델에 새로운 데이타를 적용했기 때문에 변화되는 데이타 특성을 반영하지 못한다. 만약 더 많은 데이타가 입력이 되고 더 먼 시점을 예측한다면 MQR 기법의 에러율은 점진적으로 증가할 것으로 예상할 수 있다. IMQR 기법에 1285개의 샘플이 검증 데이타로 사용되었지만 모델 갱신을 위해 ε 값을 초과하는 경우는 41번으로 모델 갱신은 41번 이루어졌다. IMQR 기법은 41번의 모델 갱신을 통해 다른 기법보다 우수한 결과를 얻었다. 표 1과 그림 4의 결과와 같이 RMSE 결과도 RME 결과와 유사한 정확도를 보였다.

선형적이면서 시간에 따라 분포가 변화되는 태풍 데이타 실험 결과 표 2와 같이 제안한 기법인 IMLR이 가장 우수하였다.

실험 결과 위도의 경우 MLR과 SVR은 각각 8.26%

표 1 Mackey-Glass Time Series 데이타 실험 결과

구분		DES	SVR	MLR	MQR	IMQR
1시점	RME	0.1801	0.087	0.0956	0.0477	0.0461
	RMSE	0.1925	0.1131	0.1016	0.0459	0.0455
2시점	RME	0.3361	0.1117	0.1301	0.0532	0.0489
	RMSE	0.3192	0.1526	0.1332	0.0524	0.0485
3시점	RME	0.3354	0.1102	0.1327	0.0692	0.0521
	RMSE	0.3190	0.1525	0.1342	0.0682	0.0532
4시점	RME	0.3267	0.1328	0.1357	0.0758	0.0617
	RMSE	0.2931	0.1664	0.1377	0.0788	0.0622
5시점	RME	0.4129	0.1725	0.1647	0.0946	0.0776
	RMSE	0.3699	0.2082	0.1643	0.1031	0.0818



(a) RME 측정 그래프 (b) RMSE 측정 그래프

그림 4 스트림 데이타 예측 실험 그래프

표 2 태풍 NESAT 데이타 실험 결과

구분		NESAT		
		위도	경도	중심기압
MLR	RME	0.0826	0.0106	0.0041
	RMSE	22.9628	15.545	5.2285
SVR	RME	0.0127	0.0048	0.0039
	RMSE	4.2361	8.1920	4.8677
IMLR	RME	0.0102	0.0026	0.0034
	RMSE	3.4808	6.3365	4.1698

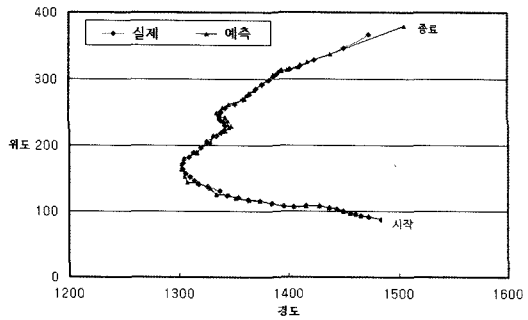


그림 5 NESAT 이동경로 VS 예측경로

와 1.27%의 상대 에러율을 보였다. MLR은 한번의 학습으로 생성된 모델에 새로운 데이타를 적용했기 때문에 다른 기법보다 에러율이 높음을 알 수 있다. 또한 일반적으로 성능이 좋은 SVR보다 점진적으로 모델을 갱신하는 IMLR 기법이 1.02%의 에러율로 더 좋은 성능을 보였다. 새로운 데이타에 대해 점진적으로 모델을 갱신해줌으로써 예측 정확도를 높였다. 표 2의 결과와 같이 경도, 중심기압의 결과도 유사한 정확도를 보였다. 그림 5는 태풍 NESAT의 실제 이동 경로와 제안한 기법인 IMLR을 통해 예측한 경로를 보여준다. 그림을 통해 거의 정확하게 예측을 하였음을 알 수 있다.

6. 결 론

최근 센서와 네트워크 기술의 발달로 인해 시간과 공간의 제약성 없이 데이타를 실시간으로 수집하고 분석하여 의사결정에 반영할 수 있게 되었다. 센서 네트워크로부터 수집이 되는 스트림 데이타는 시간에 따라 데이타의 분포가 변화될 수 있고 짧은 시간에 많은 데이타가 연속적으로 수집이 되는 특징을 가지고 있다. 또한 스트림 데이타는 시간 속성을 갖기 때문에 시계열 데이타로 간주할 수 있다. 그러나 스트림 데이타가 갖는 특징 때문에 기존의 시계열 데이타를 이용하여 예측을 수행할 수 없다. 따라서 이 논문은 시간에 따른 데이타 분포의 변화를 반영하여 미래 예측을 수행하는 점진적인 다중 이항 회귀분석(IMQR)을 제안하였다.

IMQR은 다중 회귀 모델에 데이타를 입력하기 위해 차원 분열을 먼저 실시하고, 단계적 모델 갱신을 위해 투플 기반 슬라이딩 윈도우 기법을 사용하였다. 모델 갱신에 적용될 데이타가 탐색이 되면 고정 크기의 큐에 저장이 되고 큐의 처음과 마지막 데이타와 기존의 모델 정보를 이용하여 회귀 모델을 갱신한다. 이 기법은 행렬에 최소 정보만을 유지하여 모델 갱신을 수행함으로써 공간 복잡도를 줄였다. 또한 데이타 분포를 모델에 반영할 수 있기 때문에 에러율의 증가를 최소화 시킨다. 다른 기법과의 비교 실험 결과 가장 낮은 에러율을 보여 제안한 기법의 타당함을 증명하였다. 제안한 기법은 비선형적이며 시간에 따라 데이타의 분포가 변화될 수 있는 도메인에 적용 가능하며 이차항 계산 부분을 제외하면 선형적인 데이타에도 적용 가능하다.

참 고 문 헌

[1] M. J. Franklin and S. R. Jeffery etc., "Design Considerations for High Fan-In Systems: The HiFi Approach," Conference on Innovative Data Systems Research, pp. 290-304, 2005.

[2] A. Manjeshwar and D. P. Agrawal, "TEEN: A routing protocol for enhanced efficiency in wireless sensor networks," International Workshop Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, pp. 2009-2015, 2001.

[3] R. C. Olover and K. Smettem etc., "Field Testing a Wireless Sensor Network for Reactive Environmental Monitoring," Intelligent Sensors, Sensor Networks and Information Processing, pp. 7-12, 2004.

[4] B. Xu. and O. Wolfson, "Time-Series Prediction with Application to Traffic and Moving Objects Databases," ACM Workshop on Data Engineering for Wireless and Mobile Access, pp. 56-60, 2003.

[5] B. Babcock, S. Babu, and M. Datar, et al., "Models and Issues in Data Stream Systems," Invited paper in Proc. of PODS, 2002.

[6] L. Golab, M. Tamer Ozsu, "Issues in Data Stream Management," In SIGMOD Record, Volume 32, Number 2, 2003.

[7] S. Babu, J. Widom, "Continuous queries over data streams," In ACM SIGMOD Record, pp. 109-120, 2001.

[8] 오광우, 이성덕, 이우리, "시계열 분석 입문 및 응용", 탐진, 2000.

[9] N. Davey, S. P. Hunt, and R. J. Frank, "Time Series Prediction and Neural Networks," In Journal of Intelligent and Robotic Systems, 2001.

[10] X. Hao, D. XU, "Time Series Prediction based on Non-Parametric Regression and Wavelet-Fractal," In Proc. of ISCP04, pp. 388-391, 2004.

[11] S. Sarkka, A. Vehtari, and J. Lampinen, "Time Series Prediction by Kalman Smoother with Cross-Validated Noise Density," In Proc. of IJCNN, pp. 1653-1658, 2004.
[12] O. B. Yaik, C. H. Yong, and F. Haron, "Time Series Prediction using Adaptive Association Rules," In Proc. of DFMA05, pp. 310-314, 2005.
[13] B.-K. Yi, ND Sidiropoulos, and T. Johnson, et al, "Online Data Mining for Co-Evolving Time Sequences," In Proc. of ICDE2000, pp. 13-22, 2000.
[14] 조동연, 장병탁, "순차적 베이지안 진화 연산을 이용한 시계열 예측", 한국정보과학회 추계 학술발표논문집, Vol.27, No.2, pp. 311-313, 2000.
[15] D. Kibler, D. W. Aha, and M. Albert, "Instance-based prediction of real-valued attributes," In Computational Intelligence, pp. 51-57, 1989.
[16] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning," In Artificial Intelligence Review, pp. 11-73, 1997.
[17] S. Schaal, C. G. Atkeson, and S. Vijayakumar, "Real-Time Robot Learning With Locally Weighted Statistical Learning," In Proc. of the IEEE International Conference on Robotics and Automation, pp. 288-293, 2000.
[18] W. D. Smart, L. P. Kaelbling, "Practical reinforcement learning in continuous spaces," In Proc. of the 17th International Conference on Machine Learning, pp. 903-910, 2000.
[19] 박성현, "회귀분석", 민영사, 1997.
[20] S. C. Chapra, R. P. Canale, "Numerical Method for Engineers, Third Edition," McGraw-Hill Korea, 1999.
[21] G. W. Flake, S. Lawrence, "Efficient SVM Regression Training with SMO," In Machine Learning, pp. 271-290, 2002.
[22] C. C. Chang, C. J. Lin, LIBSVM, http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/, 2001.
[23] S. Ruping, mySVM, Computer Science Dep. AI Unit Univ. of Dortmund, 2000.
[24] Typhoon Research Center, http://www.typhoon.or.kr, 2001.

김 성 현
2005년 충북대학교 통계학과(이학사). 2007년 충북대학교 전자계산학과(공학석사) 2006년~현재 SPSS Korea 근무. 관심분야는 데이타 마이닝, 스트림 데이타 예측 및 분류

김 룡
2000년 연변과학기술대학교 전자전산학과(공학사). 2003년 충북대학교 대학원 전자계산학전공(이학석사). 2007년 충북대학교 대학원 전자계산학전공(이학박사). 2007년~현재 한국전자통신연구원 연구원. 관심분야는 스트림 데이타마이닝, 시공간 데이타 마이닝, 시공간 데이타베이스, 스트림 데이타 처리, 센서 데이타 처리

류 근 호
1988년 연세대학교 대학원 전산전공(공학박사). 1976년~1986년 육군군수지원사 전산실(ROTC 장교), 한국전자통신연구원(연구원), 한국방송통신대 전산학과(조교수) 근무. 1989년~1991년 Univ. of Arizona Research Staff(TempIS 연구원, Temporal DB). 1986년~현재 충북대학교 전기전자 및 컴퓨터공학부 교수. 관심분야는 시간 데이타베이스, 시공간 데이타베이스, Temporal GIS, 지식기반 정보검색 시스템, 유비쿼터스컴퓨팅 및 스트림데이타처리, 데이타마이닝, 데이타베이스 보안, 바이오인포메틱스