韓國 컴퓨터情報學會 論文誌
第10卷 第2號, 2005. 5.

2005-10-2-4-1

범주형 시퀀스 데이터의 K-Nearest Neighbor 알고리즘

오 승 준*

A K-Nearest Neighbor Algorithm for Categorical Sequence Data

Seung-Joon Oh *

요 약

최근에는 단백질 시퀀스, 소매점 거래 데이터, 웹 로그 등과 같은 상업적이거나 과학적인 데이터의 폭발적인 증가를 볼 수 있다. 이런 데이터들은 순서적인 면을 가지고 있는 시퀀스 데이터들이다. 본 논문에서는 이런 시퀀스 데이터들을 분류하는 문제를 다룬다. 분류 기법으로는 의사결정 나무나 베이지안 분류기, K-NN 방법 등 여러 종류가 있는데, 본 연구에서는 K-NN 방법을 이용하여 시퀀스들을 분류한다. 또한, 시퀀스들간의 유사도를 구하기 위한 새로운 계산 방법과 효율적인 계산 방법도 제안한다.

Abstract

TRecently, there has been enormous growth in the amount of commercial and scientific data, such as protein sequences, retail transactions, and web-logs. Such datasets consist of sequence data that have an inherent sequential nature. In this paper, we study how to classify these sequence datasets. There are several kinds techniques for data classification such as decision tree induction, Bayesian classification and K-NN etc. In our approach, we use a K-NN algorithm for classifying sequences. In addition, we propose a new similairty measure to compute the similarity between two sequences and an efficient method for measuring similarity.

▸ Keyword : 데이터 마이닝(Data Mining), 분류(Classification), 시퀀스(Sequences)

• 제1저자 : 오승준
• 접수일 : 2005.03.18, 심사완료일 : 2005.05.12
* 경기공업대학 산업경영시스템과 교수

Ⅰ. 서 론

최근에는 상업적이거나 과학적인 데이터의 폭발적인 증가를 볼 수 있다. 이들 중 웹 로그, DNA나 단백질 시퀀스, 소매점 거래 데이터 등과 같은 분야의 데이터들은 순서적인 면을 가지고 있는 시퀀스 데이터(또는 시퀀스)들이다. 즉, 데이터의 항목들간에 순서가 존재하는 것이다. 예를 들어, 두 개의 시퀀스들이 동일한 항목들로 이루어졌더라도 항목들간의 순서가 다르면 서로 다른 시퀀스들로 여긴다.

항목들간에 순서가 존재하는 시퀀스들을 분류하는 것은 많은 면에서 유용하다. 예를 들면, 웹 사용자들의 사이트 방문기록을 보관한 웹 로그 파일들을 이용하여 새로운 웹 사용자들을 분류하는 것은 웹 사용자들의 행동을 미리 예측하는데 도움을 준다. 또한, DNA나 단백질 시퀀스들을 분류하는 것은 유전자나 단백질의 기능들에 대한 중요한 통찰력을 얻는데 사용될 수 있다[1].

여러 가지 분류 기법들 중에서 K-nearest neighbor(K-NN) 분류 기법은 패턴 분류 문제 영역에 있어서 잘 알려진 분류 기법중 하나이다. 특히, 중요한 특성중의 하나가 데이터들간의 거리 또는 유사도만이 필요하다는 것이다. 따라서, 본 연구에서는 K-NN 방법으로 시퀀스 데이터들을 분류하는 문제를 다룬다.

시퀀스들 사이의 유사도를 계산하는 방법에는 edit distance 방법[2,3,4]과 seqeuce alignment 방법[2,5]이 있으며, 수학적인 관점에서 보면, edit distance 방법과 sequence alignment 방법은 동일하다[2]. edit distance 방법은 유사도 계산시 시퀀스 전체를 고려하기 때문에, 때로는 중요한 특성을 나타내는 서브 시퀀스들을 고려하지 못하며, 다수의 edit operations 조합이 가능하다. sequence alignment 방법은 scoring scheme에 의존적이며, 항목 값들의 종류가 적은 경우에만 효율적이다. 그래서 본 연구에서는 이들 기존 방법들의 단점을 개선한 새로운 유사도 계산 방법을 제안하고, 제안하는 유사도를 K-NN 방법에 적용하여 시퀀스들을 분류한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존연구를 살펴보고, 3장에서는 시퀀스들간의 유사도를 계산하는 새로운 방법을 제안한다. 4장에서는 3장에서 제안한 유사도를 이용한 K-NN 방법에 대하여 설명한다. 다음으로 5장에서는 splice 데이터 셋으로 제안하는 방법의 성능을 보이고, 6장에서는 결론을 제시한다.

Ⅱ. 기존 연구

분류 기법에는 의사결정 나무나 베이지안 분류기, K-NN을 이용한 방법, 사례기반 추론, 러프 셋을 이용한 방법 등 여러 종류가 있다[6]. 이중에서 K-NN을 이용한 방법은 문서[7], 침입탐지 시스템[8][13][14], 스트링[9], 생물학 분야의 시퀀스[1] 분류 등 여러 분야에서 활용이 되고 있다.

또한, K-NN 방법의 개선 및 확장에 대한 연구도 많이 이루어져왔다. 예를 들면, Y. Wu et al.[10]은 K-NN 방법의 분류 속도를 증가시키기 위해 템플릿 압축과 사전 처리 단계를 이용하였으며, M. Khan et al. [11]은 P-tree라는 자료구조를 사용하여 공간 데이터 스트림들을 효율적으로 분류하는 방법을 제안하였다.

K-NN 방법을 이용하여 시퀀스들을 분류한 연구로는 M. Deshpande et al.[1]과 A. Juan et al. [9]이 있는데, 두 논문 모두 시퀀스들간의 유사도를 계산하는데, edit distance 방법을 이용하였다. 따라서, 본 연구에서는 시퀀스들간의 유사도를 계산하는데 있어, 새로운 유사도 방법을 제안하고, 이것을 이용하여 시퀀스들을 분류한다.

Ⅲ. 시퀀스들간의 유사도

3.1 유사도 측정

K-NN 방법을 이용하여 시퀀스들을 분류하기 위해서는, 시퀀스들 간의 유사도(혹은 거리)를 측정해야 한다. 일반적으로 두 시퀀스들간의 유사도는 공통 항목이 많을수록, 또한 항목들의 순서가 동일할수록 높다고 할 수 있다. 따라서, 이 두 가지 요소를 동시에 고려하기 위해서는 두 시퀀스 사이에 동일 서브셋들이 얼마나 많이 존재하느냐를 고려한다. 본 연구에서는 동일 서브셋들을 찾기 위해 순서를 가지는

두 항목 쌍들을 이용한다. 즉, 두 시퀀스들 사이에 동일 항목 쌍들이 많을수록 유사도가 높게 나오는 성질을 이용한다.

〔예 3.1〕 두 시퀀스 S1 = 〈A B C D〉, S2 = 〈A C D E〉 가 있다. S1의 두 항목 쌍들의 모임은 (AB, AC, AD, BC, BD, CD)이고 S2 의 두 항목 쌍들의 모임은 (AC, AD, AE, CD, CE, DE)이다. S1, S2에 동일한 두 항목 쌍들이 많을수록 유사도는 높다. ((AC, AD, CD)가 공통 두 항목 쌍들임)

3.2 유사도 계산 방법

데이터베이스 D는 시퀀스들의 집합이고, 시퀀스 S는 n개의 항목들의 모임이며 〈 x1 x2 ··· xi ··· xj ··· xn 〉으로 표시하고, 여기서 xi 는 범주형 값을 가지는 항목이다. S의 크기는 S에 있는 항목들의 개수이며, | S | 로 나타낸다. 시퀀스 S에서 순서를 가지는 2개의 항목들로 구성된 xi xj (i<j)를 시퀀스 요소 ek 라고 하며, ek 들의 모임을 E = (e1, e2 , ... , ek, ...)라 한다. E의 크기는 E에 있는 요소들의 개수이며, | E | 로 나타낸다.

〔예 3.2〕 시퀀스 S = 〈 A B C E 〉에서 | S | = 4이고, 시퀀스 요소들의 모임은 E = (AB, AC, AE, BC, BE, CE)이며, | E | = 6이다.

시퀀스내의 항목들뿐만 아니라 항목들간의 순서도 고려를 해서 식(3.1)과 같이 유사도 계산 방법을 제안한다.

〔정의 3.1〕 두 시퀀스 S1 = 〈 a1 a2 ··· an 〉과 S2 = 〈 b1 b2 ··· bm 〉 의 시퀀스 요소들의 모임을 각각 E1 = (ea1, ea2, ··· , eai , ···), E2 = (eb1, eb2, ··· , ebj , ···)라고 하면, S1, S2 의 유사도 sim(S1, S2)는 다음과 같이 정의한다.

$$sim(S_1, S_2) = \frac{|E_1 \cap E_2|}{\frac{|E_1| + |E_2|}{2}} \quad \cdots\cdots (3.1)$$

여기서, $|E_1 \cap E_2|$ 는 E1과 E2의 공통 요소들의 개수이며, E1과 E2사이에 공통 항목들이 많을수록 유사도는 높고, 이 값을 $(|E_1| + |E_2|)/2$로 나누는 것은 유사도를 0과 1사이의 값을 갖도록 하기 위해서이다.

〔예 3.3〕 두 시퀀스 S1 = 〈 A B D A 〉, S2 = 〈 A C D A C 〉에서 시퀀스 요소들의 모임은 각각 E1 = (AB, AD, AA, BD, BA, DA) 과 E2 = (AC, AD, AA, AC, CD, CA, CC, DA, DC, AC) 이며, | E1 | = 6, | E2 | = 10, E1∩E2 = (AD, AA, DA), | E1∩E2 | =3이다. 따라서, 두 시퀀스의 유사도 sim(S1, S2)는 3/8이다.

유사도 측정을 위하여 3항목 이상의 시퀀스 요소를 사용할 수도 있다. 그러나, 본 연구에서 2항목 시퀀스 요소들만 고려하는 이유는 첫째, 3항목 이상으로 구성된 시퀀스 요소들도 세분화해 보면 2항목 시퀀스 요소들로 모두 표현할 수 있기 때문이다. 둘째, 2항목 시퀀스 요소를 고려하는 것이 3개 이상의 항목들로 시퀀스 요소들을 구성하는 것보다 계산량에 있어 훨씬 효율적이기 때문이다. 예를 들어, 시퀀스 S = 〈x1 x2 ··· xi ··· xn〉의 2항목 시퀀스 요소들의 수는 nC2이지만, 3항목 시퀀스 요소들의 수는 nC3이 된다. n>5 에서는 2항목 시퀀스 요소들의 개수보다 3항목 시퀀스 요소들의 개수가 커지기 때문에 공통된 시퀀스 요소들을 찾는데 있어 계산량은 더욱 늘어나게 된다.

다음과 같은 세 시퀀스 S1 = 〈 A B C X Y Z 〉, S2 = 〈 X Y Z A B C 〉, S3 = 〈 X Y Z P Q R 〉가 있다. sim(S1,S2), sim(S1,S3)를 edit distance 방법을 이용하여 구하면, sim(S1,S2)=6, sim(S1,S3)=6이다. 그러나, 제안하는 유사도 방법을 이용하면, sim(S1,S2)=6/15, sim(S1,S3)=3/15이다. 즉, 일부 서브시퀀스들의 자리가 바뀔 경우 (블럭 operations), 제안하는 유사도 방법은 edit distance 방법보다 효율적으로 유사도를 측정할 수 있다. 또한, 제안하는 유사도 방법은 시퀀스 요소를 이용하여 두 시퀀스 사이의 유사도를 계산하므로, edit distance 방법처럼 다수의 edit operation 조합이 생성되지 않는다.

3.3 효율적인 유사도 계산 방법

이 절에서는 유사도 계산을 효율적으로 수행하기 위한 방법에 대하여 설명한다. 식(3.1)에서 보면, 시퀀스들간의 유사도를 계산하는데 두 시퀀스들간의 공통된 시퀀스 요소들의 개수를 구하는 것이 중요하다. 따라서, 공통된 시퀀스 요소들의 개수를 효율적으로 구하는 〔성질 3.1〕을 이용하면 시퀀스들간의 유사도를 효율적으로 계산할 수 있다.

〔성질 3.1〕 두 시퀀스 S1= 〈 a1 … ai … ck … an 〉, S2= 〈 b1 … bj … cl … bm 〉가 있다. ck, cl 항목들은 S1, S2 에 공통으로 있는 항목들이며, ai, bj 항목들은 각각 S1, S2 에만 있는 항목들이다. ck 항목들로만 이루어진 시퀀스를 S3라 하고, cl 항목들로만 이루어진 시퀀스를 S4라 하자. 또한, E1, E2, E3, E4 를 각각 S1, S2, S3, S4 의 시퀀스 요소들의 모임이라고 하면, S1, S2 의 유사도 sim(S1, S2)는 다음과 같이 계산된다.

$$sim(S_1, S_2) = \frac{|E_3 \cap E_4|}{\frac{|E_1| + |E_2|}{2}} \quad \cdots\cdots (3.2)$$

〔증 명〕 식(3.1)에 의하여,

$$sim(S_1, S_2) = \frac{|E_1 \cap E_2|}{\frac{|E_1| + |E_2|}{2}}$$ 이다.

여기서 |E1∩E2| = |E3∩E4| 이다. 왜냐하면, |E1∩E2| 는 S1과 S2에 공통으로 존재하는 시퀀스 요소들의 개수이기 때문에 S1, S2에서 서로 자신들에게만 존재하는 항목들을 제외한 S3, S4 의 공통 시퀀스 요소들의 개수인 |E3∩E4| 를 구하여도 마찬가지의 결과를 얻는다.

그러므로,

$$sim(S_1, S_2) = \frac{|E_3 \cap E_4|}{\frac{|E_1| + |E_2|}{2}}$$

〔예 3.4〕 두 시퀀스 S1 = 〈A B C F Z A〉, S2 = 〈A F C H〉의 유사도 sim(S1, S2)를 계산해 보자. S1, S2 로부터 직접 유사도를 계산하는 경우에는 S1의 시퀀스 요소들인 (AB, AC, AF, AZ, AA, BC, BF, BZ, BA, CF, CZ, CA, FZ, FA, ZA)를 S2의 시퀀스 요소들인 (AF, AC, AH, FC, FH, CH)와 비교하여 유사도를 계산한다.

그러나, 〔성질 3.1〕을 사용하기 위해, S1, S2의 공통 항목들로만 구성된 S3=〈A C F A〉과 S4=〈A F C〉를 구한다. S3는 다음과 같이 생성을 한다. S1의 모든 항목들을 차례대로 S2의 항목들과 비교하여 동일 항목이 존재하면 S3에 추가시키면서, S3을 생성한다. 마찬가지 방법으로 S2의 모든 항목들을 차례대로 S1의 항목들과 비교하여 S4를 생성한다. 그후, S3의 시퀀스 요소들인 (AC, AF, AA, CF, CA, FA)와 S4의 시퀀스 요소들인 (AF, AC, FC)를 이용하여 유사도를 계산한다. 그러므로, S1과 S2로부터 직접 유사도를 계산하는 것보다 훨씬 효율적으로 유사도를 계산할 수 있다.

3.4 제안하는 유사도 계산방법의 타당성

본 연구에서 제안하는 유사도 계산 방법 (식(3.1))의 타당성을 검토한다. 시퀀스 S = 〈 x1 x2 … xi xj … xn 〉에서 xj를 xi의 di,j-다음 항목이라고 하며, xj와 xi (또는 xi와 xj) 사이를 간격이 di,j라고 한다. 여기서 di,j= |j-i| 이다. 또한, 두 시퀀스 S1, S2에서 공통이 되는 항목들의 모임을 CI(S1,S2)라고 하며, |CI(S1,S2)| 는 CI(S1,S2)의 요소들의 개수이다.

〔예 3.5〕 두 시퀀스 S1 = 〈 A B C D 〉, S2 = 〈 A B E C 〉가 있다. 시퀀스 S1에서 항목 D를 항목 B의 2-다음 항목이라고 하며, 항목 B와 D사이의 간격은 2이다. CI(S1,S2) = (A, B, C)이며, |CI(S1,S2)| = 3 이다.

〔성질 3.2〕 공통항목의 개수가 크면 유사도가 높다 시퀀스 S1, S2, S3에서 |S2| = |S3| 이고, |CI(S1,S2)| ≥ |CI(S1,S3)| 이면, sim(S1,S2) ≥ sim(S1,S3) 이다. (여기서, CI(S1,S2)들의 순서와 CI(S1,S3)들의 순서는 S1의 항목들 순서와 동일)

〔증 명〕 공통 시퀀스 요소들은 순서를 갖는 공통 2항목 요소들로 구성이 되기 때문에, 공통 항목들의 개수가 커질수록 공통 시퀀스 요소들의 개수는 증가한다. 그런데, 조건에서 |CI(S1,S2)| ≥ |CI(S1,S3)| 이므로, |E1∩E2| ≥ |E1∩E3| 이고, sim(S1,S2) ≥ sim(S1,S3) 이다.

〔예 3.6〕 S1=〈A B E C D〉, S2=〈A B C P D〉, S3=〈A B C P Q〉에서 S1와 S2는 4개의 공통 항목을 가지며, S1와 S3는 3개의 공통 항목을 가지고 있다. 식(3.1)에 의하여 유사도를 구하면, sim(S1,S2) = 6/10, sim(S1,S3) = 3/10 이 되어, sim(S1,S2) 〉 sim(S1,S3)이 된다.

〔성질 3.3〕 항목들의 간격이 크면 유사도가 낮다 시퀀스 S1, S2, S3에서 S1의 한 항목을 간격이 i(j)인 항목 다음으로 이동시켜 생성된 시퀀스를 S2(S3)라 하자 (i<j). 그러면, sim(S1,S2) > sim(S1,S3) 이다.

〔증 명〕 |S1| = |S2| = |S3|, |E1∩E2| = |S1| - i 이고, |E1∩E3| = |S1| - j 이다.
그런데, i<j 이므로 |E1∩E2| > |E1∩E3| 이고, sim(S1,S2) > sim(S1,S3)이다.

〔예 3.7〕 S1=〈A B C D E〉, S2=〈A C B D E〉, S3=〈A C D E B〉에서 S2는 S1의 항목 B를 간격이 1인 항목 C 다음으로 이동시킨 시퀀스이고, S3는 S1의 항목 B를 간격이 3인 항목 E 다음으로 이동시킨 시퀀스이다. 식(3.1)에 의하여 유사도를 구하면, sim(S1,S2) = 9/10, sim(S1,S3) = 7/10 이 되어, sim(S1,S2) > sim(S1,S3)이 된다.

Ⅳ. K-NN 알고리즘

K-NN을 이용한 분류 기법은 알고리즘의 단순함과 비교적 낮은 오분류율로 인해 패턴 인식 분야에서 널리 사용되고 있는 기법중의 하나이다. 본 연구의 K-NN 알고리즘 단계는 (그림 1)과 같다.

단계 1. 분류하고자 하는 시퀀스 Si에 대하여 유사도가 가장 높은 순서대로 K 이웃 시퀀스들을 구한다.
단계 2. K개의 이웃 시퀀스들이 가장 많이 속해 있는 클래스 레이블을 구한다.
단계 3. 단계 2에서 구한 클래스 레이블로 시퀀스 Si를 분류한다.
단계 4. 분류하고자 하는 시퀀스가 남아있으면, 단계 1로 가고, 아니면 알고리즘을 끝낸다.

그림 1. K-NN 알고리즘
Fig 1. K-NN algorithm

단계 1에서는 시퀀스 Si에 대하여, 트레이닝 셋에서 유사도가 가장 높은 순서대로 K 이웃 시퀀스들을 구한다. 이때, 시퀀스들간의 유사도는 본 연구에서 제안하는 유사도를 사용한다. 단계 2에서는 단계 1에서 구한 K개의 이웃 시퀀스들이 가장 많이 속해 있는 클래스 레이블을 구한다. 단계 3에서는 단계 2에서 구한 클래스 레이블로 시퀀스 Si를 분류한다. 단계 4에서는 분류하고자 하는 시퀀스가 남아 있는지를 검사하여, 미분류 시퀀스가 남아 있으면 단계 1로 가고, 그렇지 않으면 알고리즘을 끝낸다.

〔예 4.1〕 6개의 시퀀스들로 이루어진 트레이닝 셋이 (그림 2)와 같다.

S1 = 〈 A B C J 〉,S2 = 〈 A C J K 〉,
S3 = 〈 A B J 〉, S4 = 〈 G J K I 〉,
S5 = 〈 H I J K 〉, S6 = 〈 G H I 〉

그림 2. 트레이닝 셋의 시퀀스들
Fig 2. Sequences of training set

여기서, S1, S2, S3는 클래스 레이블이 C1이고, S4, S5, S6는 클래스 레이블이 C2이다. (그림 1)의 알고리즘(K=3)으로, S7 = 〈 A B C K 〉의 클래스 레이블을 구해보자. 단계 1에서 S7과 유사도가 가장 큰 세 개의 이웃 시퀀스들을 구하면, 각각 S1, S2, S3이다. 단계 2에서 S1, S2, S3의 클래스 레이블을 구하면 모두 C1이다. 단계 3에서, 단계 2에서 찾은 클래스 레이블 C1을 S7의 클래스 레이블로 정한다.

Ⅴ. 실험결과

본 연구에서 제안하는 방법의 성능을 평가하기 위해, splice 데이터 셋으로 실험을 하였다. 본 실험은 인텔 2.4GHz 사양의 펜티엄 Ⅳ 컴퓨터에서 C++ 언어로 코딩을 하여 수행하였다.

splice 데이터셋은 UCI KDD 아카이브에 포함되어 있는 데이터셋이다[12]. 이 데이터셋은 60개의 항목을 가진 뉴클레오타이드(nucleotide) 시퀀스들을 포함하고 있으며, 각각의 시퀀스들은 엑손/인트론 경계 (exon/intron, EI라 부름)나 인트론/엑손 경계 (intron/exon, IE라 부름)에 속하는 클래스 레이블을 가진다. EI에 속하는 시퀀스들이 767개이며, IE에 속하는 시퀀스들이 768개이다.

본 연구에서는 splice 데이터 셋을 〈표 1〉과 같이 두 개의 데이터 셋으로 만든 후, 실험을 하였다. DS1은 splice 데이터 셋의 40%가 트레이닝 셋으로, 나머지 60%가 테스트 셋으로 구성이 되며, DS2는 splice 데이터 셋의 60%와 40%가 각각 트레이닝 셋과 테스트 셋으로 구성이 된다.

표 1. 실험용 데이터 셋의 구성
Table 1. Composition of experimental dataset
(단위: 트랜잭션 수)

데이터 셋		EI에 속하는 시퀀스	IE에 속하는 시퀀스	합계
DS1	트레이닝 셋	307	307	614
	테스트 셋	460	461	921
DS2	트레이닝 셋	460	460	920
	테스트 셋	307	308	615

두 개의 데이터 셋 DS1과 DS2를 본 연구에서 제안하는 유사도를 이용하여 K-NN방법으로 분류를 수행하였다. 그 후, 오분류된 시퀀스들의 개수를 구하였으며, 실험 결과는 (그림 3)과 (그림 4)에 있다.

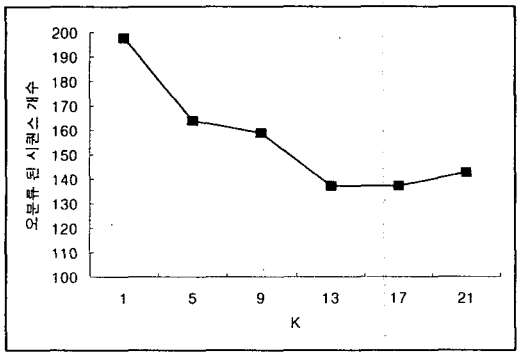

그림 3. DS1에 대한 실험결과
Figure 3.. Result for DS1

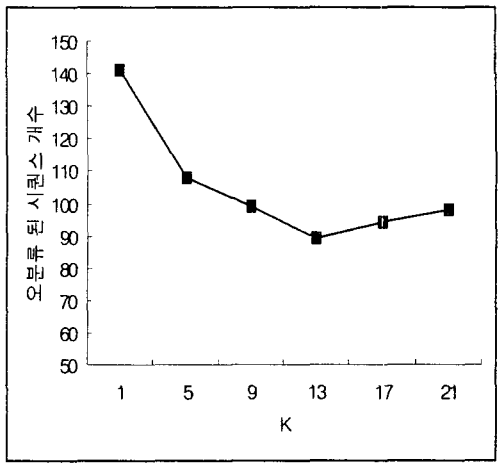

그림 4. DS2에 대한 실험결과
Figure 4.. Result for DS2

(그림 3)과 (그림 4)의 실험 결과에서 보면, K값이 1에서 13까지 증가할수록 오분류 시퀀스들의 개수가 줄어들고, 그 이후는 증가함을 알 수 있다. DS1과 DS2 모두 K값이 13일 때 오분류 시퀀스들의 개수가 가장 적었다.

Ⅵ. 결론

본 논문에서는 범주형 항목들이 순서를 가지는 시퀀스들의 분류 문제를 연구하였다. 본 문제를 풀기 위하여 새로운 유사도 계산 방법을 제안하였다. 본 연구에서 제안하는 시퀀스들간의 유사도는 순서를 가지는 두 항목 쌍들이 비교 대상의 두 시퀀스들 사이에 얼마나 많이 포함되어 있느냐에 따라 계산이 된다. 또한, 유사도 계산과정을 효율적으로 수행할 수 있는 유사도 계산 방법과 제안하는 유사도 계산방법을 K-NN 알고리즘에 이용하여 시퀀스들을 분류하였다.

참고문헌

[1] M. Deshpande and G. Karypis, "Evalutation of Techniques for Classifying Biological Sequences", PAKDD 2002, Taiwan, 2002.

[2] D. Gusfield, Algorithm on Strings, Trees, and Sequences, Press Syndicate of the University of Cambridge, New York, 1997.

[3] D. S. Hirschberg, Pattern Matching Algorithms, Oxford University Press, 1997.

[4] P. Moen, Attribute, Event Sequence, and Event Type Similarity Notions for Data Mining, PhD Thesis, Univ. of Helsinki, Dept. of Com. Sci., 2000.

[5] K. Charter, J. Schaeffer and D. Szafron, "Sequence Alignment using FastLSA", Proc. 2000 Int. Conf. Math. and Eng. Tech. in Medicine and Biological Sci., Las Vegas, Nevada, pp. 239-245, 2000.

[6] J. Han and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2001.

[7] E. -H. Han, G. Karypis and Vipin Kumar, "Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification", PAKDD 2001, Hong Kong, 2001.

[8] Y. Liao and V. R. Vermuri, "Use of K-Nearest Neighbor Classifier for Intrusion Detection, Computers & Security", Vol. 21, No. 5, pp. 439-448, 2002.

[9] A. Juan and E. Vidal, "On the Use of Normalized Edit Distance and an Efficient k-NN Search Technique (k-AESA) for Fast and Accurate String Classification", Int'l Conf. on Pattern Recognition, Spain, 2000.

[10] Y. Wu and K. Ianakiev, V. Govindaraju, "Improved k-Nearest Neighbor Classification, Pattern Recognition", Vol. 35, pp. 2311-2318, 2002.

[11] M. Khan, Q. Ding and William Perrizo, "K-Nearest Neighbor Classification on Spatial Data Streams Using P-Trees", PAKDD 2002, Taiwan, 2002.

[12] C. L. Blake and C. J. Merz, UCI Repository of Machine Learning Databases, 1998.

[13] 최인수, 차홍준, "대규모 네트워크를 위한 침입탐지결정 모듈 설계," 컴퓨터정보학회 논문지, 제7권, 제2호, 2002.

[14] 김강, 전종식, "보안정책 기반 침입탐지 시스템 모델 설계", 컴퓨터정보학회 논문지, 제8권, 제4호, 2003.

저 자 소 개

오 승 준

2004년 8월 한양대학교 산업공학과, 공학박사

2005~현재 경기공업대학 산업경영 시스템과 교수