논문 13-38B-07-01

한국통신학회논문지 '13-07 Vol.38B No.07
http://dx.doi.org/10.7840/kics.2013.38B.7.501

퍼지를 이용한 클라우드 기반의 소셜 네트워크 서비스 계층적 시각화

박 선◆, 김 용 일°, 이 성 로*

Hierarchical Visualization of Cloud-Based Social Network Service Using Fuzzy

Sun Park◆, Yong-Il Kim°, Seong Ro Lee*

요 약

현재 대부분의 소셜 네트워크 서비스에 대한 시각화방법들은 네트워크 자료를 시각화하여 표현하는 것에만 중점을 두고 있으며, 기하급수적으로 증가하는 소셜 네트워크의 빅데이터 처리에 대한 계산량 및 효율적인 처리속도는 전혀 고려하지 않고 있다. 본 논문은 소셜 네트워크의 사용자 노드 간의 계층 관계를 사용자 중심으로 시각화하는 클라우드 기반의 방법을 제안한다. 제안방법은 퍼지를 이용하여 소셜 네트워크 노드의 계층 관계를 표현함으로써 사용자의 사회관계를 직관적으로 이해할 수 있으며, 소셜 네트워크에서의 사용자들의 중심 역할 관계를 쉽게 파악할 수 있다. 또한 클라우드 기반의 하둡(hadoop)과 하이브(hive)를 이용하여 시각화 알고리즘을 분산병렬 처리함으로써 소셜 네트워크의 빅데이터를 신속히 처리할 수 있다.

Key Words : SNS(social network service), fuzzy, cloud, visualization, hierarchy

ABSTRACT

Recently, the visualization method of social network service have been only focusing on presentation of visualizing network data, which the methods do not consider an efficient processing speed and computational complexity for increasing at the ratio of arithmetical of a big data regarding social networks. This paper proposes a cloud based on visualization method to visualize a user focused hierarchy relationship between user's nodes on social network. The proposed method can intuitionally understand the user's social relationship since the method uses fuzzy to represent a hierarchical relationship of user nodes of social network. It also can easily identify a key role relationship of users on social network. In addition, the method uses hadoop and hive based on cloud for distributed parallel processing of visualization algorithm, which it can expedite the big data of social network.

Ⅰ. 서 론

클라우드 컴퓨팅의 사전적 의미는 정보가 인터넷상에 영구적으로 저장되고, 인터넷에 접속할 수 있는 클라이언트가 정보를 일시적으로 보관하여 활용하는 컴퓨터 환경을 의미한다. 이는 이용자의 모든

※ 이 논문은 2012년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2009-0093828)
※ 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(NIPA-2012-H0301-12-2005)
◆ 주저자 : 목포대학교 정보산업연구소, sunpark@mokpo.ac.kr, 정회원
° 교신저자 : 호남대학교 인터넷콘텐츠학과, yikim@honam.ac.kr, 정회원
* 목포대학교 정보전자공학과, srlee@mokpo.ac.kr, 정회원
논문번호 : KICS2013-04-166, 접수일자 : 2013년 4월 8일, 최종논문접수일자 : 2013년 7월 1일

정보를 인터넷 상의 서버에 저장하고, 이 정보를 각종 IT(internet technology) 기기를 통하여 언제 어디서든 접근하여 이용하는 것이다[1]. 그러나 클라우드 컴퓨팅에서 가장 중요한 것은 실제 업무를 수행하고, 데이터를 분석하고 사용자 요청에 비즈니스 로직을 수행하는 애플리케이션이다[2].

소셜 네트워크 서비스(SNS, social network service)는 사람들 사이의 사회적 관계나 사회적 네트워크를 반영 또는 구성하는 것을 목적으로 하는 사이트, 플랫폼 등의 온라인 서비스이다. 정보통신과 스마트 단말기의 발전으로 현실의 사회적 네트워크 관계들이 온라인상의 사회적 관계형성으로 신속히 이동하고 있다. 또한 현실에서는 제한적이던 사회적 관계들이 온라인상에 형성되면서 시간 및 지역, 계층, 국가 등의 시공간의 제약을 극복하며 다양한 계층의 목적에 맞추어 형성되고 있다. 특히 사회관계 중요한 공동체나 중심적인 역할을 수행하는 사용자를 네트워크상에서 그래프로 표현하는 시각화방법이 소셜 네트워크의 중요한 분석방법으로 많은 선호를 받고 있다. 즉, 소셜 네트워크의 시각화 방법을 이용하여서 소셜 네트워크상의 중요한 공동체나 중심적인 역할을 수행하는 사용자의 검색은 다양한 분야의 기초분석 자료로 활용할 수 있다[3-5].

현재 소셜 네트워크 시각화를 위한 연구의 대표적인 접근방법으로는 노드 링크(NL, node-link) 접근방법[3], 행렬 그래프(MAT, matrix graph) 접근방법[4], 노드 링크와 행렬 그래프의 혼합형 접근방법(hybrid of NL and MAT)[5, 6]이 주로 연구되고 있으며, 이외에도 다양한 기법들을 기존방법들에 적용하여 성능을 향상시키는 확장형 접근방법들이 있다[7-9]. 노드 링크를 이용한 방법의 경우 네트워크의 전체 구조를 시각화하는데 유용한 방법이다. 그러나 시각화되는 노드들이 서로 겹치고 에지들이 서로 엇갈리는 문제를 가지고 있다. 즉, 노드 링크를 이용하여 표현하는 사용자간의 관계가 많아질수록 사용자의 관계를 파악할 수 없는 문제가 있다. 이러한 문제를 해결하기위해서는 샘플링, 필터링, 군집과 같은 후처리를 통하여 필요한 부분만을 정제하여 해결할 수 있는 다양한 방법들이 제안되었으나, 역시 시각화되는 결과가 이해하기 어렵거나 비용이 많이 드는 문제를 가지고 있다[3,5,6].

행렬 그래프 기반방법은 노드 링크 기반방법의 시각화 결과에 대한 가독성 문제를 해결하기 위해서 제안된 방법이다. 행렬 그래프 기반방법은 노드 링크 기반방법에 비해서 쉽게 노드를 파악할 수 있으며, 노드들 간의 경로도 역시 쉽게 파악할 수 있다. 그러나 행렬 형태로 표현되는 노들이 행렬 상에 희소행렬(sparse)로 표시되어 많은 공간을 사용하기 때문에 공간에 대한 낭비가 발생하며, 노드 링크 기반방법과 마찬가지로 경로 탐색이 어려운 문제를 가지고 있다. 또한 행렬 그래프로 표현되는 노드들이 이해하기 힘든 문제를 가지고 있다[3,4].

노드 링크와 행렬 그래프의 혼합형 방법으로는 대표적으로 MatLink[5]와 MatTrix[6]가 있다. MatLink 방법은 노드 링크 방법과 행렬 그래프 방법의 경로를 해결하기 위해서 제안된 방법이다. 이 방법은 노드 링크 방법의 에지를 행렬 그래프 방법에 겹쳐서 행렬 그래프 방법의 성능을 향상시키고 있다. 그러나 이 방법 역시 행렬에 연결된 링크들이 복합하게 구성되어 있기 때문에 이해하기 어려운 단점을 가지고 있다[3,5,6]. MatTrix 방법은 MatLink 방법의 이해하기 어려운 문제점을 해결하기 위해서 제안된 방법이다. 이 방법은 행렬 그래프를 더 작은 여러 개의 행렬 그래프로 분해해서 링크를 연결한다. 그러나 이 방법 역시 행렬 그래프를 기반으로 하고 있기 때문에 행렬 상에 나타나는 노드의 관계를 이해하기 어렵다[4]. 본 논문의 저자들은 퍼지 연관 곱과 상관관계 행렬을 이용하여 네트워크의 노드와 링크를 트리의 계층 형으로 표현하여 시각화하는 방법을 제안하였다. 이 방법은 소셜 네트워크 시각화 방법에 대한 가독성 문제를 해결하였으나, 네트워크의 규모가 커질수록 계산 량이 많아지는 문제를 가지고 있다[8,9].

소셜 네트워크의 시각화를 위해서는 크게 두 가지 방향에 대하여 생각해 보아야 한다. 첫 번째는 소셜 네트워크의 자료를 효율적으로 시각화하는 알고리즘에 관한 것이다. 즉, 시각화되어 표현되는 자료들이 가독성이 높아 사용자에 의해서 직관적으로 이해할 수 있도록 하는 효율적인 시각화 알고리즘을 연구해야 한다. 두 번째로 고려해야 하는 것은 기하급수적으로 늘어나는 소셜 네트워크 자료를 신속히 처리할 수 있는 방법을 고려해야 한다. 현재 기존 대부분의 시각화방법들은 소셜 네트워크 자료를 시각화로 나타내는 방법만을 중점적으로 연구고 있으며, 기하급수적으로 증가하는 소셜 네트워크의 빅데이터 처리에 대한 계산량 및 시간은 전혀 고려하지 않고 있다. 즉, 소셜 네트워크의 빅데이터를 신속히 계산하여 시각화 표현에 적용할 수 있는 저렴한 비용의 효율적인 분산병렬처리 방법을 연구해

야 한다. 요즘은 컴퓨팅 자원의 발전에 의해 단일 컴퓨터를 이용하더라도 충분히 일정 규모 이상의 자료를 처리할 수 있다. 그러나 큰 시스템 하나를 만드는 것은 작은 시스템 여러 대를 구매하는 것과 비교하면 터무니없이 비싸다. 또한 인터넷의 발전으로 저장되는 자료들이 대량으로 증가하면서 단일처리 방법을 이용하여 대용량 빅데이터를 처리하는 것이 어려운 일이다. 이 때문에 큰 데이터를 효율적으로 접근하는 방법으로 클라우드 기반의 분산병렬 컴퓨팅 방법이 저렴한 비용과 효율적인 처리방법으로 많이 연구되어 활용되고 있다. 즉, 분산병렬처리 방법은 단일처리 방법의 계산을 분산하여 계산하기 때문에 단일 알고리즘의 계산문제가 근원적으로 바뀌게 되었으며, 이 때문에 단일처리 알고리즘을 분산병렬처리 알고리즘으로 재설계하는 것이 새로운 연구 분야로 재 각광받고 있다.

본 논문은 소셜 네트워크 시각화 방법의 두 가지 요구사항을 만족시킬 수 있도록 소셜 네트워크의 사용자 관계를 퍼지를 사용하여 사용자관계를 계층적으로 시각화하여 표현하는 새로운 클라우드 기반의 방법을 제안한다. 제안방법은 퍼지의 관계 곱에 의해 계산된 내부관계와 네트워크 노드들 간의 연결 정보에 의한 외부관계를 이용하여서 소셜 네트워크 관계를 사용자중심의 계층적으로 표현하여 시각화한다. 또한 제안된 방법은 클라우드 기반의 분산병렬 처리를 위하여 하둡(hadoop)[2]을 이용하여 데이터를 분산 저장 및 처리하며, 하이브(hive)[2]를 이용하여 하둡에 분산저장된 자료를 병렬처리로 분석 및 시각화하여 처리속도를 향상시킨다. 마지막으로 분산병렬처리된 시각화 계산 결과는 JSON(javascript object notation)으로 저장되고 D3[10]를 이용하여 웹브라우저에 계층적 그래프로 시각화되어 인터넷을 통하여 시각화 결과를 접근할 수 있다. 하둡은 분산 파일 시스템(distribution file system)과 분산 컴퓨팅을 위한 맵리듀스(MapReduce)를 포함하여 개발된 분산병렬처리 시스템이다[2]. 하이브는 페이스북이 개발한 하둡 기반의 데이터웨어하우징 시스템으로 SQL과 매우 유사한 HiveQL이라는 쿼리를 제공한다[2]. D3는 마이크 보스탁이 만든 자바스크립트 라이브러리로 데이터 집합의 문맥 안에 HTML(hyper text markup language), SVG(scalable vector graphics), Canvas 같은 웹 페이지 요소를 데이터에 따라 보여주고, 삭제하며, 편집할 수 있도록 지원한다[10].

Ⅱ. 퍼지 이론

퍼지집합은 보통집합이 0과 1로 명백하게 드러나는 것과는 다르게, 많은 개수의 멤버십 정도를 고려하여 기존 집합론보다 더 넓은 범위를 가진다. 멤버십의 개념은 전체집합 X의 원소 x에서 전체집합 Y의 두 원소 중의 하나로 사영이다. 퍼지 관계를 보통관계의 확장으로 서로 다른 곱 공간에서 퍼지 관계를 곱을 통해 서로 결합되며 퍼지 관계 곱(fuzzy relational product)에 의해 생성된다[11].

퍼지 관계 곱은 크리스프 함의 연산자 (crisp implication operator)를 확장하여 퍼지에 적용한 것으로서, 크리스프 함의 연산자는 {0, 1}×{0, 1}→{0, 1}로 정의되는데 반해, 퍼지 함의 연산자는 [0, 1]×[0, 1]→[0, 1]로서 단위 구간의 다치 논리로 확장된 것이다. 퍼지 함의 연산자의 종류는 무수히 많으며 본 논문에서는 대표적인 Kleene-Diense 퍼지 함의 연산자를 사용하며 예는 다음 식(1)과 같다[11].

$$a \rightarrow b = (1-a) \vee b = max(1-a, b), \quad a=0 \sim 1,\ b=0 \sim 1 \quad (1)$$

집합이론에서 "A⊆B"는 "∀x , x∈A → xB "와 같고 "A∈℘(B)"와도 같다. 여기서 ℘(B)는 B의 멱집합 (power set) 이다. 따라서 퍼지 집합에서의 "A∈B인 정도"는 A∈℘(B) 인 정도이므로 μ℘(B)A로서 나타낼 수 있으며 다음과 같이 정의된다.

(정의1) 퍼지 함의 연산자 →와 크리스프 전체집합 U의 퍼지 집합 B가 주어진 상태에서 B의 퍼지 멱집합의 멤버십 함수 μ℘B는 다음과 같이 주어진다.

$$\mu_{\wp B}A = \bigwedge_{x \in U} (\mu_A x \rightarrow \mu_B x) \qquad \blacklozenge$$

(정의2) U_1, U_2, U_3는 유한한 전체 집합이라 하고, R은 U_1에서 U_2로의 퍼지관계이고, S는 U_2에서 U_3로의 퍼지관계이다. 즉, R은 U_1U_2의 퍼지 부분집합이고 S는 U_2U_3의 퍼지 부분집합이다. 퍼지 관계 곱은 a는 U_1이고 c는 U_3일 때, a가 c에 관련되어 있는 정도를 나타낼 사용되는 퍼지연산이다. U_1에서 U_3으로의 퍼지관계인 삼각논리곱 ,◁, 는 다음과 같이 정의된다. 여기서 i는 U_1에 포함되는 원소의 수이며, k는 U_3에 포함되는 원소의 수이고, j는 U_2에 포함되는 원소의 수이다. 또한 N_j는 j의 총 수를 나타낸다.

$$(R \triangle S)_{ik} = \frac{1}{N_j} \sum_j (R_{ij} \rightarrow S_{jk}) \quad (2)$$

이것을 퍼지 관계 곱 (fuzzy relational products)이라 한다. ◆

(정의3) 퍼지 함의 연산자는 주어진 문제의 범주에 따라 달라진다. a∈U_1에 대한 후위집합 (afterset) aR는 a와 연관된 y∈U_2로 구성된 U_2의 퍼지 부분집합이며 그 멤버십 함수는 $\mu_{aR}(y)=\mu_R(a,y)$로 주어진다. c∈U_3에 대한 전위집합 (foreset) Sc는 c에 연관된 y∈U_2로 구성된 U_2의 퍼지 부분집합이며 그 멤버십 함수는 Sc(y)=S(y,c)로 주어진다. aR이 Sc의 부분집합인 평균정도는 y∈aR의 멤버십 정도가 y∈Sc의 멤버십 정도를 함의하는 평균정도로서 다음과 같이 정의된다.

$$\pi_m(aR \subseteq Sc) = \frac{1}{N_{U_2}} \sum_{y \in U_2} (\mu_{aR}(y) \rightarrow \mu_{Sc}(y)) \quad (3)$$

여기서 π_m은 평균 정도를 나타내는 함수이다.

위의 평균 정도는 R◁S에 의해서 a가 c에 관련되는 정도를 나타낼 수 있다[10-12].

Ⅲ. 본 론

본 장에서는 제안방법인 클라우드 기반의 소셜 네트워크 서비스의 계층적 시각화에 대하여 설명한다. 제안 방법은 그림1과 같이 시각화 방법 알고리즘 모듈, 분산병렬 처리 모듈, 시각화 모듈로 구성된다.

3.1. 시각화 알고리즘 모듈

그림1(1)의 시각화 알고리즘은 전처리된 소셜 네트워크 자료를 본 논문의 저자들이 이전에 제안한 시각화 방법[7]을 확장하여서 그림1(2)의 분산병렬 처리 모듈을 통하여 분산병렬로 계산하고, 계산결과인 시각화 표현 구조를 JSON구조 파일로 저장하여 그림1(3)의 시각화 표현 모듈에 전달한다.

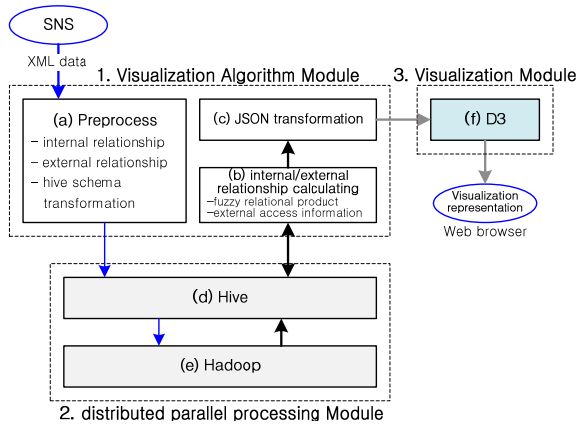


그림 1. SNS 계층 시각화를 위한 클라우드 기반의 제안 방법

Fig. 1. Proposed method based on cloud for SNS hierarchy visualization

3.1.1. 전처리

그림1(a) 전처리 단계는 소셜 네트워크의 XML 자료에 포함되어 있는 정보를 수집하여 사용자 노드의 내외부관계를 계산할 수 있도록 자료를 전처리하여 행렬로 만들고, 이 행렬 집합을 하이브 스키마에 적합한 관계형 자료로 변환하여 하이브를 통하여 하둡에 분산 데이터로 저장한다.

내부관계는 소셜 네트워크 사용자들이 대화한 메시지들 간의 관계정보를 이용하여 용어-사용자노드 빈도 행렬을 생성한다. 즉, 내부관계의 예를 트위터(twitter)로 보면 다음과 같다. 트위터는 자기와 비슷한 생각을 지닌 사람 간에 정보나 생각, 취미, 관심사 등을 실시간으로 공유할 수 있다. 공유되는 정보를 퍼뜨리거나 자신의 생각을 추가하여 다른 사람에게 공유하거나 댓글을 달 수 있다. 트위터에서의 내부관계는 공통관심사에 대한 게시글이나, 다른 사람에 글에 대한 댓글 등 공통 주제에 대한 사용자들 간에 게시되는 글 자체를 내부관계라 할 수 있다. 한글 메시지의 경우에는 한글 형태소 분석 도구[12]를 사용하여 용어를 추출한 다음에 빈도 행렬을 구성하며, 영문 메시지의 경우 불용어 제거, 어근 추출한 후에 빈도 행렬을 구성한다. 영문의 불용어는 Rijsbergen의 불용어 목록[13]에 정의 되어있는 불용어를 어휘 분석하여 무의미한 용어를 제거한다. 영문의 어근 추출은 Porter의 어근 추출 알고리즘[13]을 이용하여 영어의 파생어들 중에서 가중 중심이 되는 용어인 어근으로 변환한다. 용어-사용자노드 빈도 행렬의 $M_{*j} = [t_{1j}, t_{2j}, ..., t_{nj}]^T$는 사용자 메시지인 j번째 열에 포함된 용어의 출현 빈도를 나타낸다. 여기서 용어빈도 t_{ij}는 j번째 사용자 메시지에

출현한 i번째 용어의 출현 빈도이다.

외부관계는 소셜 네트워크상에서 노드들 간에 접근되는 정보로 네트워크 노드들 간에 전송되는 메시지의 양과 사용자간 메시지가 참조되는 횟수, 기간 등을 이용하여 만든다. 트위터를 통한 외부관계의 예는 다음과 같다. 트위터에서 자신과 비슷한 생각을 지닌 사람인 팔로어(follower)의 수, 특정 주제에 대한 게시글의 수, 게시된 글에 대한 댓글이나 격려 메시지의 수, 특정 글을 다른 사용자들에 퍼뜨리는 수 등을 외부관계라 할 수 있다.

다음 표1은 전처리되는 외부관계 기호들의 정의를 나타낸다. 여기서, mn은 한사용자가 다른 사용자에게 보내는 메시지의 개수, tmn은 사용자들 간 참조되는 메시지의 총 개수, rmn은 사용자가 재전송하는 메시지의 개수, unrm은 메시지를 재 참조하는 사용자의 개수, di는 i번째 일의 하루 동안 받은 메시지의 총 개수, tdme는 메시지를 주고받는 총 일 수, tnme는 두 사용자가 사용한 메시지에 포함된 명사의 개수를 각각 나타낸다.

표 1. 외부관계 기호에 대한 정의
Table 1. Definition of external relationship symbol.

symbol	signification
mn	· message number of sending between user's nodes
tmn	· total message number
rmn	· resend message number
unrm	· user number of re-reference message
di	· message number of *i*th day
tdme	· total day of message exchange
tnme	· term number of message exchange between user's nodes

그림2는 소셜 네트워크의 XML 자료를 하이브 스키마에 적합한 관계형 구조로 변환하기 위한 UML(unified modeling language)로 나타낸 것으로 내부관계 및 외부관계에 이용되는 변수 및 함수를 정의하였다. 그림2에서 시각화 클래스는 소셜 네트워크의 내부관계와 외부관계를 참조한다. 내부관계는 소셜 네트워크의 사용자 메시지의 용어-사용자노드 빈도에 대한 변수 termUserFrequency를 정의하였으며, 내부관계함수 ir()를 이용하여 퍼지 연관 곱을 이용하여 내부관계를 계산한다. 표1에서 정의된 외부관계 변수와 외보관계함수 er()을 이용하여 외부관계 계산한다.

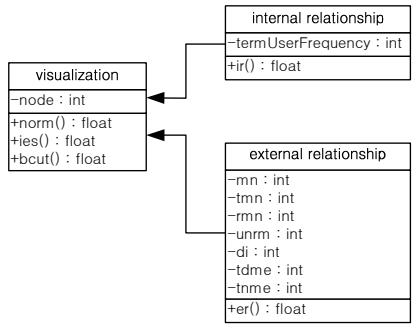


그림 2. 내외부관계 UML 다이어그램
Fig. 2. UML diagram of internal and external relationship

3.1.2. 내외부관계 계산

(1) 내부관계 계산

소셜 네트워크 노드의 내부관계는 네트워크에서 전송되는 사용자 노드의 메시지에 포함된 정보를 얼마나 반영되는지를 나타내는 것이다. 즉, 사용자의 메시지는 하나의 사용자 노드로 정의할 수 있으며, 이 노드와 사회관계를 갖는 노드는 이 메시지에 대한 댓글, 답신 메시지, 격려 메시지 등으로 정의할 수 있다. 이러한 메시지들은 특정 주제에 부합되는 용어의 집합으로 구성된다. 이 때문에 용어-사용자노드 빈도행렬이 그 노드의 내부관계를 계산할 수 있는 정량화된 값이라 할 수 있다. 이 정량화 자료에 퍼지 연관곱을 이용하여 노드와 노드간의 포함관계를 정량적으로 계산할 수 있다. 퍼지 관계 곱을 이용하여 두 노드 간의 포함관계를 다음과 같이 계산한다. 전처리된 용어-사용자노드 빈도행렬에 식(1)의 Kleene-Diense 퍼지 함의 연산자를 기반으로 한 식(4)의 퍼지 연관 곱을 계산하여서, 사용자와 사용자 간의 포함관계를 기반으로 사용자 내부 관계를 계산한다. 일반적으로 식(3)의 퍼지 관계 곱을 적용하여 노드들 간의 퍼지 연관 관계, $n_i \rightarrow nj$를 유도할 수 있다. 그러나 n_i에 멤버십 값($\mu_{ni}(x)$)이 작은 원소 x가 많으면, $n_i \subseteq n_j$의 포함여부와 관계없이 항상 1에 가까운 값이 나오는 문제점이 있다. 따라서 두 노드의 내부관계 함수 ir()을 다음과 같이 정의하여 퍼지 집합의 포함 관계 $\mu_{m,\beta}(n_i \subseteq n_j)$를 이용하여 계산한다.

$$ir(n_i,n_j) = \mu_{m,\beta}(n_i \subseteq n_j) = \quad (4)$$
$$(R^T \triangle_\beta R)_{ij} = \frac{1}{|n_{i\beta}|} \sum_{K_{i\beta} \in n_{i\beta}} (R_{ik}^T \rightarrow R_{kj})$$

여기서, K_i 는 i번째 용어이고, n_i, n_j는 i번째와 j

번째 노드이며, $n_{i\beta}$는 n_i의 β-제약, $\{x|\mu_{nj}(x) \geq \beta\}$ 이고 $|n_{i\beta}|$는 $n_{i\beta}$의 원소의 개수다. R은 m×n행렬로서 R_{ij}는 $\mu_{nj}(K_i)$,즉, $K_i \in n_j$인 정도이다. R^T는 행렬 R의 전치 행렬로서 $R_{ij} = R^T_{ji}$이다.

(2) 외부관계 계산

외부관계는 소셜 네트워크상에서 참조되는 사용자 노드의 메시지의 양이 소셜 네트워크에서 얼마나 반영되었는가를 나타내며, 참조되는 메시지를 이용하기 때문에 방향성이 고려해야 한다. 즉, 사용자 노드 a의 입장에서는 사용자 노드 b에게 보내는 메시지의 비율이 증가할수록 a에서 b로 노드의 외부관계가 증가한다. 또한 최근에 참조하는 메시지일수록 외부관계가 증가한다. 이외에도 메시지 참조는 메시지를 게재하는 사용자 노드나 참조하는 사용자 노드나 모두 참조 행위를 인지할 있는데 비해, 재참조는 재전송하는 b노드만이 누구에게 보내는지 인지할 수 있고, a노드는 다른 사용자 노드의 재전송 메시지참조 유무는 알 수 없다. 이 때문에 참조 메시지에 비해서 재 참조되는 메시지는 외부관계에 중요도가 적게 반영된다. 본 논문에서는 이러한 소셜 네트워크의 노드 접근 정보를 반영하여서 식(5)와 같이 외부관계(external relationship)함수 er()를 이용하여 외부관계를 계산한다.

$$er(a,b) = \left(\frac{mn \times tnme}{tmn}\right) \times \sum_{i=1}^{tdme} \left(\frac{di}{tdme-(1-i)}\right) + \left(\frac{r\,mn}{tmn \times unrm}\right) \quad (5)$$

제안된 외부관계는 메시지의 양에 기반을 두고 있어 많은 양의 허위 메시지가 발생하는 경우 왜곡이나 편향된 시각화 결과를 나타낼 수 있다. 본 논문에서는 시각화가 왜곡되는 결과를 최소화 시기키 위하여 외부관계에 내부관계를 반영하여 시각화한다. 즉, 사용자 노드가 허위 메시지를 보내더라도 연관된 다른 사용자 노드와의 포함관계가 낮음으로 내부관계의 값이 낮아진다. 이러므로 내부관계나 외부관계 값이 한쪽만 높음으로써 노드간의 관계가 편향되는 것을 최소화 시킬 수 있다.

(3) 계층적 시각화 구조

사용자 노드의 계층적 시각화 구조 단계는 다음과 같다. 첫째, 계산된 내부관계와 외부관계에 식(6)을 이용하여 정규 화한다. 둘째, 정규화된 내외부관계에 식(7)을 이용하여 합산한다. 셋째, 식(8)의 bcut()을 이용하여서 내외부관계의 합산 값에서 평균이상의 값을 가지는 사용자 노드들만 선택한다. 그래프 노드에 포함된 관계 값의 합을 이용하여 사용자의 계층관계를 구성한다. 즉, 사용자의 노드는 노드에 접속되는 경로의 관계 합이 큰 쪽이 계층도의 상위노드가 되며, 복합관계로 연결된 노드인 경우 노드에 포함된 부모노드와 자식노드의 관계 합이 큰 노드가 상위 노드가 된다. 노드와 노드는 한 번만 연결되는 것이 원칙이나 필요에 의해 다중 연결도 가능하다. 마지막으로 합산 행렬의 평균값을 기준으로 사용자 계층관계의 중요도를 조정하여 소셜 네트워크의 관계 계층의 크기를 조절 할 수 있다. 즉 bcut()의 값이 크면 소셜 네트워크의 관계 계층의 크기가 작아지며, bcut()의 값이 작의면 관계 계층의 크기는 커진다.

내부관계와 외부 접근정보가 나타내는 원소 값의 정량적 비율이 다르기 때문에 이를 시각화에 반영하면 내부관계나 외부관계한쪽으로 치우쳐서 시각화되는 문제가 발생할 수 있다. 이러한 문제를 해결하기 위해서 내외부관계의 값이 전체 사용자들 중에서 차지하는 비율로 정규화(normalization)하여서 원소 값의 불균형 문제를 해결한다. 다음 식(6)은 내부관계 및 외부관계를 정규화 하는 식이다.

$$norn(a,b) = \frac{n_{ij}}{\sum_{i=1}^{k}\sum_{j=1}^{l} n_{ij}} \quad (6)$$

여기서 norm()은 관계를 정규화 시키는 함수며, (a, b)는 a노드를 참조하는 b노드를 나타내며, k는 a노드를 참조하는 사용자의 총노드수, l은 b노드를 참조하는 사용자의 총노드수, nij는 i번째 a노드를 참조하는 j번째 b노드의 값을 나타낸다.

내외부관계를 정규화 한 후에 이들을 시각화 구조에 반영하기 위해서는 내부관계와 외부관계를 다음 식(7)과 같이 합산해야 한다.

$$ies(a \rightarrow b) = norm(ir(a,b)) + norm(er(a,b)) \quad (7)$$

여기서 ies()는 내부관계와 외부관계를 합산시키는 함수이며, norm()은 정규화 함수, ir(a, b)는 a노드에서 b노드 대한 내부관계, er(a, b)는 a노드에서

b노드에 대한 외부관계를 나타낸다.

다음 식(8)은 노드의 관계를 계층적으로 시각화 시 기준점을 제시하는 식이다. 본 논문에서는 내외부관계의 합산 값에서 평균이상의 값을 이용한다.

$$bcut(n_i) = \frac{\sum_{j=1}^{l} n_{ij}}{l} \quad (8)$$

여기서 n_i는 i번 노드를 나타내며, l은 사용자의 수, n_{ij}는 i번 노드와 j번 노드의 관계를 나타낸다.

3.1.3. JSON 변환

그림1(c)의 JSON 변환은 그림1(b)의 내외부관계 계산 결과인 사용자 노드 계층구조를 JSON(javascript object notation)형태로 변환하여 그림1(3)의 시각화 모듈에 전달한다. 다음 표2는 내외부관계 계산의 결과인 사용자 노드 시각화 계층구조를 JSON 형식으로 변환한 것의 일부분을 보여주고 있다.

표 2. 시각 구조로부터의 JSON 변환 결과의 일부분
Table 2,. A part of result of JSON transformation from visualization structure

```
{
 "name": "n22",
 "children": [
  {
   "name": "n2",
   "children": [
    {
     "name": "n3",
     "children": [
      {
       "name": "n11",
       "children": [
           {"name": "n28", "size": 8740} ]
      }
     ]
    }
  }
```

3.2. 분산병렬 처리 모듈

그림1(2)의 분산병렬 처리 모듈은 그림1(d) 하이브와 그림1(e)의 하둡으로 구성된다. 분산병렬 처리 모듈은 그림1(1)의 시각화 방법모듈의 전처리에서 변환된 자료를 하이브를 통하여 하둡에 분산 데이터로 저장하며, 그림1(b) 내외부관계 계산과 연결되어 하이브를 통해 저장된 자료를 하둡에서 분산병렬로 계산한다. 하둡의 분산병렬 처리 구성환경은 다음 같다. 본 논문의 분산병렬 처리 시스템은 인텔 3i 기반의 4대의 개인용 컴퓨터를 이용하여 구성하였다. 하둡의 분산 서버 구성정보로 4대의 개인용 컴퓨터를 이용하여 네임노드 1대, 보조 네임노드와 데이터노드 공용 1대, 데이터노드 2대로 구성하였다.

다음 표2는 HiveQL 스키마를 정의한 것의 일부분을 나타낸 것이다. 본 논문에서는 internalInformation, internalRelation, externalRelation, visualization 등 4개의 스키마를 정의하였다. internalInformation 스키마는 사용자노드, 용어, 용어 빈도수, 퍼지 멤버십이 정의되어 내부관계 계산에 기본 자료로 이용된다. internalRelation 스키마는 사용자, 퍼지 관계 곱 값 등의 내부관계 스키마가 정의되었다. externalRelation 스키마는 표2의 외부관계 변수(mn, tmn, rmn, unrm, di, tdme, tnme)와 외부관계계산 함수 값 er 등의 외부관계 스키마가 정의되었다. visualization 스키마는 internalRelation과 externalRelation을 이용하여 사용자와 사용자의 계층 구조 시각화 표현을 위한 사용자1과 사용자2 간의 정규화 값 norm, 내외부관계 합 ies, 계층적 시각화의 기준점 값 bcut가 스키마로 정의되었다. visualization 스키마의 값들은 D3에 전송되어 사용자의 계층적 구조로 시각화되어 표현한다.

표 3. HiveQL 스키마 정의 일부분
Table 3.. Definition of HiveQL schema

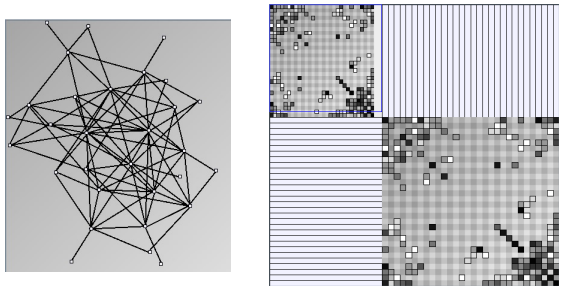
```
CREATE EXTERNAL TABLE internalInformation
(node INT, term STRING, count INT, membership FLOAT)
ROW FORMAT DELIMITED
        FIELDS TERMINATED BY ','
        LINES TERMINATED BY '\n'
        STORED AS TEXTFILE
LOCATION '/user/hadoop/sns';

CREATE EXTERNAL TABLE internalRelation
(node INT, fuzzy FLOAT)
ROW FORMAT DELIMITED
        FIELDS TERMINATED BY ','
        LINES TERMINATED BY '\n'
        STORED AS TEXTFILE
LOCATION '/user/hadoop/sns';
```

3.3. 시각화 모듈

그림1(3)의 시각화 모듈은 시각화 알고리즘의 결과인 사용자 노드의 시각화 계층 구조인 JSON(javascript object notation) 자료를 D3의 자바스크립트 라이브러리를 이용하여 웹브라우저에서 그래픽으로 시각화하여 표현한다. 그림3은 [14]의 소셜 네트워크 시각화 예 및 평가 자료 중 실제 자료

인 이메일 자료를 노드 링크와 행렬 그래프를 이용하여 시각화한 것을 보여준다.



(a) node link visualization
(a) 노드 링크 시각화

(b) matrix graph visualization
(b) 행렬 그래프 시각화

그림 3. 소셜 네트워크의 이메일 교류에 관한 실제 자료의 시각화[14]
Fig. 3. Real data visualization regarding email exchange of social network[14]

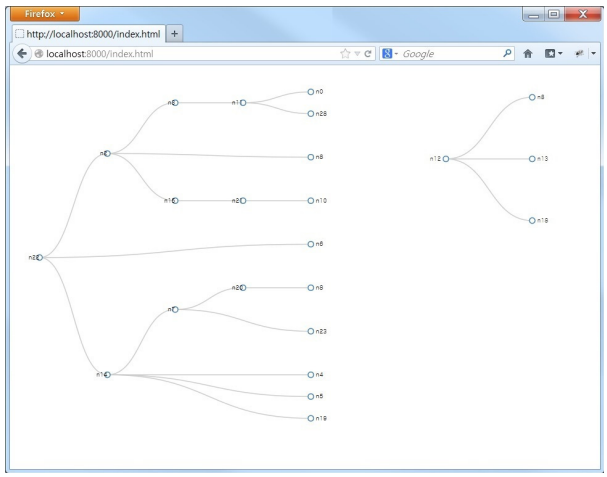


그림 4. 그림 3의 계층적 시각화
Fig. 4. Hierarchy visualization of Figure 3

그림4는 제안방법을 이용하여 그림3의 시각화 결과를 하나의 계층 트리로 보여주고 있다. 첫 번째 계층형 트리는 n22, n2, n6, n14, n3, n8, n15, n11, n21, n0, n28, n10, n4, n5, n7, n19, n20, n23, n8 등 19개의 노드로 구성되어 있다. 두 번째 계층형 트리는 n12, n8, n13, n18 등 4개의 노드로 구성되어 있다. 그림3의 노드링크 구조에서는 총30개의 노드로 구성되어 있는데 비하여 제안 방법의 결과인 그림4의 계층형 트리구조에서는 23개의 노드로 구성되어 있으며 이중 n8은 3번 중복되어 표현되어 있다. 이는 시각화 알고리즘의 bcut()함수에 의하여 평균이사의 상호작용 관계를 갖는 노드들은 중요도가 거의 없는 것으로 평가하여 제거했으며, n8 노드의 경우 다중관계를 갖고 있기 때문에 중복되어 계층형 트리에 표시되었다. 그림3과 그림4를 비교해 보면, 그림4의 각 노드의 중요도를 직관적으로 파악할 수 있는 것을 알 수 있다.

Ⅳ. 실 험

본 논문에서는 성능평가 자료로 Social Network Generation 사이트의 소셜 네트워크 실제 자료를 이용한다[14]. 평가 자료는 다음과 같이 emailDay.xml, emailWeek.xml, emailMonth.xml, emailYear.xml, emailGDay.xml, emailGWeek.xml, emailGMonth.xml, emailGYear.xml, 등의 8개의 XML 파일로 구성된다[14].

표 4. 그래픽 표현의 가독성을 위한 Ghoniem의 평가 작업
Table 4. Ghoniem's evaluation task for readability of a graphic representation

task	generic tasks
1	· approximate estimation of the number of nodes in the graph, referred to as 'nodeCount'
2	· approximate estimation of the number of the links in the graph, referred to as 'edgeCount'
3	· finding the most connected node, referred to as 'mostConnected'
4	· finding the node given its label, referred to as 'findNode'
5	· finding a link between two specified nodes, referred to as 'findLink'
6	· finding a common nighbor between two specified node, referred to a 'findNeighbor'
7	· finding a path between two nodes, referred to as 'findPath'

본 논문에서는 실험에 대하여 가독평과 수행속대를 평가한다. 가독성 평가의 척도는 표4 Ghoniem의 작업 척도[3]를 이용하며, Ghoniem은 노드의 수, 링크의 수, 가중 중심이 되는 노드, 주어진 레이블에 적합한 노드 찾기, 특별한 노드 사이의 링크 찾기, 특별한 노드 사이의 인접 노드 찾기, 두 노드 사이의 경로 찾기 등 총 7개의 평가 작업을 제시하였다. 평가 값은 0부터 3사이의 값으로 점수를 부여하며, 점수가 0인 경우 에러이고 1인 경우 최고점수가 된다. 점수 부여 방법은 평가자가 시각화결과에 대한 평가척도 항목을 즉시 검색하는 경우 최고점인 3을, 주저하거나 지체하면 시간에 따라서 감점하며, 인식하지 못하거나 너무 많은 시간을 지체하면 0점을 부여한다[14]. 평가는 목포대학교 정보전자학과 1학년에서 4학년까지 학생들을 고르게 분포

하도록 혼합하여 100명의 평가자로 평가 그룹을 구성하였다[14-16].

평가비교는 제안방법인 FRH(fuzzy relationship hierarchy)을 NL, MAT, MatLink, MatTrix 방법을 비교하여 평가하였다. 여기서 NL은 Ghoniem의 노드링크에 의한 시각화 방법이며[3], MAT는 매트릭스에 의한 시각화방법이고[4], MatLink[5]와 Matrix[6]는 Henry가 제안한 노드링크와 매트릭스의 혼합형 방법이다.

실험1) 그림5는 5가지 시각화 방법의 평과 결과를 비교하여 보여준다. 7가지 평가 척도에 대한 평균의 비교결과 제안방법인 FRH가 NL방법에 비하여 가독성이 52.16% 우수, MAT방법에 비하여 36.12% 우수, MatLink방법에 비하여 24.64% 우수, MatTrix에 비하여 16.67%가 우수하다. 결과를 분석해보면 다음과 같다. 비교평가 결과 최하위 결과를 보이는 것은 NL방법이다. NL방법을 제외하고 MAT, MatLink, MatTrix의 경우 작업1, 2, 3에서는 제안방법과 큰 차이를 보이지 않으나, 작업 4, 5, 6, 7에는 큰 차이를 보이고 있다. 이것은 NL방식의 경우 노드와 경로가 증가할 수 록 서로 중복되어 가독성이 떨어지며, MAT방식의 경우 경로가 없기 때문에 평가자들이 판독하기 어려운 것으로 분석된다. 또한 NL방법의 경우 중심 역할의 노드들이 한곳에 모여 겹치기 때문에 쉽게 검색할 수 없으며, MAT, MatLink, MatTrix 방법들의 경우 매트릭스 기반이기 때문에 중심 역할을 평가자들이 신속히 판별하기는 어려운 것으로 분석된다.

실험2) 두 번째 평가는 시각화 방법의 수행시간으로 전처리 시간을 제외한 시각화 알고리즘의 결과가 JSON 파일이 만들어질 때까지의 시간을 기록하여 평가하였다. 평가방법은 단독으로 개인 컴퓨터에서 시각화 방법을 수행한 시간과 4대의 개인 컴퓨터에 분산병렬 처리하여 수행한 시간을 비교하여 평가하였다. 비교 평가에 사용된 5대의 개인 컴퓨터는 동일한 환경으로 cpu는 인텔 i3 3.3Ghz, 램 8G, 하드디스크 1Tbyte로 구성되어 있다. 평가 자료는 가독성 평가와 같이 8개의 XML 평가 자료를 이용하였다. 실험결과 단독으로 제안방법을 수행시 평균 54분 8초의 처리시간이 소요되었으며, 분산병렬로 처리시 12분 47초가 소요되었다. 즉, 제안방법을 분산병렬로 처리하는 경우 단독으로 시각화 구조를 계산하는 것에 비하여 4.2배의 시간이 단축되었다.

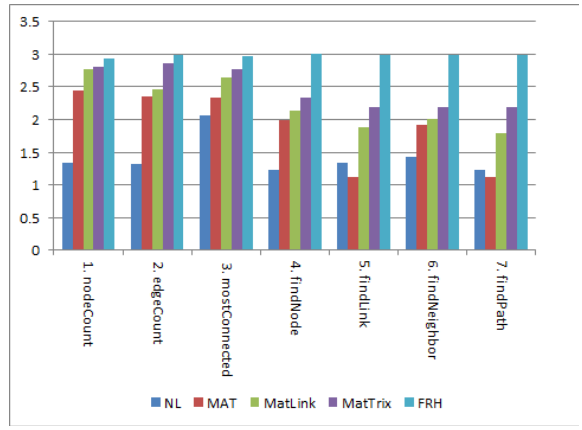


그림 5. 실험결과 비교
Fig. 5. Comparison of evaluation results

V. 결 론

이전 소셜 네트워크의 시각화 방법들은 시각화 결과 사용자 관계를 쉽게 읽거나 이해하기 어렵고, 네트워크 노드들의 경로를 탐색하는 것이 어려운 문제가 있다. 이 외에도 사용자 노드의 메시지가 시각화에 반영도지 않기 때문에 사용자들의 경향이나 쟁점을 쉽게 파악할 수 없으며, 소셜 네트워크 자료가 빅데이터일 경우 처리 효율이 좋지 않은 문제를 가지고 있다. 본 논문은 이러한 문제를 해결하기 위해서 네트워크상의 내외부관계를 반영하여 사용자간의 관계를 사용자 중심으로 계층적 시각화하는 새로운 클라우드 기반의 분산병렬처리 방법을 제안하였다. 제안방법은 다음과 같은 장점을 갖는다. 첫째, 제안방법은 사회적 관계를 계층적으로 시각화하기 때문에 사용자들의 관계 및 중요도를 쉽게 파악 및 분석할 수 있다. 둘째, 사용자가 작성한 메시지를 네트워크의 내부관계에 반영함으로써 소셜 네트워크상에 쟁점이 되는 사안들을 사용자관계 계층에 반영할 수 있다. 셋째, 네트워크 노드의 관계를 외부 접근정보에 반영함으로써 노드간의 경로를 쉽게 파악할 수 있다. 넷째, 사용자 중심으로 사용자관계를 계층적으로 표현하기 때문에 소셜 네트워크상에서 중요한 공동체나 중심적인 역할을 수행하는 사용자를 쉽게 찾을 수 있으며, 사용자 관계의 계층적 시각화로 이전 연구들의 가독성 문제를 해결할 수 있다. 마지막으로 제안 방법은 하둡(hadoop)과 하이브(hive)를 이용하여 분산저장 및 병렬로 계산되어 결과는 D3를 이용하여 계층적 그래프로 시각화함으로써 기존의 시각화 방법에 비하여 빠른 결과를 얻을 수 있다.

References

[1] NAVER encyclopedic knowledge, *Cloud Computing*, Retrieved Mar., 30, 2013, from http://terms.naver.com/entry.nhn?cid=200000000&docId=1350825&mobile&categoryId=20000756.

[2] J. W. Jeong, *Beginning Hadoop Programming: Developement and Operations*, Wikibooks, 2012.

[3] M. Ghoniem, J.-D. Fekete, and P. Castagliola, "On the readability of graphs using node-link and matrix based representations: a controlled experiment and statistical analysis," *Inform. Visualization*, vol. 4, no. 2, pp.114-135. July 2005.

[4] S. Wasserman and K. Faust, *Social Network Analysis*, Cambridge University Press, 1994.

[5] N. Henry and J.-D. Fekete, "MatLink: enhanced matrix visualization for analyzing social networks," *Lecture Notes in Computer Science*, vol. 4663, pp. 288-302, Sep. 2007.

[6] N. Henry, J. Fekete, and M. J. McGuffin, "NodeTrix: a hybrid visualization of social networks," *IEEE Trans. Visualization Comput. Graphics*, vol. 13 no. 6, pp. 1302-1309, Nov.-Dec. 2007.

[7] J. Heer and D. Boyd, "Vizster: visualizing online social networks," in *Proc. IEEE Symp. Inform. Visualziation (INFOVIS 2005)*, pp. 32-39, Minneapolis, U.S.A., Oct. 2005.

[8] S. Park, J. J. G. Jeong, M. S. Yoe, and S. R. Lee, "Visualization method of user hierarchy of among SNS users," *J. Korea Inst. Inform. Commun. Eng.*, vol. 16, no. 8, pp. 1717-1724, Aug. 2012.

[9] S. Park, J. W. Kwon, M. A. Jeong, Y. W. Lee, and S. R. Lee, "Hierarchy visualization method of SNS user fuzzy relational," *J. Inst. Electron. Eng. Korea*, vol. 49, no. 9, pp. 76-84, Sep. 2012.

[10] D3, *Data-Driven Documents*, Retrieved June, 30, 2013, from http://d3js.org.

[11] W. Bandler and L. Kohout. "Semantics of implication operators and fuzzy relational products," *Int. J. Man-Mach. Stud.*, vol. 12, no. 1, pp. 89-116, Jan. 1980.

[12] K. N. Han and K. W. Nam, *Beginning of Korea information processing : for understanding Korean language by computer*, Communicationbooks, 2007.

[13] W. B. Frankes and B. Y. Ricardo, *Information Retrieval : Data Structure & Algorithms*, Prentice-Hall, 1992.

[14] N. Henry, J. D. Fekete, Social Network Generation, Retrieved June, 30, 2013, from http://www.infovis-wiki.net/index.php/Social_Network_Generation#Real_Social_Networks.

[15] J. T. Oh, "Personal environment service and technology based on smart phone," *J. Korean Inst. Commun. Inform. Sci. (KICS)*, vol. 38, no. 5, pp. 454-463, May 2013.

[16] Y. J. Park, H. S. Cho, and J. W. Son, "Transmitting/receiving of standard health data using bluetooth HDP on the Android platform," *J. Korean Inst. Commun. Inform. Sci. (KICS)*, vol. 38, no. 5, pp. 464-470, May 2013.

박 선 (Sun Park)

1996년 2월 전주대학교 전자계산학과 이학사

2001년 2월 한남대학교 정보통신학과 공학석사

2007년 2월 인하대학교 컴퓨터정보공학과 공학박사

2008~2009년 호남대학교 컴퓨터공학과 전임강사

2010년 전북대학교 인력양성사업단 박사후 과정

2010년 12월~현재 목포대학교 정보산업연구소 연구전임교수

<관심분야> 정보검색, 데이터마이닝, 데이터베이스, 해양IT정보융합

김 용 일 (Yong-Il Kim)

1984년 전남대학교(이학사)
1986년 한국과학기술원(공학석사)
1986년~1994년 한국원자력연구소 선임연구원
1994년~2000년 초당대학교 컴퓨터학과 조교수
2002년~현재 호남대학교 인터넷콘텐츠학과 조교수
<관심분야> 빅데이터 처리, 지능형정보검색, 클라우드 컴퓨팅, 지능형 에이전트 등

이 성 로 (Seong Ro Lee)

1987년 2월 고려대학교 전자공학과 공학사
1990년 2월 한국과학기술원 전기및전자공학과 공학석사
1996년 8월 한국과학기술원 전기및전자공학과 공학박사
1997년 9월~현재 목포대학교 공과대학 정보전자공학과 교수
<관심분야> 디지털통신시스템, 이동 및 위성통신시스템, USN/텔레매틱스응용분야, 임베디드시스템