호적 데이터베이스에서 가중치방식 동일인 추적

구슬기*, 우병일*, 문성업*, 이상원*, 손병규**
*성균관대학교 컴퓨터공학과
**성균관대학교 동아시아학술원
e-mail : aristrain@gmail.com

Weight-based identification of individuals In Chosun family register database

Seul-Ki Koo*, Byung-Il Woo*, Sungup Moon*, Sang-Won Lee*, Byung Giu Son**
*Dept. of Computer Engineering, SungKyunKwan University
**Academy of East Asian Studies, SungKyunKwan University

요 약

호적은 사회문화적으로 많은 가치를 지닌 데이터이나 체계적으로 전산화되어 있지 않기 때문에 사용에 많은 제약이 존재했다. 또한 호적은 직접적인 세금 징수 및 군역 관리를 위한 기록이기 때문에 인구연구에 사용하기에 부정확하고 결여되거나 불일치한 데이터가 다수 존재한다. 따라서 인구학 연구에 중요한 동일인 정보에 대해 직접적인 비교로 동일인을 찾는 것은 불가능하다. 본 논문에서는 웹 서버와 데이터베이스를 사용해 가중치방식을 통한 동일인추적을 가능하게 하는 것으로 호적의 연구자원으로서의 가치를 증가시킨다. 조선시대 단성현의 93,803 개의 데이터를 대상으로 최적 가중치 한계와 소모시간의 단축을 위한 방법에 대해 서술한다.

1. 서론

호적은 일정 지역내의 개인과 가족을 호(戶)의 형태로 묶어 저장한 공문서를 말한다. 주기적으로 기록된 호적은 지역내의 개인과 가족의 탄생, 이동, 소실을 기록하는 것으로 시대의 사회적, 문화적 상황과 변동을 추적하는 것을 가능하게 한다. 하지만 기본적으로 호적은 세금 징수 및 군역관리를 위한 기록이므로 사회적,문화적 연구에 사용되기에는 알맞은 형태가 아니다. 또한 호적은 체계적으로 전산화 되어 있지 않기 때문에 사회문화적 연구에 그리 많이 사용되지 않는다. 특정 연도에 등장한 개인이 언제 재등장하고, 소실되는지에 대한 체계적이고 자동적인 접근이 이루어지지 않았기 때문에 기존에는 직접 수동으로 탐색해야만 했고 많은 시간과 노력이 소모되었다. 이 작업이 자동화 된다면 호적이 연구 자료로써 더 많은 가치를 가질 것이며 역사학자들이 과거의 생활상 및 변천과정을 연구하는데 큰 도움이 될 것이다.

본 논문에서는 조선시대 경상도 단성현의 호적을 대상으로 하여 데이터베이스를 구축하고 개개인의 동일인을 추적하는 방안에 대해 소개하고 그 중 최적의 방법을 제안하고자 한다. 2 장에서는 호적 데이터의 특징과 알고리즘, 구현 방식을 포함한 전체적인 시스템을 소개하고 3 장에서는 최적의 동일인 추적을 위한 여러 방식에 대한 실험 및 결과를 보이며 마지막 4 장에서는 결론과 향후 연구에 대해 기술한다.

2. 호적 데이터 정리 시스템 소개

2.1 호적 데이터의 특성

단성현의 호적 데이터는 64 개의 속성을 가지는 93,803 개의 레코드로 가족 단위로 정렬되어있다. 이 93,803 개의 레코드는 조사된 연도에 따라 1678 년부터 1818 년까지 26 개의 연도로 나눠져 있으며 각각의 연도를 식년이라 부른다. 이 레코드는 다음과 같은 네 가지 특징을 지닌다.

첫 번째, 동일인을 찾는데 사용하는 속성이 14 개로 제한되어 있다. 호적 데이터내의 총 속성 수는 64 개지만 이 중에는 동일 데이터를 한글/한자로 두 번 나타내거나 동일인을 찾을 때는 사용할 필요가 없는 속성이 다수 존재한다. 양반층 외의 기록은 호적에 기록할 때 한글에 대해 한자의 음독, 훈독을 사용하지만, 기록시 일정한 기준이 있는 것이 아니기 때문에 동일인물의 데이터라도 다른 식년에 기록된 한자명이 다른 경우가 많다. 이것을 고려해 모든 한자/한글이 모두 존재하는 데이터는 한글만을 사용하도록 한다. 총 64 개의 속성 중 동일인을 찾을 때 사용하는 속성은 14 개로 연도정보, 개인정보, 가족정보, 데이터 고유정보의 4 가지로 구분 할 수 있다.

두 번째, 데이터에 빈 값이 많다. 호적 데이터내의 총 속성 수 6,003,392 개 중 데이터가 들어있지 않거나 부지(不知:알 수 없음)로 기록된 속성은 총

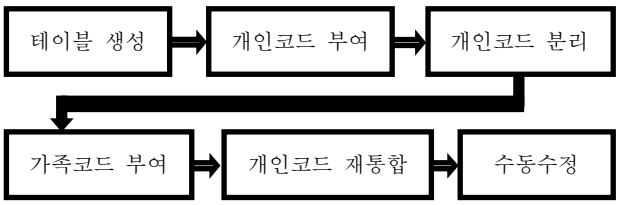
3,192,588 개로 53.17%에 이른다. 동일인 찾기에 사용되는 14 개 속성에 한정할 경우, 총 수 1,500,848 개 중 빈 값은 351,284 개로 23.4%가 된다. 모든 데이터가 온전한 속성은 연도, 레코드번호, 호 내 위상, 호주의 이름의 4 개 속성뿐이다.

세 번째, 같은 데이터를 다르게 표현 하는 경우가 다수 존재한다. 앞에서 말한 훈독,음독에 의한 부 정확성 외에도 분명한 동일인물이 연도가 변함에 따라 속성이 변하는 경우가 존재한다. 수작업에 의한 오기(誤記)나 아명(兒名)을 사용하여 분가 전과 분가 후의 이름이 다른 경우도 다수 존재한다.

네 번째, 여성의 경우 많은 경우 이름이 정확히 기록되지 않는다. 대부분의 기혼여성은 이름이 기록되지 않고 '씨', '소사', '성'이 이름 대신 표기된다. 미혼 여성의 경우도 마찬가지로 '악지'나 '아지' (아기라는 의미)가 이름 대신 표기된다. 통계적으로 93,803 개의 레코드 중 이름 속성으로 '씨'라는 값을 가지는 레코드는 10,568 개가 되며 소사(7,374 개),성(4,127 개), 악지 (1,996 개), 아지(770 개)가 그 뒤를 잇는다.

2.2 동일인 추적 알고리즘

동일인 추적은 6 단계로 이루어져 있으며 전체적인 구조는 그림 1 과 같다.



(그림 1) 전체적인 구조

본 논문에서는 테이블 생성부터 개인코드 통합을 다루고 있으며 마지막 단계인 수동수정을 최소화 하는 것을 목표로 하고 있다.

2.1 절에 설명한 호적 데이터의 특성에 의해 직접적으로 속성들을 비교해 동일인을 구하는 것은 한계가 있다. 동일인 탐색에 사용되는 14 개 속성이 모두 일치하는 동일인을 찾을 경우 동일인 탐색비율은 3.4%에 불과하다. 모든 동일인을 자동으로 추적하는 것은 불가능하므로 최대한 수동수정을 줄이는 것을 목표로 한다. 따라서 본 논문에서는 동일인을 탐색을 위해 가중치 방식을 통한 개인코드 부여를 제안한다. 가중치 방식을 통한 개인코드 부여는 각 속성의 중요도 및 정확도에 따라 다른 가중치를 주어 일치하는 항목과 가중치를 곱한 수의 합이 일정 기준 이상이 될 때 동일인으로 판단하는 방식이다.

개인코드 부여 모듈은 표 1 의 속성별 가중치를 사용한 식(1)을 사용한다.

$$\sum \text{일치여부}\left(\text{일치할 경우 1, 일치하지 않을 경우 0}\right) \times \text{일치하는 항목의 가중치} \geq \text{가중치 기준} \quad (1)$$

속성	가중치
성	2
이름	4
간지	4
호 내 위상	2
부 이름	1
외조부 이름	1
조부 이름	1
증조부 이름	1
총합	16

<표 1> 속성별 가중치

예를 들면 특정 인물과 성,이름,간지와 부명, 외조부명이 같고 나머지가 다른 두 개의 레코드가 존재하며 가중치 기준이 11 이라고 하자. 속성별 가중치를 고려해 계산하면 가중치 합은 1*2(성) + 1*4(이름) + 1*4(간지) + 0*2(호 내 위상) + 1*1(부 이름) + 1*1(외조부 이름) + 0*1(조부 이름) + 0*1(증조부 이름) = 12 ≥ 11 이므로 두 레코드는 동일인으로 취급할 수 있다. 각 속성별 가중치는 빈 값 등장 확률 및 기존 수동 동일인 탐색시 사용한 방법을 고려하여 결정하였다.

가중치를 통한 동일인 검색으로 검색된 레코드들에게 같은 개인코드를 부여한다. 개인코드는 각 개인을 나타내는 코드로 이 후 작업에 도움을 준다.

이 방법을 사용할 경우 일괄적으로 개인코드의 부여가 가능하지만 타인을 동일인으로 판단 할 가능성이 있다. 이 경우 시스템이 판단 할 수 없기 때문에 전문가(역사학자)의 판단이 필요하지만 동일한 연도에 동일인물이 두 번 호적에 등장하지는 않으므로 같은 연도에 등장하는 레코드는 명백히 동일 인물이 아니라고 판정할 수 있다. 그러므로 개인코드를 전체적으로 순회하여 같은 연도를 가지면서, 같은 개인코드를 가지는 레코드를 찾아 개인코드를 분리하도록 한다. 또한 (등장연도 - 나이)로 구할 수 있는 탄생연도가 3 년 이상 차이 나는 경우도 타인일 가능성이 크므로 개인코드를 분리하도록 한다.

개인코드 분리가 끝나면 명백히 타인이 동일인으로 판단되는 경우를 막을 수 있다. 하지만 반대로 동일인이 타인으로 분류되는 경우는 막을 수 없다. 이 문제를 해결하기 위해 가족의 추적을 통해 동일인을 찾는 방법을 사용한다. 예를 들어 연도 A 에 ㄱ,ㄴ이라는 부부가 존재하고 연도 B 에 ㄷ,ㄹ이라는 부부가 있다고 가정하면 ㄱ과 ㄷ이 동일인으로 취급되어 같은 개인코드를 가진다면 ㄴ과 ㄹ이 타인으로 취급되어 다른 개인코드를 가진다 해도 동일인일 가능성이 크다. 이런 경우에는 개인 코드 부여시 사용한 가중치 기준보다 낮은 기준을 사용하여 다시 확인하여 동일인 여부를 판별 할 수 있다.

가족추적을 통해 동일인을 찾기 위해서는 가족 단위로 정렬되어 있는 레코드의 특성을 이용해 가족코드를 만들어 부여해야 한다. 가족을 판별하기 위해서는 연도, 주호 명, 레코드번호를 사용하도록 한다. 가족코드가 부여된 후 같은 개인코드를 가지는 개인

의 가족코드를 통합하여 가족의 추적이 가능하도록 한다. 앞에서 말한 방법을 통해 가족코드를 추적해 개인코드를 통합한다.

개인코드 재통합까지 완료되면 이후는 전문 연구원(역사학자)들이 판단하여 수동으로 수정하도록 한다.

2.3 시스템 구현

호적 데이터베이스 시스템은 여러 연구자가 동시에 접근해 수정 및 연구가 가능하도록 해야 하고 매 갱신이 동일하게 유지되어야 한다는 특성을 고려해 웹을 통해 구현하도록 하였다. 우분투 리눅스 10.04 를 서버로 사용했으며 제작 언어는 PHP 를 사용하였고 데이터베이스는 MySQL 5.1.41 을 사용하였다.

데이터베이스는 3 개의 테이블로 이루어져 있다. 첫 번째 테이블은 원본 호적을 저장하는 테이블로 93,803 개의 레코드의 64 개 속성을 모두 저장하고 있으며 레코드 번호를 주 키로 가진다. 사용자는 이 테이블을 통해 실제적인 데이터에 접근이 가능하다. 두 번째 테이블은 개인코드를 저장하는 테이블로 개인코드 번호, 등장/퇴장 연도, 개인이 포함되는 가족의 코드 번호 등 동일인으로 취급된 개인의 정보를 저장한다. 개인코드 테이블은 개인코드 번호를 주 키로 가지며 개인코드 테이블의 주 키는 원본 호적 테이블의 외래 키 이기도 한다. 사용자는 이 관계를 사용하여 개인코드만으로도 원본에 빠른 접근이 가능하다. 마지막 테이블은 가족코드를 저장하는 테이블이다. 가족코드 테이블은 가족코드 번호와 등장/퇴장 연도의 3 개 속성을 가지며 주 키는 가족코드 번호이다. 가족코드 테이블의 주 키는 개인코드 테이블의 가족코드 번호 속성과 주 키-외래 키 관계로 연결되어있다. 한 개인은 분가 전/후의 2 개의 가족을 가지므로 개인코드 하나는 최대 2 개의 가족코드를 가질 수 있도록 구성하였다.

3. 실험 결과

3.1 최적 동일인 정확성 및 성능 평가

2.1 절에서 설명한 데이터의 특징에 의해 가중치 방식으로 동일인을 찾을 때 가장 중요한 변수는 가중치 한계 값과 빈 값 처리 방식이다. 가중치 한계 값이 높다면 정확도는 올라가지만 동일인을 찾는 비율은 낮아질 것이고 반대로 가중치 한계 값이 낮다면 동일인을 찾는 비율은 높아질 것이지만 정확도는 떨어지게 된다. 적절한 가중치 한계 값을 선택하여 최소한의 정확성 손실로 최대한 많은 동일인을 찾도록 해야 한다.

앞에서 서술한 것처럼 데이터에 빈 값이 많기 때문에 빈 값의 처리방법이 큰 변수가 된다. 서로 다른 레코드의 빈 값을 동일한 값으로 취급한다면 정확성은 떨어지지만 비교적 높은 가중치 한계 값에서도 많은 동일인을 찾을 수 있고, 연도나 호 내 위상처럼 빈 값은 없지만 실질적 가치가 크지 않은 속성의 영향이 적어질 것이다. 반면 다른 값으로 처리한다면 정확도는 높아지겠지만 빈 값이 없는 속성의 영향이 커지기 때문에 높은 가중치 한계 값에서 적은 동일인만을 찾을 것이며, 낮은 가중치 한계 값에서는 연도, 호 내 위상과 같이 빈 값은 없지만 실질적 가치가 크지 않은 속성의 영향이 커져 올바른 결과를 찾기 힘들어진다.

각 방식을 평가하기 위해 연도별 새 동일인 추가 확률의 평균과 상위 10 개 개인코드의 카운터 합을 구하도록 했다. 연도별 새 동일인 추가 확률은 식년의 레코드 중 기존 개인코드에 통합되지 않고 새로 개인코드를 받는 레코드의 확률을 의미한다. 이것을 매년 구해 그 평균을 구하는 것으로 동일인을 얼마나 많이 찾는지를 평가 할 수 있다. 동일인을 더 많이 찾을 수록 이 확률은 낮아진다.

상위 10 개 개인코드의 카운터 합은 가장 많은 레코드를 가진 상위 10 개의 개인코드가 가진 레코드의 수를 합친 것이다. 타인이 동일인으로 판단되는 경우가 많을 수록 카운터가 커지므로 이 수치가 낮으면 타인이 동일인으로 판단되는 경우가 적은 것으로 판단이 가능하다.

1 차 테스트에는 서로 다른 레코드의 빈 값을 같은 것으로 취급하여 정확성보다 많은 동일인을 찾는 것을 우선하여 테스트하였다. 한계 가중치를 14 에서 10 까지 수정하면서 테스트한 1 차 테스트의 결과는 다음과 같다.

한계 가중치	연도별 새 동일인 추가 확률 평균	상위 10개 개인 코드의 카운터 합
14	60.42%	852개
13	53.48%	1537개
12	52.09%	1571개
11	27.02%	12422개
10	18.77%	17115개

<표 2> 빈 값 동일화 비 제거 테스트 결과

1 차 테스트 결과 가중치를 낮추면 동일인 추가 확률이 낮아지는 것을 확인 할 수 있었지만 개인코드의 카운터 합이 크게 늘어나 정확도가 떨어지는 것을 확인 할 수 있다.

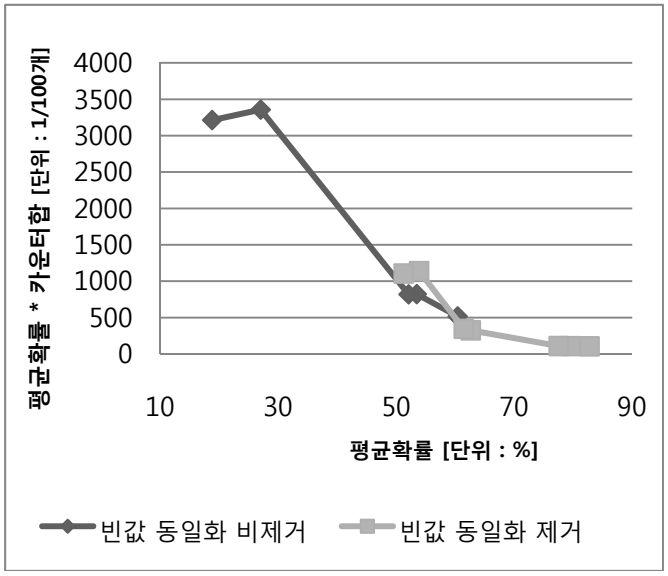
1 차 테스트와는 달리 2 차 테스트는 서로 다른 레코드의 빈 값을 다른 값으로 취급하여 많은 동일인을 찾는 것 보다 정확성을 우선하여 테스트하였다. 한계 가중치를 14 에서 8 까지 수정하면서 테스트한 2 차 테스트의 결과는 표 3 과 같다.

한계 가중치	연도별 새 동일인 추가 확률 평균	상위 10개 개인 코드의 카운터 합
14	82.77%	126개
13	79.57%	135개
12	77.52%	141개
11	62.63%	518개
10	61.44%	562개
9	53.88%	2112개
8	51.25%	2157개

<표 3> 빈 값 동일화 제거 테스트 결과

1 차 테스트와는 달리 연도별 동일인 추가 확률은 줄었지만 개인코드의 카운터 합이 감소하여 정확성이 향상된 것이 확인되었다.

이를 바탕으로 1 차 테스트 결과와 2 차 테스트 결과를 비교해 보기 위해 상위 10 개 개인코드의 카운터의 합과 연도별 새 동일인 추가 확률 평균의 곱을 구해 그래프를 그려 보면 그림 2 와 같다.



(그림 2) 1 차,2 차 테스트 비교

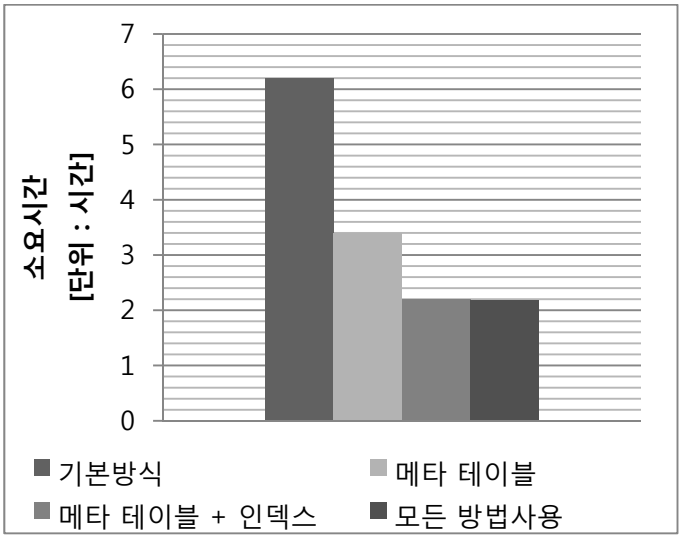
연도별 새 동일인 추가 확률의 평균이 60%이상 인 경우에는 빈 값 동일화 제거 방식이 카운터의 수가 적지만 그 이하에서는 오히려 빈 값 동일화 비 제거 방식이 증가하는 현상을 보였다.

그러므로 높은 정확성을 필요로 할 경우에는 한계 가중치 10 에서 14 사이의 빈 값 동일화 제거 방식을 사용하고 반대로 높은 동일인 검색확률을 원한다면 13 이하의 빈 값 동일화 비 제거 방식을 사용하는 것이 효율적임을 알 수 있다.

3.2 동일인 탐색시간 최적화

동일인 탐색 시 각 레코드가 등장한 연도 이후 등장 하는 모든 레코드의 대부분의 속성을 대상으로 탐색을 실시하기 때문에 시간이 오래 걸리는 문제가 발생했다. 기본적인 방식 사용시 6 시간 이상의 시간이 소비되었다. 이러한 소요시간에 대해 단축을 위해 동일인 찾기에 일부 속성만 사용 되는 데이터의 특성을 고려해 동일인 찾기에 사용되는 속성만 복사한 메타 테이블을 만들었다. 테이블의 크기가 기존 26.6Mb 에서 11Mb 로 줄어들어 탐색 대상이 약 58.7% 줄어드는 효과를 얻을 수 있다. 또한 원본에는 삽입/삭제가 없다는 특징을 이용해 메타 테이블의 각 속성에 대해 인덱스를 사용하였다. 마지막으로 의미 없는 탐색을 줄이기 위해 해당 식년으로부터 110 년 이후의 레코드는 대상에서 제외하도록 했다. 각 방식의 소요시간을 테스트해 본 결과는 그림 3 에 나타나 있다.

110 년 이후 레코드는 대상에서 제외하도록 한 방식은 큰 효과가 없었으나 메타 테이블을 사용할 경우 검색대상인 레코드의 크기가 줄어들어 소요시간이 46.2% 감소하였다. 인덱스를 사용하는 방식도 35.3%의 소요시간 감소를 보였다.



(그림 3) 소요시간 테스트

4. 결론 및 향후 연구

본 논문에서는 조선시대 호적의 동일인 분석을 위한 데이터베이스 구축 및 동일인 분석 방식에 대해 제안하였다.

호적은 체계적으로 정리되어 있지 않고 세금징수 및 군역 관리에 집중되어 있어 결여되거나 부정확한 데이터가 많아 직접적인 비교만으로는 동일인을 찾을 수 없다. 이를 해결하고자 본 논문에서는 가중치 방식을 통한 동일인 탐색을 제안하고 실험을 통해 원하는 정확도에 맞는 최적 한계 가중치와 빈 값 처리 방안에 대해 설명하였다. 또한 탐색시간을 최적화 시키기 위한 방안으로 메타 테이블 및 인덱스의 사용에 대해 실험하여 그 결과를 보였다.

차후에는 더 빠르고 정확한 방법을 얻기 위해 가중치를 사용하지 않고 레코드의 각 속성의 유무에 따라 분기해 다른 방법으로 동일인을 찾는 방안에 대해서도 연구가 필요하다. 또한 각 방식을 정밀하게 평가할 수 있는 새로운 평가 방법에 대해서도 연구가 필요하다.

참고문헌

[1] 이춘식, “데이터베이스 설계와 구축 : 성능까지 고려한 데이터 모델링”, 한빛 미디어, 2005

[2] 손병규, “호적 - 1606~1923, 호구기록으로 본 조선의 문화사”, 휴머니스트, 2007.

[3] 호적대장 연구팀, “단성호적대장연구”, 성균관대 대동문화연구원, 2003.

[4] 손병규, “조선후기 상속과 가족형태의 변화”, 대동문화연구 61 호, 2008.