Particle Swarm Optimization 알고리즘을 이용한 바이오칩 데이터의 군집화 및 분류화 기법

이윤경*°, 윤혜정*, 이민수*, 윤경오**, 최혜연**, 김대현**, 이근일**, 김대영**
이화여자대학교 컴퓨터학과*, (주)마크로젠**

Clustering and Classifying DNA Chip Data using Particle Swarm Optimization Algorithm

Yoon-Kyung Lee*°, Hyejung Yoon*, Minsoo Lee*, Kyong Oh Yoon**,
Hye Yeon Choi**, Dae Hyun Kim**, Keun il Lee**, Dae Young Kim**
Dept. of Computer Science Ewha womans University, Macrogen Inc

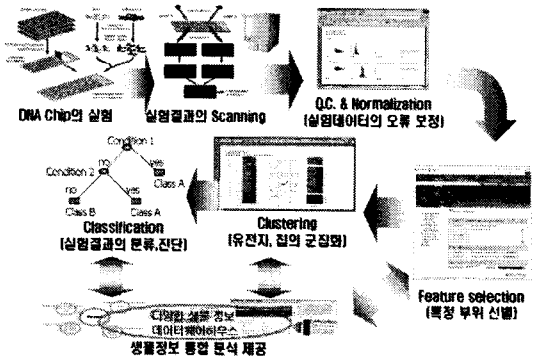
요 약

바이오 칩 분석 시스템은 다양한 종류의 바이오칩에서 자료를 추출하고 유용한 정보를 얻기 위해 데이터를 분석하는 시스템이다. 데이터를 분석하는 다양한 기법 중 대표적인 것이 클러스터링과 분류화(classification)이다. 클러스터링은 비슷한 개체들을 한 집단으로 묶는 방법이고, 분류화는 미리 정해진 클래스에 데이터를 해당하는 클래스로 분류하는 기법이다. 다양한 알고리즘을 통해서 데이터를 클러스터링 및 분류화를 할 수 있는데 바이오칩과 같이 데이터의 양이 방대한 경우는 생태계를 모방한 알고리즘을 적용하는 것이 효율적이다. 본 논문에서는 생태계 모방 알고리즘 중 하나인 PSO 집단 알고리즘을 사용하여 바이오칩 데이터로부터 클러스터의 중심을 찾아 클러스터링을 하고, 분류 규칙을 발견하여 이를 바이오데이터에 적용, 분류해 주는 시스템을 기술하고 있다.

1. 서론

다양한 생물 정보에 대한 분석 기술은 편리성과 신속성, 정확성이 더욱 중요해지고 있다. 따라서 실험 정보를 담은 바이오 칩[1]에 대한 통합 분석의 필요성이 크게 대두되고 있다. 기존의 널리 이용되는 유전자 분석 방법으로는 DNA와 DNA 그리고 DNA와 RNA 사이의 결합을 이용한 Southern 과 Northern blot 방법이 있다. 대부분의 Southern blot은 서로 같은 유전 정보를 가졌는가를 밝히는데 쓰이고, Northern blot은 특정 유전자가 얼마나 발현되는가를 알아내는데 쓰인다. 이러한 방법들은 한 번에 수십 개 이상의 유전자들을 검색하기가 어려워서 질병 진단과 같이 즉각적인 분석 결과가 필요한 경우에 시간 및 인력 소요가 불가피하다는 문제점이 있다. 이러한 문제점을 극복한 것이 바이오칩으로써 바이오칩은 기존의 분자생물학적 지식에다 고도의 기계 및 전자공학 기술을 접목하여 만들어졌다. 언제 어디서든 바이오 칩을 손쉽게 분석할 수 있도록 바이오 칩 스캐너의 소형화가 중요하며 또한 대용량의 바이오 데이터를 빠른 시간 내에 처리하기 위해서는 네트워크로 연결된 분석 시스템들 간에 서로 협력할 수 있는 분산 처리 환경이 지원되어야 한다. 그리고 생태계를 모방한 알고리즘들을 분석 과정에 도입함으로써 효율적이고 정확한 분석 결과를 얻는 것이 중요하다. 생물학자들은 바이오칩을 사용하여 유전자 실험을 한 뒤에 바이오칩으로부터 데이터를 추출하고 유전자 사이에 유사성 유형을 발견하기 위해 데이터를 조사하는 등의 분석 과정을 필요로 한다.

본 연구에서는 바이오칩의 다양한 정보를 보다 효율적으로 분석하기 위해서 생태계 모방 알고리즘의 하나인 Particle Swarm Optimization 알고리즘을 데이터마이닝과 접목시켜서 클러스터링 및 분류화(classification)를 하고자 한다.



(그림 1) 바이오칩 분석 시스템의 프로세스

논문의 구성은 다음과 같다. 2장에서 관련연구로 클러스터링과 분류화(classification), PSO (Particle Swarm Optimization) 알고리즘을 소개하고, 3장에서는 PSO 알고리즘에 기반한 클러스터링 시스템, 4장에서는 분류화 시스템을 설명한다. 5장에서는 구현 결과를 설명하고 마지막으로 6장에서는 결론과 향후 연구 방향을 기술한다.

2. 관련 연구

2.1 클러스터링 기법

클러스터링[2]이란 비슷한 개체들을 한 집단으로 묶는 과정을 뜻한다. 클러스터링 알고리즘으로는 다음과 같은 알고리즘들이 있다. K-means 알고리즘은 거리에 기반을 둔 클러스터링 방법으로 가까운 곳에 있는 데이터들끼리 같은 군집으로 묶는다. 계층적 알고리즘은 처음에 각각의 데이터 점을 하나의 클러스터로 설정한 후 이 둘 쌍 간의 거리를 기반으로 하여 분할, 합병해 나가는 상향식 방식으로 운영된다.

2.2 분류화 기법

분류화(classification)[2]는 널리 사용되는 데이터 마이닝 기법 중 하나로 크게 두 단계로 이루어져있다. 첫 번째 단계에서는 클래스 값이 있는 훈련 데이터로 훈련을 하여 분류 규칙을 발견하고 두 번째 단계에서는 분류 규칙을 클래스 값이 없는 테스트 데이터에 적용하여 데이터가 해당될 클래스를 예측하는 것이다. 예측된 클래스들은 실질적으로 데이터가 해당되어야 할 클래스들과 비교하여 분류 규칙의 정확도를 측정하게 된다.

2.3 Particle Swarm Optimization 알고리즘

PSO(Particle Swarm Optimization) 알고리즘[3]은 생태계 모방 알고리즘의 하나로 군집을 이루는 동물들, 예를 들어 새떼 등의 사회적 행동을 관찰하고 그 행동 패턴을 모태로 하여 개발된 최적화 기법이다. PSO는 문제를 해결하는 면에서 population 기반의 EC (Evolutionary Computation)와 유사한 과정을 거친다. EC는 시간이 지나면서 population이 점점 더 나은 population으로 진화를 하는 형태를 갖고, PSO는 각 군집이 더 나은 군집으로 발전되는 형태를 갖는다. 주어진 최적화 문제에 대해서 PSO는 세 가지 요소인 particle, 속도, fitness 함수를 정의함으로써 해결할 수 있다. Particle은 군집을 구성하고 있고, 문제에 대한 하나의 솔루션을 말하며 각각은 메모리를 갖고 있다. Fitness 함수는 각 particle들이 얼마나 좋은지를 나타내주는 함수이다. 이 fitness 함수를 이용하여 각 particle들은 자신이 시간에 따라 방문한 particle 중에서 가장 좋은 fitness값을 갖는 particle을 메모리에 기억하고 이를 지역 최적 값이라고 부른다. 그리고 군집은 모든 particle에 대한 지역 최적 값에 대해서 가장 좋은 fitness값을 갖는 particle을 전역 최적 값이라고 부른다. 이 때, 속도는 각 particle들이 다음 시간에 어느 방향으로 움직일지를 결정하는데, 이는 이전의 속도와 지역 최적 값과 전역 최적 값에 의해 정의된다. 시간 t에서 particle i의 속도를 $v_i(t)$라고 하고 particle i에 대한 지역 최적 값을 $q_i(t)$, 전역 최적 값은 particle g의 지역 최적 값이라고 하면, $v(t+1)$는 다음과 같이 정의된다.

$$v(t+1)=\omega \cdot v(t)+c_1\Phi_1(q_i(t)-p_i(t))+c_2\Phi_2(q_g(t)-p_i(t)).$$

이를 자세히 보면 지역 최적 값과 전역 최적 값의 방향과 현재의 자신의 속도를 결합하여 다음 방향을 결정하게 된다. 따라서 PSO는 속도에 의한 particle의 진화를 통해 더 나은 솔루션을 찾는다는 것을 알 수 있다.

3. PSO 클러스터링 알고리즘

PSO 알고리즘은 무리가 먹이를 찾아가는 과정에서 무리 전체가 정보를 공유한다는 가설과 무리 내부의 개체가 지금까지의 자기의 경험과 무리 전체에 공유되어 있는 정보를 기초로 하여 행동한다는 개념을 최적화 과정에 도입한 방법으로 이 방법을 이용한 clustering 알고리즘은 다음과 같다.

```
PSO clustering algorithm

For each particle
    Initialize particle
END

Do
    For each particle
     Calculate fitness value
     If the fitness value is better than the best fitness
value (pBest) in history
     set current value as the new pBest
    End

    Choose the particle with the best fitness value of
all the particles as the gBest
    For each particle
       Calculate particle velocity according equation

       Update particle position according equation
    End
While maximum iterations or minimum error criteria is
not attained
```

가장 먼저 초기화를 통해 각 입자의 위치와 속도 벡터를 난수를 이용해 설정한다. 각 particle은 무작위로 클러스터의 중심을 선택한다. 이후, 명시된 종료 시점까지 반복 실행하면서 각 particle에 속하는 각 데이터 벡터는 모든 클러스터의 중심까지의 거리를 계산하고 거리가 가장 짧은 cluster에 포함된다. 이후 전체에 fitness 함수를 적용하여 적합도를 계산한다. 이 작업이 모든 데이터 벡터에 대해 완료되면 global best 값과 local best값을 update하고 PSO search를 이용하여 cluster의 중심을 update한다. 적용되는 fitness function은 다음과 같다.

$$J_e = \frac{\sum_{j=1}^{N_c}\left[\sum_{\forall Z_p \in C_{ij}} d(Z_p, m_j)/|C_{ij}|\right]}{N_c}$$

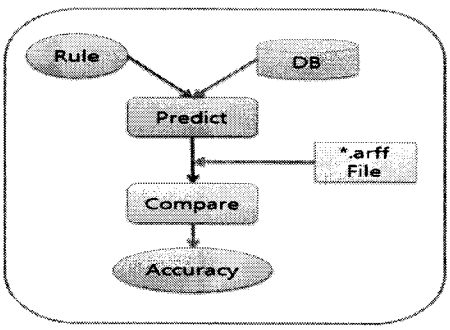
이 값이 작을수록 적합도는 크다고 평가된다.

4. PSO 분류화 알고리즘

PSO 알고리즘에서의 규칙 발견 알고리즘은 데이터 집합을 training set과 test set으로 나누면서 시작된다. 입자 또는 개체라 불리는 무리 내부의 개체 하나하나는 규칙을 나타내며 이들은 임의적으로 생성되게 된다. 생성된 개체들은 무리가 되어 움직이거나 진화를 하게 된다. 규칙이 미리 지정한 quality criteria를 만나게 되면 가지치기 과정을 수행하게 되며 이 과정을 통해서 불필요한 속성들이 제거가 되고 이 규칙이 최종 규칙 집합에 저장되게 된다. 개체들은 새로이 발견된 규칙에 의해 분류가 되며 분류가 된 것들은 training set에서 제거가 되며 그 후 새로운 목표 클래스(target class)가 선택된다. 그러면 새로운 진화 알고리즘이 수행되어 마찬가지로 새로운 규칙을 찾게 된다. 이러한 과정을 미리 지정한 수의 개체가 training set에 남게 될 때 까지 반복 수행을 한다. PSO 알고리즘의 particle을 분류화의 Rule 이라고 할 때, $P_i(t) = (P_i^1(t), P_i^2(t))$ 는 t시간일 때의 Class 1, Class 2 에 대한 i번째 individual 이 된다. 그리고 유전자의 개수를 파라미터로 보고 각 유전자에 대한 조건은 발현 값에 해당이 된다. 이를 수식으로 표현하면 다음과 같다.

$$P_i^1(t) = (P_{i,1}^1(t), P_{i,2}^1(t), P_{i,3}^1(t), \ldots, P_{i,100}^1(t))$$

속도는 발현 값의 변화량으로 정의 되는데, 이것은 이전 속도와 이전 지역 최적 값과의 발현 값의 차, 전역 최적 값과 발현 값의 차이로 계산을 하게 된다. 이를 앞에서 말한 PSO의 위치업데이트 수식에 적용을 시켜 계산을 하게 된다. 결국 PSO 알고리즘에 의해 처음 만들어 냈던 규칙에 있는 유전자의 발현 값이 조금씩 업데이트(변화)되면서, 특정 유전자이 어떤 발현 값이 되어야 가장 좋은 fitness를 가지게 되는지를 찾아낸다. 따라서 정확한 결과 또는 오차가 적은 rule을 찾을 수 있다.



(그림 2) 규칙을 적용한 분류 과정

정규화와 품질관리 과정을 거친 바이오칩 데이터는 데이터베이스에 저장이 되며 훈련 데이터로써 사용되어진다. Normalization_ID는 실험 ID이며 한 실험에는 수많은 유전자가 있게 된다. probe name은 유전자 이름을 나타내며, expression_value 유전자가 해당 실험에서 발현된 값을 나타낸다. 각각의 실험은 같은 수의 유전자를 갖는다. 각 실험의 클래스 값은

파일로 입력을 받게 되며 파일의 확장자는 *.arff이다. 클래스는 사용자가 정할 수 있으며 암이 있을 경우 1, 없을 경우를 -1로 하였다. 저장된 데이터와 파일의 클래스 정보를 가지고 PSO 알고리즘을 이용해서 훈련을 시키면 많은 분류 규칙이 결과로 나오게 된다. 결과로 나온 분류 규칙 중 신뢰도와 정확도가 높은 규칙이 선택되어져 클래스가 없는 테스트 집합에 분류 규칙을 적용하여 클래스들을 예측하게 된다. 예측된 클래스들은 해당 데이터가 본래 분류되어야 할 클래스 값과 비교하여 정확도를 측정하게 된다.

5. 실험 및 결과

PSO 알고리즘을 기반으로 한 클러스터링 시스템과 분류 시스템은 C로 구현되었다.

Normalization_ID	Probe_name	Expression_value
N000289	341276	-.68560427
N000289	341287	-1.4155682
...	...	...
N000304	316843	-.98317541

```
@relation AB 1700 mouse chip
@address 203.255.177.137:1521:rome
@id bio
@pw ant
@range 10

@class
N000287 -1
N000288 1
N000289 1

N000310 1
```

(그림 4) 훈련 데이터와 클래스 정보 파일

100개의 유전자와 발현값, 24개의 실험 데이터를 가지고 실험한 결과 아래와 같은 클러스터링 결과 및 분류 규칙이 생성이 되었다. 클러스터의 수는 총 5개로 각 particle들은 아래의 표와 같이 클러스터링 되었다.

표 안에 들어있는 값 들은 100개의 유전자중 몇 번째 유전자에 해당하는 것인지를 나타낸다. 첫 번째 클러스터에는 1번째 유전자부터 99번째 유전자까지 총 58개의 유전자가 들어있으며 두 번째 클러스터에는 8번, 62번, 87번, 88번째 유전자가 클러스터링 되어 있다는 것을 알 수가 있다. 분류화 결과에서 301972은 유전자 이름(probe name)을 말하며 -0.6은 binning을 사용한 유전자의 발현 값(expression_value)을 의미한다. 생성된 분류 규칙을 테스트 집합에 적용한 결과 66.7%의 정확도를 나타내었다.

Cluster 1	1, 4, 6, 9, 10, 11, 13, 15, 17, 18, 19, 21, 22, 23, 25, 26, 27, 28, 29, 31, 34, 35, 36, 37, 38, 39, 40, 43, 44, 45, 46, 47, 48, 51, 54, 55, 56, 58, 59, 60, 64, 68, 69, 73, 74, 78, 79, 81, 82, 83, 84, 86, 89, 91, 92, 94, 98, 99,
Cluster 2	8, 62, 87, 88
Cluster 3	61, 67, 95
Cluster 4	63, 85
Cluster 5	2, 3 ,5 ,7, 12, 14, 16, 20, 24, 30, 32, 33, 41, 42, 49, 50, 52, 53, 57, 65, 66, 70, 71, 72, 75, 76, 77, 80, 90, 93, 96, 97, 100

(그림 5) 클러스터링 결과

```
IF 301972=-0.6 AND 303177=-2.1 THEN '1'
DEFAULT -1
```

(그림 6) 분류 규칙 결과

6. 결론 및 향후 연구

생태계 모방 알고리즘 중의 하나인 PSO 알고리즘을 기반으로 구현된 클러스터링 및 분류 시스템은 바이오칩 데이터에서 클러스터의 중심을 찾아 데이터들을 클러스터링 하고, 분류할 규칙을 찾아내어 테스트 집합에 적용시켜 테스트 집합의 클래스를 예측하고 정확도를 측정한다. 방대한 양의 바이오칩 데이터를 PSO 알고리즘을 이용하여 효율적으로 클러스터링을 하고, 분류 규칙을 찾을 수 있었다. 앞으로는 수행 속도의 향상을 위해 최적화를 하고, 분류 규칙의 정확도를 높이기 위해 binning 과정을 향상시켜 보다 높은 정확도를 갖는 분류 규칙을 생성할 수 있게 연구할 것이다.

[참고문헌]

[1] Sorin Craghici. Data Analysis Tools for DNA Microarrays. Chapmen & Hall, 2003

[2] Jiawei Han, Micheline Kamber. Data Mining Concepts and Techniques. Morgan Kaufmann, 2001.

[3] A Cooperative approach to particle swarm optimization, Bergh., Engelbrecht, A.P. Evolutionary Computation, IEEE Transactions on Volume 8, Issue 3, June 2004 Page(s):225 - 239