진화프로그래밍을 이용한 이상 유전자 분류 방법 제안

김영지* · 배상현*

*조선대학교 컴퓨터통계학과

Suggestion Method of Classific System of Abnormal Genetic using EP

Young-Gie Kim* · Sang-Hyun Bae*

*Computer Science & Statistics, Chosun University

E-mail : syan264@lycos.co.kr, shbae@chosun.ac.kr

요 약

DNA 기술의 발달로 얻어진 대량의 유전자 정보를 손쉽게 이상 값을 가진 유전자의 정확한 분류와 진단을 할 수 있는 방법인 Microarray 기술에 대한 기대가 커지고 있다. 정확한 분류를 하기 위해서는 추출된 유전자에 들어 있는 많은 잡음 즉 이상 값을 가진 유전자만을 추출할 필요가 있다. 따라서 본 논문에서는 세 가지 dataset에 대해 기존 연구방법의 여러 가지 유전자 추출 방법을 조사하고 Matlab으로 구현한 진화프로그램을 이용하여 새로운 데이터의 분류방법과 모델링 방법을 제안한다.

ABSTRACT

It is expect that Microarray technique be direct classification and diagnosis of Genetic data have abnomal data value because DNA technique. It is necessary that many noses that is abnomal data in sampling genetic data. So in this paper reported sampling method in exiting study then suggests new data classific system and modeling method using EP by Matlab about three dataset.

키워드

Microarray, 이상 유전자, EP, Modeling, Clustering

Ⅰ. 서 론

복잡한 조절 기능을 갖는 생명 현상에 대한 분자 수준의 단편적인 이해는 한계가 있기 때문에 Human Genomic Project(HGP)와 같이 전체적인 이해를 위한 연구의 필요성이 대두되었다. 이 과정에서 염기서열의 기능을 이해하는 것은 필수적이므로 DNA 칩이 개발되었다. 최근의 cDNA Microarray와 Oligonucleotide Microarray 기술의 발전은 엄청난 양의 유전자 정보를 제공하고 있다. DNA 칩 기술의 발전은 유전자 정보의 대량 생산을 가능하게 하였고, 특정한 실험 환경과 조건에 따른 수천 개의 유전자 발현 정도를 동시에 파악할 수 있고, 이를 대량으로 처리함으로써 수천 개의 유전자 정보를 굉장히 빠르고 정확하게 분석할 수 있게 되었다[1][2].

유전자 Microarray 기술의 발전은 이상 유전자의 정확한 예측과 진단 분야에도 적용되어 많은 도움을 줄 것으로 예상된다. 특히 이상 유전자에 대해 정확한 분류 할 수 있다는 것은 이상 유전자의 치료 방법에 매우 중요하고 혁신적인 방법을 제공할 수 있기 때문에 이 문제에 관한 많은 연구가 진행되고 있다[3][4]. 기존의 방법 중에 분류에 필요한 정보를 주는 유전자를 선택하기 위한 다양한 특징 추출 방법과 여러 분류기를 이용하여 실험한 결과를 분석하고, 특징의 상관관계를 기준으로 각 분류기의 결과를 결합함으로써 분류 성능을 향상시키려는 연구가 있었다[4]. 하지만 하나의 벤치마크 데이터에 대하여 실험하였기 때문에 실험 결과에 대한 충분한 검증이 되지 않았다. 따라서 다양한 벤치마크 데이터를 이용하여 분류기의 성능을 체계적으로 분석해 볼 필요가 있다.

본 논문에서는 Leukemia dataset, Colon dataset, Lymphoma dataset 등 세 가지의 벤치마크 된 데이터 집합에 기존의 분류방법에 대해 조사

하고 Matlab으로 구현한 진화프로그램을 이용하여 새로운 데이터의 분류방법과 클러스터링 방법을 제안 하고자 한다.

II. 본 론

2.1 유전알고리즘(GAs : Genetic Algorithms)

유전알고리즘(GAs)은 모든 진화 기반의 탐색 알고리즘 가운데 아마도 잘 알려진 기법으로 탐색 과정에 자연 도태와 유전학의 원리를 사용한 최초의 알고리즘은 아니지만, 오늘날 가장 널리 사용되고 있다. GAs는 유연하고 강인한 능력으로 인해 불연속 함수 문제를 포함한 최적화 및 다양한 디자인 문제를 해결할 수 있는 대안으로 각광받고 있다.

2.2 진화프로그래밍(EP : Evolutionary Programming)

진화프로그래밍(EP)은 자연계에서 볼 수 있는 무성생식과 유성생식을 모방한 것으로 유전정보의 교환을 위해 교배로 알려진 재결합 연산자가 사용되며, 무작위 효과를 위해 돌연변이 연산도 사용된다. 모든 연산이 부모개체의 선택이 필요하며, 이 선택은 개체의 적합도에 의존하기 때문에 개체의 적합도를 측정할 수 있는 적합도 함수가 필요하게 된다. 또 집단 내의 개체를 무작위 값으로 초기화함으로써 초기 다양성을 확보하여 진화가 이루어지는 과정에서는 돌연변이 연산을 통해 다양성을 유지한다[6]. 그림 1은 진화프로그래밍의 계산 과정을 가상코드 형태로 정리한 것이다.

$t=0$; //시간을 초기화하고 시작

initializePopulation :
$P(t)=a_1(t), a_2(t), ..., a_n(t)$
$a_k=(x_{1k}, x_{2k}, ..., x_{nk}, v_{1k}, v_{2k}, ..., v_{nk})$
//임의의 값으로 개체 집단을 초기화

evalPopulation :
$P=\{\Phi(a_1(t)), \Phi(a_2(t)), ..., \Phi(a_n(t))\}$
$\Phi(a_k(t))=T(F(x_{1k}, x_{2k}, ..., x_{nk}), \delta)$
//집단 내 모든 개체의 적합도를 평가

while$(not$ 종료조건$)$ *do*{
Mutate :

$a'k=m'(a)(x_{1k}, ..., x_{nk}, \sigma_{1k}, ..., \sigma_{nk}), k=1, 2, ..., \mu$
//개체 집단에 돌연변이 적용

evaluate :
$P'(t)=\{\Phi(a_1(t)), \Phi(a_2(t)), ..., \Phi(a_n(t))\}$
$\Phi(a_k(t))=T(F(x_{1k}, x_{2k}, ..., x_{nk}), \delta)$
//새로운 개체 집단의 적합도를 평가

$P(t+1)=select\ s_{(q)}(P(t)+P(t))$
//실제 적합도로부터 확률적으로 생존 개체의 선택
//$s_{(q)}$는 q-승자승 선택

$t=t+1$; // 세대 수의 증가
}
//종료 조건(시간 또는 적합도)을 만족 못하면 계속 수행

end;
/**$m'(a)$는 돌연변이 연산자로 표준 정규분포 돌연변이, 로그정규분포 돌연변이, 평균 돌연변이 및 자기-적응성을 갖는 평균 돌연변이 가운데 하나를 사용.*/

그림 1. 진화프로그램의 계산 과정

2.2.1 개체의 선택 - 차분진화

차분진화란 집단에 속한 개체 벡터의 거리와 방향정보를 사용하는데, 그 구조와 연산이 간단한 반면 수렴성이 뛰어나다[5]. 목적변수에 해당하는 개체를 집단으로 사용하는 탐색 방법으로 집단의 크기는 진화 동안 변하지 않는다. 핵심은 변화된 개체를 발생하는 체계에 있는데, 교배로 만들어진 새 개체가 이전 것보다 우수하면 새 개체가 살아남는다.

$$F(x_i)=100(x_1^2-x_2)^2+(1-x_1)^2 \quad \cdots\cdots(1)$$

```
% 교배율에 의해 교배 대상 위치 결정
% 교배율 보다 작은 값을 가지면 1로, 그렇지 않으면
0으로
cross = rand(np, nx) < Cr;
nocross = cross < 0.5;  % cross와 0과 1의 값을
반대로 가짐

if (nde == 1)  % 식(3.1)에 의한 교배
   % 교배용 개체 생성
   crosspop = crossarr1 + F * (crossarr2 - crossarr3);
else
   nde = 2;  % 식(3.4)에 의한 교배
end

% 교배 대상 popold와 교배용 멕터 crosspop의 교배
연산
cross = oldpop .* nocross + crosspop .* cross;

% 탐색범위를 벗어난 경우, 해당 변수의 경계 값으로
설정
for i = 1: np
  for j = 1: nx
    if (cross(i, j) < xbound(i, j))
      cross(i, j) = xbound(1, j);
    elseif (cross(i, j) > xbound(i, j))
      cross(i, j) = xbound(2, j);
    end
  end
end

% 새로 만들어진 집단 cross의 적합도 평가
for i = 1: np
  [x obj] = feval(fname, cross(i, :), parm);  % 개체
평가
  tempfit = obj;
  nfeval = nfeval +1;

if (fType == 0)
  if (tempfit <= fit(i))  % 이전 개체의 목적함수 값과
비교
    % 더 작으면 새 개체를 다음 세대 개체(자식)로 인정
    pop(i, :) = cross(i, :);
    fit(i) = tempfit;  % 목적함수 값 저장

      if (tempfit < bestfit)
        bestpop = i;
        bestfit = fit(i);  % 더 우수한 목적함수 값 저장
      end
    end
  else
    if (tempfit >= fit(i))  % 이전 개체의 목적함수 값과
비교
```

```
        % 더 크면 새 개체를 다음 세대 개체(자식)로 인정
        pop(i, :) = cross(i, :);
        fit(i) = tempfit;  % 목적함수 값 저장

          if (tempfit > bestfit)
            bestpop = i;
            bestfit = fit(i);  % 더 우수한 목적함수 값
저장
          end
      end
    end
end

objhist(gen) = bestfit;
bestx = pop(bestpop, :);  % 더 우수한 개체(해) 저장

gen = gen + 1;
```

그림 2. 차분진화 프로그래밍

```
function [x, obj] = Func(x, parm)
obj = 100 * (x(2) - x(1)^2)^2 + (1 - x(1))^2;
```

그림 3. 식(1) - 차분진화 프로그래밍

2.2.2 진화프로그래밍(EP)

EP는 샘플들의 탐색 공간에서의 벡터를 양수로 스케일링하여 확률적인 요소를 포함함으로써 그 벡터의 적합도를 부여한다. 자연계에서 볼 수 있는 특정한 유전 방식을 모방한다. 기존의 유전자 프로그래밍에서는 교배연산과 돌연변이연산을 모두 사용하는 방법과는 달리 부모와 자식 간의 관계를 강조하는 돌연변이 연산만을 사용한다.

EP의 진화과정은 μ개의 부모 개체 각각을 돌연변이 시켜서 μ개의 자식 개체를 만든 후, 부모와 자식 개체의 합한 2μ개체로부터 확률적인 $q-$승자승 선택($q \geq 1$)을 통해 다음 세대를 위한 μ개의 부모 개체를 선택한다. $q-$승자승 선택 원리는 그림 4와 같다.

1. 개체 a_i에 대해 2μ개의 개체로부터 q개의 개체를 랜덤하게 선택한다.
2. 선택된 q개의 개체와 a_i의 적합도를 비교해서 q개 가운데 몇 개의 개체가 a_i의 적합도보다 열등한지를 세어서 이를 점수 $w_i \in \{0, 1, ..., q\}$로 준다.
3. 앞의 두 과정을 모든 개체 $a_i (i = 1, 2, ..., 2\mu)$에 대해 실행한다.
4. 2μ개의 개체를 $w_i (i = 1, 2, ..., q)$값에 따라 내림차순으로 정렬한다.

그림 4. $q-$승자승 선택 원리

w_i가 높은 값을 갖는 μ개의 개체를 다음 세대의 부모 개체로 선정한다. 점수 w_i는 식 (2) 과 같이 계산된다.

$$w_i = \sum_{j=1}^{q} \begin{cases} \text{만약 } \Phi(a_i) \geq \Phi(a_{x_j}) \text{이면 } 1, & i = 1, 2\mu, x_j \in \{1, ..., 2\mu\} \\ \text{아니면 } 0 \end{cases} \quad (2)$$

x_j는 $q-$승자승에 포함될 개체를 지정하기 위한 균일분포의 정수형 랜덤 변수이고, $\Phi(a_{x_j})$는 랜덤변수에 의해 지정된 개체의 적합도이다. 승자승 크기인 q 가 증가함에 따라 선택 체계는 $(\mu+\mu)$ 진화전략에서처럼 결정론적으로 변해간다. 가장 우수한 개체는 최대 적합도 점수인 q를 받기 때문에 항상 생존하며, 이는 엘리트주의를 사용한 것과 같은 효과를 나타낸다. 진화프로그래밍의 진화 과정은 그림 5와 같이 이루어지며, 종료 조건(만족할 만한 해를 찾았거나 계획된 반복 횟수에 도달한 경우)을 만족할 때까지 계속된다.

A. 초기 집단을 임의의 값으로 초기화한다. 집단에서 개체 (해) 의 숫자는 최적화 속도에 큰 영향을 주지만, 집단 크기를 얼마로 해야 적당하지에 대한 답은 없다.
B. 개체는 새로운 집단으로 복제되고, 이 개체가 돌연변이 된다. 돌연변이의 강도는 부모 개체에 할당된 변화 요구에 의존한다. 메타-진화프로그래밍에서는 목적변수의 돌연변이에 앞서 표준편차를 먼저 돌연변이해야 한다. 돌연변이 연산자로는 표준 정규분포 돌연변이, 로그정규 분포 돌연변이, 평균 돌연변이 및 자기-적응성을 갖는 평균 돌연변이 가운데 하나를 사용한다.
C. 자식 개체는 적합도 평가를 받고, 확률적인 승자승 선택을 사용하여 다음 세대 집단을 구성한다.

그림 5. 진화프로그램의 진화 과정

집단 크기가 일정해야 할 필요는 없으며, 부모 개체가 하나 이상의 자식 개체를 발생할 수도 있다.

2.3 모델링(Model)

입출력 사이의 관계를 기술 할 수 있는 퍼지 규칙을 생성할 수 있는 것을 퍼지 모델링이라 부르며, 이 규칙을 인식할 때 전건부와 후건부를 동시에 퍼지 분할하는 방법으로 퍼지 클러스터링을 사용한다[6]. 퍼지 클러스터링은 데이터를 그룹화하기 위한 무감독 학습전략(unsupervised learning strategy)으로 사용되지만, 데이터로부터 퍼지 규칙(if ~ then ~)을 생성할 때에도 유용하게 사용된다. 현재까지 가장 널리 사용되고 있는 퍼지 클러스터링 방법은 FCM(Fuzzy C-Means) 알고리즘이다. 이 방법은 특정 데이터가 모든 클러스터에 속하는 정도를 합하면 1이 되도록 하는 알고리즘으로 그림 6과 같이 전개된다.

(1) 분할 수 $c(2 \leq c \leq n)$와 m의 값을 선택한다.
(2) 퍼지 분할 행렬 a_i의 초기값을 결정한다.
(3) 식 (3)을 이용하여 클러스터의 중심 V를 계산한다.

$$v_i^{(t+1)} = \frac{\sum_{k=1}^{n} (\mu_{ik}^{(t)})^m X_k}{\sum_{k=1}^{n} \mu_{ik}^{(t)}} \quad (m > 1, i = 1, ..., c) \quad .....(3)$$

(4) 식 (4)를 이용하여 퍼지 분할 행렬을 갱신한다.

$$\mu_{ik} = \frac{1}{\sum_{j}^{c} \left(\frac{|X_k - V_i|^2}{|X_k - V_j|^2} \right)^{\frac{1}{(m-1)}}} \quad (i = 1, ..., c,\ k = 1, ..., n) \quad ....(4)$$

(5) $|U^{(t+1)} - U^{(t)}| < \delta$을 만족하면 이 과정을 종료하고 그렇지않으면 단계 (3)으로 복귀해서 이 과정을 반복한다. 통상 δ값으로 10^{-3}을 사용한다.

그림 6. FCM 클러스터링 알고리즘

이상의 알고리즘을 코드로 만든 것이 그림 7이다.

```
while (loop_cond >= eps)
```

```
iter = iter +1;
U0 = U;
tmp = U0 .^ m;  % 분할 행렬 원소의 계승 계산
% 클러스터별로 모든 데이터의 분할 행렬의 값
ttmp = sum(tmp');

% 식(3) - 클러스터 중심 V를 계산
V = X * tmp';
for i = 1:c
  V(:, i) = V(:, i) / ttmp(i);  % 클러스터 중심 계산
end
for i = 1:c
  for j = 1:d
    % V0를 X와 차원이 같게 만듦
    V0(j, :) = V(j, i) * ones(1, n);
  end
  tmp2 = X - V0;
  tmp2 = tmp2 .* tmp2;
  dist(i, :) = sum(tmp2);  % 클러스터 중심과
                           데이터와 거리
end
% 클러스터 중심과 데이터와의 거리가 최소 거리
  보다 작은 데이터를 찾아 최소 거리로 설정.
% 이는 새로운 분할 행렬을 계산할 때 0으로
  나누어지는 것을 방지
i = find(dist < min_dist);
dist(i) = dist(i) - dist(i) + min_dist;

% 식(4) - 새로운 분할 행렬 계산
for i = 1:c
  % 데이터가 가장 가까운 클러스터에 속하는 정도를
    1로 바꿈
  U(i, :) = 1./ ((dist(i, :)./ sum(dist)).^ (1/(m-1)));
end
% 데이터가 모든 클러스터에 속하는 정도의 합을 1로
  만듦
sumU = sum(U);
for i = 1:c
  U(i, :) = U(i, :) ./ sumU;
end
loop_cond = max(max(abs(U - U0)));  % 종료조건
                                    계산
end
```

그림 7. FCM 클러스터링

III. 결 론

3.1 기존의 실험 환경 및 결과

기존의 실험 방법에서 Leukemia, colon, lymphoma dataset에 대하여 7 가지의 유전자 선택 방법과 6 가지의 분류기를 사용한 결과와 42개의 분류 결과 중에서 3개의 분류기를 선택하여 결합한 결과는 Leukemia dataset의 경우에는 각 분류기의 가장 좋은 성능과 결합방법의 가장 좋은 성능이 같지만, Colon dataset의 경우와 Lymphoma dataset의 경우에는 결합방법의 결과가 더 나은 성능을 보이는 것을 알 수 있다. Colon dataset의 경우 분류기의 최고 인식률을 83.87%인데 반하여, 결합방법의 최고 인식률은 93.55%로 분류 성능이 향상된 것을 볼 수 있다. 또한 Lymphoma dataset의 경우에도 분류기의 최고 인식률은 92.00%인데 반해, 결합방법의 최고 인식률은 96.00%로 분류 성능이 향상되었음을 알 수 있다[1].

3.2 향후 연구 방향

현재 실험 단계인 제안된 진화프로그래밍은 기존의 방법 보다 인식률이 향상 될 수 있을 것이라 예상하고 있다[6]. 이러한 진화프로그래밍의 활용 분야는 패턴분류가 있으며 구체적으로는 시계열 데이터의 패턴 분류, 환자의 나이와 방사선 사진의 특징으로부터 유방암을 찾는 선형 식별 모델과 신경회로망을 학습시켜 분류 시스템의 설계할 수 있다. 모델링 및 제어방법에서는 수술중인 환자의 혈압 제어, 바다 음향의 모델링, 진화에 의한 시스템 동정, 카오스 모델(카오스 방정식의 파라메타 추정)을 만들고, 이를 활용하여 관측된 시계열 데이터에 카오스 신호의 존재 유무를 확인, 비선형 시스템의 퍼지 모델링 및 제어를 위해 퍼지 규칙베이스를 동정, 최적 제어 등에 사용할 수 있다. 최적화 부분을 중점적으로 하면 VLSI 채널의 라우팅 설계, 여행 일정 최적화, 연료 분배 최적화를 예상할 수 있으며 의사결정으로는 게임에서 지능적인 의사결정, 지진 발생 진원지 결정, 재고 문제에서 활용할 수 있을 것이다. 또한 음성 신호의 특징 해석, 신경회로망의 진화, 모의 DNA 칩으로부터 DNA 순서 정보 구성, 유한 요소 문제에 응용, 다국적 언어로 된 정보를 복구할 수 있도록 질문을 번역할 것으로 예상한다.

참고문헌

[1] 원홍희, 조성배, "암 분류를 위한 기계학습 분류기의 성능평가", 제18회 한국정보처리학회 추계 논문집 제9권 제2호

[2] M. B. Eisen and P. O. Brown, "DNA arrays for analysis of gene expression", Methods Enzymbol, vol. 303, pp. 179-205, 1999.

[3] L. Li, C. R. Weinberg, T. A. Darden and L. G. Pedersen, "Gene selection for sample classification based on gene expression data: Study of sensitivity to choice of parameters of the GA/KNN method", Bioinformatics, vol. 17, no. 12, pp. 1131-1142, 2001.

[4] S. Dudoit, J. Fridlyand, and T. P. Speed, "Comparison of discrimination methods for the classification of tumors using gene expression data", Journal of the American Statistical Association, vol. 97, pp. 77-87, 2002.

[5] J. W. Ryu and S. B. Cho, "Towards optimal feature and classifier for gene expression classification of cancer", Lecture Note in Artificial Intelligence, vol. 2275, pp. 310-317, 2002.

[5] 황희수, "진화계산 및 진화디자인", 내하출판사, pp. 79-87, 2002.

[6] 황희수, "퍼지, 진화컴퓨팅 프로그래밍", 내하출판사, pp. 64-126, 2006.