제2회 컴퓨테이셔널 인텔리젼스 합동학술대회

확률 라이브러리 모델 상에서의 조건부 확률 계산 방법
A Method for Computing Conditional Probabilities in Probabilistic Library Model

허민오*, 장병탁

서울대학교 전기컴퓨터공학부 moheo@bi.snu.ac.kr*, btzhang@cse.snu.ac.kr

요약

확률 라이브러리 모델(Probabilistic Library Model)은 DNA컴퓨팅 방법론에 기반하여, 라이브러리를 이루는 원소들의 빈도로 결합확률분포를 표현하고자 하는 모델이다. PLM에서 조건부 확률을 계산하는 방법이 필요해짐에 따라, 본 논문에서는 *in-vitro*에서 DNA를 이용하여 임의의 조건부 확률을 계산하는 두 방법, 두 라이브러리를 이용한 방법, 은닉 PLM을 이용한 방법을 제시하고, 시뮬레이션을 통하여 실제 확률과 1% 이내로 동일한 결과를 얻었다. 은닉 PLM 방법이 실험 단계를 줄일 수 있음을 설명하였다.

1. 서론

확률 라이브러리 모델(Probabilistic Library Model)은 DNA컴퓨팅 방법론에 기반하여[4], 라이브러리를 이루는 원소들의 빈도로 결합확률분포를 표현하고자 하는 모델이다.[2,3] 이 모델은 *in-vitro*에서 DNA를 이용하여 구현해낼 수 있으며, 분자생물학 실험실에서 흔히 쓰이는 실험기법인 PCR과 dilution을 이용하여 나타내고자 하는 값들을 조정할 수 있다.[1] 이러한 PLM의 연산자를 고려할 때, 실험실에서 실제로 시도할 만한 가능성을 가진다. 특히, 어떤 패턴을 조회하고자 할 때 쉽게 자신이 찾는 패턴을 나타내는 원소를 발견할 수 있어야 하는데, 실제로 DNA 분자는 화학적으로 강력한 안정성을 갖는 것과 더불어 상보적인 쌍을 갖는 성질이 있어, 원하는 서열을 갖는 DNA 분자를 높은 신뢰도로 발견할 수 있고 화학적 안정성이 있으므로 연산모델의 원소로 사용하기에 적절하다.

PLM은 거시적인 관점에서 볼 때, 통계물리학적인 분자 정보처리 모델로서 시스템이 가지고 있는 확률 분포의 시간적인 진화를 다룬다.[2, 3] 즉, DNA 형태의 일반화된 논리식들(wDNF)과 혼성화 반응을 통해 이중 DNA구조를 형성하고 그들을 선별적으로 추출하여 복제함으로써 라이브러리에 있는 분자들의 확률분포를 조정하면서 학습하게 된다.[1]

하지만, PLM에서는 학습될 훈련 데이터 집합의 데이터들이 차례로 DNA로 표현된 후, 잘려져서 입력되게 되므로[1], 학습된 라이브러리로부터 임의의 변수에 대한 조건부 확률을 계산하는 것은 직관적이지 않다. 그래서, 본 논문에서는 이러한 계산을 수행하고자, 임의의 조건부 확률을 계산하는 두 가지 방법을 제시한다.

2절에서는 PLM 시뮬레이션 조건과 매개변수에 대해 언급하고 3절에서 PLM을 사용한 조건부 확률을 계산하는 방법을 제시, 시뮬레이션을 통해 얻은 결과를 평가한다. 그리고 4절에서는 은닉 확률 라이브러리 모델을 이용한 계산 방법을 제시하고, 시뮬레이션 결과를 평가하며 3절에서 제시한 방법과 비교하고, 마지막 5절에서 결론을 제시한다.

2. PLM의 도입 및 사용된 매개변수

실제로 분자를 이용하여 직접 실험실에서 실험하기에는 비용과 시간의 문제가 크므로, 적절한 제약을 가한 상태에서 시뮬레이션 실험으로 대체하였다. 아래에 시뮬레이션을 위해 세운 네 가지의 가정들을 나열하였다.

1. 모든 DNA 분자는 다른 모든 DNA 분자와 접촉할 가능성을 충분히 갖는다.
2. 화학반응이 일어나는 용기의 부피와 각

분자의 부피는 고려되지 않는다. 즉, 연산상의 문제만 아니라면 무한히 많은 분자의 존재가 가능하다.

3. 조금이라도 상보적이지 않은 DNA들은 서로 절대로 결합하지 않으며, 접힘이나 헤어핀과 같은 2차구조로 인해 성질이 변하거나 다른 쌍이 생기는 문제는 일으키지 않는다. 또한, 분자의 성질은 위치 좌표에 완전히 무관하며 상보적인 쌍 외에는 다른 힘 – 이를테면, 전자기력, 중력의 영향을 받지 않는다.
4. 모든 실험 과정에서 용액은 완전히 섞여 있어서, 용액 전체에 모든 성분의 분자가 각 성분마다 완전히 균등 분포를 갖는다.

아래에 PLM 시뮬레이션 상에서 필요한 매개변수들을 정리하였다.

Order: order가 커질수록 동시에 많은 변수들간의 상관도를 고려하게 된다. 실제 실험에서는 DNA 분자의 수에 제한이 있으므로 n 비트 상의 실험에서 order i까지 고려할 경우, $\sum_n C_i$ 개의 경우의 수를 고려하게 된다. 이 실험에서는 1, 2 order에서 가능한 경우의 수를 완전히 모두 사용하였다.

PCR rate(Learning rate): 학습률과 같은 역할을 한다. PLM에서는 학습 데이터가 입력되면 모든 X_i 단위로 완전히 조각나며, 미리 준비된 라이브러리에서 각각의 조각난 X_i 값들에 해당하는 DNA를 찾아 적정 비율로 증폭을 시킨다. 원하는 DNA 분자를 생성하여 원하는 개수만큼 실험용기에 넣는 실험기술은 아직 실용화 되지 않았기 때문에, 산술적 덧셈은 부적절하며, PCR을 이용한 곱셈 연산이 적당하다.
이러한 부분을 구현하기 위해 X_i 에 해당되는 조각마다 라이브러리 안의 분자 개수에 특정 비율을 곱하였다. 이 실험에서는 1.00002, 1.000002의 두 학습률을 사용하였다.

Decision Rule: 결과값을 알고자 할 때, 어떤 방법을 이용할 것인가에 관한 부분이다. 이 실험에서는 Majority Voting을 이용하였다.

Dilution : 학습률에 따라 계속 개수를 늘리기만 하면 학습이 진행됨에 따라 용기 내의 분자수가 무한히 증가한다. 이러한 일은 실제로 실험 상에서 불가능하고, 시뮬레이션 상에서도 변수의 최대값을 넘어서는 일이 발생하게 된다. 그러므로, dilution을 통해 적정 수의 분자를 버려서 전체 분자 개수를 유지한다. 실제 실험상에서도, 일괄적으로 모든 라이브러리의 분자가 섞여있는 전체 용액의 일부를 버리고 그만큼 다시 용매를 채워, 분자의 숫자를 유지하는 것이 쉽다. 이 실험에서는 다음의 식을 모든 라이브러리 내 원소의 수에 곱하였다.

$$1.0-\frac{\alpha-1.0}{N_L N_C} \qquad (\text{식 } 1)$$

α : PCR rate

N_L: 라이브러리 내 원소의 가짓수

N_C: 클래스 가지 가짓수

전처리 및 사용한 데이터: 사용한 데이터는 UCI Machine Learning Repository의 OptDigit 데이터이다.[5] 이것은 손으로 쓴 숫자를 32×32의 데이터로 가진 데이터집합이다. 이 데이터 자체로는 실험하기에 너무 크기 때문에 8×8로 줄였다. 16개의 픽셀 중에 8개 이상 채워져 있으면 1, 그 이하이면 0으로 결정하였다. 그림 1은 실험에 사용한 데이터의 클래스 별로 digit들의 분포를 어두운 정도로 나타낸 그림이다.

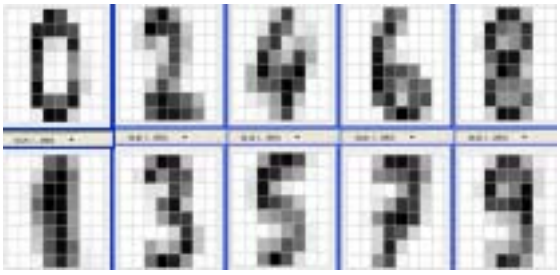

그림 1: 클래스 별 Digit 데이터의 평균 이미지. 0~255의 흑백이미지로 1과 0의 빈도를 나타내었다.

매개변수	값
초기 Library 내의 동일한 DNA 수	1,000,000개
사용된 Order	1, 2
PCR rate	1.00002, 1.000002
전처리, 데이터집합	8×8 이진 데이터
Decision Rule	Majority Voting
Dilution	모든 Library에 동일하게 특정 비율로 줄임.

표 1: 사용된 매개변수와 값

상기 사용된 전략과 매개변수들을 요약하여 표 1에 간단히 정리하였다.

3. 조건부 확률 계산 알고리즘

단일 라이브러리만으로는 학습된 데이터에 대한 조건부 확률을 계산하기 어렵다. 그 이유는 아래와 같다.

1. PCR rate에 따라 증폭되고 dilution을 통해 희석이 되므로 증폭된 정확한 DNA 개수를 알 수 없다.
2. 조건에 맞는 모든 DNA 개수를 세는 것은 어려운 일이며, 개수를 정확히 알게 되더라도 그로부터 확률을 알아낼 수 없다.

하지만, 원하는 조건부 확률에서 쓰이는 변수를 모두 표현할 수 있을 정도의 order를 사용한다면, 단일 라이브러리만 사용하는 경우가 아니라면, PLM으로부터 확률을 계산할 수 있다.

그림 2에 두 개의 라이브러리를 이용하여 PLM에서 확률을 구하는 방법을 제시한다.

제시한 알고리즘이 작동하는 원리는 단순하다. 두 라이브러리의 PCR rate 차이에 따라 나타나는 DNA 개수 차이의 비율은 그대로 유지되기 때문이다. 이 비율의 값을 통하여 본래의 임의의 확률을 얻어낼 수 있다.

위의 알고리즘은 나눗셈 연산과 뺄셈 연산, 정규화 과정을 거쳐야 하므로, 외부적 별도 연산의 도입이 불가피하다. 표 2는 x22와 x30, y를 변수로 두고 위의 알고리즘을 이용하여 계산한 결과이다.

1. 학습을 완료한 이후에도 풍부한 표현력을 지닐 정도로 작은 값의 PCR rate를 두 개를 정한다.
2. 동일한 학습 데이터로 두 라이브러리에 학습을 시킨다.
3. 조건부에 해당하는 변수와 값, 원하는 확률을 나타내는 변수들과 값들에 해당하는 DNA들을 모두 두 라이브러리에서 꺼낸다.
4. 더 큰 PCR rate를 사용한 라이브러리에서 얻은 DNA 수를 N_A, 작은 PCR rate를 사용한 라이브러리에서 얻은 DNA 수를 N_B 라고 할 때, $N_A / N_B - 1$ 값을 계산하고 같은 변수들끼리 정규화 한다.
5. 정규화된 $N_A / N_B - 1$ 값이 찾고자 하는 확률이 된다

그림2 : 조건부 확률 계산 알고리즘1 (라이브러리 2개 사용)

확률	시뮬레이션 결과	실제 확률	오차
P(x22=0 | y=0)	0.9711553	0.9654255	0.00573
P(x22=1 | y=0)	0.0288447	0.0345745	0.00573
P(x30=0 | y=0)	0.9343055	0.9281915	0.00611
P(x30=1 | y=0)	0.0656945	0.0718085	0.00611
P(x22=0, x30=1)	0.0311807	0.032958	0.00178
P(x22=1, x30=0)	0.0370399	0.0389746	0.00194
P(x22=1, x30=1)	0.0236869	0.0251112	0.00142
P(x22=0, x30=0 | y=0)	0.9162426	0.9069149	0.00933
P(x22=0, x30=1 | y=0)	0.0549653	0.0585106	0.00355
P(x22=1, x30=0 | y=0)	0.0183223	0.0212766	0.00295
P(x22=1, x30=1 | y=0)	0.0104700	0.0132979	0.00283

표 2: 조건부 확률 계산 알고리즘1의 시뮬레이션 결과. 두 개의 라이브러리를 사용. 실제 확률과 결과는 1% 이내의 오차 안에서 동일하다.

4. 은닉 확률 라이브러리 모델을 이용한 조건부 확률 계산 알고리즘

은닉 확률 라이브러리 모델(Latent PLM)은 표현할 변수 외에도 별도의 추가 변수를 둘 수 있는 모델을 의미한다.

3절에서 제시한 알고리즘은 DNA 개수를 세어야 하고, 두 개의 라이브러리를 사용해야 하는 불리함이 있다. 이 두 문제점은 은닉 PLM을 도입하여 제거할 수 있다.

먼저, 변수 Z_1을 추가로 도입한다. 이 변수는 Class 변수 y와 같이 모든 라이브러리에 포함되며, 값은 0 또는 1의 이진값을 갖게 한다. 단, 이 과정으로 인해 라이브러리의 복잡도는 2배 증가한다. 그림 3에 은닉 PLM을 도입한 알고리즘을 제시한다.

표 3과 같이 은닉 PLM을 도입한 확률 계산 알고리즘2도 조건부 확률 계산 알고리즘1과 같은 수준의 성능을 보이고 있다. 그러나, 은닉 PLM을 도입한 조건부 확률 계산 알고리즘2는 한 개의 라이브러리만 사용해도 되며, 단지 Z_1=0인 DNA수와 Z_1=1인 DNA수의 비율만 알면 되므로 더욱 우수하다.

1. 학습을 완료한 이후에도 풍부한 표현력을 지닐 정도로 작은 값의 PCR rate를 두 개(A_1, A_2)를 정한다. (단, $A_1 > A_2$)
2. 학습 데이터를 하나의 라이브러리에 학습을 시킨다. 단, Z_1=0인 것에 A_1의 학습률을 적용하고, Z_1=1인 것에 A_2의 학습률을 적용한다
3. 조건부에 해당하는 변수와 값, 원하는 확률을 나타내는 변수들과 값들에 해당하는 DNA를 모두 라이브러리에서 꺼낸다
4. 과정 3에서 얻은 Z_1=0인 DNA 수를 N_A, Z_1=1인 DNA 수를 N_B 라고 할 때, $N_A/N_B - 1$ 연산을 하고 같은 변수끼리 정규화를 한다.
5. 정규화된 $N_A/N_B - 1$ 값이 찾고자 하는 확률이 된다.

그림 3: 조건부 확률 계산 알고리즘2 (은닉 PLM 도입)

확률	시뮬레이션 결과	실제 확률	오차
P(x22=0 | y=0)	0.9711553	0.9654255	0.00573
P(x22=1 | y=0)	0.0288447	0.0345745	0.00573
P(x30=0 | y=0)	0.9343055	0.9281915	0.00611
P(x30=1 | y=0)	0.0656945	0.0718085	0.00611
P(x22=0, x30=1)	0.0311807	0.032958	0.00178
P(x22=1, x30=0)	0.0370399	0.0389746	0.00194
P(x22=1, x30=1)	0.0236869	0.0251112	0.00142
P(x22=0, x30=0 | y=0)	0.9162426	0.9069149	0.00933
P(x22=0, x30=1 | y=0)	0.0549653	0.0585106	0.00355
P(x22=1, x30=0 | y=0)	0.0183223	0.0212766	0.00295
P(x22=1, x30=1 | y=0)	0.0104699	0.0132979	0.00283

표 3: 조건부 확률 계산 알고리즘2의 시뮬레이션 결과. 은닉 PLM 사용. 실제 확률과 결과는 1% 이내의 오차 안에서 동일하다.

5. 결론

본 논문에서는 확률 라이브러리 모델에서 임의의 조건부 확률을 계산하는 알고리즘을 두 가지 제시하였다. 두 방안의 성능은 거의 동일하였으나, 은닉 PLM을 활용하는 편이 실험단계를 줄인다는 면에서 더 우수하다.

감사의 글

이 논문은 교육부 BK21 사업 및 과학기술부 국가지정연구실사업(NRL)에 의하여 지원되었음.

참고문헌

[1] B.-T. Zhang and H.-Y. Jang, Molecular learning of wDNF formulae, *Preliminary Proceedings of the Eleventh International Meeting on DNA Computing (DNA 11)*, pp. 185-195, 2005.

[2] B.-T. Zhang and H.-Y. Jang, A Bayesian algorithm for in vitro molecular evolution of pattern classifiers, *Lecture Notes in Computer Science*, 3384:458-467, 2005.

[3] B.-T. Zhang and H.-Y. Jang, Molecular programming: evolving genetic programs in a test tube, *The Genetic and Evolutionary Computation Conference (GECCO 2005)*, vol. 2, pp. 1761-1768, 2005.

[4]장병탁., 바이오분자 컴퓨터 기술, *물리학과 첨단기술*, 12(5):13-19, 2003

[5] ftp://ftp.ics.uci.edu/pub/machine-learning-databases/optdigits/