은닉 확률 라이브러리 모델에서의 조건부 확률의 계산

허민오° 장병탁

서울대학교 컴퓨터공학부

moheo@bi.snu.ac.kr°, btzhang@snu.ac.kr

Computing Conditional Probabilities in a Latent Probabilistic Library Model

Min-Oh Heo° and Byoung-Tak Zhang

School of Computer Science and Engineering, Seoul National University

요 약

확률 라이브러리 모델(Probabilistic Library Model)은 DNA컴퓨팅 방법론에 기반하여, 라이브러리를 구성하는 원소들의 빈도를 이용하여 결합확률분포를 표현하고자 하는 모델이다. PLM에서 결합확률분포 외에도 조건부 확률을 계산하는 방법이 필요해짐에 따라, 본 논문에서는 *in-vitro*에서 DNA를 이용하여 임의의 조건부 확률을 계산하는 방법으로 은닉 확률라이브러리모델(Latent Probabilistic Library Model)을 이용한 방법을 제시하고, 이전 논문에서 미비한 부분인 알고리즘의 타당성 증명을 보완하였다. 또한, 시뮬레이션을 통하여 실제 확률과 1% 이내로 동일한 결과를 얻었다.

1. 서론

확률 라이브러리 모델(Probabilistic Library Model)은 DNA컴퓨팅 방법론에 기반하여[4], 라이브러리를 구성하는 원소들의 빈도를 이용하여 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 수행할만한 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 비교적 높은 신뢰도로 발견할 수 있고 화학적 안정성이 있으므로 연산모델의 원소로 사용하기에 적절하다.[5]

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 DNA 구조를 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 되고[1], 연산과정에 Dilution 과정이 있어 증폭된 원소의 정확한 개수를 알 수 없으므로[5], 학습된 라이브러리로부터 임의의 변수에 대한 조건부 확률을 계산하는 것은 어렵다. 그래서, 본 논문에서는 이러한 계산을 수행하고자, 임의의 조건부 확률을 계산하는 방법과 증명을 제시한다.

2절에서는 PLM 시뮬레이션 조건과 매개변수에 대해 언급하고 3절에서 은닉 확률 라이브러리 모델을 이용한 조건부 확률을 계산하는 방법을 제시하고 그 타당성을 보인 후, 시뮬레이션 실험결과를 보여준다. 그리고, 마지막 4절에서 결론을 제시한다.

2. PLM의 도입 및 사용된 매개변수[5]

실제로 분자를 직접 이용하여 실험실에서 실험하기에는 비용과 시간이 많이 소모되므로, 적절한 제약을 가한 상태에서 시뮬레이션 실험으로 대체하였다. 시뮬레이션을 위해 세운 네 가지의 가정들을 아래에 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, 점힘이나 헤어핀과 같은 2차구조로 인해 성질이 변하거나 다른 쌍이 생기는 문제는 일어나지 않는다. 또한, 분자의 성질은 위치 좌표에 완전히 무관하며 상보적인 쌍 외에는 다른 힘 - 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, 라이브러리에서 각각의 조각난 X_i 값들에 해당하는 DNA를 찾아 적정 비율로 증폭을 시킨다. 실제 PCR 실험에서는 n회의 PCR 사이클에 따른 2^n 배의 증가만이 가능한데, 전체 라이브러리 용액에서 적정 부피만 덜어내어 PCR 과정을 거치고 다시 그 결과물을 본래 라이브러리 용액에 첨가하는 것으로 원하는 수치의 곱셈 연산을 수행할 수 있다. 이 실험에서는 1.00002, 1.000002의 두 PCR rate를 사용하였다.

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

Dilution : PCR rate에 따라 계속 개수를 늘리기만 하면 학습이 진행됨에 따라 용기 내의 분자수가 무한히 증가한다. 이러한 일은 실제로 실험 상에서 불가능하다. 그러므로, Dilution을 통해 적정 수의 분자를 버려서 전체 분자 개수를 유지하도록 조정을 해야 한다. 실험상에서도, 모든 라이브러리의 분자가 섞여있는 전체 용액에서, 일괄적으로 일부를 버리고 그만큼 다시 용매를 채워, 분자의 숫자를 유지하는 것이 쉽다. 이 실험에서는 다음의 식을 모든 라이브러리 내 원소의 수에 곱하는 것으로 표현할 수 있다.

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

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.[1] 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.

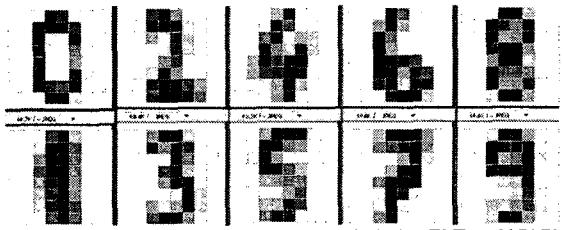

그림 1: 클래스 별 Digit 데이터의 평균 이미지. 0~255의 흑백이미지로 1과 0의 빈도를 나타내었다.

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

표 1: 사용된 매개변수와 값

상기 사용된 전략과 매개변수들을 요약하여 표 1 에 정리하였다.

3. 은닉 확률 라이브러리 모델을 이용한 조건부 확률 계산 알고리즘

3.1 알고리즘[5]

은닉 확률 라이브러리 모델(Latent PLM)은 표현할 변수 외에도 별도의 추가 변수를 둘 수 있는 모델을 의미한다. 변수 Z_1을 추가로 도입하여, 모든 라이브러리 원소에 포함시키고 그림 2에서 명시한 방법을 따른다.

1. 학습을 완료한 이후에도 풍부한 표현력을 지닐 정도로 작은 값의 동일하지 않은 PCR rate를 두 개(r_A, r_B) 정한다. (단, $r_A > r_B$)
2. 학습 데이터를 하나의 라이브러리에 학습을 시킨다. 단, Z_1=0인 것에 r_A의 PCR rate를 적용하고, Z_1=1인 것에 r_B의 PCR rate를 적용한다.
3. 조건부에 해당하는 변수와 값, 원하는 확률을 나타내는 변수들과 값들에 해당하는 원소를 모두 라이브러리에서 꺼낸다.
4. 과정 3에서 얻은 Z_1=0인 원소 수를 N_A, Z_1=1인 원소 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 2: Latent PLM을 이용한 조건부 확률 계산 알고리

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

증

3.2 알고리즘 증명

S 를 원하는 조건부 확률에 기여하는 항들을 모두 가진 집합이라 하고, 조건 C_i는 집합 S 에 속하는 i번째 원소, 이에 따라 생기는 각 조건 C_i 마다 증폭되는 횟수를 α_{Ci} 라 하자. 그리고, 과정 4 와 같이, 각 Z_1=0, Z_1=1 인 원소를 각각 윗첨자 A, 윗첨자 B 를 써서 표시한다.

학습 데이터에 따라 r_A, r_B의 PCR rate 로 α_{Ci} 번 증폭하면 각각 식 2, 식 3 과 같이 표현할 수 있다.

$$N^{A}_{new,C_i} = N^{A}_{init,C_i} \cdot r_A^{\alpha_{C_i}} \quad \text{(식 2)}$$

$$N^{B}_{new,C_i} = N^{B}_{init,C_i} \cdot r_B^{\alpha_{C_i}} \quad \text{(식 3)}$$

이 때, 모든 초기값은 동일하다는 조건을 고려하여, 조건 C_k 라는 조건에 부합하는 확률을 알고리즘의 3, 4, 5 번 과정에 따라 표현하면 다음과 같다.

$$P(C_k) = \frac{\frac{N^{A}_{init,C_k} r_A^{\alpha_{C_k}}}{N^{B}_{init,C_k} r_B^{\alpha_{C_k}}} - 1}{\sum_{C_i \in S} \left(\frac{N^{A}_{init,C_i} r_A^{\alpha_{C_i}}}{N^{B}_{init,C_i} r_B^{\alpha_{C_i}}} - 1\right)} = \frac{\left(\frac{r_A}{r_B}\right)^{\alpha_{C_k}} - 1}{\sum_{C_i \in S} \left\{\left(\frac{r_A}{r_B}\right)^{\alpha_{C_i}} - 1\right\}} \quad \text{(식 4)}$$

여기서, $\frac{r_A}{r_B} \approx 1$ 이므로, $\left(\frac{r_A}{r_B}\right)^{\alpha_{Ck}}$ 에 대하여 테일러 전개를 하면 다음과 같은 근사값을 얻을 수 있다.

$$P(C_k) \approx \frac{1 + \alpha_{C_k} \ln \frac{r_A}{r_B} - 1}{\sum_{C_i \in S} \left\{1 + \alpha_{C_i} \ln \frac{r_A}{r_B} - 1\right\}} \approx \frac{\alpha_{C_k}}{\sum_{C_i \in S} \alpha_{C_i}} \quad \text{(식 5)}$$

식 5 에서 볼 수 있듯이 알고리즘 연산의 결과가 상대적 빈도를 표현하고 있다. 이는 r_A / r_B 가 1 에 가까울수록 통계적 확률 정의에 의한 확률에 가까워진다.

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 2: Latent PLM 을 이용한 조건부 확률 계산 알고리즘의 시뮬레이션 결과. 실제 확률과 결과는 1% 이내의 오차 안에서 동일하다.

3.3 실험결과

이 알고리즘은 뺄셈 연산과 정규화 과정을 거쳐야 하므로, 외부적 별도 연산의 도입이 불가피하다. 하지만, 원소 수 N_A와 N_B를 직접 실험실에서 얻는 것은 거의 불가능한데 비해, N_A/N_B 라는 원소 수의 비는 실험적인 측정이 가능하므로 이를 통해 조건부 확률을 계산할 수 있게 되었다. 이 알고리즘도 이 부분에 중점을 두어 개발되었다. 표 2 는 임의로 선택된 변수인 x22 와 x30, y 를 변수로 두고 위의 알고리즘을 이용하여 계산한 결과이다. 실제 확률과 알고리즘을 통해 얻은 결과가 1% 오차 이내에서 동일하다. 실제 확률과의 오차는 테일러 전개에서 근사값을 구했기 때문에 발생한다. 테일러 전개 시 2 차항 이후는 무시하였는데, 이 때 무시된 항들로 인하여 오차가 나타난다.

4. 결론

본 논문에서는 DNA를 이용한 분자 컴퓨팅의 한 모델인 확률 라이브러리 모델 상에서 계산하기 어려웠던 조건부 확률을 계산하는 방법을 은닉 확률 라이브러리 모델을 도입하여 조건부 확률 계산 알고리즘을 제시하였고, 알고리즘의 타당성을 보였다. 시뮬레이션 실험 결과 1% 오차 이내에서 실제 확률과 동일한 값을 얻었다. 실제 실험 상에서, 두 개의 PCR rate를 선정하는 문제가 중요할 것이다. 두 PCR rate 값이 동일하지 않으면서도 두 값의 비가 1에 가까워야 하고, 실험 수행 상의 난이도가 최대한 적어야 하기 때문이다.

감사의 글

이 논문은 과학기술부 국가지정연구실사업(NRL)에 의하여 지원되었음.

참고 문헌

[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] 허민오, 장병탁, 확률 라이브러리 모델 상에서의 조건부 확률 계산 방법, 제 2회 컴퓨테이셔널 인텔리젼스 합동학술대회, pp.290-294, 2006