확률라이브러리모델 기반의 Hypernetwork 학습에 의한 패턴완성시의 Subsampling 효과 분석

김주경° 장병탁
서울대학교 컴퓨터공학부
{jkkim°, btzhang}@bi.snu.ac.kr

Analysis of Subsampling Effects in Pattern Completion by Hypernetwork Learning Based on Probabilistic Library Model

Joo-Kyung Kim°, Byoung-Tak Zhang
School of Computer Science and Engineering, Seoul National University

요 약

패턴완성(Pattern Completion)은 사용되는 패턴 성분들 사이의 higher-order correlation 정보가 중요한 의미를 가질 수 있는 기계학습 문제 중 하나이다. higher-order correlation은 확률라이브러리모델(Probabilistic Library Model)로 구현되는 hypernetwork 개념을 도입해서 나타낼 수 있다. 하지만 확률라이브러리모델을 사용하여 higher-order 정보를 나타내려할 때 초기라이브러리가 모든 가능한 조합의 원소들을 가지도록 구성하기는 쉽지 않다. 그 대안으로 초기라이브러리 구성 시 학습패턴들을 subsampling하여 적은 숫자의 원소들만으로 higher-order correlation의 근사치를 나타내게 할 수 있다. 본 논문에서는 이와 같이 subsampling이 사용되어 구성된 확률라이브러리모델을 이용한 패턴완성시의 correlation의 order에 따른 효과를 분석하여 본다.

1. 서 론

패턴완성이란 기본 패턴들을 입력 받아 학습한 후 일부분 훼손된 패턴이 주어졌을 때 정상적인 패턴의 모양에 가깝도록 복원하는 것으로서 대표적인 방법들로 neural network기반인 Hopfield network [1]이나 Boltzmann machine [2] 등이 있다. 패턴완성을 효과적으로 수행하려면 주어지는 패턴의 성분들 사이의 higher-order correlation 정보가 학습되고, 사용될 수 있어야 하는 경우가 많다 [3]. 하지만 기존의 neural network 기반의 방법들을 사용하려면 입력공간이나 적용할 correlation의 order가 증가함에 따라 기하급수적인 메모리 및 계산시간이 필요하다. 본 논문에서는 패턴의 성분들 간의 효과적인 higher-order correlation 모델링을 가능하게 하는 hypernetwork 모델과 이를 기본적인 연산과 대규모 병렬처리를 통해 구현할 수 있는 확률라이브러리모델(Probabilistic Library Model, PLM) [4]을 소개하고 패턴완성시 적절한 correlation order에 대한 분석을 해본다.

2. Hypernetwork 모델

Hypergraph [5]는 하나의 edge가 1이상 임의의 수의 vertices를 가질 수 있는 graph이다. 즉 $G=(V,E)$일 때 $V=\{v_1,v_2,..,v_n\}$, $E=\{E_1,E_2,..,E_m\}$, $E_i=\{v_{i1},v_{i2},..,v_{ik}\}$ 형태로 나타내진다. 이때 E_i는 hyperedge이고 $k\geq 0$의 cardinality를 가지는 V의 부분집합이다. 그림 1은 7개의 vertices, 5개의 hyperedges로 구성되는 hypergraph의 예이다.

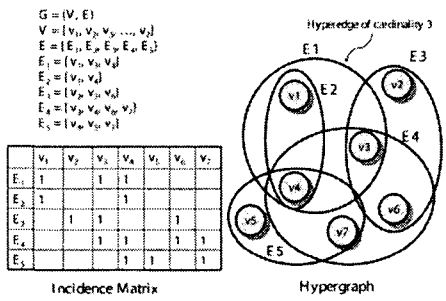

그림 1. hypergraph 예제

Hypergraph의 각 hyperedges에 weight가 도입된 것을 hypernetwork이라 한다. 여기서 패턴의 성분들을 vertices로, 여러 성분들 간의 k-order correlation을 hyperedge에 포함된 vertices와 그것의 weight값으로 대응시킴으로서 hypernetwork으로 k-order correlation을 나타낼 수 있다.

3. Hypernetwork의 PLM기반 표현

어떤 hypernetwork이 $V=\{v_1,v_2,..,v_n\}$, $E=\{E_1,E_2,..,E_m\}$, $E_i=\{v_{i1},v_{i2},..,v_{ik}\}$로 나타내어 질 때 E를 library, E_i들을 library의 원소들로 대응시켜서 PLM으로 구현할 수 있다. 본 논문에서는 모든 E_i들의 cardinality는 k로 동일, 즉 k-uniform 하도록 정한다. PLM 모델에서 hyperedges의 weight는 동일한 형태의 hyperedges의 개수로 나타낼 수 있다. 예를 들어 E_1의 weight가 E_2의 weight보다 3배 크다는 걸 표현하려면 library상에서 E_1 형태의 원소수가 E_2의 원소수보다 3배 많이 존재하게 만들면 된다. 즉 hypernetwork상에서 weight값들의 변화를 library안의 특정원소들의 비율 증감으로 나타낼 수 있다.

4. PLM기반 Hypernetwork의 학습 및 패턴완성 수행

다음은 PLM을 이용한 Hypernetwork의 학습, 패턴완성의 수행과정이다. 이 때 훈련패턴은 0이나 1을 성분값으로 가지는 벡터형태이다.

-1. 각 훈련패턴들에 대해 k개의 성분값을 추출해내는 subsampling을 n번 수행하여 cardinality가 k인 library원소들을 $n\times$훈련패턴수 만큼 가지는 초기 library를 구축한다.

-2. 훈련패턴과 library원소들을 비교해서 library원소에 포함되어 있는 성분값들이 훈련패턴의 해당 성분값들과 t_1 이하의 오차이내에서 동일한 원소들은 복제를 통해 개수를 증가시킨다. 이 과정을 모든 훈련패턴에 대해 반복한다.

-3. 패턴완성을 수행할 대상 패턴들을 입력받아서 library원소들과 비교해서 library원소에 포함되어 있는 성분값들이 입력된 패턴의 해당 성분값들과 t_2 이하의 오차이내에서 동일한 원소들만 남기고 그렇지 못한 원소들은 제외시킨다.

-4. 남아있는 library원소들을 분석해서 각 성분별로 해당 성분의 값이 1인 원소들의 개수를 구한다. 이 때 값이 1인 원소들의 개수가 상대적으로 많은 성분은 패턴완성의 결과물내의 해당 성분의 값이 1인 확률을 높게 되고 그렇지 못한 성분은 값이 0이 될 확률이 높게 된다.

1번 과정에서 library를 초기화 할 때 전체 입력공간내에서 가능한 모든 조합의 성분들을 library원소로 갖고 있도록 하는 것이 이상적일 것이다. 그래야 지정된 cardinality에서 가능한 모든 경우의 correlation에 대한 정보를 가질 수 있기 때문이다. 하지만 가능한 모든 조합의 성분을 포함하도록 하려면 library의 원소수가 $2^k\times {}_dC_k$ (d는 입력공간의 차원수, k는 cardinality) 만큼 필요하므로 d나 k가 커지는 경우 컴퓨팅 파워의 제약 때문에 구현이 쉽지 않다. 대신에 초기 library를 훈련패턴들의 subsampling을 통해 구성할 경우 적은 수의 원소수만 갖게 되더라도 실제로 훈련패턴에서 쓰이고 있는 correlation에 대한 정보는 상당부분 가지고 있도록 초기화 되므로 실제 패턴완성 시에도 필요한 주요 correlation들은 가지고 있게 초기화하는 효과가 있다.

2,3번 과정에서 일정오차를 허용한 상태에서 주어진 패턴과 library원소가 일치하게 되는 경우 해당 library원소를 증가 또는 패턴완성 결과에 반영시키는 것은 일종의 일반화 또는 feature extraction의 과정이라고 볼 수 있다. 만약 오차를 허용하지 않을 경우 패턴완성 수행이 단지 훈련패턴들을 메모리에 저장해놓고 입력패턴과 가장 닮은 훈련패턴들의 평균을 결과물로 내놓는 단순한 형태가 될 것이다. 그렇게 되면 패턴완성 수행 시 훈련패턴들과 거의 동일한 형태의 입력패턴에 대해서는 잘 작동하지만 훈련패턴들과 어느 정도 형태의 차이가 나는 패턴이 입력되는 경우에는 입력과 비슷한 훈련패턴을 찾기가 어려우므로 제대로 작동하기 힘들 것이다.

5. 시뮬레이션

0부터 9까지의 클래스를 가지는 UCI machine learning deposit의 Optical Recognition of Handwritten Digits[1]의 총 3876개의 비트맵 데이터에 대하여 훈련패턴의 분포도를 동일하도록 3760개를 선정하여 64비트 이진벡터로 변환한 후 훈련시키고 총 1797개의 테스트 데이터를 입력패턴으로 넣어서 패턴완성을 수행시켜본다. 시뮬레이션에서는 총 64개의 입력공간내의 성분 중 실제로 의미 있는 49개의 성분만 selection하여 사용하였다.

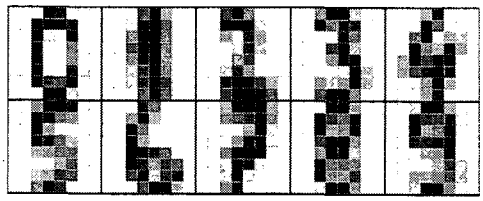

그림 2. 훈련패턴들의 클래스 별 평균 이미지

그림 2는 8×8 행렬로 나타내 본 훈련패턴들의 각 클

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

래스 별 평균 이미지이다. 여기에서 진하게 표시되는 성분일수록 훈련패턴들 내에서 성분값이 1인 확률이 높은 것이다.

그림 3은 test데이터에 대해 일정비트수 만큼 훼손시켜 만든 패턴을 입력으로 넣고 패턴완성 수행 후 원래 패턴과의 차이를 나타낸 것이다. 수행과정에서 알고리즘의 2,3번 단계에서 필요한 허용오차들은 cardinality의 20%에 해당하는 값을 적용하였다. x축은 훼손정도, y축은 원래패턴과의 차이. 즉 에러의 정도가 된다. 각 그래프는 library원소의 cardinality에 따른 결과의 차이를 나타낸다. 그리고 cardinality가 다르더라도 사용되는 전체 메모리공간은 동일하게 되도록 cardinality가 높은 경우에는 초기library 생성 시 한 훈련패턴에 대해 subsampling 되는 횟수를 제한하였다. (제한하지 않는 경우 높은 cardinality의 그래프들의 기울기가 약간 완만해지지만 기본적인 경향은 그림3과 동일하다.) 그림 3에서 볼 수 있듯이 원본패턴으로부터 훼손의 정도가 적을 때에는 높은 cardinality를 사용할 때 결과의 에러가 적음을 알 수 있다. 하지만 훼손의 정도가 커질수록 높은 cardinality를 사용한 경우에 에러가 급격히 커짐을 볼 수 있는데 이는 입력패턴에 훼손이 많이 된 경우 cardinality가 너무 높으면 입력패턴과 일정오차 안에서 일치되는 library원소를 찾기가 어려워져서 제대로 된 결과를 만들기 어려워지기 때문이다.

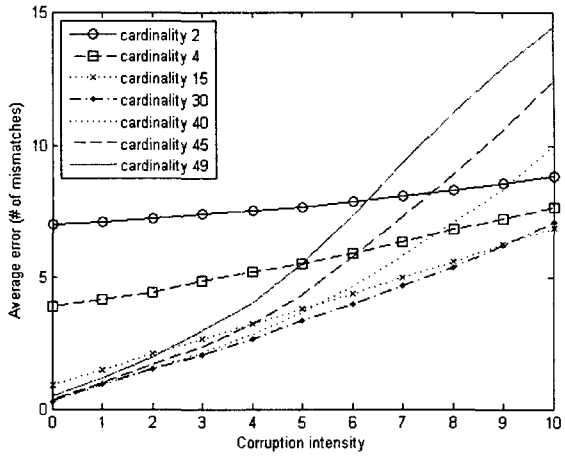

그림 3. 입력패턴의 훼손정도에 따른 패턴완성 결과의 에러

6. 결론

본 논문에서는 PLM을 기반으로 hypernetwork을 구성하여 패턴완성을 수행하는 방법을 제시하였다. PLM을 사용 시 이상적으로는 cardinality가 어느 정도 높으면서도 모든 가능한 조합의 library원소가 library내에 존재하도록 해야 가장 패턴완성의 성능이 뛰어날 것으로 예상한다. 단 적정 수준 이상의 큰 cardinality는 별 의미가 없을 것으로 예상되는데 그 정도의 cardinality를 가지면, 즉 그 정도 order의 correlation 정보를 가지면 주어진 패턴들에 대해 패턴완성을 제대로 수행하기 충분하기 때문이라고 생각할 수 있다. 하지만 실제 구현 시에는 컴퓨팅 파워의 제약으로 인해 모든 가능한 조합의 library 원소를 가지도록 할 수 없었고 이에 대한 대안으로 훈련패턴들을 subsampling하여 초기화 하는 방법을 사용하였다. 이 경우 일반화 측면에서 문제가 발생하지만, 즉 입력패턴의 훼손정도가 심할수록 패턴완성의 성능이 떨어지게 되지만 훼손이 적은 편인 경우 효과적으로 패턴완성을 수행할 수 있다. 즉 입력공간의 크기, 입력패턴들의 분포나 훼손정도에 따라 최적의 cardinality를 찾아서 패턴완성을 수행해야 할 것이다. 그리고 이때의 cardinality가 곧 해당 문제에 대해 패턴완성을 수행 시에 적절한 correlation의 order가 될 것임을 예상할 수 있다. 또한 PLM은 애초에 DNA Computing [6]의 강력한 대규모 병렬연산 기능을 이용하는 것을 염두로 만들어진 모델이므로 DNA Computing을 이용하여 컴퓨팅 파워에 덜 제약적인 조건에서 패턴완성을 수행하게 되는 것을 기대할 수도 있다.

감사의 글

이 논문은 과학기술부 국가지정연구실사업(NRL)에 의하여 지원되었음

7. 참고문헌

[1] Hopfield, J.J., "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences, USA*, vol.79, 1982, pp. 2554-2558.

[2] Hinton, G.E., Sejnowski, T. J., "Learning and Relearning in Boltzmann Machines," In D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*, Cambridge: MIT Press, 1986, pp. 282-317.

[3] MacKay, David J.C., *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003.

[4] Zhang, B.-T. and Kim, J.-K., "DNA Hypernetworks for Information Storage and Retrieval", *DNA12*, 2006. (accepted)

[5] Berge, C. *Graphs and Hypergraphs*, North-Holland Publishing, Amsterdam, 1973.

[6] Adleman, "Molecular computation of solutions to combinatorial problems", *Science*, 266:1021-1024. (Nov. 11). 1994.