구조학습 기반의 서열 데이터 재현 기법

김병희○ 장병탁
서울대학교 컴퓨터공학부
bhkim@bi.snu.ac.kr, btzhang@bi.snu.ac.kr

Structure Learning Method for Recalling Sequences

Byoung-Hee Kim○ Byoung-Tak Zhang
School of Computer Science and Engineering, Seoul National University

요 약

본 고에서는 서열 내의 구조 학습 및 패턴 매칭에 기반한 서열 집합 재현 기법을 제안한다. 제안하는 기법은 인지과정에 대한 모델링 기법이며, 다양한 길이의 서열 조각의 집합으로 메모리가 형성되고 재현되는 과정을 각각 학습 및 추론/예측 단계로 포함한다. 특히, 반복 학습을 통한 정보 압축 및 특징적 패턴의 자동 추출을 특징으로 하는 기법이다. 멜로디 학습 및 재현 실험을 통해, 제안 기법의 유용성을 보이며, 학습된 모델 분석을 통해 결과에 대한 해석 및 핵심 정보 추출이 가능함을 보인다.

1. 서 론

순차적 또는 동적인 패턴에 대한 모델링 기법은 인공지능 및 기계학습 분야의 지속적인 연구 주제이자, 컴퓨팅 자원의 기하급수적 증가에 따라 다양한 분야에서 활발히 연구되는 주제이다[1]. 이산적 서열 데이터를 주로 다루는 자연언어 처리 분야와[2] 음악 인지 분야[3]에서도 관련 연구가 활발하다. 특히, 최근에는 뇌 자체가 근본적으로는 예측 기계로 보는[4] 연구의 흐름에 따라 더욱 주목받고 있다.

그러나, 여전히 많은 연구가 예측 성능에 초점을 주로 초점을 두었으며, 상대적으로 예측성능을 설명하기 위한 특징적 패턴 파악 측면은 주목을 받지 못하였다.

본 고에서는 다양한 길이의 서열 조각으로 구성된 메모리 모델을 기반으로 한 서열 데이터 재현 기법을 제안한다. 반복 학습을 통해 서열 데이터의 특징적 구조를 자동으로 추출하며 정보를 압축하고, 해석 가능한 학습 결과 모델을 구축한다. 음악 멜로디 학습에 적용하여 재현 성능과 기법의 정량적, 정성적 특징을 정리한다.

2. 제안 기법

2.1 관련 연구

서열 생성을 위한 가장 직관적인 모델은 이전 서열의 문맥을 고려하여 다음에 올 심볼을 예측하는 방식이다. 이는 유한한 알파벳 Σ로 구성된 서열에 대해 문맥 $s \in \Sigma^*$이후에 심볼 σ가 나타날 확률 $p(\sigma|s)$을 학습하는 문제로 표현된다. Shannon의 기념비적인 연구 이래[5], 언어 모델 및 알고리즘 작곡 분야에서는 다양한 *n*-그램 모델, 즉 *n*-1차 마코프 모델이 연구되어 왔다[2]. *N*-그램과 함께, 다양한 차수의 연관성을 고려하는 VMM (variable-order Markov model) 또한 활발하게 연구되어 왔으며, 빈도 계산, 스무딩(미관측 이벤트에 대한 확률 부여 방식) 및 가변 길이 모델의 세 요소[6]를 어떤 방식으로 구현하느냐에 따라 수많은 알고리즘이 개발되었다. 대부분의 알고리즘은 학습 데이터의 압축된 표현형으로서 접미어 나무(suffix-tree)를 기반으로 표현되며, 접미어 나무 전체를 유지하는 무손실 압축 방식과, 나무의 가지치기를 통해 모델을 간략화하는 손실 압축 방식으로 크게 구분할 수 있다.

본 논문에서 제안하는 서열 데이터 재현 기법은 VMM 계열의 손실 압축 계열의 알고리즘으로서도 해석 가능한 방식이며, 차별점은 베이지안 필터링 관점에서 예측-교정 순환(prediction-correction cycle) 요소를 추가한 점이다. 특히, 실세계 적응형 모델로서 점진적 학습 모델로의 용이한 확장성을 고려하였다.

2.2 제안하는 알고리즘

제안하는 기법과 기존 VMM과의 차이는 다양한 서열 조각의 학습 데이터에서의 관측 빈도를 기반으로 예측 확률을 계산하는 대신, 생성을 위한 학습(learning to generate)을 반복하는 과정에서 서열 조각의 가중치를 동적으로 변하도록 휴리스틱을 적용한 점이다.

제안하는 알고리즘은 그림 1과 같다.

10~12행의 예측 단계는 다음과 같은 세부 단계를 포함한

1 [**초기 모델 구성**]
2 - 서열 집합 $Q = \{q_i | q_i \in \Sigma^*\}$의 각 서열 q_i에서 추출
3 가능한 유일한 *n*-그램($2 \le n \le D$) 전체 Q_D중 일정
4 비율 L의 *n*-그램을 임의추출
5 - 각 *n*-그램에 기본 가중치(w_0) 부여
6 [**반복**] 다음의 부분적 학습/전역적 학습 조합을 일정
7 회수(epoch) 반복
8 [**부분적 학습**: 각 서열($q_i = x_{1:T_i}$)별 생성 위한 학습]
9 - $x_{1:D-1}$을 시작 문맥으로 설정
10 - 문맥 기반 패턴 매칭을 이용한 예측(prediction):
11 $\sigma^* = \arg\max_\sigma p(x_t = \sigma | s),\ s = x_{t-1:t-D+1}$,
12 $D \le t \le T_i$
13 - 원곡과 비교하여 예측에 참여한 *n*-그램의 가중치 교
14 정(correction)
15 [**전역적 학습**]
16 - 모델에 포함된 모든 *n*-그램의 가중치를 정규화
17 - 일정 비율(ρ)의 *n*-그램을 모델에서 제거하고 Q_D에
18 서 임의추출 후 기본 가중치 부여

그림 1. 학습 알고리즘

다: (1) 현재 모델 내에서 문맥의 접미어에 패턴이 매칭되는 모든 *n*-그램을 추출하여 집합 M 구성 (2) M으로 구성한 '접미어 나무(suffix tree)'를 기반으로 $p(x_t = \sigma|s)$를 PPM-C([6]) 방식으로 계산한다.

13~14의 교정 단계는 다음과 같은 세부 단계를 포함한다: (3) symbol이 일치하면$(x_t = \sigma^*)$ (1)에서 추출한 각 *n*-gram별로 길이에 비례하여 가중치 조정 $\triangle w = w_{t-1} + n$, (4) 일치하지 않으면$(x_t \neq \sigma^*)$ $\triangle w = w_{t-1} - n$

3. 서열 재현 실험 및 분석

3.1 실험의 목표

본 절에서는 2절에서 제안한 기법의 특성 및 성능을 파악하기 위한 실험에 대해 요약한다. 제안하는 기법의 특징으로 서열 집합의 손실형 정보 압축, 학습 과정 및 결과 모델의 해석 가능성을 둘 수 있다. 실제 데이터 셋 기반의 학습과정을 분석하여 복잡성, 모호성 하에서 서열 재현 성능 및 알고리즘의 특성을 파악하는 실험을 구성하였다.

실험은 학습 능력, 학습 과정 중 모델의 변화 분석 및 모델의 압축 성능 확인 과정으로 이어진다.

3.2 데이터 구성 및 실험 세팅

제안한 기법의 적용 예로서 멜로디 곡 모음의 재생성 예를 보인다. 실험 데이터로는 비틀즈의(Beatles) 곡 중 4/4박자 30곡을 선별하였다. 각 곡의 MIDI 파일에서 메인 멜로디를 추출하고, C장조(A단조)로 일괄 변경 후, 음 높이(pitch)와 음의 길이(duration)의 카테시안 조합으로서 멜로디를 단일 서열로 표현한다($\Sigma \subset R^p \times R^q$). 쉼표를 포함한 전체 길이는 9152개 음표이며, $|\Sigma| = 302$이다.

각 서열을 반으로 나누어, 앞쪽 반을 학습 데이터, 뒤쪽 반을 테스트 데이터로 지정하며, 적중률(hit ratio)로 성능을 평가한다. 많은 곡이 간주를 포함하는 전/후 반복 구조 또는 유사 구조를 가짐을 고려할 때, 이러한 설정은 약한 일반화 성능 평가 방법으로 볼 수 있다.

3.3 실험 결과 및 분석

반복 학습 과정 중의 성능 변화를 살펴본 결과, 학습 데이터 재현율 반복 학습을 통해 꾸준히 증가하며, 시험 데이터 재현율은 일정 시점 이후 하락하는, 전형적인 학습 곡선을 얻을 수 있었다. 그림 2(a)에 정리한 예에서는 대략 20회 반복 학습 후 '과다적응(overfitting)'이 발생하는 것을 볼 수 있다.

모델의 구조 변화 과정을 추적하여 과다적응 발생의 원인과 패턴을 보다 상세하게 살펴볼 수 있다. 그림 2(a)에서와 동일한 설정의 학습 과정에서 모델의 구조 변화를 살펴본 결과를 그림2(b)에 정리하였다. 20회 반복학습 이전과 이후의 차이가 극명히 드러남을 확인할 수 있다. 여러 곡에서 공통적으로 나타나는 '짧은 패턴(차수 2~4)'의 증가 추세가 이 시점에서 반대로 돌아서며 각 곡별로 특화된 패턴을 기억하기에 유리한 '긴 패턴(차수 7~10)'의 비중이 점차 커지는 것을 확인할 수 있다.

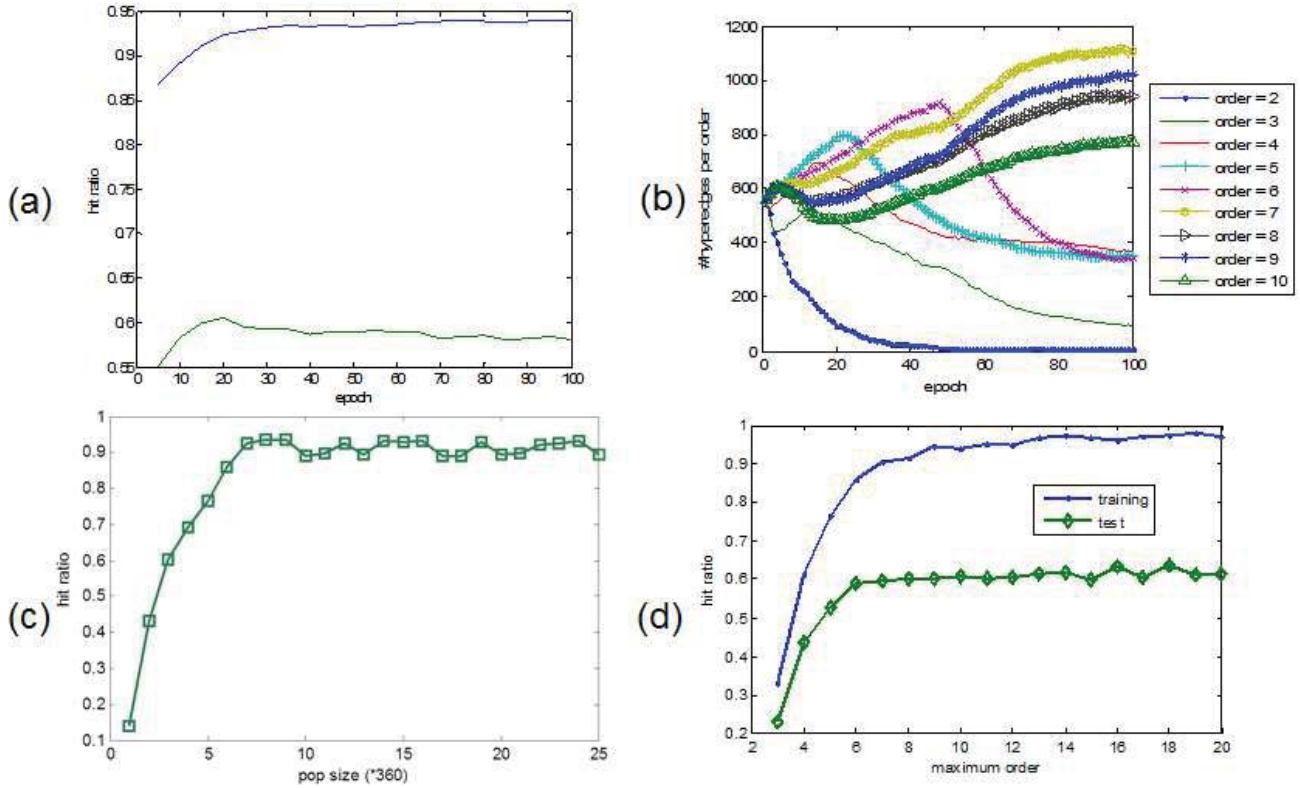

그림 2. 비틀즈(Beatles) 곡 중 선별한 30곡에 대한 멜로디 재현 실험 결과. (a)(b)학습 곡선 및 학습 과정에서의 모델의 구조 변화(압축률 22.5%, 최대 차수 10) (c)압축률에 따른 재생성 성능 변화(최대 차수 10) (d)최대 차수에 따른 성능 변화(압축률 22.5%)

압축률에 따른 성능 변화를 살펴본 결과 그림 2(c)와 같이 일정 압축률 이하에서는 큰 성능의 변화가 없음을 확인하였다. 이는 짧은 패턴의 특성을 킨 패턴이 포함한다는 점에서 이해할 수 있는 결과이며, 제안하는 기법을 통해 다양한 길이의 패턴이 자동으로 선별되고 적절한 조합 구성이 가능함을 볼 수 있는 결과이다.

지금까지의 결과에서 볼 수 있듯이, 최대 차수는 서열 재현 성능을 결정하는 매우 중요한 인자이다. 그림 2(d)에서 볼 수 있듯이, 최대 차수가 증가함에 따라 학습 데이터 재현율이 올라가는 것을 확인할 수 있다. 그러나, 시험 데이터 재현 측면에서는 최대 차수 8 이후는 큰 영향을 주지 않았다. 학습 데이터와 시험 데이터 간의 유사한 멜로디 패턴을 표현하는 데는 차수 4~6의 패턴이 결정적인 요인임을 파악할 수 있다.

4. 맺음말

본 논문에서는 서열 집합 저장 및 재현 모델로서 다양한 차수의 서열 조각의 조합 학습 및 재현 기법을 제안하였다. 제안하는 기법은 혼합 차수 마코프 모델 기반의 인지 메모리 구조 학습에 초점을 두었으며, 불균일한 서열 집합의 정보를 손실압축하는 기법으로도 볼 수 있다. 적용 예로서 비틀즈의 팝송 모음을 압축하고 재현한 결과를 살펴보았으며, 30곡의 시험 데이터 대비 0.6 정도의 재현율 및 과다학습 방지를 위한 학습 중단 기준을 확인하였다.

현재 모델은 많은 부분이 휴리스틱에 기초하고 하다. 보다 원리적인 모델 구축을 위해 정보병목기법[7]을 결합한 정보 압축 기법 및 서열 데이터의 지속적인 유입에 따른 동적인 학습 기법을 이어지는 연구로서 추진하고 있다.

참고문헌

[1] D. Barber, T. Cemgil, and S. Chappa (eds), “Bayesian Time Series Models,” Cambridge University Press, 2011.

[2] Y. Teh, A hierarchical Bayesian language model based on Pitman-Yor processes, In *Proceedings of the 21st International Conference on ACL/COLING*, pp. 985-992, 2006.

[3] M.A. Rohrmeier, S. Koelsch, Predictive information processing in music cognition. A critical review, *International Journal of Psychophysiology*, 83:164-175, 2012.

[4] A. Clark, Whatever next? Predictive brains, situated agents, and the future of cognitive science, *Behavioral and Brain Sciences*, 2012 (in press).

[5] C. E. Shannon. A mathematical theory of communication, *Bell System Technical Journal*, 27:379−423, 623−656, 1948.

[6] R. Begleiter, R. El-Yaniv, and G. Yona, On prediction using variable order Markov models, *Journal of Artificial Intelligence Research*, 22:385−421, 2004.

[7] N. Tishby, F.C. Pereira, and W. Bialek: “The Information Bottleneck method,” In *Proceedings of The 37th annual Allerton Conference on Communication, Control, and Computing*, pp. 368−377, 1999