



Improving Molecular Design by Stochastic Iterative Target Augmentation

☰ Category	Molecule
☰ BCI 관련	X
☰ Conference / Journal	
☰ Year	2020
☰ dataset	
☰ 작성 완료 or 작성중	작성완료
👤 작성자	 한지웅
☰ 비고	

Improving Molecular Design by Stochastic Iterative Target Augmentation

Generative models in molecular design tend to be richly parameterized, data-hungry neural models, as they must create complex structured objects as outputs. Estimating such models from data may be challenging due to the lack of

 <https://arxiv.org/abs/2002.04720>



Abstract

1. Molecular design 에서 생성 모델은 parameterized 의 경향이 크고, data 가 언제나 부족하다
2. 즉, Lack of sufficient 가 있다는 것
3. In this paper
 - a. Self-training 접근법을 이용했는데,
 - b. 반복적으로 추가적인 target molecule 을 붙여나갈 수 있도록 작업을 해주는 것
4. 방법으로는
 - a. Simple property predictor 를 이용해서 generative model 을 훈련
 - b. Property predictor 의 경우 likelihood model 로써 사용되는데 generative model 로 부터 생성된 후보들을 걸러주는 역할을 함

- c. 걸러주는 방법은 **stochastic EM iteration** 을 사용하여 **log-likelihood** 를 최대화 시키는 방법을 사용한다
5. 이러한 심플한 방법으로도 충분하다?

1. Introduction

1. 우리가 원하는 것 → 원하는 **property** 를 가지고 있는 **molecule** 을 만드는 것
2. 주로 Generative 모델을 사용하는데 이에 대한 단점으로는 매우 큰 **parameter** 가 요구 됨, 이는 복잡한 (그래프) 구조로 표현됨
3. 결론적으로 데이터가 많이 필요함
4. **Our challenge**
 - a. **Data-sparse** 한 상황에서도 높은 퀄리티의 분자를 생성하는 것
 - b. (본인들이 주장하기에) **Simple** 하고 놀랍도록 높은 효과를 보이는 **self-training 접근법** 을 만들었다고 한다
5. 저자들이 만든 **stochastic iterative target augmentation** 방법은 아래와 같다.
 즉, **input molecule** 이 들어오면, **생성모델을 이용해 새로 디자인 할 분자들의 후보를 뽑아주고, filter 를 이용해 후보자들을 걸러줌**. Model 의 input 이 없는 **unconditional setting** 에서는, **model** 과 **filter** 를 이용해 간단하게 **output** 을 내뿜는 것이다.

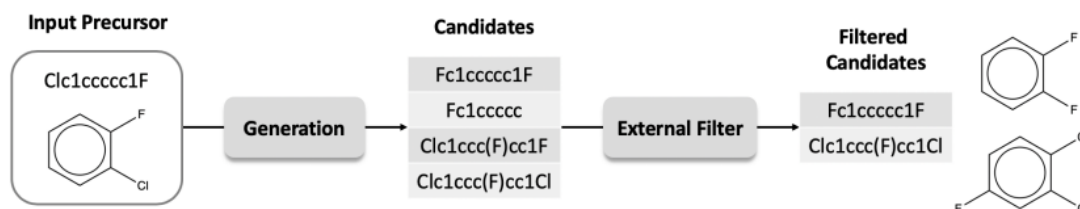


Figure 2. Illustration of data generation process for conditional molecular design. Given an input molecule, we first use our generative model to generate candidate modifications, and then select sufficiently similar molecules with high property score using our external filter. In the unconditional setting where the model takes no input, we simply sample outputs from the model and filter by property score.

그에 대한 방법을 살펴보자

- a. 먼저 **Property predictor** 를 이용해서 적은 양의 **supervised dataset** 대해 **generative model** 을 **pre-training** 하는 것으로 시작한다.
- b. 이렇게 훈련된 Property predictor 는 후보 물질들을 **filtering** 하는 **likelihood** 모델로써 사용된다.
- c. 이러한 filtering 과정을 거친 **Filtered candidate** 들은 이후 training epoch 를 위한 training 의 data 에 포함된다 → 이래서 **Self-training 접근법** 이라고 하는 듯
- d. 이론적으로 이러한 과정들은 **stochastic EM** 의 한 iteration 으로 보여질 수 있는데, 이는 로그라이클리 후드를 최대화 시키는 방법을 말한다.

6. 그래서 우리의 모델은 2가지 시나리오로 테스트 되는데,
 - a. Molecular generative modeling (Uncinditional molecular design) = **아예 땡으로 생성**
 - b. **Graph-to-graph translation** = 어떠한 특성을 가지고 있는 이미 존재하는 분자의 특징을 더 좋게 수정하는 conditional design problem 에 대한 것 (아래를 참조)

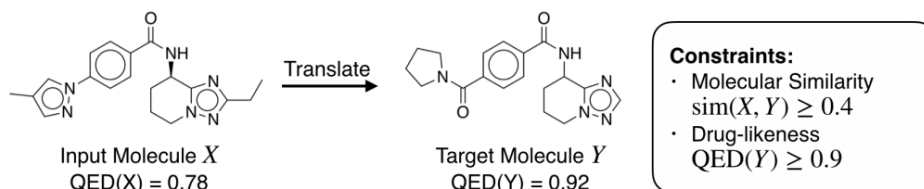


Figure 1. Illustration of conditional molecular design. Molecules can be modeled as graphs, with atoms as nodes and bonds as edges. Here, the task is to train a translation model to modify a given input molecule into a target molecule with higher drug-likeness (QED) score. The constraint has two components: the output Y must be highly drug-like, and must be sufficiently similar to the input X .

7. 마지막으로, (별로 안 중요 한 것 같긴한데) molecular domain 에만 한정된 것은 아니다

2. Stochastic Iterative Target Augmentation

1. 우리의 연구는 **주어진 분자 X 를 화학적 특징이 개선된 다른 물질 Y 로 변환하는 동시에, Y가 X 와 유사하게 유지되도록 하는 Conditional molecular design** 방법으로 부터 시작,
 - a. 이에 대응하는 Unconditional task 는 input 을 받지 않고 우리가 원하는 특징을 가진 분자만을 생성하려고 한다.
 - b. 우리의 모델은 input conditioning 을 없애는 것 또한 가능하다
2. 주어진 X 에 대해, 모델은 Y 를 생성하고자 함
 - a. 어떠한 제약 조건 c 를 가짐 $\rightarrow f(X, Y)$ 의 value 를 나타냄 \rightarrow 즉 $C = \text{constrain}$
 - b. ex) Conditional 상황에서, $c = 1$ 이라면, Y 는 X와 충분히 유사하면서 우리가 지정한 임계값 (property 를 말하는) 넘긴다는 뜻
 - c. 이러한 augmentation 방식은 이미 존재하는 Dataset $D = \{(X_i, Y_i)\}$ 를 이용해 어떠한 훈련된 translation model P 에도 적용 될 수 있음
 - d. 그림 2 에 나타난 것과 같이 모델은 2개의 단계가 반복적으로 나타나는 절차를 따름
3. Figure 2

Algorithm 1 Stochastic iterative target augmentation

Input: Data $\mathcal{D} = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$, model $P^{(0)}$

```
1: procedure AUGMENTDATASET( $\mathcal{D}, P^{(t)}$ )
2:    $\mathcal{D}_{t+1} = \mathcal{D}$   $\triangleright$  Initialize augmented dataset
3:   for  $(X_i, Y_i)$  in  $\mathcal{D}$  do  $\rightarrow$  모든 dataset sample 들에 대해
4:     for attempt in  $1, \dots, C$  do  $\rightarrow$  C번 동안 후보와 생성
5:       Apply  $P^{(t)}$  to  $X_i$  to sample candidate  $Y'$ 
6:       만족하는  $\left[ \begin{array}{l} \text{후보물질이} \\ \text{있다면} \end{array} \right. \text{if } c = 1 | X_i, Y' \text{ and } (X_i, Y') \notin \mathcal{D}_{t+1} \text{ then}$ 
7:         Add  $(X_i, Y')$  to  $\mathcal{D}_{t+1}$ 
8:       그 다음 if  $K$  successful translations added then
9:          $\left[ \begin{array}{l} \text{step에 추가} \end{array} \right. \text{break from loop}$   $\rightarrow$  k번만큼 성공하면 break 하므로
10:   return augmented dataset  $\mathcal{D}_{t+1}$ 

11: procedure TRAIN( $\mathcal{D}$ )
12:   for epoch in  $1, \dots, n_1$  do  $\triangleright$  Regular training
13:     Train model on  $\mathcal{D}$ .  $\rightarrow$  Generation을 위한 training
14:   for epoch in  $1, \dots, n_2$  do  $\triangleright$  Augmentation
15:      $\mathcal{D}_{t+1} = \text{AUGMENTDATASET}(\mathcal{D}, P^{(t)})$ 
16:      $P^{(t+1)} \leftarrow$  Train model  $P^{(t)}$  on  $\mathcal{D}_{t+1}$ .
```

a. Augmentation step

- i. 주어진 조건을 만족하면 Dataset 에 추가,
- ii. 만족하는 생성물질이 없다면 추가 X, 그냥 data를 두배로 뿔려줌

b. Training step

- i. 일반적인 Generation 을 위한 training
- ii. Augmentation

4. Test step

- a. 테스트 시에는, 필터를 사용해서 예측된 아웃풋을 선별 (스크리닝) 한다
- b. 주어진 입력 X 가 Y 로 제대로 translation 되었는지 확인하기 위해, 제약조건 c 를 만족시킬 때 까지, L 개의 아웃풋을 샘플링 한다.
- c. 만약 모든 L 번의 시도가 실패하면, 그냥 '첫번째' 것을 output 으로 낸다

5. Conditional 과 관련된 추가적인 사항으로
 - a. 해당되는 Y 가 없는 즉, 레이블이 없는 입력 X 에 대한 augmentation 을 수행 할 수 있는지 관찰함
 - b. 그런고로, augmentation step 에서 테스트셋의 입력을 포함시킴으로써 transductive setting 에서 훈련 데이터셋을 추가로 증강시키거나, 단순히 레이블이 없는 입력을 semi-supervised setting 으로 이용 할 수 있다.

3. Algorithm Motivation

1. 성공적인 translation 이 가능한 모델의 확률을 높이는것에 집중
2. 라벨을 부여하고, 모수를 계산하는 법을 반복적으로 시행하는 EM 알고리즘으로 모델을 특정지었다
3. 먼저 알아야 할 것은
 - a. external Filter c 는 이진 랜덤 변수라는 것이다 $f(X,Y)$ 의 아웃풋
 - b. 이는 Y 가 인풋 X 와 관계를 만족하는지를 제한하는 역할을 함
4. 또한 우리의 목적은 log-likelihood 를 최대화 하는 것

$$\mathbb{E}_X [\log P_\theta(c = 1 | X)]$$

5. 대부분의 경우 주어진 X 에 대해 Y 는 하나 이상일 것이므로, Y(들)을 latent variable 이라고 생각 하겠다. 그러므로 위의 식을 확장하면

$$\log \sum_Y P_\theta(Y|X) \cdot p(c = 1|X, Y)$$

가 된다.

6. 위의 식은 이산적인 latent variable 인 Y(들)을 포함하고 있으므로, 우리는 표준 EM 알고리즘, 특히 Incremental 을 이용하여 위의 식을 최대화 시키도록 하겠다.
(IEM(Incremental EM) 알고리즘은 자료의 개수가 매우 클 때 정규혼합모형을 추정하기 위해 자주 사용됨)
7. EM 알고리즘을 여기 연구에 적용 할 수 있는데,
 - a. E : Augmentation step → 필터에 의해 걸러지는 샘플로부터 Posterior 샘플이 drawn 됨
 - b. M : Training step
8. 좀 더 자세하게는
 - a. $P_\theta(t)(y|x) = t$ epoch 후의 translation 모델

- b. epoch $t+1$ 에서, $P_{\theta}^{(t)}$ 를 이용해서 각각의 X 인풋에 대해 augmentation step 은 C 개의 서로 다른 후보물질들을 샘플링 하고
- c. 제약조건 c 에 미치지 못하는 후보물질은 제거하고, 남은 후보물질은 (그다음 epoch 로 훈련 될 수 있는 데이터로) interpretable 하게 된다. 이때 현재의 posterior \sim 를 이용한다.
- d. 결과적으로 training step 은 확률적 경사하강법을 통해 EM 을 최대화 한다.

4. Experiments

Conditional molecular design 으로 부터 실험을 시작한다.

4.1 Conditional Molecular Design

목표 → 기존에 존재하는 Molecules의 화학적 properties 를 증가시키는 것

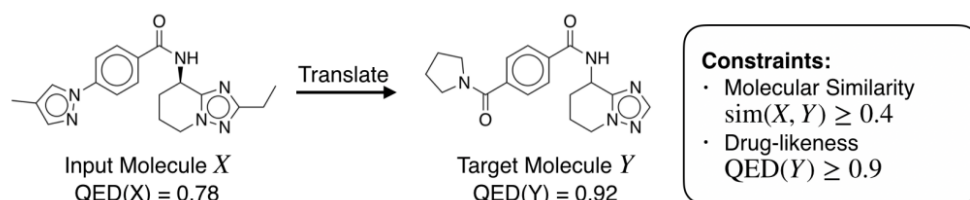


Figure 1. Illustration of conditional molecular design. Molecules can be modeled as graphs, with atoms as nodes and bonds as edges. Here, the task is to train a translation model to modify a given input molecule into a target molecule with higher drug-likeness (QED) score. The constraint has two components: the output Y must be highly drug-like, and must be sufficiently similar to the input X .

이는 Graph-to-graph 문제로 공식화 될 수 있음

Dataset → molecular pairs $D = \{(X_i, Y_i)\}$

1. 각 분자들은 property score 가 추가로 라벨링 됨
2. 우리의 방법은 conditional molecular design 에 매우 적합함, 왜냐하면 target molecule 이 유니크하지 않기 때문
→ 각각의 precursor 는 property 를 최적화하기 하기 위한 **다른 다양한 방법**으로 modify 될 수 있음
3. 그러므로 Data augmentation 동안 precursor 당 새로운 target 을 발견 할 수 있다

External Filter

1. Constrain → 2가지 종류로 이루어져있음
 - a. Y 의 chemical property 는 임계점 베타를 넘어야함
 - b. X 와 Y 의 similarity 는 임계점 감마를 넘어야함
→ $\text{sim}(X, Y)$ 를 이용하는데, Morgan fingerprints 의 Tanimoto similarity 를 이용
→ 2개의 molecule 간의 structural overlap 을 이용
2. 레알월드 세팅에서는, experimental assay 를 통해 chemical property 의 ground truth 를 측정 한다, 이는 너무 비싸고 시간이 오래걸리는 작업이다, 그러니까 돈을 아끼기 위해서 in silico 상으

로 Property predictor F1 을 만들었음

- a. F1 의 역할 : True property evaluator F0 을 근사화
- b. Predictor 를 훈련 시키기 위해 training set 의 molecule 과 그것의 라벨된 property 를 이용하였다
- c. F1 은 GCN 으로 파라미터화 되어있고 Chemprop package 를 이용해 훈련되 있었음
- d. Data augmentation setting 에서는 F1 을 필터로 사용했음
- e. F1 은 임계치 베타를 충족하지 못하는 output 을 걸러줌

4.1.1 Experimental Setup

두 가지의 Conditional molecular design task 를 evaluation 하기 위해 아래 task를 사용

1. QED optimization task

- a. 주어진 compound X에 대해 Improve drug-likeness (QED)
- b. 그냥 만들어진게 얼마나 약이랑 비슷한지 보는 것 같음

2. DRD2 optimization task

- a. 도파민 타입2 리셉터의 biological activity 와 비슷한 정도를 측정
- b. DRD2(Y)는 주어진 모델의 Biological activity 의 예측된 확률???
- c. 그냥 Biological activity 를 측정하나봐 ㅋ

3. in silico evaluator 의 output 을 ground truth 로 사용

4. 레알 월드 시나리오의 test-time 으로만 사용

Evaluatuion Metrics

2가지의 metric을 사용해서 성능을 측정

1. Success

- a. 20 번의 trails
- b. 각 output 중 성공으로 측정된 X 의 fraction
- c. **Main Metric**

2. Diversity

- a. 각각의 분자 X에 대해, $Y_1 \sim Y_Z$ 중 성공적으로 translate 된 합성물의 세트에서 페어들 간의 Tanimoto distance ($1 - \text{sim}(Y_i, Y_j)$) 로 정의) 를 측정
- b. 하나 이하의 성공적인 translation 이 있는 경우 diversity 는 0으로 정의
- c. 모든 test X 에 대해 평균값을 구함

- 결과

Graph based representation

Improving Molecular Design by Stochastic Iterative Target Augmentation				
Model	QED Succ.	QED Div.	DRD2 Succ.	DRD2 Div.
VSeq2Seq	58.5	0.331	75.9	0.176
VSeq2Seq+ (Ours)	89.0	0.470	97.2	0.361
VSeq2Seq+, semi-supervised (Ours)*	95.0	0.471	99.6	0.408
VSeq2Seq+, transductive (Ours)*	92.6	0.451	97.9	0.358
HierGNN	76.6	0.477	85.9	0.192
HierGNN+ (Ours)	93.1	0.514	97.6	0.418

average distance between two different correct output for the same input

sequence to sequence using 'string representation' of molecules

Improving Molecular Design by Stochastic Iterative Target Augmentation							
Model	Train-Aug	Train+	Test+	QED Succ.	QED Div.	DRD2 Succ.	DRD2 Div.
VSeq2Seq	✗	✗	✗	58.5	0.331	75.9	0.176
VSeq2Seq(test)	✗	✗	✓	77.4	0.471	87.2	0.200
VSeq2Seq(train)	✓	✓	✗	81.8	0.430	92.2	0.321
VSeq2Seq+	✓	✓	✓	89.0	0.470	97.2	0.361
VSeq2Seq(no-filter)	✓	✗	✗	47.5	0.297	51.0	0.185

Model	Top-1
MLE	71.91
MLE + RL + Beam Search	77.12
MLE+ (Ours)	85.02

강타-학습보다
올라!!!

Conclusion

1. Molecular Design 을 위한 Stochastic iterative target augmentation framework
2. Stochastic iterative target augmentation 가 상호보완적으로 architecture 를 크게 improvement 했으며,
3. Filter 가 상당히 Robust 하였다
4. 끝!