Flow Network based Generative Models for Diverse Candidate Generation

Iryna Zabarianska

Moscow Institute of Physics and Technology
Intelligent Systems Department

Moscow, November 26, 2024

Preliminaries

1. Set of states $\mathcal{S}$ - discrete;
2. Set of terminal states $\mathcal{X} \subset \mathcal{S}$;
3. Alphabet of actions $\mathcal{A}$ - finite;
4. $\mathcal{A}(s) \subseteq \mathcal{A}$ set of allowed actions at state s;
5. $\mathcal{A}^*(s)$ set of all sequences of actions allowed after state s;
6. $R(x)$ reward for a terminal state x;
7. Policy π chooses from state $s \in S$ an allowable action $a \in \mathcal{A}(s)$ with probability $\pi(a|s)$:

$$\pi(x) \approx \frac{R(x)}{\sum_{x' \in \mathcal{X}} R(x')}$$

Our goal

To learn policies such that $\pi(x) \propto R(x)$ when sampled.

Preliminaries

1. Set of states $\mathcal{S}$ - discrete;
2. Set of terminal states $\mathcal{X} \subset \mathcal{S}$;
3. Alphabet of actions $\mathcal{A}$ - finite;
4. $\mathcal{A}(s) \subseteq \mathcal{A}$ set of allowed actions at state s;
5. $\mathcal{A}^*(s)$ set of all sequences of actions allowed after state s;
6. $R(x)$ reward for a terminal state x;
7. Policy π chooses from state $s \in S$ an allowable action $a \in \mathcal{A}(s)$ with probability $\pi(a|s)$:

$$\pi(x) \approx \frac{R(x)}{\sum_{x' \in \mathcal{X}} R(x')}$$

Our goal

To learn policies such that $\pi(x) \propto R(x)$ when sampled.

Preliminaries

1. Set of states $\mathcal{S}$ - discrete;
2. Set of terminal states $\mathcal{X} \subset \mathcal{S}$;
3. Alphabet of actions $\mathcal{A}$ - finite;
4. $\mathcal{A}(s) \subseteq \mathcal{A}$ set of allowed actions at state s;
5. $\mathcal{A}^*(s)$ set of all sequences of actions allowed after state s;
6. $R(x)$ reward for a terminal state x;
7. Policy π chooses from state $s \in S$ an allowable action $a \in \mathcal{A}(s)$ with probability $\pi(a|s)$:

$$\pi(x) \approx \frac{R(x)}{\sum_{x' \in \mathcal{X}} R(x')}$$

Our goal

To learn policies such that $\pi(x) \propto R(x)$ when sampled.

Preliminaries

1. Set of states $\mathcal{S}$ - discrete;
2. Set of terminal states $\mathcal{X} \subset \mathcal{S}$;
3. Alphabet of actions $\mathcal{A}$ - finite;
4. $\mathcal{A}(s) \subseteq \mathcal{A}$ set of allowed actions at state s;
5. $\mathcal{A}^*(s)$ set of all sequences of actions allowed after state s;
6. $R(x)$ reward for a terminal state x;
7. Policy π chooses from state $s \in S$ an allowable action $a \in \mathcal{A}(s)$ with probability $\pi(a|s)$:
$$\pi(x) \approx \frac{R(x)}{\sum_{x' \in \mathcal{X}} R(x')}$$

Our goal

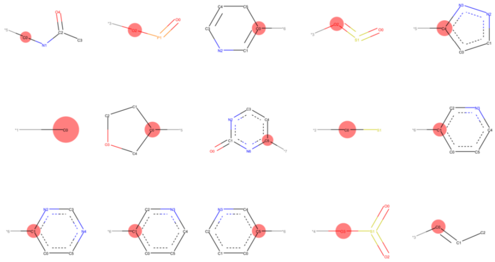
To learn policies such that $\pi(x) \propto R(x)$ when sampled.

Structure of Graph

Function $C : \mathcal{A}^* \rightarrow S$ maps action sequence $\vec{a} \triangleq (a_1, \ldots, a_n)$ to single x:

- If C - bijective, the generative process is the traversal of a tree: from root node to a leaf;
- If C - non-injective, instead of a tree, we get a directed acyclic graph or DAG.

Example: Molecules can be seen as DAG.



Poblem of Non-injective Case

Proposition

Consider

- Function $\tilde{V} : S \to \mathbb{R}^+ : \ \tilde{V}(s) = \sum_{\vec{b} \in \mathcal{A}^*(s)} R(s + \vec{b}) > 0$;
- Policy π starts in state $s_0 = C(\emptyset)$:

$$\pi(a|s) = \frac{\tilde{V}(s+a)}{\sum_{b \in \mathcal{A}(s)} \tilde{V}(s+b)}.$$

Then the following is obtained

- $\pi(s) = \sum_{\vec{a}_i : C(\vec{a}_i) = s} \pi(\vec{a}_i)$;
- If C is bijective: $\pi(s) = \frac{\tilde{V}(s)}{\tilde{V}(s_0)}$ and $\pi(x) = \frac{R(x)}{\sum_{x' \in \mathcal{X}} R(x')}$ for $x \in \mathcal{X}$;
- If C is non-injective: $\pi(x) = \frac{n(x)R(x)}{\sum_{x' \in \mathcal{X}} n(x')R(x')}$, where $n(x) \triangleq |\{\vec{a}_i : C(\vec{a}_i) = x\}|$.

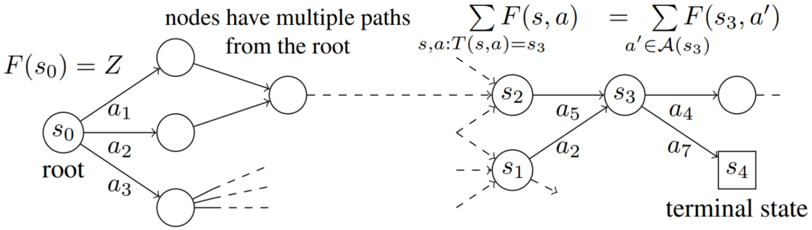
Flow Network Alternative

Main difference: learn a flow F, rather than estimating the value $\tilde{V}$.

Some notations:

- $T(s, a) = s'$: state-action pair (s, a) leads to s';
- $F(s, a)$ - flow between node s and node $s' = T(s, a)$;
- $F(s)$ - total flow going through s.

The incoming flow equals the outgoing flow!



Flow Network: good property

Proposition

Consider

- Policy π starts in state s_0: $\pi(a|s) = \frac{F(s,a)}{F(s)}$;
- $F(s) = R(s) + \sum_{a \in \mathcal{A}(s)} F(s,a)$;
- Flow consistency equation is satisfied:

$$\sum_{s,a:T(s,a)=s'} F(s,a) = R(s') + \sum_{a' \in \mathcal{A}(s')} F(s',a')$$

Then the following is obtained

- $\pi(s) = \frac{F(s)}{F(s_0)}$;
- $F(s_0) = \sum_{x \in \mathcal{X}} R(x)$;
- $\pi(x) = \frac{R(x)}{\sum_{x' \in \mathcal{X}} R(x')}$.

Objective Functions for GFlowNet

Main idea: approximate the flow F such that the flow consistency equations are respected with enough capacity.

Objective: for a trajectory τ

$$\tilde{\mathcal{L}}_\theta(\tau) = \sum_{s' \in \tau \neq s_0} \left(\sum_{s,a: T(s,a)=s'} F_\theta(s,a) - R(s') - \sum_{a' \in \mathcal{A}(s')} F_\theta(s',a') \right)^2$$

Modified objective: train predictor to estimate $F_\theta^{log}(s,a) = \log F_\theta(s,a)$.

$$\mathcal{L}_{\theta,\varepsilon}(\tau) = \sum_{s' \in \tau \neq s_0} (\log \left[\varepsilon + \sum_{s,a: T(s,a)=s'} \exp F_\theta^{log}(s,a) \right]$$

$$- \log \left[\varepsilon + R(s') + \sum_{a' \in \mathcal{A}(s')} \exp F_\theta^{log}(s',a') \right])^2$$

Objective Functions: good property

Proposition

Suppose that

- Trajectories τ (used to train F_θ) sampled from policy P: $P(a|s) = \frac{F^*(s,a)}{F^*(s)}$;
- F^* satisfies flow consistency equation;
- Exist $\theta : F_\theta = F^*$;
- $\theta^* \in argmin_\theta \mathbb{E}_{P(\tau)}[L_\theta(\tau)]$ a minimizer of the expected training loss

Then the following is obtained

- $F_{\theta^*} = F^*$ and $L_{\theta^*}(\tau) = 0, \forall \tau \sim P(\theta)$;
- If $\pi_{\theta^*}(a|s) = \frac{F_{\theta^*}(s,a)}{\sum_{a' \in \mathcal{A}(s)} F_{\theta^*}(s,a')}$, then

$$\pi_{\theta^*}(x) = \frac{R(x)}{\sum_{x' \in \mathcal{X}} R(x')}.$$


ATTENTION
THANKS FOR
YOUR ATTENTION