MassInvTree

#academic #generative-models #ai4science #mass-spectrum #monte-carlo-tree-search

> This work is basically a new work of mine.

Background

Mass spectrum data is very important and useful in identifying molecules as well as characterizing molecular properties. However, solving the spectrum data requires lots of human experiences, and data centric methods are still inferior to the expers.

This paper is focusing on the **inverse** problems. Current approaches includes

- **Autoregressive** approaches, which are trained to convert tokenized spectral informations into SMILES strings as outputs. One representative methods is Spec2Mol. However, due to the **autoregressive nature**, these methods fail to capture the **permutation invariant** nature of spectral data, and they do not **incorporate other information about the molecules.**
- **Indirect** approaches, which uses intermediate representations of molecules before generating chemical structures. These methods are more chemically interpretable, but **not necessarily lead to better performance**.

A crucial problem making the inverse problem hard to solve is that, even for the same spectrum, there are possibly multiple molecule candidates which can satisfy the constrains of the spectrum well. Current methods directly uses **beam search** or other naive methods like DiffMS and select top-k candidates with high probabilities, which fails to fully explore the whole output space, resulting in low novelty and low diversity.

To solve the **one-to-many** problem, we would like to more efficiently yet thoroughly explore the output space with some **heuristic** methods. Inspired by the success of Monte Carlo Tree Search in the protein inverse folding problems (to generate molecular structures given 3D structures) like ProtInvTree, we would like to utilize the probability distribution learnt in the **discrete diffusion** model to do MCTS.

Method

People usually generate the molecular structures by generating

- **Fingerprints**, which is not exactly the structures, but **abstractions of molecular properties**, and the task is then modeled as a regression task -- to generate a high-dimensional vector, like what they do in MIST;
- **SMILES**, which is the representation of molecular structures, and the problem becomes a sequence generation task, which is usually modeled in an autoregressive manner like Spec2Mol; and
- **Connectivity graph**, which uses connectivity graphs of atoms to directly represent the molecular structure, like DiffMS.

To do Monte Carlo Tree Search, graph based representations seem to be superior for the permutation invariant property. In this work, we do MCTS based on DiffMS, and the whole process can be divided into several steps, including

1. **Feature extraction**, which directly uses the encoder module in DiffMS to extract useful embeddings of the original spectral data;
2. **Monte Carlo Tree Search**, which can be divided into four stages
    1. **Selection**, select node to expand based on **UCT**;
    2. **Expansion**, expand the selected node for K times;
    3. **Evaluation**, evaluate each of the expanded nodes through **jumpy denoising** and an external reward model;
    4. **Backpropagation**, which update the value of nodes on the path with the leaf values.

Selection

In this step, we calculate the **UCT** value for all leaf nodes, and select the one with highest **UCT** value for future expansion. The **UCT** is defined as

$$UCT(s_t) = V(s_t) + w\sqrt{\frac{\ln N(p)}{N(s_t)}},$$

where $V(\cdot)$ denotes the node value, $N(\cdot)$ denotes the visit count of nodes. w is a hyperparameter to balance the **exploitation** and **exploration**.

After this step, we can get the node to expand at the m-th iteration, which is

$$s_m = \arg \max_{s \in \text{leaf nodes}} UCT(s).$$

Expansion

> The expansion step for the original ProtInvTree seems to be a little bit problematic. Writing of that article is bad.

After selecting the node with the highest UCT score, we expand the selected node. We generate X child nodes of the selected node, and then for each child, we decode one

> Here the X is a constant over all expansion steps. This is a hyperparameter, and the value of X will affect the tradeoff between diversity and generation quality.

Here, the expand function $\mathcal{E}(\cdot)$ defines a set of child nodes. In the original ProtInvTree, the expansion is done in two steps

- **Focus**, which selects K_t positions via a special function, and the K_t is a cosine scheduled value.
  - Here, the function may vary, which can be a pure random selection function or an entropy-based function. In the original work of ProtInvTree, the performance of random function is surprisingly good.
- **Grounding**, which randomly sample amino acids of the corresponding positions selected in the **focus** stage.

> Though the action is defined to be the posterior condition of x_{t+1} conditioned on x_t and c in the ProtInvTree paper, they are doing in a totally different way. The decoupling strategy of **focus** and **grounding** will make the joint distribution of the two sampling processes does not equal to the original distribution, especially when they are trying different **focus** strategies.

However, since the DiGress is not a masked diffusion model, no focus-grounding technique is needed. For each action, it is just a transition matrix applied to all the entries in the adjacency matrices.

In the language of reinforcement learning, the action is defined as

$$a_t = \{(i_k, x_{i_k})\}_{k=1}^{n \times n \times d},$$

which is a set of multiple decoding results, and the policy is then defined as

$$\pi(a_t|s_t) = p_\theta(x_{t+1}|x_t, c).$$

> Here n is the number of atoms (nodes of the graph), while d is the number of different chemical bonds (subgraph). a_t means the transition matrix of each entry.

> What should we do if the same child node is generated during the expansion stage?

Evaluation

In the evaluation step, we do two things:

1. Do one-step-denoising in a DDIM way, and get an approximation of the final molecule;
2. Use an external reward model to evaluate the quality of the approximated molecule -- Here we firstly use MassFormer to get the estimated mass spectrum, then we calculate the similarity between the estimation and the ground truth sequence.

> The most commonly used similarity measure is cosine similarity between the binned spectrums -- However, there are many aggregation tricks in this field; details are in the MassFormer paper.

If we use the function $\mathcal{F}$ to represent the MassFormer model, the whole reward function should be the cosine similarity between the predicted and the ground truth spectrum. To avoid the similarity scores being dominated by a few significant peaks, we firstly

take logarithm before calculating the similarity, and the reward becomes

$$r_m = \frac{\ln \mathcal{F}(x_m^T) \cdot \ln y_{GT}}{\| \ln \mathcal{F}(x_m^T)\| \cdot \| \ln y_{GT}\|},$$

where x_m^T is the approximation generated by jumpy denoising from x_m.

Backpropagation

Update the visit count and the value, and then update all the values of nodes on the path.

For all the nodes on the path, the relevant values are updated as

$$N_{new}(s_j) = N_{old}(s_j) + 1,$$

$$V_{new}(s_j) = \frac{V_{old}(s_j) \cdot N_{old}(s_j) + r_{t+1}}{N_{new}(s_j)}.$$

To understand, $N(\cdot)$ keep tracks of how many nodes are used to update the value of the node, while the $V(\cdot)$ is the average reward scale.

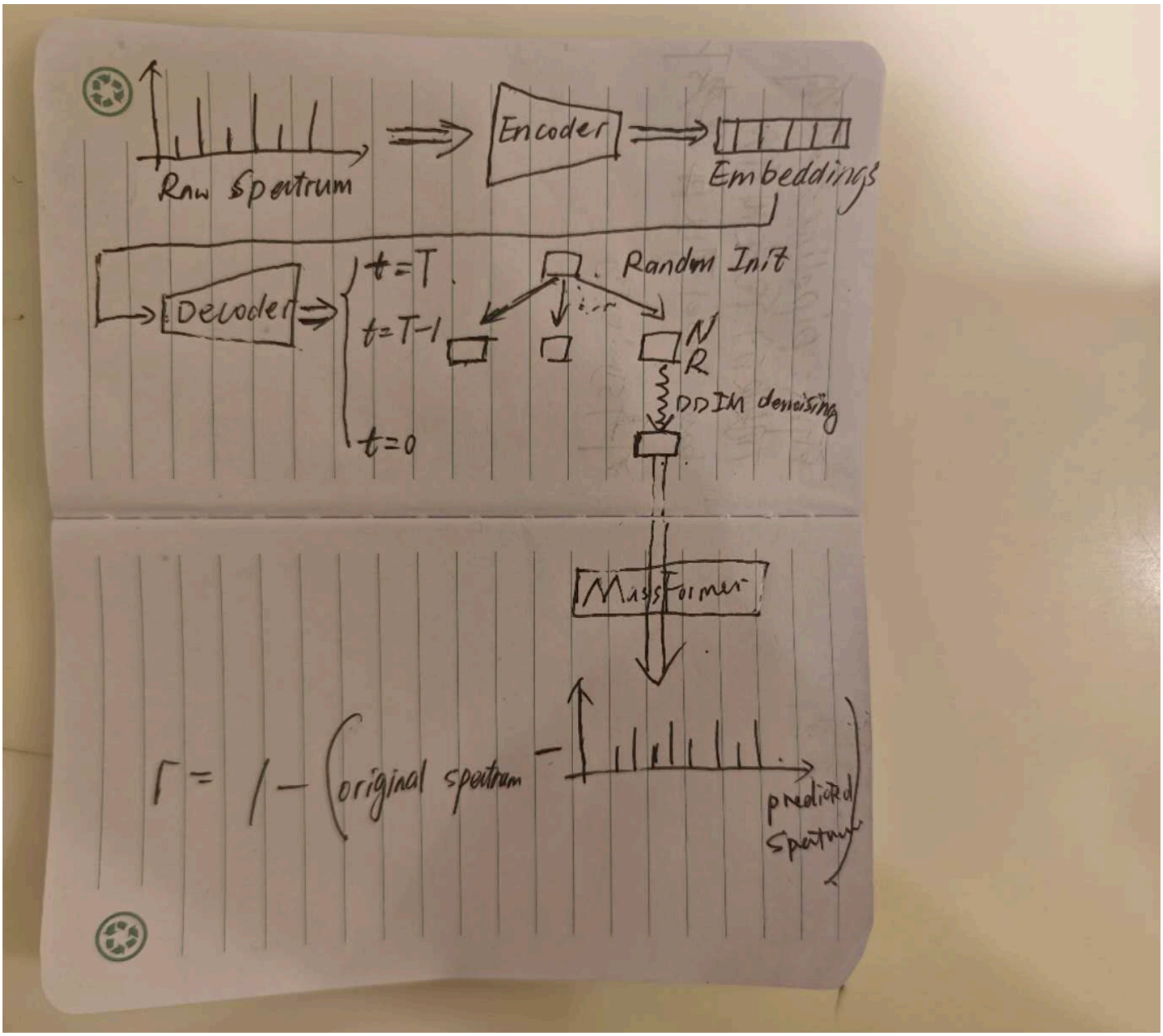
Pseudocode

Algorithm 1 Monte Carlo Tree Search

Input: Input spectrum y, DiffMS encoder $\mathcal{E}$, DiffMS decoder parameters θ, MassFormer $\mathcal{F}$, max tree search iteration M, expansion number N, reward threshold τ

Output: A set $\mathcal{S}$ containing all possible solutions Monte Carlo Tree Search

$c \leftarrow \mathcal{E}(y)$
for $m \leftarrow 1$ to M **do**
  $s_m \leftarrow \underset{s \in \{\text{Leaf Nodes}\}}{\arg\max}\ V(s) + w \cdot \sqrt{\frac{\ln N(p)}{N(s)}}$ ▷ UCT based selection
  for $n \leftarrow 1$ to N **do**
    $a_t^n \sim \pi_\theta(a_t | s_t, c, K_t)$ ▷ Expansion
    Apply a_t^n on s_t to get s_{t+1}^n, and add that to the search tree
    $a_{t+1}^n \sim \pi_\theta(a_t | s_{t+1}^n, c, \#\text{ of all the rest positions})$ ▷ Jumpy denoising
    Apply a_{t+1}^n on s_{t+1}^n to get s_T^n
    $r_{t+1}^n \leftarrow \frac{\ln \mathcal{F}(s_{t+1}^n) \cdot \ln y}{\| \ln \mathcal{F}(s_{t+1}^n)\| \cdot \| \ln y\|}$ ▷ Evaluation
    $V(s_{t+1}^n) \leftarrow r_{t+1}^n$
    for $p \in \{\text{Parent Nodes of } s_{t+1}^n\}$ **do** ▷ Backpropagation
      $N_{new}(p) \leftarrow N_{old}(p) + 1$
      $V_{new}(p) \leftarrow \frac{V_{old}(p) \cdot N_{old}(p) + r_{t+1}^n}{N_{new}(p)}$
    end for
    if s_{t+1}^n is a leaf node and $r_{t+1}^n \geq \tau$ **then**
      Add s_{t+1}^n into the answer set $\mathcal{S}$
    end if
  end for
end for
return $\mathcal{S}$



Acc@k Sim@k

1 10 1~10

100次 100

10 Branch -- Decoding path

average out the collision energy -- to merge the gap between missing info nad task

Meta information of the spectrum generation

I am eager to contribute to the NeurIPS community in several ways:

1. Community and Volunteer Support: I am prepared to offer practical support to the conference. As I am based in the US, I can serve as a designated presenter for posters from my international collaborators who may face visa challenges, ensuring their valuable research is represented and discussed. Furthermore, I am keen to volunteer on-site, whether by assisting workshop organizers with logistics or helping moderate Q&A sessions to ensure events run smoothly for all attendees. Meanwhile, as one who can fluently speak Chinese, Cantonese and English, I can help to communicate between different people familiar to different languages.

2. Enhancing Diversity: I would contribute to the meeting's diversity of perspective. Coming from a low-income background in a region with limited access to global research opportunities, my journey has been non-traditional. I believe my presence helps represent the growing pool of talent from underrepresented socioeconomic backgrounds in AI and can add a valuable viewpoint to scientific discussions.

3. Inspiring Future Talent: My attendance would have an impact that extends beyond the conference itself. I am committed to being a role model and sharing my experiences and the knowledge gained from NeurIPS with aspiring students in my home community. By demonstrating that overcoming economic barriers is possible, I hope to encourage more young talent to pursue science and contribute to our field in the future.

Meta information

Concerns

1. What if the selected leaf node for expansion is not expandable? -- Then maybe we should end the search, or we select the second biggest UCT one.
2. M is controlling the total number of tree-deepening process. What if after M iterations, no one valid leaf node is generated? -- Then maybe we should apply normal decoding strategies on these states.
3. How to deal with repetitive nodes appearing in the Expansion stage? -- Ignore the problem and count that as a normal node?

Dataset

Model	Dataset	Specials
DiffMS	NPLIB1, MassSpecGym	Adduct is needed, energy does not matter
ICEBERG (old)	NPLIB1, NIST20	Adduct is needed, energy does not matter
ICEBERG (new)	NIST20	Both adduct and energy are needed
MARASON	NIST20, MassSpecGym	Both adduct and energy are needed
SIRIUS	MassBank, GNPS, NIST17	Takes in the spectra and adduct info, returns the chemical formulae
MIST	--	The pretrained version does not matter

Conclusion

DiffMS - NPLIB1 <=> ICEBERG (old) - NPLIB1
DiffMS - MassSpecGym <=> MARASON - MassSpecGym

do a proof of concept

Extend to other molecule generative model: Spec2Mol, etc

Solutions

- Directly use DiffMS, without further training
  - Use old version of ICEBERG to do NLPIB1, use MARASON to do MassSpecGym
    - Not a very good idea
  - Retrain the spectrum prediction model on the datasets DiffMS is trained on
    - Still, not a very good idea -- spectrum prediction models are quite heavy, requiring too much memory
- Train DiffMS on NIST20
  - We only include NIST20 in our experiments
    - Not a very good idea
  - We can do experiment on NIST20 and MassSpecGym with MARASON