Graph Optimization - Model Outline

Octav Dragoi

June 5, 2020

1 Introduction

The problem is stated as a **molecule optimization** problem: given one molecule, we wish to come up with another one that improves on certain desirable metrics.

The dataset consists of N pairs of molecules, one being the improved version of the other:

$$D = \{(\mathcal{X}_i, \mathcal{Y}_i) : 1 \leq i \leq N\} \tag{1}$$

We describe all the steps of the model we are planning to implement. By and large, these are:

- Node embedding generator
- Point cloud optimizer
- Point cloud resizer
- Molecular discretizer

2 Node Embeddings

A given molecular graph consists of atoms and interatomic bonds, each with their own set of characteristics. For each such graph, we generate a set of point cloud embeddings in $\mathbb{R}^d, d \in \mathbb{N}$, where each point corresponds to one of the atoms. Thus, the number of points will be equal to the number of atoms in the molecule, and their relative positions in the metric space $\mathbb{R}^d$ should encode the bonds and relationships between them.

Mathematically, denote by $\mathcal{G}$ the set of all molecular graphs, and let $\mathscr{D}_2(\mathbb{R}^d)$ be the set of all (weighted) point clouds in $\mathbb{R}^d$, i.e.

$$\mathscr{D}_2(\mathbb{R}^d) = \left\{ \sum_{i=1}^{n} \lambda_i \delta_{x_i} : n \in \mathbb{N}, \lambda_i \in \mathbb{R}, x_i \in \mathbb{R}^d \right\}$$

We wish to learn a function f such that:

$$f : \mathcal{G} \to \mathscr{D}_2(\mathbb{R}^d)$$

and for which a good pseudoinverse f^{-1} can be deduced. In this manner, f acts as an encoder, and f^{-1} as a decoder, of molecules in the space $\mathscr{D}_2(\mathbb{R}^d)$. The goal of this embedding is to execute an optimization process within that space, in order to find optimal points and transform them back to molecular graphs.

We encode our dataset D, defined in (1), and transfer the problem to the Wasserstein space:

$$\begin{aligned} f(D) &= \{(f(\mathcal{X}_i), f(\mathcal{Y}_i)) : 1 \leq i \leq N\} \\ &= \{(X_i, Y_i) : 1 \leq i \leq N\} \end{aligned}$$

For the encoding step, we define f to be a **Graph Convolutional Network**.

3 Wasserstein Optimization

Within the Wasserstein space of point clouds $\mathscr{D}_2(\mathbb{R}^d)$, knowing the geometry will help design optimization algorithms. We shall consider a sub-case for a slice of $\mathscr{D}_2(\mathbb{R}^d)$ where all the point clouds have the same number of points, namely for:

$$\mathscr{D}_2^k(\mathbb{R}^d) = \left\{ \sum_{i=1}^{k} \lambda_i \delta_{x_i} : \lambda_i \in \mathbb{R}, x_i \in \mathbb{R}^d \right\}$$

3.1 Point Clouds with Exactly n Points

In this case, the process of transforming one molecule into another is easier to parametrize. Let $X, Y \subset \mathscr{D}_2^n(\mathbb{R}^d)$, then the optimal transport plan between X and Y is actually a transport map (see the Wasserstein writeup for a more detailed explanation of this fact).

In other words, if $X = \{x_1, x_2, \ldots, x_n\}, Y = \{y_1, y_2, \ldots, y_n\}$, then the optimal transport map $T : X \to Y$ induces a permutation σ such that:

$$T(x_i) = y_{\sigma(i)}$$

A second crucial observation is that, under this consideration, the **barycenter** $B \subset \mathscr{D}_2^n(\mathbb{R}^d)$ of X and Y with weights $1 - t$ and t is defined by:

$$\begin{aligned} B &= \bigcup_{i=1}^{n} \delta_{(1-t)x_i + t y_{\sigma(i)}} \\ &= \bigcup_{i=1}^{n} \delta_{x_i + t(y_{\sigma(i)} - x_i)} \end{aligned}$$

As $t \in [0, 1]$, this defines a **geodesic** curve between X and Y.

This means that, within the space $\mathscr{D}_2^n(\mathbb{R}^d)$ and under the W_2 Wasserstein metric, the directions in which a point cloud X can be deformed are parametrized by the velocities within R^d in which every point of X is deformed (i.e. the vectors $y_{\sigma(i)} - x_i$). These directions are the **tangent vectors** at X within $\mathscr{D}_2^n(\mathbb{R}^d)$.

We would like to train a model that, given a molecule embedding X, will return a set of directions for the points of X in which to displace them and improve the molecule.

3.2 Resizing Point Clouds

To resize a point cloud X with n points into a point cloud X' with $m \neq n$ points, one can take the barycenter of X with m points.

In the case where $m < n$, this is equivalent to the k-means problem.

4 Discretization

After arriving at an improved molecule embedding $\hat{Y}$ from X as described in the previous section, we need to apply an algorithm that transforms $\hat{Y}$ into a molecule $\hat{\mathcal{Y}} \in \mathcal{G}$.