A Supplement to the Details of the DTisT Method

Having It Both Ways: Single Trajectory Embedding for Similarity Computation with Pairwise Learning

A. Adaptable Base Encoder

1) Short-term Perception: Given the original trajectory $T^a = [p_1^a, \cdots, p_{|T^a|}^a]$, the hidden sequence $\mathbf{x}_{p^a} = [x_{p_1^a}, x_{p_2^a}, ..., x_{p_{|T^a|}^a}]$ is encoded from sequential point information as follows:

$$\begin{aligned}
x_{p_i^a} &= \mathbf{W}_x \mathbf{Norm}(p_i^a) \\
z_{p_i^a} &= \sigma(\mathbf{W}_z(x_{p_{i-1}^a} \oplus x_{p_i^a}) + \mathbf{b}_z) \\
r_{p_i^a} &= \sigma(\mathbf{W}_r(x_{p_{i-1}^a} \oplus x_{p_i^a}) + \mathbf{b}_r) \\
\tilde{x}_{p_i^a} &= tanh(\mathbf{W}_h(r_{p_i^a} x_{p_{i-1}^a} \oplus x_{p_i^a}) + \mathbf{b}_c) \\
x_{p_i^a} &= (1 - z_{p_i^a}) x_{p_{i-1}^a} + z_{p_i^a} \tilde{x}_{p_i^a}
\end{aligned}$$

where $\mathbf{W}_z, \mathbf{W}_r, \mathbf{W}_h \in R^{d\times 2d}$, $\mathbf{b}_z, \mathbf{b}_r, \mathbf{b}_c \in R^d$, and $\mathbf{W}_x \in R^{d\times 2}$ are parameters of the GRU. The operator $\oplus$ symbolizes the vector concatenation function. The function $\mathbf{Norm}(\cdot)$ is employed for the normalization of trajectory point coordinates. The term z serves as the update gate, instrumental in accumulating and disseminating node information throughout the trajectory. The reset gate, represented as r, plays a pivotal role in effectively segregating information across nodes spanning large intervals.

2) Long-term Perception: In the Long-term Perception encoding, we use the Transformer Encoder to encode the grid node sequence $\mathbf{x}_{n^a}$ into o^a. Prior to this, we integrate positional encoding into the $\mathbf{x}_{n^a}$ sequence to augment the node representation, defined as:

$$\begin{aligned}
\mathbf{PE}_{n_i^a}[\mathrm{2d}] &= sin(\frac{i}{10000^{\mathrm{2d}/d}}) \\
\mathbf{PE}_{n_i^a}[\mathrm{2d}+1] &= cos(\frac{i}{10000^{\mathrm{2d}/d}}) \\
\bar{x}_{n_i^a} &= x_{n_i^a} + \mathbf{PE}_{n_i^a}
\end{aligned}$$

where $\mathbf{PE}_{n_i^a}[\mathrm{2d}]$ and $\mathbf{PE}_{n_i^a}[\mathrm{2d}+1]$ denote the even and odd dimensions of $\mathbf{PE}_{n_i^a}$, respectively. This results in the augmented sequence $\bar{\mathbf{x}}_n^a = [\bar{x}_{n_1^a}, \cdots, \bar{x}_{n_L^a}]$.

Subsequently, the serialized representation is encoded into the grid node sequence through the following process using the Transformer Encoder: $o^a = \mathbf{TransformerEncoder}(\bar{\mathbf{x}}_n^a)$.

B. Negative Sampling Example

To facilitate understanding, let's simplify the scale of this problem: Consider a set of nine trajectories, $\mathcal{T}^a = \{T_0, \ldots, T_8\}$, with the anchor trajectory T_a, and sample three positive and three negative trajectories ($\eta = 3$) from it. The distances from the anchor trajectory to the trajectories in the set are $\{f(T_a, T_i)|T_i \in \mathcal{T}\} = \{0.5, 10, 15, 25, 50, 1.5, 7, 800, 100\}$.

First, according to the positive trajectory sampling rule, the nearest trajectories are sampled in order, resulting in the set $\mathcal{S}^+(T^a, \mathcal{T}, f) = \{T_0, T_5, T_6\}$. The remaining trajectories form the sample space for negative trajectory sampling, $\mathcal{T}_*^a = \{T_1, T_2, T_3, T_4, T_7, T_8\}$, with distances $\{f(T_a, T_i)|T_i \in \mathcal{T}_*^a\} = \{10, 15, 25, 50, 800, 100\}$. Using these trajectory similarities/distances, the normalized probabilities are calculated as follows:

$$p_f^a(T^i) = f(T^i, T^a) / \sum_{T^j \in \mathcal{T}_*^a} f(T^j, T^a)$$

and resulting in:

$$\{p_f^a(T^i)|T_i \in \mathcal{T}_*^a\} = \{0.01, 0.015, 0.025, 0.05, 0.8, 0.1\}$$

Based on this probability distribution, three trajectories are sampled without repetition:

$$\mathcal{S}^-(T^a, \mathcal{T}, f) = \left\{T^i | T^i \sim p_f^a(T^i), T^i \in \mathcal{T}_*^a, 1 \le i \le \eta\right\}$$

For example, $\mathcal{S}^-(T^a, \mathcal{T}, f) = \{T_7, T_3, T_4\}$. Finally, the sampling results are sorted by similarity, resulting in $\mathbf{T}_a^- = \{T_7, T_4, T_3\}$.