Multi-Scale Representation Learning for Spatial Feature Distributions using Grid Cells

Gengchen Mai\(^1\), Krzysztof Janowicz\(^1\), Bo Yan\(^2\), Rui Zhu\(^1\), Ling Cai\(^1\), Ni Lao\(^3\)
\(^1\)STKO Lab, UC Santa Barbara; \(^2\)LinkedIn Corporation; \(^3\)SayMosaic Inc.

Embedding clustering of different location encoding models:
(a)-(d) baselines (e)-(f) **Space2Vec**

GitHub Repo: [https://github.com/gengchenmai/space2vec](https://github.com/gengchenmai/space2vec)
Unsupervised Text Encoding

Position Encoding: encode word positions with sinusoid functions of different frequencies

Transformer (Vaswani et al., 2017) BERT (Devlin et al., 2019)
Unsupervised Text Encoding

Position Encoding: encode word positions in sentences with multiple sinusoid functions with different frequencies

BERT (Devlin et al. 2019)
Unsupervised Text Encoding

Word2Vec (Mikolov et al., 2013)\(^1\)  
BERT (Devlin et al. 2019)

\(^1\)http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/
Unsupervised Location Encoding

1. Radial Basis Function (RBF)
   
   \[ K(x, x') = \exp \left( -\frac{||x - x'||^2}{2\sigma^2} \right) \]

   - choosing the correct scale is challenging
   - Need to memorize the training samples

2. Tile-based approaches (Berg at al. 2014):
   discretize the study area into regular grids
   - choosing the correct scale is challenging
   - does not scale well in terms of memory

3. Directly feed the coordinates into a FFN (inductive single-scale location encoder)
   - hard to capture fine grained distributions

Geo-aware Image Classification (Mac Aodha et al., 2019)
Key challenge for location encoding

- Joint modeling distributions with very different characteristics
- \( \Rightarrow \) multi-scale location representations

![Women's Clothing](Clustered Distribution)
![Education](Even Distribution)

Renormalized Ripley's K for different POI types
Grid Cell Based Multi-Scale Location Encoding

- **Grid cells** in mammals provide a multi-scale periodic representation that functions as a metric for location encoding.

- It can be simulated by summing three cosine grating functions oriented 60 degree apart (a simple Fourier model of the hexagonal lattice).

Mean grid spacing for all modules (M1–M4) in all animals (colour-coded)
Space2Vec

Given a location $\mathbf{x}$:

$$
\text{Enc}_{\text{theory}}^{(x)}(\mathbf{x}) = \text{NN}(PE^{(t)}(\mathbf{x}))
$$

$$
PE^{(t)}(\mathbf{x}) = [PE_0^{(t)}(\mathbf{x}); ...; PE_s^{(t)}(\mathbf{x}); ...; PE_{S-1}^{(t)}(\mathbf{x})]
$$

$$
PE_s^{(t)}(\mathbf{x}) = [PE_{s,1}^{(t)}(\mathbf{x}); PE_{s,2}^{(t)}(\mathbf{x}); PE_{s,3}^{(t)}(\mathbf{x})]
$$

$$
PE_{s,j}^{(t)}(\mathbf{x}) = [\cos(\frac{\langle \mathbf{x}, \mathbf{a}_j \rangle}{\lambda_{\text{min}} \cdot g^{s/(S-1)}}); \sin(\frac{\langle \mathbf{x}, \mathbf{a}_j \rangle}{\lambda_{\text{min}} \cdot g^{s/(S-1)}})] \forall j = 1, 2, 3;
$$
# Point of Interest Type Classification

<table>
<thead>
<tr>
<th>POI Groups</th>
<th>Clustered ((r \leq 100m))</th>
<th>Middle ((100m &lt; r &lt; 200m))</th>
<th>Even ((r \geq 200m))</th>
</tr>
</thead>
<tbody>
<tr>
<td>direct</td>
<td>0.080 (-0.047)</td>
<td>0.108 (-0.030)</td>
<td>0.084 (-0.047)</td>
</tr>
<tr>
<td>wrap</td>
<td>0.106 (-0.021)</td>
<td>0.126 (-0.012)</td>
<td>0.122 (-0.009)</td>
</tr>
<tr>
<td>tile</td>
<td>0.108 (-0.019)</td>
<td>0.135 (-0.003)</td>
<td>0.111 (-0.020)</td>
</tr>
<tr>
<td>rbf</td>
<td>0.112 (-0.015)</td>
<td>0.136 (-0.002)</td>
<td>0.119 (-0.012)</td>
</tr>
<tr>
<td>theory</td>
<td>0.127 (-)</td>
<td>0.138 (-)</td>
<td>0.131 (-)</td>
</tr>
<tr>
<td># POI</td>
<td>16,016</td>
<td>7,443</td>
<td>3,915</td>
</tr>
<tr>
<td>Root Types</td>
<td>Restaurants; Shopping; Food; Nightlife; Automotive; Active Life; Arts &amp; Entertainment; Financial Services</td>
<td>Beauty &amp; Spas; Health &amp; Medical; Local Services; Hotels &amp; Travel; Professional Services; Public Services &amp; Government</td>
<td>Home Services; Event Planning &amp; Services; Pets; Education</td>
</tr>
</tbody>
</table>

(a) direct  (b) tile  (c) wrap  (d) rbf \((\sigma=1k)\)  (e) \(\lambda_{min}=1k\)  (f) \(\lambda_{min}=500\)  (g) \(\lambda_{min}=50\)
Geo-Aware Image Classification

(Mac Aodha et al., 2019)

<table>
<thead>
<tr>
<th></th>
<th>BirdSnap†</th>
<th>NABirds†</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Prior (i.e. uniform)</td>
<td>70.07</td>
<td>76.08</td>
</tr>
<tr>
<td>Nearest Neighbor (num)</td>
<td>77.76</td>
<td>79.99</td>
</tr>
<tr>
<td>Nearest Neighbor (spatial)</td>
<td>77.98</td>
<td>80.79</td>
</tr>
<tr>
<td>Adaptive Kernel (Berg et al., 2014)</td>
<td>78.65</td>
<td>81.11</td>
</tr>
<tr>
<td>tile (Tang et al., 2015) (location only)</td>
<td>77.19</td>
<td>79.58</td>
</tr>
<tr>
<td>wrap (Mac Aodha et al., 2019) (location only)</td>
<td>78.65</td>
<td>81.15</td>
</tr>
<tr>
<td>rbf (σ=1K)</td>
<td>78.56</td>
<td>81.13</td>
</tr>
<tr>
<td>grid (λ_min=0.0001, λ_max=360, S = 64)</td>
<td><strong>79.44</strong></td>
<td>81.28</td>
</tr>
<tr>
<td>theory (λ_min=0.0001, λ_max=360, S = 64)</td>
<td>79.35</td>
<td><strong>81.59</strong></td>
</tr>
</tbody>
</table>


GitHub Repo: [https://github.com/gengchenmai/space2vec](https://github.com/gengchenmai/space2vec)
Geo-Aware Image Classification

Our multi-scale location encoding (grid and theory) can outperform 1) RBF (rbf); 2) tile-based approaches (tile); 3) single-scale location encoding (wrap).

<table>
<thead>
<tr>
<th>Prior Type</th>
<th>BirdSnap†</th>
<th>NABirds†</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Prior (i.e. uniform)</td>
<td>70.07</td>
<td>76.08</td>
</tr>
<tr>
<td>Nearest Neighbor (num)</td>
<td>77.76</td>
<td>79.99</td>
</tr>
<tr>
<td>Nearest Neighbor (spatial)</td>
<td>77.98</td>
<td>80.79</td>
</tr>
<tr>
<td>Adaptive Kernel (Berg et al., 2014)</td>
<td>78.65</td>
<td>81.11</td>
</tr>
<tr>
<td>tile (Tang et al., 2015) (location only)</td>
<td>77.19</td>
<td>79.58</td>
</tr>
<tr>
<td>wrap (Mac Aodha et al., 2019) (location only)</td>
<td>78.65</td>
<td>81.15</td>
</tr>
<tr>
<td>rbf (σ=1k)</td>
<td>78.56</td>
<td>81.13</td>
</tr>
<tr>
<td>grid (λ_{min}=0.0001, λ_{max}=360, S = 64)</td>
<td><strong>79.44</strong></td>
<td>81.28</td>
</tr>
<tr>
<td>theory (λ_{min}=0.0001, λ_{max}=360, S = 64)</td>
<td>79.35</td>
<td><strong>81.59</strong></td>
</tr>
</tbody>
</table>


GitHub Repo: [https://github.com/gengchenmai/space2vec](https://github.com/gengchenmai/space2vec)