Problem Definition

Goal 1: train unsupervised word embeddings that capture word similarity and word analogy, like GloVe

Goal 2: additionally capture lexical entailment
- use the hyperbolic space for training
- no supervision during training
- provide mathematically sound way to perform inference for the three evaluation tasks

Prior work & Baselines

1) GloVe embeddings (Pennington et al, 2014)
- Euclidean point embeddings
- capture word similarity and analogy

2) Gaussian Word Embeddings (Vilnis et al, 2014)
- represent words in the space of diagonal Gaussians
- capture lexical entailment, but no indication of their performance on word analogy tasks

Poincaré GloVe

Loss: $J = \sum_{i=1}^{n} L(X_i) = -\log P(Y_i | X_i) + b_i - \log X_i)^2$

- $d_{Poincare}(x, y) = \sqrt{\sum_{i=1}^{n} d_{Poincare}(x_i, y_i)^2}$ for training in the Cartesian product of Poincaré balls

Connection to word2gauss

Let $\Sigma = diag(\sigma^2)$, $\Sigma' = diag(\sigma'^2)$ and $d_{H}(\cdot)$ be the half-plane distance. Then $d_{Poincare}(X \mid \mu, \Sigma, X' \mid \mu', \Sigma') = \sqrt{\sum_{i=1}^{n} 2d_{H}((\mu_i / \sqrt{\Sigma_{ii}}, (\mu'_i / \sqrt{\Sigma'_{ii}})))}$ (Costa et al, 2015).

Word analogy via Parallel Transport

"king is to queen what man is to woman"

A→B ⇒ C→D, and A→C ⇒ B→D

d_A = e ⊕ gyr(e) ⊕ (x ⊕ b), and d_B = b ⊕ gyr(b) ⊕ (x ⊕ e)

Solution: interpolate between the two points obtained with parallel transport

Lexical entailment

Poincaré ball → Translation → Rotation → Half-plane → Gaussians

Translation & rotation parameters chosen using two approaches:
1) semi-supervised (using a sample of WordNet generic/specific words)
2) unsupervised (using sample of frequent/rare words)

Let P, Q be two words with Gaussian embeddings ($\mu$, $\Sigma$) and ($\mu'$, $\Sigma'$),

LE score: $\text{Le}(P, Q) = \log(V_{y}) - \log(V_{x}) = \sum_{i=1}^{n} \log(\sigma_{i}) - \log(\sigma_{i})$

Intuition:
Large variance ⇒ Generic word
Small variance ⇒ Specific word

Experiments

- Word similarity and analogy results (highlighted: the best and the 2nd best).

<table>
<thead>
<tr>
<th>Experiment name</th>
<th>Random word sim.</th>
<th>SimLex</th>
<th>SimVerb</th>
<th>Google</th>
<th>MSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>100D GloVe</td>
<td>0.3778</td>
<td>0.5011</td>
<td>0.2583</td>
<td>0.1492</td>
<td>0.5011</td>
</tr>
<tr>
<td>500D Poincare GloVe</td>
<td>0.4187</td>
<td>0.6209</td>
<td>0.3208</td>
<td>0.1915</td>
<td>0.6339</td>
</tr>
<tr>
<td>500D Poincare GloVe</td>
<td>0.4276</td>
<td>0.6234</td>
<td>0.3181</td>
<td>0.189</td>
<td>0.6045</td>
</tr>
<tr>
<td>500D Poincare GloVe</td>
<td>0.4104</td>
<td>0.5782</td>
<td>0.3022</td>
<td>0.1685</td>
<td>0.6300</td>
</tr>
</tbody>
</table>

- Hyperplex results (Spearman correlation) for different model types ordered according to their difficulty.

Table 1

<table>
<thead>
<tr>
<th>MODEL</th>
<th>TYPE</th>
<th>METHOD</th>
<th>UNSUPERVISED HYPERPENNY SCORE</th>
<th>SUPERVISED HYPERPENNY SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>500D Poincare GloVe</td>
<td>A→C</td>
<td>$= \exp(x)$, init trick (190k)</td>
<td>0.380</td>
<td>0.402</td>
</tr>
<tr>
<td>500D Poincare GloVe</td>
<td>A→C</td>
<td>$= \exp(x)$, init trick (190k)</td>
<td>0.344</td>
<td>0.423</td>
</tr>
<tr>
<td>500D Poincare GloVe</td>
<td>A→C</td>
<td>$= \exp(x)$, init trick (190k)</td>
<td>0.284</td>
<td>0.341</td>
</tr>
</tbody>
</table>

Resources

Code: [https://github.com/alex-tifrea/poincare_glove](https://github.com/alex-tifrea/poincare_glove)

References

- Pennington et al. “GloVe: Global vectors for word representation”, EMNLP 2014
- Chang et al. “Distributional inclusion vector embedding for unsupervised hypernymy detection”, NAACL 2018