Fair Word Embeddings using Mixed-Curvature Models

Ayushi Agarwal
Department of Computer Science
UCLA
UID: 705496407
ayushi15@ucla.edu

Vishnu Vardhan Bachupally
Department of Computer Science
UCLA
UID - 005525521
vishnu1911@ucla.edu

Satvik Mashkaria
Department of Computer Science
UCLA
UID - 405729026
satvikm@ucla.edu

Vaibhav Kumar
Department of Computer Science
UCLA
UID - 305710616
vaibhavk@ucla.edu

Abstract

Language has a heterogeneous structure – It has several orders of hierarchies and spherical structures. While most word embeddings for these languages are represented in the Euclidean space Pennington et al. [2014], Mikolov et al. [2013a], there have been work done to embed words in hyperbolic space Tifrea et al. [2018] to better represent the hierarchical structure and spherical space Meng et al. [2019] to capture the symmetric structure in the language. We hypothesize that a fixed-curvature space is not enough to capture this heterogeneous structure. We propose mixed-curvature space – Cartesian product of the three, Euclidean, Spherical and Hyperbolic space to embed words to capture the diverse semantic and syntactic information in text.

Github: https://github.com/ayu15031/nlp-mixCurvature-Glove

1 Introduction

Now a days, most of the Natural Language Processing tasks require word embeddings. The word embeddings are essential because they capture the similarity of words in a vector representation. A lot of work has been done till date to develop models for creating these word embeddings. These include Word2Vec Mikolov et al. [2013b], Glove Pennington et al. [2014] that learn the embeddings efficiently in an unsupervised fashion.

However, these word embeddings are represented in Eucledian space. This assume the fact that all words are henceforth equal. But languages mostly have heterogeneous structure. As shown in Tifrea et al. [2018] and Meng et al. [2019], models have been developed in hyperbolic and spherical spaces to capture different structures of word. However, we hypothesize that the language model does not follow a similar structure and a fixed curvature model is not enough. So we propose a mixed curvature space model to create word embeddings.

Also, as we know that word embeddings contain high bias in terms of stereotypes and predujices Papakyriakopoulos et al. [2020]. We plan to apply some debiasing algorithms to create fair word embeddings using our multi-mixture model.

The code for Glove has been adopted from git.

1.1 Mixed-Curvature Space

A Mixed-Curvature space can simply be represented as cartesian product of several fixed curvature spaces. Formally, a mixed-curvature space $\mathbb{P}$ can be formulated as follows:

$$\mathbb{P} = \times_{i=1}^{k} \mathcal{M}_i^{d_i} \tag{1}$$

where $\times_{i=1}^{k}$ represents the Cartesian product of k fixed-curvature manifolds $\mathcal{M}_i^{d_i}$ with dimension d_i for the i^{th} manifold.

1.2 Operations in Mixed-Curvature Space

Point-wise operations in mixed-curvature space can be performed by decomposing and applying the operator in the fixed-curvature space and then aggregating them. For the aggregation function, using weighted average with learnable weights. Following are a few operations in the hyperbolic/spherical space we will be using:

1. **Exponential Map**: $\exp_0^K(\mathbf{v}) = \tanh(\sqrt{K}\|\mathbf{v}\|)\frac{\mathbf{v}}{\sqrt{K}\|\mathbf{v}\|}$
2. **Logarithmic Map**: $\log_0^K(\mathbf{x}) = \tanh^{-1}(\sqrt{K}\|\mathbf{x}\|)\frac{\mathbf{x}}{\sqrt{K}\|\mathbf{x}\|}$
3. **Distance**: $d_{\left(\mathcal{M}_K^d\right)}(\mathbf{x},\mathbf{y}) = \frac{2}{\sqrt{K}}\tanh^{-1}\left(\sqrt{K}\,\|-\mathbf{x}\oplus_K \mathbf{y}\|\right)$
4. **Mobius Addition**: $\mathbf{x}\oplus_K \mathbf{y} = \frac{\left(1+2K\mathbf{x}^T\mathbf{y}+K\|\mathbf{y}\|^2\right)\mathbf{x}+\left(1-K\|\mathbf{x}\|^2\right)\mathbf{y}}{1+2K\mathbf{x}^T\mathbf{y}+K^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2}$

Any of the above operations can be applied in mixed-curvature space as a weighted average of operation results in individual space. For example, the distance function in the mixed curvature space $\mathbb{P}$ can be written as: $d_{\mathbb{P}}^l(x,y) = \sum_{i=1}^{k} d_{\mathcal{M}_i^i}^l\left(x^{(i)}, y^{(i)}\right)$

2 Our Approach

We evaluate the Glove embeddings in 4 different spaces.

1. Euclidean space
2. Hyperbolic space
3. Spherical space
4. Mixed Curvature space

2.1 GloVe in Euclidean space

The GLOVE model suggests to learn embeddings as to satisfy the equation $w_i^T\tilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i)$. Bias term needs to be added to make P_{ik} symmetric and hence absorbing $\log(X_i)$ into i 's bias

$$w_i^T\tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik})\,.$$

Finally, the authors suggest to enforce this equality by optimizing a weighted least-square loss:

$$J = \sum_{i,j=1}^{V} f(X_{ij})\left(w_i^T\tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

where V is the size of the vocabulary and f down-weights the signal coming from frequent words (it is typically chosen to be $f(x) = \min\{1, (x/x_m)^\alpha\}$, with $\alpha = 3/4$ and $x_m = 100$).

2.2 GloVe in Mixed Curvature Space

From the above work we try to replace the Euclidean distance with the distance in metric space to train mixed curvature embeddings. We can rewrite the Eq. (1) with the Euclidean distance as $-\frac{1}{2}\left\|w_i-\tilde{w}_k\right\|^2+b_i+\tilde{b}_k=\log\left(X_{ik}\right)$, where we absorbed the squared norms of the embeddings into the biases. We thus replace the GLOVE loss by:

$$J=\sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(-k_1 * h_1\left(d\left(w_i,\tilde{w}_j\right)\right)-k_2 * h_2\left(d\left(w_i,\tilde{w}_j\right)\right)-k_3 * h_3\left(d\left(w_i,\tilde{w}_j\right)\right)+b_i+\tilde{b}_j-\log X_{ij}\right)^2$$

where k_1, k_2, k_3 are the attentions given to Euclidean space, hyperbolic space and spherical space respectively. h is a function to be chosen as a hyperparameter of the model, and d can be any differentiable distance function. Although the most direct correspondence with GloVe would suggest $h(x)=x^2/2$

2.3 Hyperbolic and Spherical space

These spaces are subsets of above mentioned mixed curvature space. Similar loss functions are used to train the models in these spaces as well.

2.4 Solving Analogies

The task of solving analogies involve finding a word W_d give words W_a, W_b, W_c where there is a semantic relation between these four words as: "Word a is to b as c is to d". There are several ways to solve analogies Sousa et al. [2020] but in general the task of solving analogies in Euclidean space can be formulated as $\overrightarrow{v_{d'}}=\arg\max_w \mathcal{S}(w,\overrightarrow{v_b}-\overrightarrow{v_a}+\overrightarrow{v_c})$ where $\mathcal{S}$ is some similarity metric and $\overrightarrow{v_x}$ is the learned vector for the word x. For euclidean space, the similarity function $\mathcal{S}$ can simply be cosine similairty between vectors, however, for non-zero curvature space we use negative distance as the distance metric. Finally, to translate vectors we use Mobius operators as detailed in Tifrea et al. [2018]. For mixed curvature space, we simply perform the operation in all the underlying spaces to find their correspondig vector $\overrightarrow{v_{d'\mathcal{M}_i}}$ in manifold $\mathcal{M}_i$. The final vector $\overrightarrow{v_{d'}}$ is the tuple of the underlying vectors.

3 Dataset

The task of training embeddings on large datasets is quite expensive. Hence we have limited the size of our dataset and used the Harry Potter dataset to train our embeddings different spaces. The dataset has been created from the Harry Potter books. We have preprocessed the dataset using a tool called spacy.

4 Experiments and Results

4.1 Experiment Setting

We have created Glove embeddings using the harry potter dataset in the euclidean space, hyperbolic space, spherical space and finally mixed curvature space. Here are the parameters using which we have trained the models: Context size: 3, embedding size: 128, epochs: 150, batch size 128, learning rate: 2e-4. For non-zero curvature spaces we use the Reimannian counterpart of the Adam Bécigneul and Ganea [2018] optimizer and train the weights using double-precision floating point numbers. All of our implementations are written using the PyTorch library Paszke et al. [2019] in Python 3.7. The loss plots show that our models are converge in their respective spaces in Figure 1.

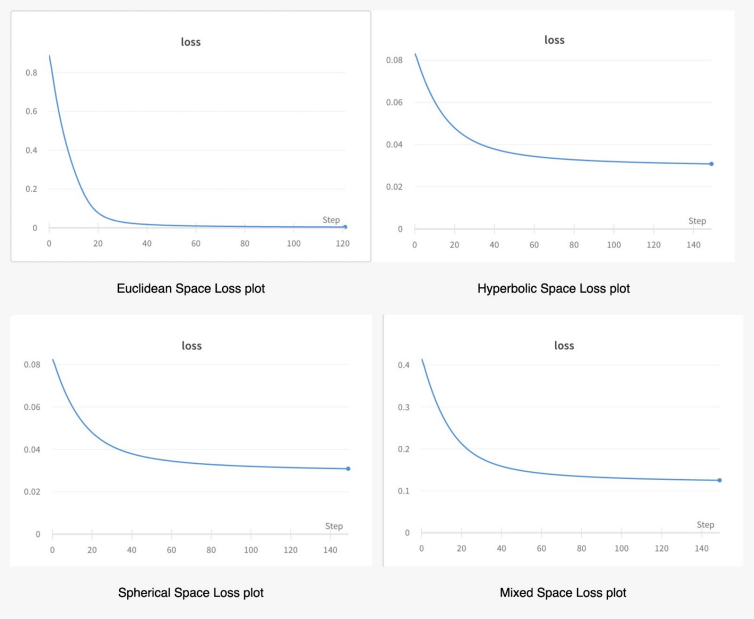

Figure 1: Loss Plots

Analogies
roger:davies :: dudley:dursley, remus:lupin :: barty:crouch, cho:ravenclaw :: cedric:hufflepuff, xenophilius:lovegood :: argus:filch, anthony:ravenclaw :: neville:gryffindor, bellatrix:slytherin :: colin:gryffindor, seamus:gryffindor :: padma:ravenclaw, sirius:black :: fred:weasley, peeves:poltergeist :: dobby:elf, trevor:toad :: bogrod:goblin

Table 1: Example of Some Analogies.

4.2 Ranked Analogy Test

We use the harry potter analogy dataset to evaluate our models. It contains various analogies with relations such as firstname-lastname, child-father, wizard-country, name-pet, founder-relic and more. Table 1 shows some analogy task examples.

Due to insignificant training resources, our GloVe models are extremely under-fitted. Therefore, for an intermediate comparison we use a modification of analogy task called – Ranked Analogy Task. We describe it as follows: Given a 4-tuple of word pairs (a, b, c, d) such that a-is-to-b as c-is-to-d. Instead of solving for d and reporting if there was an exact match, we report the rank of the word d in the neighborhood of $d^{'}$ where $d^{'}$ is computed by solving the analogy task through a, b and c as described in Section 2.4. In Table 2 we record the rank's average, standard deviation and median over the harry potter analogy dataset. Hyperbolic GloVe seems to be performing the best however, the standard deviation is so high that gaining any inference from these results wouldn't be wise.

Model	**Rank Mean**	**Rank Standard Deviation**	**Rank Median**
GloVe	112.7	53.6	117.0
Mixed GloVe	92.2	56.1	92.0
Hyperbolic GloVe	**47.4**	**51.4**	**32**
Spherical GloVe	59.1	56.3	49

Table 2: Ranked Analogy Test for the four GloVe models.

5 Conclusion and Future Work

We hypothesized that the Mixed-curvature space implementation of GloVe would offer some edge over the existing Vanilla, Poincare and Spherical GloVe. However, we observe no such gains. This could be because of difficulty in finding a minima for a very complex loss in mixed-curvature space or because of the slow-training and convergence over double precision floating point numbers required to train models in non-Euclidean space Tifrea et al. [2018]. For future work we would like to spend some more compute power on this problem and perform a more exhaustive study.

6 Link to the code

https://github.com/ayu15031/nlp-mixCurvature-Glove.git

References

Glove Implementation. `https://github.com/noaRricky/pytorch-glove`.

Pablo Badilla, Felipe Bravo-Marquez, and Jorge Pérez. Wefe: The word embeddings fairness evaluation framework. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 430–436. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.

Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. *arXiv preprint arXiv:1810.00760*, 2018.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 4356–4364, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Masahiro Kaneko and Danushka Bollegala. Gender-preserving debiasing for pre-trained word embeddings. *ArXiv*, abs/1906.00742, 2019.

Yu Meng, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance Kaplan, and Jiawei Han. Spherical text embedding. *Advances in Neural Information Processing Systems*, 32, 2019.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013b.

Orestis Papakyriakopoulos, Simon Hegelich, Juan Carlos Medina Serrano, and Fabienne Marco. Bias in word embeddings. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 446–457, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Tiago Sousa, Hugo Gonçalo Oliveira, and Ana Alves. Exploring different methods for solving analogies with portuguese word embeddings. In *9th Symposium on Languages, Applications and Technologies (SLATE 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincar\'e glove: Hyperbolic word embeddings. *arXiv preprint arXiv:1810.06546*, 2018.

Shen Wang, Xiaokai Wei, Cicero Nogueira Nogueira dos Santos, Zhiguo Wang, Ramesh Nallapati, Andrew Arnold, Bing Xiang, Philip S Yu, and Isabel F Cruz. Mixed-curvature multi-relational graph neural network for knowledge graph completion. In *Proceedings of the Web Conference 2021*, pages 1761–1771, 2021.

Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. Learning gender-neutral word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4847–4853, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1521. URL `https://aclanthology.org/D18-1521`.