Riemannian Optimization for Skip-Gram Negative Sampling

Alexander Fonarev[1,2,4], Oleksii Hrinchuk[1,2,3], Gleb Gusev[2,3], Pavel Serdyukov[2], Ivan Oseledets[1,5]

[1]Skolkovo Institute of Science and Technology, [2]Yandex LLC, [3]Moscow Institute of Physics and Technology, [4]SBDA Group, [5]Institute of Numerical Mathematics RAS

1. Problem

Training Skip-Gram Negative Sampling (SGNS) word embedding model ("word2vec") to measure the semantic similarity between them can be reformulated as a two-step procedure:

Step 1. Search for a low-rank matrix X that provides a good SGNS objective value;

Step 2. Search for a good low-rank representation $X = WC^\top$ in terms of linguistic metrics, where W is a matrix of word embeddings and C is a matrix of context embeddings.

2. Methodology

We apply Riemannian optimization for training SGNS word embedding model.

Main contributions

- We train SNGS word embedding model as a two-step procedure with clear objectives;
- We use Riemannian approach to optimize SGNS objective over low-rank matrices X for Step 1;
- We outperform state-of-the-art in both SGNS objective and the semantic similarity metric.

3. Background

SGNS as Implicit Matrix Factorization [2]

$$X = WC^T = (x_{wc}), \; x_{wc} = \langle \mathbf{w}, \mathbf{c} \rangle$$

$$\mathcal{M}_d = \{X \in \mathbb{R}^{n \times m} : \text{rank}(X) = d\}$$

$$F(X) = \sum_{w \in V_W} \sum_{c \in V_C} (\#(w,c)(\log \sigma(x_{wc}) + k\frac{\#(w)\#(c)}{|D|} \log \sigma(-x_{wc}))) \to \max_{X \in \mathcal{M}_d}$$

Riemannian Optimization

1. Projection of the gradient ascent step onto the tangent space:
$\hat{X}_{i+1} = X_i + \mathrm{P}_{\mathcal{T}_{X_i}\mathcal{M}_d} \nabla F(X_i)$
2. Retraction back to the manifold:
$X_{i+1} = R(\hat{X}_{i+1}) \in \mathcal{M}_d$

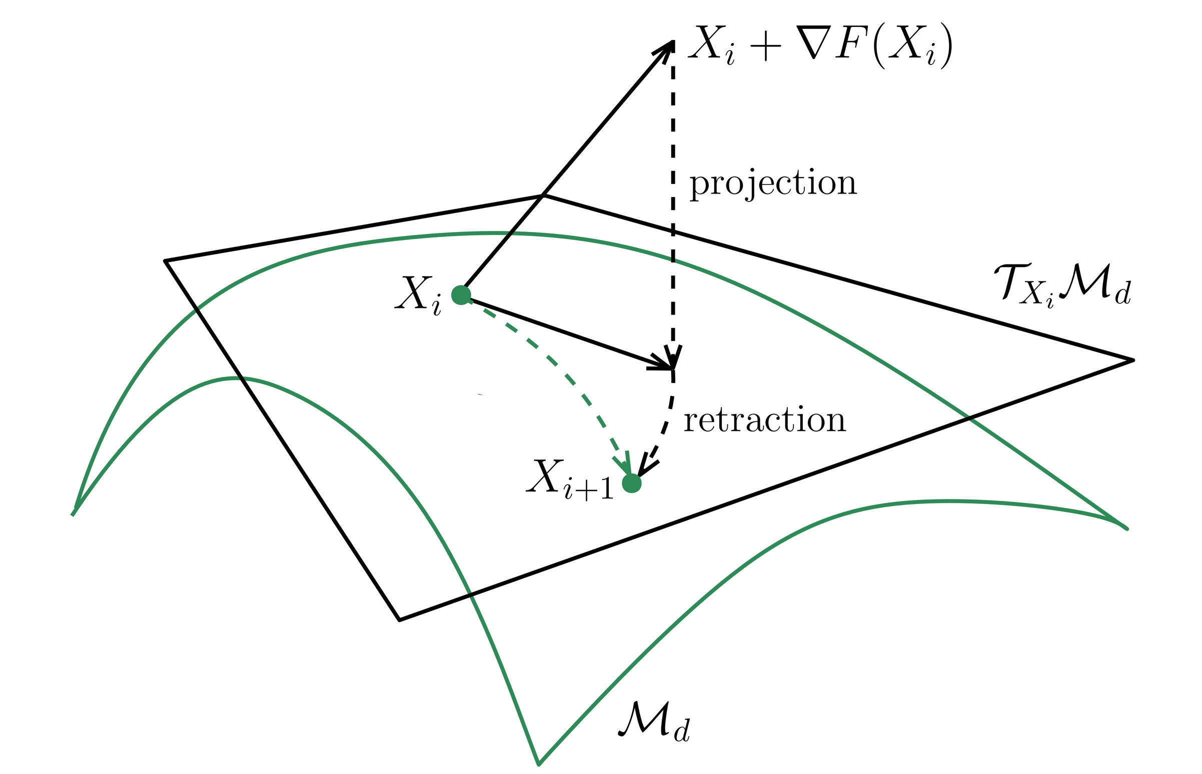

Figure 1: Geometric interpretation of one step of RO.

4. Algorithm

RO-SGNS

Require: gradient function $\nabla F : \mathbb{R}^{n \times m} \to \mathbb{R}^{n \times m}$, (W_0, C_0), maximum number of iterations K
Ensure: Word embeddings matrix $W \in \mathbb{R}^{n \times d}$

```
1:  X_0 ← W_0 C_0^T
2:  U_0, S_0, V_0^T ← SVD(X_0)
3:  for i ← 1, ..., K do
4:      X̂_i = X_{i−1} + λ∇F(X_{i−1})        λ – step-size
5:      U_i, S_i ← QR(X̂_i V_{i−1})           Retract point
6:      V_i, S_i^T ← QR(X̂_i^T U_i)           back to the
7:      X_i ← U_i S_i V_i^T                   manifold
8:  end for
9:  U, Σ, V^T ← SVD(X_K)
10: W ← U√Σ                                   Compute word embeddings
11: return W
```

Note. Highlighted retraction formulas approximate [3]:

$$X_{i+1} = R(X_i + \mathrm{P}_{\mathcal{T}_{X_i}\mathcal{M}_d} \nabla F(X_i))$$

.

5. Results

- Three different methods: RO-SGNS [1], SVD-SPPMI [2] and SGD-SGNS (original "word2vec").
- Three popular benchmarks for semantic similarity evaluation ("wordsim-353", "simlex", "men")
- Each dataset contains word pairs together with assessor-assigned similarity scores for each pair
- Original "wordsim-353" is a mixture of the word pairs for both word similarity and word relatedness tasks which we also use in our experiments ("ws-sim" and "ws-rel")

Dim. d	Algorithm	ws-sim	ws-rel	ws-full	simlex	men
$d = 100$	SGD-SGNS	0.719	0.570	0.662	0.288	0.645
	SVD-SPPMI	0.722	0.585	0.669	0.317	**0.686**
	RO-SGNS	**0.729**	**0.597**	**0.677**	**0.322**	0.683
$d = 200$	SGD-SGNS	0.733	0.584	0.677	0.317	0.664
	SVD-SPPMI	0.747	0.625	0.694	0.347	**0.710**
	RO-SGNS	**0.757**	**0.647**	**0.708**	**0.353**	0.701
$d = 500$	SGD-SGNS	0.738	0.600	0.688	0.350	0.712
	SVD-SPPMI	0.765	0.639	0.707	0.380	**0.737**
	RO-SGNS	**0.767**	**0.654**	**0.715**	**0.383**	0.732

Table 1: Spearman's correlation between predicted similarities and the manually assesed ones.

6. Examples

- We examine words, whose neighbors in terms of cosine distance change significantly.
- Table 2 contains the neighbors to the word "usa" with names of USA states marked bold.
- We represent top 11-20 nearest neighbors in Table 2, as top 10 words turned out to be exactly names of states for all three methods.

usa					
SGD-SGNS		SVD-SPPMI		RO-SGNS	
Neighbors	Dist.	Neighbors	Dist.	Neighbors	Dist.
akron	0.536	**wisconsin**	0.700	**georgia**	0.707
midwest	0.535	**delaware**	0.693	**delaware**	0.706
burbank	0.534	**ohio**	0.691	**maryland**	0.705
nevada	0.534	northeast	0.690	**illinois**	0.704
arizona	0.533	cities	0.688	madison	0.703
uk	0.532	southwest	0.684	**arkansas**	0.699
youngstown	0.532	places	0.684	**dakota**	0.690
utah	0.530	counties	0.681	**tennessee**	0.689
milwaukee	0.530	**maryland**	0.680	northeast	0.687
headquartered	0.527	**dakota**	0.674	**nebraska**	0.686

Table 2: Examples of the semantic neighbors for "usa".

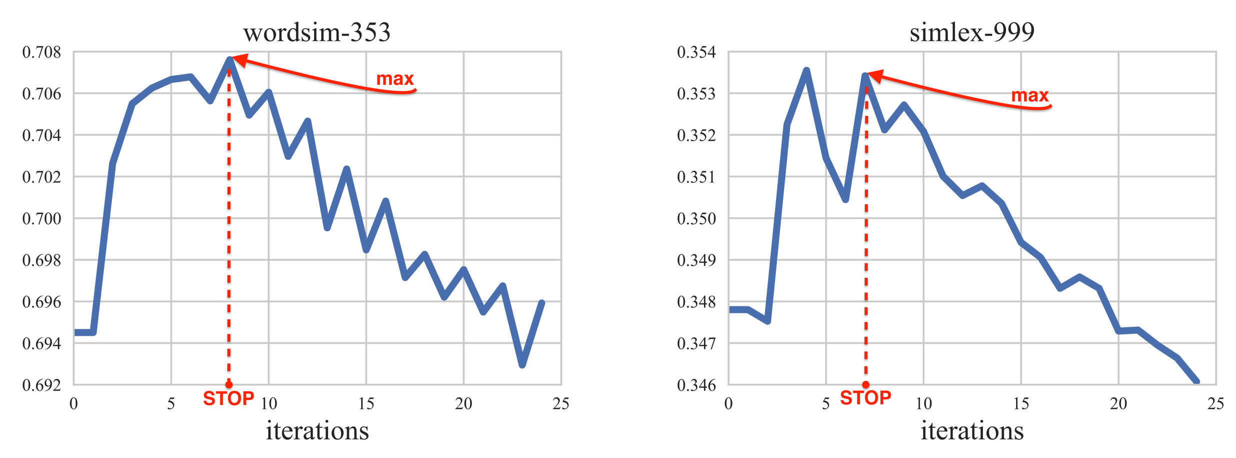

Figure 2: Spearman's correlation on each iteration.

- The best results were obtained when SVD-SPPMI was used as initialization.
- We need to stop optimization procedure on some iteration to get better model.
- Optimal value of K appeared to be the same for both test and its 10-fold cross-validation.

Source Code

- Python implementation of RO-SGNS
- templates of basic experiments
- semantic similarity datasets

`https://github.com/AlexGrinch/ro_sgns`

Future Work

- Apply more advanced optimization techniques to the Step 1 of the proposed scheme.
- Explore the Step 2 of obtaining embeddings with a given low-rank matrix.

References

[1] Alexander Fonarev, Oleksii Hrinchuk et al. Riemannian optimization for Skip-Gram Negative Sampling. In *ACL 2017*.

[2] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *NIPS 2014*.

[3] Christian Lubich and Ivan Oseledets. A projector-splitting integrator for dynamical low-rank approximation. In BIT *Numerical Mathematics 2014*.