Semantically Smooth Knowledge Graph Embedding

Shu Guo, Quan Wang, Bin Wang, Lihong Wang, Li Guo
Coordination Centre of Chine, Beijing, China

Presented by:
Avikalp Kumar Gupta, Computer Science and Engineering, IIT Kanpur

September 1, 2015

Outline

1. Introduction
    - Background
    - Purpose
2. Knowledge Graph Embedding
3. Semantically Smooth Embedding
    - Problem Formulation
    - Modelling Semantic Smoothness by LE
    - Modelling Semantic Smoothness by LLE
4. Experiments
    - Data Sets
    - Link Prediction
    - Triple Classification
5. Conclusion and Future Work
    - conclusion
    - Future Work

Background

- **Knowledge Graph (KG):** A multi-relational directed graph composed of entities as nodes and relations as edges
- Examples of Knowledge graphs: WordNet, Freebase, DB-pedia
- Application of Knowledge Graphs:
  - word sense disambiguation
  - named entity recognition
  - information extraction
- **Knowledge Graph Embedding:** A research direction which attempts to embed components of a KG into **continuous vector spaces**, so as to simplify the manipulation while preserving the inherent structure of the original graph

Purpose of this Paper

- To embed KGs consisting of entities and relations into low-dimensional vector spaces
- Requirement: learned embeddings should be compatible within each individual fact
- Aim: To also discover the intrinsic geometric structure of the embedding space

A Brief Review of KG Embedding

- KG embedding aims to embed entities and relations into a continuous vector space and model the plausibility of each fact in that space.
- In general, it consists of three steps:
  1. Representing entities (as points) and relations (as vectors, matrices or tensors) in a continuous vector space

     $\star$ Each edge in the KG is represented as a triple of fact $\langle e_i, r_k, e_j \rangle$, indicating that head entity e_i and tail entity e_j are connected by relation r_k.
  2. For each candidate fact $\langle e_i, r_k, e_j \rangle$, specifying a scoring (energy) function $f(e_i, r_k, e_j)$ to measure plausibility
  3. Learning the latent representations: To obtain the entity and relation representations, a *margin-based ranking loss* $\mathfrak{L}$ is minimized

$$\mathfrak{L} = \sum_{t^+ \in O} \sum_{t^- \in N_{t^+}} [\gamma + f(e_i, r_k, r_j) - f(e_i', r_k, e_j')]_+ \qquad (1)$$

Problem Formulation

- The entities (e) are classified into multiple semantic categories (c_e)
- An energy function on each candidate triple is defined (e.g. the energy functions listed in Table 1)
- To make the embedding space **semantically smooth**, the entity category information c_e is further leveraged (entities within the same semantic category should lie close to each other in the embedding space)
- This *smoothness assumption* is similar to the local invariance assumption exploited in manifold learning theory (i.e. nearby points are likely to have similar embeddings or labels). Thus two manifold learning algorithms **Laplacian Eigenmaps (LE)** and **Locally Linear Embedding (LLE)** are employed to model such semantic smoothness

Modelling Semantic Smoothness by LE

- **Laplacian Eigenmaps (LE):** A manifold learning algorithm that *preserves local invariance* between each 2 data points
- **Smoothness Assumption 1:** If two entities e_i and e_j belong to the same semantic category, they will have embeddings $\vec{e_i}$ and $\vec{e_j}$ close to each other.
- **Adjacency Matrix $\mathbf{W_1}$:** $w_{ij}^{(1)} = \begin{cases} 1 & \text{if } c_{e_i} = c_{e_j} \\ 0 & otherwise \end{cases}$
- **Measure of Smoothness:**

$$\mathfrak{R}_1 = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\vec{e_i} - \vec{e_j}\|_2^2 . w_{ij}^{(1)}$$

- Incorporate $\mathfrak{R}_1$ as a regularisation term in margin-based ranking loss(Eq.1), and hence minimize

$$\mathfrak{L}_1 = \frac{1}{N}\mathfrak{L} + \lambda_1\mathfrak{R}_1 \qquad (2)$$

Modelling Semantic Smoothness by LLE

Locally Linear Embedding(LLE)

- **Smoothness Assumption 2:** Each entity e_i can be roughly reconstructed by a linear combination of its nearest neighbors $N(e_i)$ in the embedding space, i.e., $\vec{e_i} \approx \sum_{e_j \in N(e_i)} \alpha_j \vec{e_j}$
- $N(e_i)$: K uniformly random entities from e_i's category
- **Weight matrix $\mathbf{W_2}$:** $w_{ij}^{(2)} = \begin{cases} 1 & \text{if } e_j \in N(e_i) \\ 0 & \textit{otherwise} \end{cases}$

  And normalize the rows so that $\forall i \sum_{j=1}^{n} w_{ij}^{(2)} = 1$
- **Measure of Smoothness:**

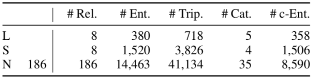
$$\mathfrak{R}_2 = \sum_{i=1}^{n} \|\vec{e_i} - \sum_{e_j \in N(e_i)} w_{ij}^{(2)} \vec{e_j}\|_2^2$$

- Incorporate $\mathfrak{R}_2$ in Eq.1, and hence minimize

$$\mathfrak{L}_2 = (1/N)\mathfrak{L} + \lambda_2 \mathfrak{R}_2 \qquad (3)$$

Data Sets

- Three data sets of different sizes:
  - L and S: small-scale data sets containing 8 relations on topics "location" and "sport" respectively
  - N 186 : a larger data set containing the most frequent 186 relations
- Entity category information is extracted from a specific relation called `Generalization`
- Table 3 gives some statistics of the three data sets after pre-processing



	# Rel.	# Ent.	# Trip.	# Cat.	# c-Ent.
L	8	380	718	5	358
S	8	1,520	3,826	4	1,506
N 186	186	14,463	41,134	35	8,590

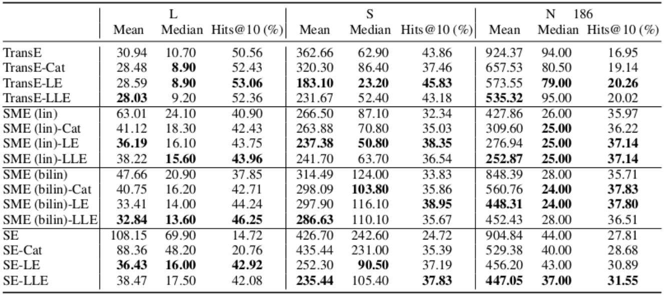
Table 3: Statistics of data sets.

- Notice that all the three data sets suffer from the data sparsity issue

Brief

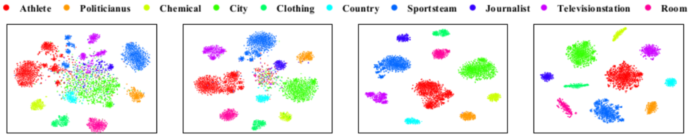
Link Prediction: To complete a triple $\langle e_i, r_k, e_j \rangle$ with e_i or e_j missing, i.e., predict e_i given (r_k, e_j) or predict e_j given (e_i, r_k).
Triple Classification: to verify whether a given triple $\langle e_i, r_k, e_j \rangle$ is correct or not

Link Prediction



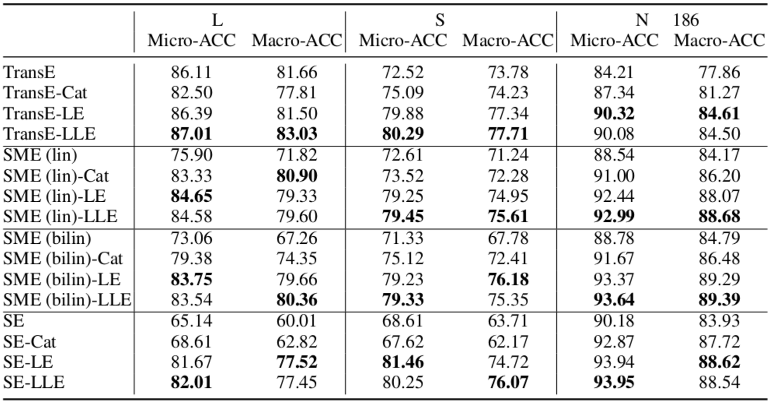
	L			S			N 186		
	Mean	Median	Hits@10 (%)	Mean	Median	Hits@10 (%)	Mean	Median	Hits@10 (%)
TransE	30.94	10.70	50.56	362.66	62.90	43.86	924.37	94.00	16.95
TransE-Cat	28.48	**8.90**	52.43	320.30	86.40	37.46	657.53	80.50	19.14
TransE-LE	28.59	**8.90**	**53.06**	**183.10**	**23.20**	**45.83**	573.55	**79.00**	**20.26**
TransE-LLE	**28.03**	9.20	52.36	231.67	52.40	43.18	**535.32**	95.00	20.02
SME (lin)	63.01	24.10	40.90	266.50	87.10	32.34	427.86	26.00	35.97
SME (lin)-Cat	41.12	18.30	42.43	263.88	70.80	35.03	309.60	**25.00**	36.22
SME (lin)-LE	**36.19**	16.10	43.75	**237.38**	**50.80**	**38.35**	276.94	**25.00**	**37.14**
SME (lin)-LLE	38.22	**15.60**	**43.96**	241.70	63.70	36.54	**252.87**	**25.00**	**37.14**
SME (bilin)	47.66	20.90	37.85	314.49	124.00	33.83	848.39	28.00	35.71
SME (bilin)-Cat	40.75	16.20	42.71	298.09	**103.80**	35.86	560.76	**24.00**	**37.83**
SME (bilin)-LE	33.41	14.00	44.24	297.90	116.10	**38.95**	**448.31**	**24.00**	**37.80**
SME (bilin)-LLE	**32.84**	**13.60**	**46.25**	**286.63**	110.10	35.67	452.43	28.00	36.51
SE	108.15	69.90	14.72	426.70	242.60	24.72	904.84	44.00	27.81
SE-Cat	88.36	48.20	20.76	435.44	231.00	35.39	529.38	40.00	28.68
SE-LE	**36.43**	**16.00**	**42.92**	252.30	**90.50**	37.19	456.20	43.00	30.89
SE-LLE	38.47	17.50	42.08	**235.44**	105.40	**37.83**	**447.05**	**37.00**	**31.55**

Table 4: Link prediction results on the test sets of L , S , and N 186.



(a) TransE. (b) TransE-Cat. (c) TransE-LE. (d) TransE-LLE.

Triple Classification



	L		S		N 186	
	Micro-ACC	Macro-ACC	Micro-ACC	Macro-ACC	Micro-ACC	Macro-ACC
TransE	86.11	81.66	72.52	73.78	84.21	77.86
TransE-Cat	82.50	77.81	75.09	74.23	87.34	81.27
TransE-LE	86.39	81.50	79.88	77.34	**90.32**	**84.61**
TransE-LLE	**87.01**	**83.03**	**80.29**	**77.71**	90.08	84.50
SME (lin)	75.90	71.82	72.61	71.24	88.54	84.17
SME (lin)-Cat	83.33	**80.90**	73.52	72.28	91.00	86.20
SME (lin)-LE	**84.65**	79.33	79.25	74.95	92.44	88.07
SME (lin)-LLE	84.58	79.60	**79.45**	**75.61**	**92.99**	**88.68**
SME (bilin)	73.06	67.26	71.33	67.78	88.78	84.79
SME (bilin)-Cat	79.38	74.35	75.12	72.41	91.67	86.48
SME (bilin)-LE	**83.75**	79.66	79.23	**76.18**	93.37	89.29
SME (bilin)-LLE	83.54	**80.36**	**79.33**	75.35	**93.64**	**89.39**
SE	65.14	60.01	68.61	63.71	90.18	83.93
SE-Cat	68.61	62.82	67.62	62.17	92.87	87.72
SE-LE	81.67	**77.52**	**81.46**	74.72	93.94	**88.62**
SE-LLE	**82.01**	77.45	80.25	**76.07**	**93.95**	88.54

Table 5: Triple classification results (%) on the test sets of L , S , and N 186.

Conclusion

- SSE imposes constraints on the **geometric structure** of the embedding space
- The semantic smoothness assumptions are constructed by using entities' **category information**, and then formulated as geometrically based regularization terms to constrain the embedding task
- By leveraging additional information besides observed triples, SSE can also deal with the **data sparsity**
- SSE *significantly and consistently outperforms* state-of-the-art embedding methods
- **Generalization:** The smoothness assumptions can actually be imposed to a wide variety of embedding models, and constructed using other information besides entities' semantic categories

Future Work

1. **Manifold regularization terms using other data sources:** Entity similarities can be derived in different ways, e.g., specified by users or calculated from entities textual descriptions.
2. **Efficiency and scalability enhancement:** Processing the manifold regularization terms can be time- and space-consuming (especially the one induced by the LE algorithm).
3. Impose the semantic smoothness assumptions on **other KG embedding methods** (e.g. those based on matrix/tensor factorization or Bayesian clustering), and even on **other embedding tasks** (e.g. word embedding or sentence embedding).

Thank You!

Questions?