Latent Distance Models and Graph Feature Models

2016-02-09

Dr. Asja Fischer, Prof. Jens Lehmann

Overview

- ▷ Last lecture we saw how neural networks/multi-layer perceptrons (MLPs) can be applied to SRL.
- ▷ After tensor-factorization methods they are the second class of latent distance models we saw in this course.
- ▷ In this lecture we will get to know a third class.
- ▷ Afterwards we will move to graph-feature models.

Latent distance models

- Belong to the class of latent variable models and are also known as **latent space models** in social network analysis.
- Basic idea: probability of relationships is based on **distance between latent presentations**.
- Approach fist proposed for modeling the probability of a relationship in a social network via

$$f(\boldsymbol{a}_j, \boldsymbol{a}_i) = -d(\boldsymbol{a}_j, \boldsymbol{a}_i)$$

where $d(\cdot, \cdot)$ is a arbitrary distance measure/metric, i.e. the Euclidean distance.

Recapitulation: Metric

Definition: Metric

Let X be an arbitrary space. A function $d : X \times X \to \mathbb{R}^+$ is called metric if it fulfills

1. $d(x, y) \geq 0$ (non-negativity)
2. $d(x, y) = 0 \Leftrightarrow x = y$ (identity of indiscernibles)
3. $d(x, y) = d(y, x)$ (symmetry)
4. $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)

▷ A well-known example is the Euclidean metric $||x - y||_2$.

Recapitulation: Norm

- Every norm $||\cdot||$ on a vector space, induces a metric $d(\boldsymbol{x}, \boldsymbol{y}) = ||\boldsymbol{x} - \boldsymbol{y}||$.

Definition: Norm

Let V be a vector space over the field F. A function $||\cdot|| : V \to \mathbb{R}^+$ is called metric for all $\boldsymbol{x}, \boldsymbol{y} \in V$ and all $s \in F$ it holds

1. $||\boldsymbol{x}|| = 0 \Leftrightarrow \boldsymbol{x} = 0$ (seperates points)
2. $||s\boldsymbol{x}|| = |s|||\boldsymbol{x}||$ (absolute homogenity)
3. $||\boldsymbol{x} + \boldsymbol{y}|| \leq ||\boldsymbol{x}|| + ||\boldsymbol{y}||$ (triangle inequality)

- The p-Norm of a real or complex vector $\boldsymbol{x} = (x_1, \ldots, x_n)$ for $1 \leq p \leq \infty$ is defined as $||\boldsymbol{x}||_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$.
- $p = 1$ results in the L_1/**Manhattan** distance: $||\boldsymbol{x} - \boldsymbol{y}||_1 = \sum_{i=1}^n |x_i - y_i|$.
- $p = 2$ results in the L_2/**Euclidean** distance: $||\boldsymbol{x} - \boldsymbol{y}||_2 = \sqrt{\sum_{i=1}^n |x_i - y_i|^2}$.

Structured Embedding (SE)[1]

- SE applies the idea of distance based score functions to multi-relational data.
- The score function is given by

$$f_{ijk}^{\text{SE}} = -||\boldsymbol{W}_k^s \boldsymbol{a}_i - \boldsymbol{W}_k^o \boldsymbol{a}_j||_1$$

  where the matrices $\boldsymbol{W}_k^s$ and $\boldsymbol{W}_k^o$ transform the global latent features of the entities to representations specific for the k-th relation.
- Higher score for existing than for non-existing entities. $\Rightarrow$ Pairs of entities in an existing relationship are closer than entities in non-existing entities.

[1] Bordes et al. Learning structured embeddings of knowledge bases. AAAI Conference on Artificial Intelligence, 2011

Structured Embedding (SE) – Knowledge extraction

- KG was constructed by extracting facts from text (Wikipedia articles).
- Link-prediction results based on SE on the KG:

e^l	people				
r	build	destroy	won	suffer	control
e^r	livelihoods	icons	emmy	sores	rocket
	homes	virtue	award	agitation	stores
	altars	donkeys	everything	treatise	emotions
	houses	cowboy	standings	eczema	spending
	ramps	chimpanzees	pounds	copd	fertility

Structured Embedding (SE) – Entity embeddings

- Closest entities from those of the top row according to the L1 distance of their embeddings:

_lawn_tennis_1	_artist_1	_field_1	_field_2	_pablo_picasso	_audrey_hepburn	_painter	_stanford_university
_badminton_1	_critic_1	_yard_9	_universal_set_1	_lin_liang	_wil_van_gogh	_artist	_univ._of_california
_squash_4	_part_7	_picnic_area_1	_diagonal_3	_zhou_fang	_signe_hasso	_printmaker	_city_univ._of_new_york
_baseball_1	_singer_1	_center_stage_1	_analysis_situs_1	_wu_guanzhong	_joyce_grenfell	_visual_artist	_stanford_law_school
_cricket_2	_prospector_1	_range_11	_positive_10	_paul_cezanne	_greta_garbo	_struct._engineer	_virginia_union_univ.
_hockey_2	_condition_3	_eden_1	_oblique_3	_yves_klein	_ingrid_bergman	_producer	_cornell_university
WordNet data				**Freebase data**			

- Embeddings can capture complex similarities.
- Embeddings can help for word-sense disambiguation of homonyms.

TransE model[2]

- The TransE model translates the latent feature presentations via a relation-specific offset r_k:

$$f_{ijk}^{\text{TransE}} = -d(\boldsymbol{a}_i + \boldsymbol{r}_k, \boldsymbol{a}_j) \ .$$

- This reduces the number of parameters over the SE model: per relation one vector $\boldsymbol{r}_k$ instead of two matrices $\boldsymbol{W}_k^o$ and $\boldsymbol{W}_k^s$.
- With $d(\cdot) = ||\cdot||_2$ and unity norm constrains on $\boldsymbol{a}_i, \boldsymbol{a}_j$ this gets

$$f_{ijk}^{\text{TransE}} = -\left(2\boldsymbol{r}_k(\boldsymbol{a}_i - \boldsymbol{a}_j) - 2\boldsymbol{a}_i^{\text{T}}\boldsymbol{a}_j + ||\boldsymbol{r}_k||_2^2\right) \ .$$

[2]Bordes et al. Translating embeddings for modeling multi-relational data, NIPS, 2013

Summary of latent feature models

We have seen different kind of latent feature models:

- ▷ tensor factorization methods
  - CP
  - RESCAL
- ▷ multi-layer perceptron (MLP) models
  - E-MLP
  - ER-MLP
- ▷ neural tensor networks (NTNs)
- ▷ latent distance models
  - structured embeddings (SE)
  - TransE

Comparison of latent feature models

Method	f_{ijk}	**Num. Parameters**
Rescal	$a_i R_{:,:,k} a_j^T$	$N_r H_e^2 + N_e H_e$
E-MLP	$r_k^T g(h_{ijk})$	$N_r(H_a + H_a \times 2H_e) + N_e H_e$
ER-MLP	$r^T g(h_{ijk})$	$H_c + H_c \times (2H_e + H_r) + N_r H_r + N_e H_e$
NTN	$r^T g([h^a_{ijk}, h^b_{ijk}])$	$N_r H_e^2 H_b + N_r(H_b + H_a) + 2N_r H_e H_a + N_e H_e$

No-Free-Lunch

Which model is the best?

- ▷ In the Knowledge Vault project[3] they found that ER-MLPs outperforms NTNs. On other link prediction tasks[4] RESCAL performs best.
- ▷ The best model depends on the data set.
- ▷ The **"no free lunch" theorem** states that in the lack of prior knowledge, any learning algorithm may fail on some learnable task (there is not an universal algorithms that performs well on all data sets).

[3] Dong et al., Knowledge Vault: A Web-scale approach to probabilistic knowledge fusion, ACM SIGKDD, 2014

[4] Yang et al. Embedding entities and relations for learning and inference in knowledege bases. CoRR, 2014

Graph feature models

- ▷ Assumption: existence of an edge can be predicted by extracting features from the observed edges in the graph.
- ▷ E.g. From the existence of the path

$$John \xrightarrow{\text{parentOf}} Anne \xleftarrow{parentOf} Mary$$

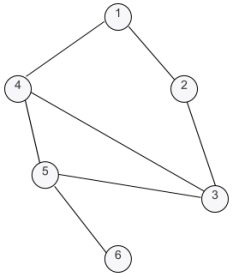
  one could predict the triple $(John, marriedTo, Mary)$.
- ▷ In contrast to latent feature models, graph feature models explain triples *directly from the observed triples in the KG.*

Some concepts from graph theory: graphs

Definition: Graph

A graph is a ordered pair $\mathcal{G} := (V, E)$ where

- V is a set of nodes (vertices)
- E is a set of edges (links).



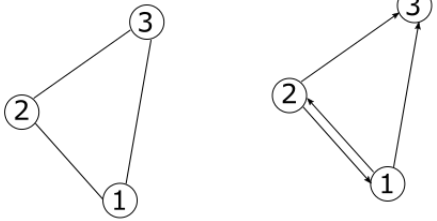
Some concepts from graph theory: (un)directed graphs

Directed graph:

- ▷ Edges are directed and correspond to ordered pairs of nodes.
- ▷ Can be used to model uni-relational data, e.g. links between web-pages.

Undirected graph:

- ▷ Edges are undirected and correspond to unordered pairs of nodes.
- ▷ Can be used to model uni-relational data with symmertic relation, e.g. friendships in social networks.



Some concepts from graph theory: multigraphs

From a mathematical perspective, linked data can be regarded as a labeled directed multigraph.

Definition: Labeled directed multigraph

A labeled directed multigraph is a tuple $\mathcal{G} := (V, L, E)$ where

- V is a set of vertices
- L is a set of edge labels
- $E \subseteq V \times V \times L$ is a set of ordered triples

Some concepts from graph theory: multigraphs

From a mathematical perspective, linked data can be regarded as a labeled directed multigraph.

Definition: Labeled directed multigraph

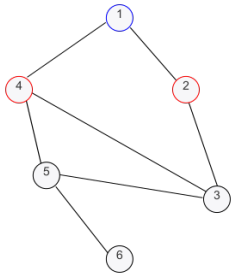
A labeled directed multigraph is a tuple $\mathcal{G} := (V, L, E)$ where

- V is a set of vertices
- L is a set of edge labels
- $E \subseteq V \times V \times L$ is a set of ordered triples

- V corresponds to the entities in the domain
- L corresponds to the different relation types
- E corresponds to the known facts about the entities (RDF triples)

Some concepts from graph theory: neighborhood

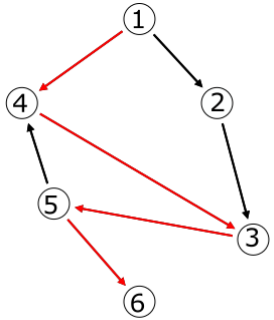
- Two vertices are called **adjacent** if they share a common edge.
- The **neighborhood** $N(v)$ of a vertex $v \in V$ in a graph $\mathcal{G}$ is the set of vertices adjacent to v



- The **degree** of a vertex is the total number of vertices adjacent to the vertex $degree(v) = |N(v)|$.
- For directed graphs one can distinguish between **indegree** and **outdegree**.

Some concepts from graph theory: paths

- A **path** in a graph represents a way to get from node v to node u traversing edges.
- Formally they can be described as a ordered list of directed edges (even in undirected graphs).



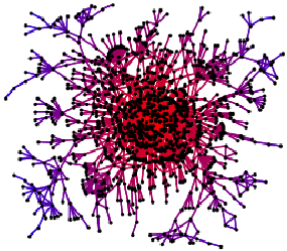
path:{(1,4),(4,3),(3,5),(5,6)}

Similarity measures for uni-relational data

- ▷ Observable graph feature models are widely used for link-prediction in graphs with a single relation.
- ▷ Graphs with single relations are e.g. often found in social network analysis (friendships between people), biology (interactions of proteins), and Web mining (hyperlinks between Web sites).
- ▷ Intuition: similar entities are likely to be related (homophily) and similarity of entities can be derived from the neighborhood of nodes or from the existence of path between notes.

Local similarity indices for uni-relational data

- Local similarity indices derive the similarity of entities from their **number of common neighbors** or their **absolute number of neighbors**.
- They are fast to compute for KGs with a single relationship and scale well to large KGs, since they only depend on direct neighborhood.
- However they can be to localized to capture long-range or global dependencies.



Local similarity indices: examples

Let $N(e_i)$ and $N(e_j)$ be the neighborhoods of nodes e_i and e_j, respectively. The probability of a link existing between both nodes depends on their similarity, which can be estimated based on

1. **Common neighbors**: the number of common neighbors

$$f_{ij} = |N(e_i) \cap N(e_j)|$$

2. **Jaccard's coefficient**: the probability of having a randomly selected neighbor in common

$$f_{ij} = \frac{|N(e_i) \cap N(e_j)|}{|N(e_i) \cup N(e_j)|}$$

Local similarity indices: examples

Let $N(e_i)$ and $N(e_j)$ be the neighborhoods of nodes e_i and e_j, respectively. The probability of a link existing between both nodes depends on their similarity, which can be estimated based on

3. **Adamic/Adar**: sum of common neighbors where rarer neighbors are weighted more heavily

$$f_{ij} = \sum_{e_h \in N(e_i) \cap N(e_j)} \frac{1}{\log(|N(e_h)|)}$$

4. **Preferential attachment**: based on the number of neighbors of each node, where the probability that new edge in a growing network involves node e_i is proportional to $|N(e_i)|$

$$f_{ij} = |N(e_i)| \cdot |N(e_j)|$$

Global similarity indices for uni-relational data

- ▷ Global similarity indices derive the similarity of entities either from the **ensemble of all paths** between them or from **random walks** on the graph.
- ▷ They often provide significantly better predictions than local indices.
- ▷ However, they are computationally more expensive.

Global similarity indices: examples

1. **Katz** is example for a global similarity index based on the ensemble of all paths between e_i and e_j. Let $\Pi^l_{e_i,e_j} = \{\text{path of length } l \text{ from } e_i \text{ to } e_j\}$.

   ▷ The similarity measure is based on the sum of all paths exponentially damped by length

   $$f_{ij} = \sum_{l=1}^{\infty} \beta^l \cdot |\Pi^l_{e_i,e_j}|$$

   where $0 \leq \beta \leq 1$.

   ▷ Very small values for β yield predictions like common neighbors.

   ▷ One can show that with adjacency matrix A the matrix of scores is given by $(\boldsymbol{I} - \beta \boldsymbol{A})^{-1} - \boldsymbol{I}$.

Global similarity indices: examples

A random work starts at a note and moves in each step to a neighbor of the current note chosen uniformly at random. Examples for global similarity indeces are then based on

1. **Hitting time**: the expected number of steps H_{ei,e_j} for reaching e_j from e_i in a random work
$$f_{ij} = -H_{e_i,e_j}$$
2. **Commute time**: the symmetric sum of hitting time from e_i to e_j and vise verse
$$f_{ij} = -H_{e_i,e_j} + H_{e_j,e_i}$$
.
3. **PageRank**: the stationary probability of e_j in a random walk that returns to e_i with probability α each step and moves to a random neighbor with probability $1 - \alpha$.

Quasi-local and high-level similarity indices

- ▷ **Quasi-local similarity indices** try to balance predictive accuracy and computational complexity by deriving the similarity estimates based on paths and random walks of **bounded length**.
- ▷ **High-level approaches based on low-rank approximations**:
  - As shown for the Katz index, the score function can often be written based on the adjacency matrix of the graph.
  - Replacing it by a low-rank approximation can be viewed as a "noise-reduction" technique keeping most of the structure in the matrix but with a greatly simplified representation.

Path Ranking Algorithm (PRA)[5]

- ▷ PRA extends the ideas of quasi-local similarity indices for uni-relational networks to large multi-relational KGs.
- ▷ Let $t = (r_1, r_2, \ldots, r_l)$ define a sequence of l edge types.
- ▷ Let $\Pi_{e_i,e_j}(t)$ denote the set of paths between e_i and e_j following the sequence t.
- ▷ Such paths can be discovered by
  - enumerating all type consistent paths from entities of the type of e_i to entities of the type of e_j.
  - random sampling in the case of too many relations.

[5] Lao and Cohen. *Relational retrieval using a combination of path-constraint random walks*. Machine Learning, 2010.

PRA: the probability of paths

- Consider a random walk where in each step we follow an outgoing link uniformly at random.
- The probability of following any path in $\Pi_{e_i,e_j}(t)$ (or following sequence t from e_i to e_j respectively) can be calculated recursively.
- For an empty sequence ($t = \{\}$ and $l = 0$) define

$$P(\Pi_{e_i,e_j}(t)) = \begin{cases} 1 & \text{, if } e_j = ei \\ 0 & \text{, otherwise} \end{cases} .$$

For an nonempty sequence with $t = (r_1, \ldots, r_l)$ let $t' = (r_1, \ldots, r_{l-1})$ and define

$$P(\Pi_{e_i,e_j}(t)) = \sum_{e_h \in range(r_{l-1})} P(\Pi_{e_i,e_h}(t'))Pr(e_j \xleftarrow{r_l} e_h) .$$

PRA: basic idea

- Key idea: Given a set of (relation-k specific) paths Π_{ijk} with maximum length L between e_i and e_j, use the corresponding probabilities as features for predicting the probability of missing edges.
- Define a feature vector

$$\boldsymbol{x}_{ijk}^{\text{PRA}} = [P(\Pi_{e_i,e_j}(t)) : \Pi_{e_i,e_j}(t) \subset \Pi_{ijk}] \ .$$

- Predict the edge probabilities based on logistic regression with relation specific regression parameters $\boldsymbol{r}_k$. That is

$$f_{ijk}^{\text{PRA}} = \boldsymbol{r}_k^{\text{T}} \boldsymbol{x}^{\text{PRA}} \ .$$

Path Ranking Algorithm (PRA): Interpretabilty

- ▷ Relation paths can be regarded as bodies of weighted rules (Horn clauses).
- ▷ For example for predicting the college a person attended, i.e. to predict triples $(p, college, c)$ observing the path $p \xrightarrow{draftedBy} t \xrightarrow{school} c$ could be a useful.
- ▷ This could be written as the Horn clause

$$(p, college, c) \leftarrow (p, draftedBy, t) \wedge (t, school, c) \ .$$

- ▷ The learned weight specifies how predictive the body of the rule is for the head.

Path Ranking Algorithm (PRA): Example

Relation Path	F1	Prec	Rec	Weight
(draftedBy, school)	0.03	1.0	0.01	2.62
(sibling(s), sibling, education, institution)	0.05	0.55	0.02	1.88
(spouse(s), spouse, education, institution)	0.06	0.41	0.02	1.87
(parents, education, institution)	0.04	0.29	0.02	1.37
(children, education, institution)	0.05	0.21	0.02	1.85
(placeOfBirth, peopleBornHere, education)	0.13	0.1	0.38	6.4
(type, instance, education, institution)	0.05	0.04	0.34	1.74
(profession, peopleWithProf., edu., inst.)	0.04	0.03	0.33	2.19

Summary

- We got to know a third group of models belonging to the class of latent feature models: latent distance models.
- The basic idea of latent distance models (e.g. Structured Embeddings and TransE) is to make the probability of the relation ship between two entities dependent on the distance of their latent representation.
- Graph feature models explain the existence of triples from features directly observed in the KG.
- Local similarity indices for uni-relational data are based on the number of (common) neighbors.
- Global similarity indices for uni-relational data are based on ensamble of all paths or on random walks.
- Quasi-local similarity indies for uni-relational data are based on paths and random walks with bounded length.
- The path ranking algorithm transfers this idea to multi-relational data bases and resulting paths can be interpreted as Horn clauses.

References

- Link prediction in social networks: Liben-Nowell and Kleinberg. The link prediction problem for social networks. *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019-1031, 2007.