A latent factor model for highly multi-relational data

Rodolphe Jenatton† Nicolas Le Roux* Antoine Bordes° Guillaume Obozinski* (RJ and NLR contributed equally)

†CMAP, UMR CNRS 7641, Ecole Polytechnique, Palaiseau, France
*INRIA-Sierra Project (INRIA/ENS/CNRS UMR 8548), 23, avenue d'Italie, 75214 Paris. France
°Heudiasyc, UMR CNRS 7253, Université de Technologie de Compiègne, France

One minute overview

- Method to model data of the form `{subject,relation,object}`
  - e.g., recommender system, social networks, NLP,...
- Interested in a setting with many relation types ($\gtrsim 10^3$)
- Main idea: Learn latent representations of subjects,relations and objects combined in a trilinear model
- Scalability: Sharing sparse latent factors among relations
- Good empirical performance:
  - Standard tensor-factorization benchmarks
  - Large-scale NLP application

Relational data modeling

Setting:

- n_s subjects $\{\mathcal{S}_i\}_{i\in[\![1;n_s]\!]}$
- n_r relations $\{\mathcal{R}_j\}_{j\in[\![1;n_r]\!]}$
- n_o objects $\{\mathcal{O}_k\}_{k\in[\![1;n_o]\!]}$

- A relationship exists for the triplet $(\mathcal{S}_i, \mathcal{R}_j, \mathcal{O}_k)$ if $\mathcal{R}_j(\mathcal{S}_i, \mathcal{O}_k) = 1$
  - e.g., a subject and a direct object linked through a transitive verb in NLP

Goal: We want to model

$$\mathbb{P}[\mathcal{R}_j(\mathcal{S}_i, \mathcal{O}_k) = 1]$$

(equivalently, approximate a binary tensor $\mathbf{X} \in \{0,1\}^{n_s \times n_o \times n_r}$)

Our approach:

- Cast the problem as matrix factorizations
- Represent the subjects and objects as vectors in $\mathbb{R}^p$
  - $\{\mathcal{S}_i\}_{i\in[\![1;n_s]\!]} \quad \rightarrow \quad \mathbf{S} \triangleq [\mathbf{s}^1, \ldots, \mathbf{s}^{n_s}] \in \mathbb{R}^{p\times n_s}$
  - $\{\mathcal{O}_k\}_{k\in[\![1;n_o]\!]} \quad \rightarrow \quad \mathbf{O} \triangleq [\mathbf{o}^1, \ldots, \mathbf{o}^{n_o}] \in \mathbb{R}^{p\times n_o}$
- Relations are matrices on which subjects and objects operate
  - $\{\mathcal{R}_j\}_{j\in[\![1;n_r]\!]} \quad \rightarrow \quad \{\mathbf{R}_j\}_{j\in[\![1;n_r]\!]} \in \mathbb{R}^{p\times p}$
- A logistic model:

$$\mathbb{P}[\mathcal{R}_j(\mathcal{S}_i, \mathcal{O}_k) = 1] \triangleq \sigma\Big(\mathcal{E}(\mathbf{s}^i, \mathbf{R}_j, \mathbf{o}^k)\Big), \quad \text{with} \quad \sigma(t) \triangleq 1/(1+e^{-t})$$

Related work

- Tensor factorization methods [Tucker, 1966; Harshman et al., 1994]
- With latent and shared/clustered attributes:
  - Collective matrix factorization [Paccanaro et al, 2001; Nickel et al., 2011]
  - Non-parametric Bayesian [Kemp et al., 2006; Sutskever et al., 2009; Miller et al, 2009; Zhu, 2012]
  - Markov-Logic networks [Kok et al., 2007]
  - Neural networks [Bordes et al., 2012]

A multiple order log-odds ratio model

- $\mathcal{E}(\mathbf{s}^i, \mathbf{R}_j, \mathbf{o}^k)$ accounts for 1-,2- and 3-way interactions
  - For instance: unigrams, bigrams and trigrams in NLP

- For some parameters $\mathbf{y}, \mathbf{y}', \mathbf{z}, \mathbf{z}' \in \mathbb{R}^p$, we define

$$\mathcal{E}(\mathbf{s}^i, \mathbf{R}_j, \mathbf{o}^k) \triangleq \overbrace{\langle \mathbf{y}, \mathbf{R}_j \mathbf{y}' \rangle}^{\text{unigram}} + \underbrace{\langle \mathbf{s}^i, \mathbf{R}_j \mathbf{z} \rangle + \langle \mathbf{z}', \mathbf{R}_j \mathbf{o}^k \rangle}_{\text{bigram}} + \overbrace{\langle \mathbf{s}^i, \mathbf{R}_j \mathbf{o}^k \rangle}^{\text{trigram}}$$

Sharing parameters across relations

- Motivation:
  - With many relation types ($n_r \gg 1$), we might have few data per relation
  - Relations can have similarities (e.g., synonyms in NLP)
  - Maybe memory expensive to store $n_r \times p^2$ elements

- Idea:
  - Decompose relations over a common set of d rank one matrices $\{\mathbf{\Theta}_r\}_{r\in[\![1;d]\!]}$
  - $\{\mathbf{\Theta}_r\}_{r\in[\![1;d]\!]}$ represent some "canonical" relations

$$\begin{cases} \mathbf{R}_j = \sum_{r=1}^{d} \alpha_r^j \mathbf{\Theta}_r, & \text{for some sparse } \ \alpha^j \in \mathbb{R}^d \\ \text{with } \ \mathbf{\Theta}_r = \mathbf{u}_r \mathbf{v}_r^\top & \text{for some } \ \mathbf{u}_r, \mathbf{v}_r \in \mathbb{R}^p \end{cases}$$

Optimization

- $\mathcal{P}/\mathcal{N}$ is the set of indices of positively/negatively labeled relations

- We maximize the following likelihood:

$$\mathcal{L} \triangleq \prod_{(i,j,k)\in\mathcal{P}} \mathbb{P}[\mathcal{R}_j(\mathcal{S}_i, \mathcal{O}_k) = 1] \quad \cdot \prod_{(i',j',k')\in\mathcal{N}} \mathbb{P}[\mathcal{R}_{j'}(\mathcal{S}_{i'}, \mathcal{O}_{k'}) = 0]$$

- After proper normalization, it leads to the minimization problem:

$$\min_{\substack{\mathbf{S},\mathbf{O},\{\alpha^j\},\\ \{\mathbf{\Theta}_r\}\mathbf{y},\mathbf{y}',\mathbf{z},\mathbf{z}'}} -\log(\mathcal{L}), \text{ with } \begin{cases} \|\alpha^j\|_1 \leq \lambda, \ \mathbf{\Theta}_r = \mathbf{u}_r \cdot \mathbf{v}_r^\top, \\ \mathbf{z} = \mathbf{z}', \mathbf{O} = \mathbf{S}, \\ \mathbf{s}^i, \mathbf{o}^k, \mathbf{y}, \mathbf{y}', \mathbf{z}, \mathbf{u}_r \text{ and } \mathbf{v}_r \text{ in } \Big\{\mathbf{w};\, \|\mathbf{w}\|_2 \leq 1\Big\} \end{cases}$$

  - The problem is non convex
  - Apply stochastic projected gradient descent
  - Use mini-batches
  - In some applications, the set $\mathcal{N}$ is not given $\rightarrow$ Sampling schemes
  - Can be useful to down-weight negative triplets

Experiments

- Matlab code and datasets available at `http://bit.ly/hdrl`

(1) Multi-relational benchmarks

- Setting:
  - Datasets: Kinships, UMLS and Nations
  - Dimensions: $n_s = n_o \approx 100$ and $n_r \approx 50$
  - 10 cross-validation for choices of $\{p, d, \lambda\}$
  - Goal: Predict relationships

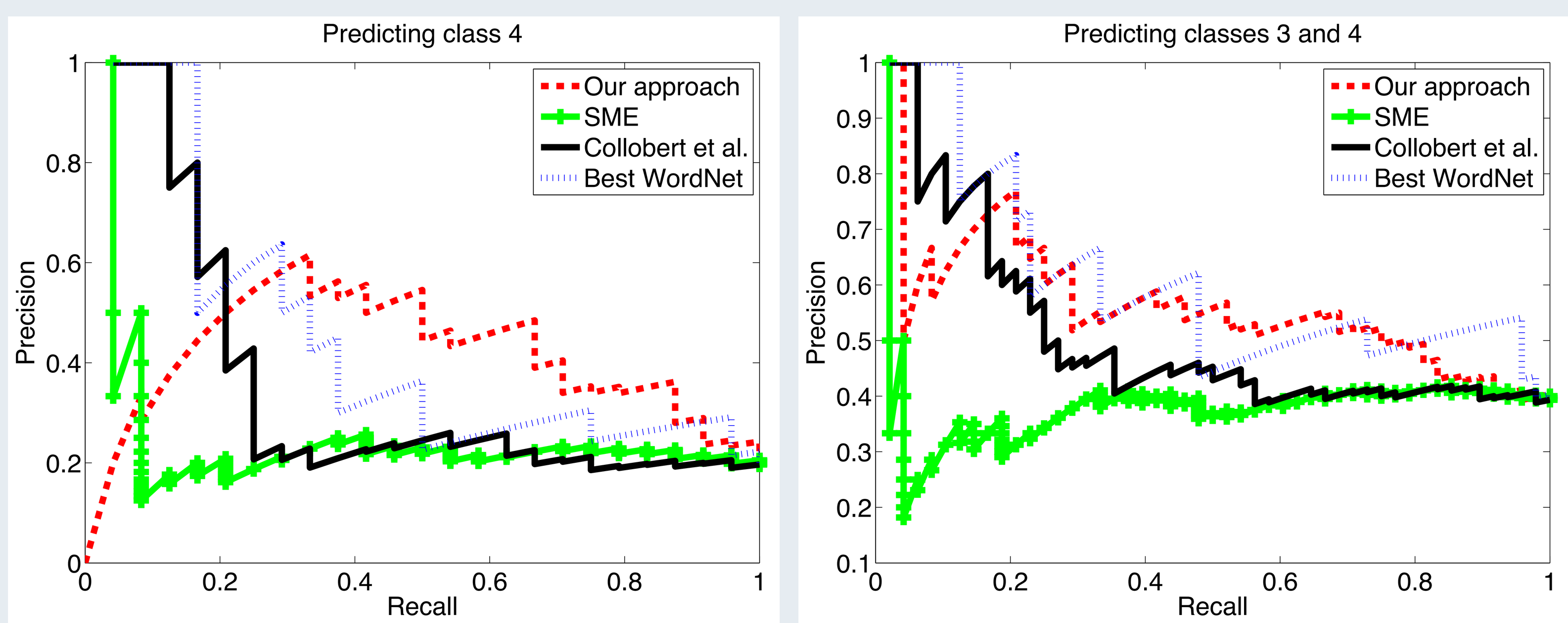
	Kinships		UMLS		Nations	
	AUC (PR)	Log-likelihood	AUC (PR)	Log-likelihood	AUC (PR)	Log-likelihood
Our approach	**0.946** $\pm$ 0.005	**-0.029** $\pm$ 0.001	**0.990** $\pm$ 0.003	**-0.002** $\pm$ 0.0003	**0.909** $\pm$ 0.009	**-0.202** $\pm$ 0.008
Nickel et al. (2011)	**0.95**	N/A	0.98	N/A	0.84	N/A
Kok et al. (2007)	0.84	-0.045 $\pm$ 0.002	0.98	-0.004 $\pm$ 0.001	0.75	-0.311 $\pm$ 0.022
Bordes et al. (2012)	0.907 $\pm$ 0.008	N/A	0.983 $\pm$ 0.003	N/A	**0.883** $\pm$ 0.02	N/A

(2) NLP application

- Setting:
  - Triplets (subject, verb, direct object) extracted from Wikipedia
  - 1,000,000/50,000/250,000 triplets for training/validation/test
  - 30,605 subjects and direct objects/**4,547** verbs
  - Goal: Predict a verb given the subject and object

	synonyms not considered			best synonyms considered		
	median/mean rank	p@5	p@20	median/mean rank	p@5	p@20
Our approach	50 / **195.0**	**0.78**	**0.95**	19 / **96.7**	**0.89**	**0.98**
Bordes et al. (2012)	56 / 199.6	0.77	**0.95**	19 / 99.2	**0.89**	**0.98**
Bigram	**48** / 517.4	0.72	0.83	**17** / 157.7	0.87	0.95

- Evaluate latent representations: Lexical similarity classification
  - Human annoted dataset from [Yang et al., 2006]
  - 130 pairs of verbs labeled with a score in $\{0,1,2,3,4\}$
    e.g., `(divide,split)` is labeled 4, while `(postpone,show)` has a score of 0
  - Idea: if $\mathbf{R}_j \approx \mathbf{R}_{j'}$, then the verbs j and j' should be similar



	AUC (class 4)	AUC (classes 3&4)
Our approach	**0.40**	0.54
Bordes et al. (2012)	0.21	0.36
Collobert et al. (2011)	0.31	0.48
Best WordNet	**0.40**	**0.59**