Experiments in Graph-based Semi-Supervised Learning Methods for Class-Instance Acquisition

Samuel Broscheit

Seminar: Graph-based Methods for Natural Language Processing (WS 2010)
Dozent: Dr. Simone Paolo Ponzetto
Seminar für Computerlinguistik, Universität Heidelberg

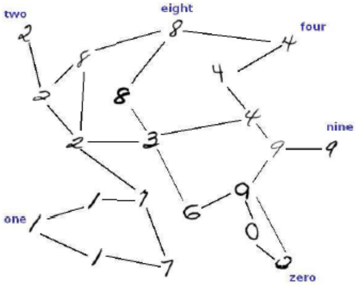
Gliederung

Motivation

- Researchers in NLP often face the problem that they only have a small amount of labelled data for their research.

Motivation

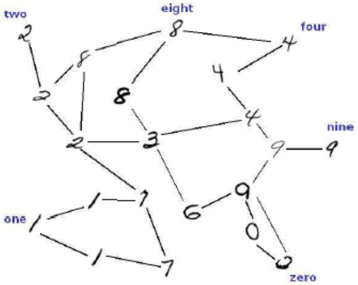
- Researchers in NLP often face the problem that they only have a small amount of labelled data for their research.



(Zhu et al. 2003)

Motivation

- Researchers in NLP often face the problem that they only have a small amount of labelled data for their research.

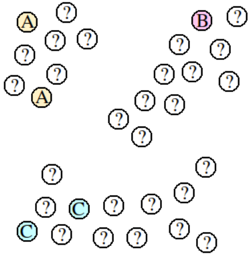


(Zhu et al. 2003)

→ Using Semi-Supervised Learning **(SSL)** we can create labeled data from labeled and unlabeled instances (transductive inference)

Semi Supervised Learning in Graphs

- Given labeled instances L and unlabeled instances U
- a weighted Graph $G = (V, E, W)$ is constructed.
- $V = V_L \cup V_U$ the union of labeled and unlabeled nodes
- $E = \{\, uv \mid W(u, v) > \epsilon\}$
- $W : V \times V \to \mathbb{R} := sim(v, w) \;\; v, w \in V$



What this talk is about

- Graphs model the paradigm of transductive inference.
- Questions:

What this talk is about

- Graphs model the paradigm of transductive inference.
- Questions:
  - × What features are used to describe the instances?

What this talk is about

- Graphs model the paradigm of transductive inference.
- Questions:
  - × What features are used to describe the instances?
  - × How is the similarity between the instances defined to construct the graph from feature space?

What this talk is about

- Graphs model the paradigm of transductive inference.
- Questions:

  × What features are used to describe the instances?

  × How is the similarity between the instances defined to construct the graph from feature space?

  → What kind of algorithms propagates the labels in a *desired* way through the graph?

What this talk is about

- Graphs model the paradigm of transductive inference.
- Questions:

  $\times$ What features are used to describe the instances?

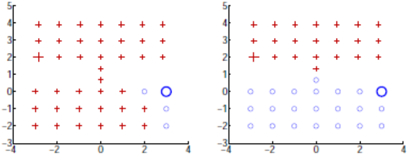
  $\times$ How is the similarity between the instances defined to construct the graph from feature space?

  $\rightarrow$ What kind of algorithms propagates the labels in a *desired* way through the graph?

  $\rightarrow$ How do such algorithms perform?

Why isn’t this trivial?

- Maxbe kNN could do this as well

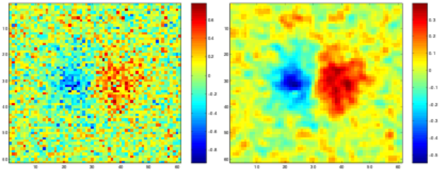


(a) (b) (Zhu et al. 2003)

- But: no notion of **smoothness**
- The mass of the labeled (big) instances get lost.

Label propagation method (LP-ZGL) (Zhu et al. 2003)

- Zhu applied mathematical proven methods used in physics to the SSL machine learning problem



(Chung 2003)

- The algorithm is derived from the model of a gaussian random field
- and formulates an objective that minimizes the error from adjacent nodes having different labels

Label propagation method (LP-ZGL) (Zhu et al. 2003)

- C is the set of labels
- D is the matrix with $D_{uu} = \sum\limits_{v \in V} W_{uv}$
- L is the Laplician $D - W$
- Y is a matrix with training labels
- $\hat{Y}$ a matrix with soft label assigments

$\rightarrow$ then the objective minimized by LP-ZGL is:

$$\min_{\hat{Y}} \sum_{l \in C} \hat{Y}_l^T L \hat{Y}_l^T = \sum_{u,v \in V, l \in C} W_{uv} (\hat{Y}_{ul} - \hat{Y}_{vl})^2 \tag{1}$$

- The node labels are then computed with loopy belief propagation.

Adsorption (Baluja et al. 2008)

- Was engineered for the video recommandation system at youtube.
- The graph was constructed by using co-views: v_1 and v_2 are connected if they have been seen by the same user.

- The algorithm iteratively updates the graph with

$$\hat{Y}_v^{t+1} = p_v^{inj} \times Y_v + p_v^{cont} \times B_v^t + p_v^{abdnmt} \times r \qquad (2)$$

- p_v^{inj} is the probability injected by the labeled node
- p_v^{cont} is the probability to continue
- p_v^{abdnmt} is the probability to stop at this node
- $p_v^{inj} + p_v^{cont} + p_v^{abdnmt} = 1$

Adsorption (Baluja et al. 2008)

$$B_v^t = \sum_u \frac{W_{uv}}{\sum_{u'} W_{u'v}} \hat{Y}_u^t \tag{3}$$

- B_v^t is the normalized injected mass from the incident neighbours
- r absorbs the injected mass of the node, if the node isn't trusted (f.ex. high degree node)

Modified Adsorption (MAD) (Talukdar and Crammer 2010)

- Talukdar and Crammer investigated if there is an objective that gets minimized by Adsorption
- There isn't one
- Thus they reformulated the basic ideas of Adsorption into an objective function that can be optimized

- $S_{vv} = 1$ if v is a labeled node

$$\min_{\hat{Y}} \sum_{l \in C} [\mu_1 (Y_l - \hat{Y}_l)^T S (Y_l - \hat{Y}_l) + \mu_2 \hat{Y}_l^T L' \hat{Y}_l + \mu_3 ||\hat{Y}_l - R_l||^2] \tag{4}$$

- μ_1 - μ_3 are the probabilities as in Adsorption

Measure

- All evaluations will use the Mean Reciprocal Rank (MRR)

$$MRR = \frac{1}{|Q|} \sum_{v \in Q} \frac{1}{rank(v)} \tag{5}$$

- Q are the tested nodes
- $rank(v)$ is the rank the gold label has in node v

Datasets

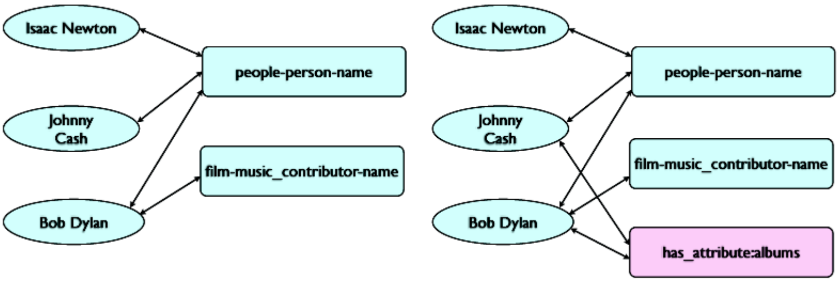
- **Freebase**

  “a large collaborative knowledge base [... which] harvests information from many open data sets (for instance Wikipedia and MusicBrainz), as well as from user contributions”
- **TextRunner**

  “an open domain IE system“ which offers extracted hypernym tuples
- **Yago Knowledge Base**

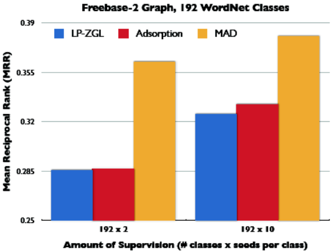
  “a light-weight and extensible ontology with high coverage [... which] contains more than 1 million entities and 5 million facts.” (Suchanek et al. 2007)

Constructed Graph

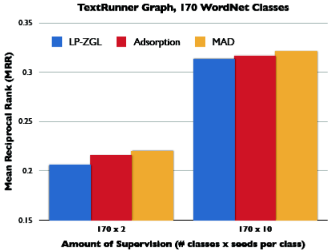


(a) (b)

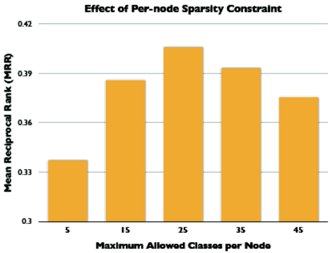
Results


Freebase-2 Graph, 192 WordNet Classes
LP-ZGL
Adsorption
MAD
0.39
0.355
0.32
0.285
0.25
Mean Reciprocal Rank (MRR)
192 x 2
192 x 10
Amount of Supervision (# classes x seeds per class)

Results

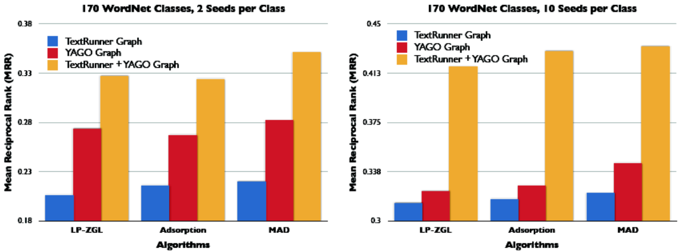

TextRunner Graph, 170 WordNet Classes
LP-ZGL
Adsorption
MAD
Mean Reciprocal Rank (MRR)
0.35
0.3
0.25
0.2
0.15
170 x 2
170 x 10
Amount of Supervision (# classes x seeds per class)

Results



Effect of Per-node Sparsity Constraint
0.42
0.39
0.36
0.33
0.3
Mean Reciprocal Rank (MRR)
5
15
25
35
45
Maximum Allowed Classes per Node

Results



170 WordNet Classes, 2 Seeds per Class
TextRunner Graph
YAGO Graph
TextRunner + YAGO Graph
Mean Reciprocal Rank (MRR)
0.38
0.33
0.28
0.23
0.18
LP-ZGL
Adsorption
MAD
Algorithms
170 WordNet Classes, 10 Seeds per Class
TextRunner Graph
YAGO Graph
TextRunner + YAGO Graph
Mean Reciprocal Rank (MRR)
0.45
0.413
0.375
0.338
0.3
LP-ZGL
Adsorption
MAD
Algorithms

Results

YAGO Attribute	**Top-2 WordNet Classes Assigned by MAD** (example instances for each class are shown in brackets)
has_currency	**wordnet_country_108544813 (Burma, Afghanistan)** wordnet_region_108630039 (Aosta Valley, Southern Flinders Ranges)
works_at	**wordnet_scientist_110560637 (Aage Niels Bohr, Adi Shamir)** wordnet_person_100007846 (Catherine Cornelius, Jamie White)
has_capital	**wordnet_state_108654360 (Agusan del Norte, Bali)** wordnet_region_108630039 (Aosta Valley, Southern Flinders Ranges)
born_in	**wordnet_boxer_109870208 (George Chuvalo, Fernando Montiel)** wordnet_chancellor_109906986 (Godon Brown, Bill Bryson)
has_isbn	**wordnet_book_106410904 (Past Imperfect, Berlin Diary)** wordnet_magazine_106595351 (Railway Age, Investors Chronicle)

Table 2: Top 2 (out of 170) WordNet classes assigned by MAD on 5 randomly chosen YAGO attribute nodes (out of 80) in the TextRunner + YAGO graph used in Figure 7 (see Section 3.6), with 10 seeds per class used. A few example instances of each WordNet class is shown within brackets. Top ranked class for each attribute is shown in bold.

Conclusion

- MAD outperforms Adsorption and LP-ZGL
- Adding attributes to the graph helps class inference
- If you want to do SSL and you can define the edge weights on your feature space, MAD might currently be the state of the art solution

Questions that haven't been answered by the author:

- How does the rate of improvement develops with increasing seed size?
- How close are the false postives and would an evaluation with precision@k be sensible?

References I

Shumeet Baluja, Rohan Seth, D. Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed Aly. Video suggestion and discovery for youtube. In *Proceeding of the 17th international conference on World Wide Web - WWW '08*, page 895, Beijing, China, 2008. doi: 10.1145/1367497.1367618. URL http://portal.acm.org/citation.cfm?id=1367618.

Moo K. Chung. Stat 992: Lecture 01, December 2003. URL http://www.stat.wisc.edu/~mchung/teaching/stat992/ima01.pdf.

Fabian Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 697–706, Banff, Alberta, Canada, 2007. ACM.

Partha Pratim Talukdar and Koby Crammer. New regularized algorithms for transductive learning, February 2010. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.153.405.

References II

Partha Pratim Talukdar and Fernando Pereira. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, page 1473–1481, Morristown, NJ, USA, 2010. Association for Computational Linguistics. URL `http://portal.acm.org/citation.cfm?id=1858681.1858830`.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-Supervised learning using gaussian fields and harmonic functions. *IN ICML*, pages 912—919, 2003. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.14.4312`.