Learning Representations in Directed Networks

O. Ivanov[1] S. Bartunov[2]

[1]Department of Computational Mathematics and Cybernetics
Lomonosov Moscow State University

[2]Department of Computer Science
National Research University Higher School of Economics

April 3, 2015

Outline

Networks in our life

- Social networks.
- Web graphs.
- Biological networks (protein–protein interaction).
- Citation networks.
- ...

Why learning representations?

- Visualization.
- Exploration of the network structure.
- Making network nodes independent.

Outline

Latent representation

- For each node we have *input* and *output* representation: $In_u, Out_u \in \mathbb{R}^D$.
- Bilinear Link Model explains local connections by latent representations of the nodes.
- For a link with a fixed source node we assign probability according to the following bilinear softmax model:

$$p(v|u, \theta) = \frac{\exp(In_u^T Out_v)}{\sum\limits_{w \in V} \exp(In_u^T Out_w)}$$

Network likelihood

- Joint links log–likelihood:

$$J_C(\theta) = \sum_{(u,v)\in E} \ln p(v|u,\theta)p(u|\theta)$$

- Maximum likelihood principle:

$$J_C(\theta) = \sum_{(u,v)\in E} \ln p(v|u,\theta) + \sum_{(u,v)\in E} \ln p(u) \to \max_{\theta,p(u)}$$

- One can find optimal $p(u)$ analytically:

$$p(u) = \frac{d_+(u)}{|E|}$$

Network likelihood

- A problem with $\sum_{(u,v)\in E} \ln p(v|u,\theta) \to \max_{\theta}$.
- Complexity of GD step is $O(|V|^2 D)$
- Complexity of SGD step on single random link is $O(|V|D)$
- Complexity of SGD epoch is $O(|V||E|D)$
- :(

Outline

Noise Contrastive Estimation

- We have a set of observed data $X = \{x_1, x_2, \dots, x_{T_d}\}, x_i \in M$ from an unknown distribution $p_d(x)$. Our goal is to find $p_d(x)$.
- We generate *noise data* set $Y = \{y_1, y_2, \dots, y_{T_n}\}, y_i \in M$ from a know distribution $p_n(y)$.
- $U = X \cup Y$. For each data point u_t we assign label
$$c_t = \begin{cases} 1, & u_t \in X \\ 0, & u_t \in Y \end{cases}$$
- We have a normalized model for observed data $p_m(x|\theta)$.

Noise Contrastive Estimation

What is a posteriori probability to observe C with given U?

$$p(u|c=1,\theta) = p_m(u|\theta) \quad p(u|c=0,\theta) = p_n(u)$$

$$\nu = \frac{p(c=0)}{p(c=1)} = \frac{T_n}{T_d}$$

$$p(c=1|u,\theta) = \frac{p_m(u|\theta)}{p_m(u|\theta) + \nu p_n(u)}$$

$$p(c=0|u,\theta) = \frac{\nu p_n(u)}{p_m(u|\theta) + \nu p_n(u)}$$

Noise Contrastive Estimation

What is a posteriori probability to observe C with given U?

$$L(\theta) = \sum_{t=1}^{T_d} \ln p(c = 1 | x_t, \theta) + \sum_{t=1}^{T_n} \ln p(c = 0 | y_t, \theta)$$

$-L(\theta)$ is also known as cross–entropy error function.
Maximizing a posteriori probability leads to approaching the X objects properties with parameter θ.

Noise Contrastive Estimation

$$J_T(\theta) = \frac{1}{T_d}\left\{\sum_{t=1}^{T_d} \ln p(c=1|x_t,\theta) + \sum_{t=1}^{T_n} \ln p(c=0|y_t,\theta)\right\} =$$

$$= \frac{1}{T_d}\sum_{t=1}^{T_d} \ln p(c=1|x_t,\theta) + \nu\frac{1}{T_n}\sum_{t=1}^{T_n} \ln p(c=0|y_t,\theta)$$

$$J_\infty(\theta) = \mathbb{E}_{x\sim p_d} \ln p(c=1|x,\theta) + \nu\mathbb{E}_{y\sim p_n} \ln p(c=0|y,\theta)$$

Noise Contrastive Estimation

$$\tilde{J}(f_m) = \mathbb{E} \ln \frac{\exp(f_m(x))}{\exp(f_m(x)) + \nu p_n(x)} + \nu \mathbb{E} \ln \frac{\nu p_n(y)}{\exp(f_m(y)) + \nu p_n(y)}$$

Theorem

$\tilde{J}$ attains a maximum at $f_m = \ln p_d$. There are no other extrema if the noise density p_n is chosen such that it is nonzero whenever p_d is nonzero.

Note

A fundamental point in the theorem is that the maximization is performed without any normalization constraint for f_m.

Noise Contrastive Estimation

Consider an unnormalized model for observed data $p_m^0(\cdot|\theta)$. $p_m^0(\cdot|\theta) \geq 0$.

$$J(\theta) = \mathbb{E} \ln \frac{p_m^0(\cdot|\theta)}{p_m^0(\cdot|\theta) + \nu p_n(x)} + \nu \mathbb{E} \ln \frac{\nu p_n(y)}{p_m^0(\cdot|\theta) + \nu p_n(y)}$$

Corollary

If the noise density p_n is chosen such that it is nonzero whenever p_d is nonzero and exists θ^* for which $p_m^0(\cdot|\theta^*) = p_d(\cdot)$, then $J(\theta)$ attains a maximum at θ^*.

Noise Contrastive Estimation

A little bit more talk about NCE:
Global optimum $\hat{\theta}^*_T$ of J_T converges in probability to θ^*, if

- exists θ^* for which $p_m(\cdot|\theta^*) = p_d(\cdot)$
- p_n is nonzero whenever p_d is nonzero
- J_T uniformly converges in probability to J
- one more condition on p_n, p_m and p_d is fulfilled

Noise Contrastive Estimation

$\sqrt{T_d}(\hat{\theta}^*_T - \theta^*)$ is asymptotically normal with mean zero.

For large sample sizes T_d, the mean squared error $\mathbb{E}(||\hat{\theta}^*_T - \theta^*||^2)$ equals C/T_d.

Noise Contrastive Estimation

Recommendations for choosing p_n and ν:

- Choose noise for which an analytical expression for $\ln p_n$ is available.
- Choose noise that can be sampled easily.
- Choose noise that is in some aspect, for example with respect to its covariance structure, similar to the data.
- Make the noise sample size as large as computationally possible.

Outline

Back to the networks

$$J_C(\theta) = \sum_{(u,v)\in E} \ln p(v|u,\theta)p(u) \to \max_{\theta}$$

Applying NCE

$$Z_u \approx \ln \sum_{w \in V} \exp(In_u^T Out_v)$$

$$p(v|u,\theta) = \exp(In_u^T Out_v - Z_u)$$

$$p(u,v|\theta) = p(u)p(v|u,\theta) \quad p_n(u,v) = p(u)p_n(v)$$

$$L_m(u,v|\alpha) = \ln \frac{p(v|u,\theta)}{p(v|u,\theta) + \nu p_n(v)}$$

$$L_n(u,v|\alpha) = \ln \frac{\nu p_n(v)}{p(v|u,\theta) + \nu p_n(v)}$$

$$J_{NCE}(\alpha) = \sum_{(u,v)\in E}^{|E|} L_m(u,v|\alpha) + \sum_{(\tilde{u},\tilde{v})\sim(p_u,p_n)}^{\nu|E|} L_n(\tilde{u},\tilde{v})$$

Introducing approximation

$$J_{NCE}(\alpha) = \sum_{(u,v)\in E}^{|E|} L_m(u, v|\alpha) + \sum_{(\tilde{u},\tilde{v})\sim(p_u,p_n)}^{\nu|E|} L_n(\tilde{u}, \tilde{v}|\alpha)$$

$$p(\tilde{u}) = p(u)$$

$$\hat{J}_{NCE}(\alpha) = \sum_{(u,v)\in E}^{|E|} \left(L_m(u, v|\alpha) + \sum_{\tilde{v}\sim p_n}^{\nu} L_n(u, \tilde{v}|\alpha) \right)$$

Introducing regularizer

J_C tends to divergence. Actually, if node u have the only outgoing link (u, v), then the corresponding component of J_C is

$$\ln \frac{\exp(In_u^T Out_v)}{\sum_{w \in V} \exp(In_u^T Out_w)}$$

One can show, that maximum obtains at $In_u = Out_v \cdot \infty$.
Surely, the same trouble we have with $\hat{J}_{NCE}$.
To approach this issue, we use regularizer $R(\alpha)$.

Introducing regularizer

We use weighted L_2–regularizer, because it has the property of isotropy for In_u and Out_u.

$$R(\alpha) = (\nu + 1) \sum_{u \in V} \left(d_+(u) ||In_u||_2^2 + d_-(u) ||Out_u||_2^2 \right)$$

Unweighted L_2–regularizer is also allowed.

$$\hat{J}_{NCE,R}(\alpha) = \hat{J}_{NCE}(\alpha) + \gamma R(\alpha)$$

Final functional

$$\hat{J}_{NCE,R}(\alpha) =$$

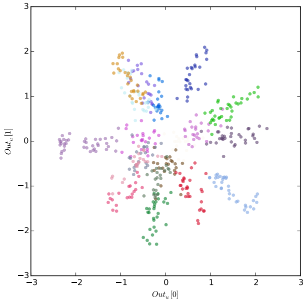
$$= \sum_{(u,v) \in E}^{|E|} \Bigg(\{ L_m(u, v|\alpha) + \gamma(||In_u||^2 + ||Out_v||^2) \} +$$

$$+ \sum_{\tilde{v} \sim p_n}^{\nu} \{ L_n(u, v|\alpha) + \gamma(||In_u||^2 + ||Out_{\tilde{v}}||^2) \} \Bigg)$$

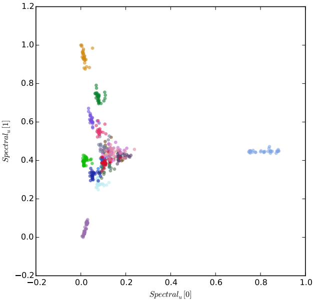
Stochastic gradient descent, parallel implementation, HOGWILD!, AdaGrad, online learning are welcome!

Outline

Random scale–free network

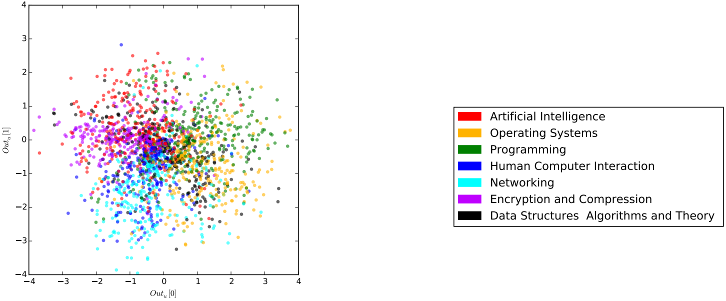


BLM output vectors



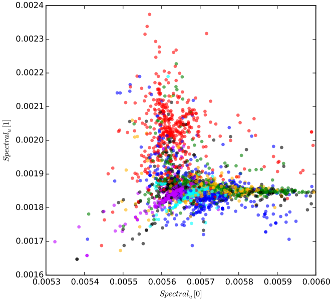
Spectral layout

Visualization of Cora citation network communities

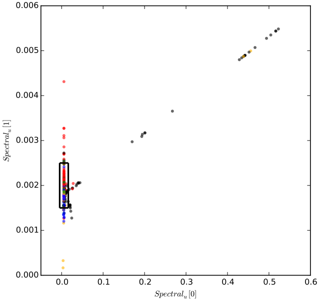


BLM output vectors

Spectral visualization of Cora citation network communities



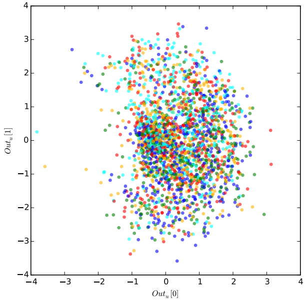
Spectral layout (zoom of the selected area on the right figure)



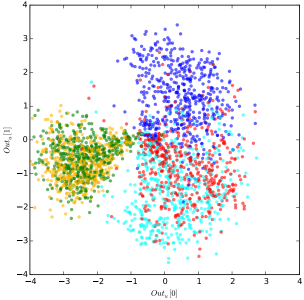
Spectral layout (all nodes)

BLM visualization of LiveJournal communities

Not more than 500 random nodes from each community.



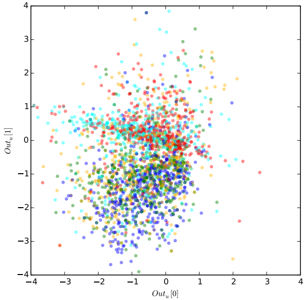
The 5 largest communities



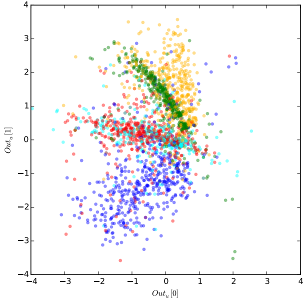
Communities 2001-st to 2005-th in descending order of size

BLM visualization of YouTube communities

Not more than 500 random nodes from each community.



The 5 largest communities



Communities from 11-th to 15-th in descending order of size

Outline

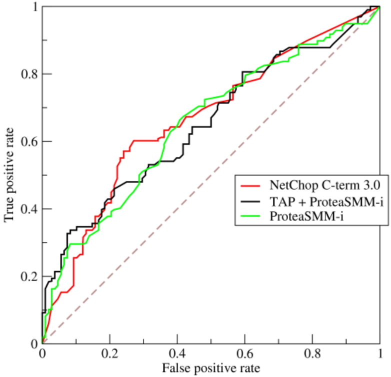
Link Prediction

We want to use representations α to get link probability for each link according to model $p(u, v|\alpha)$.
Introducing a separator p_{min} turns model into a classifier, which shows if the link probability is more than random:

$$a(u, v) = \mathbb{I}(p(u, v|\alpha) > p_{min})$$

Area Under the Curve is a base framework for measuring the quality of classifier.

Area Under the Curve



$$TPR = \frac{TP}{TP+FN} \qquad FPR = \frac{FP}{FP+TN}$$

AUC estimation

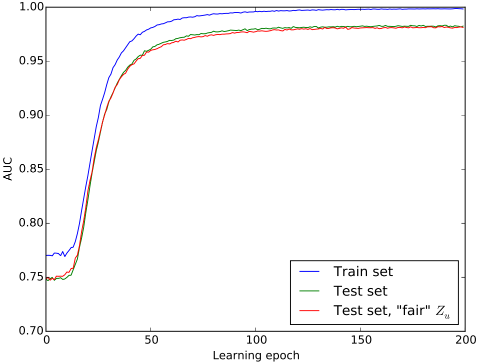
N is negative objects, P is positive.

$$AUC = \frac{1}{|N|} \sum_{i \in N} TPR_i = \frac{1}{|P||N|} \sum_{\substack{i \in N \\ j \in P}} \mathbb{I}(a(i) < a(j))$$

Too expensive to compute.
A fast stochastic estimation is applicable!
We can sample uniformly $n_1, n_2, \ldots, n_C \in N$ and $p_1, p_2, \ldots, p_C \in P$.
So the unbiased estimation of AUC is

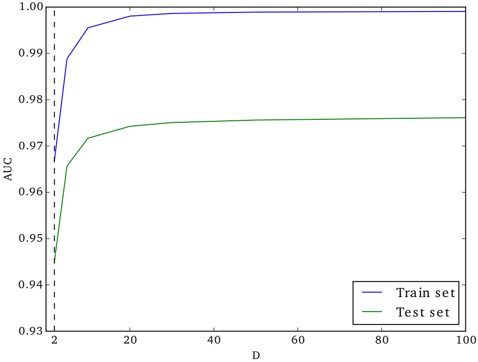
$$AUC \approx \frac{1}{C} \sum_{i=1}^{C} \mathbb{I}(p(n_i) < p(p_i))$$

AUC convergence



AUC estimation on citation cit-HepPh network during optimization (constant learning rate)

AUC dependence on D



Dependence of AUC on dimensionality of representations on LiveJournal

Baselines

- Jaccard — local similarity index, a stupid easy–implemented baseline.

$$S_{uv}^{Jaccard} = \frac{|Neigh(u) \cap Neigh(v)|}{|Neigh(u) \cup Neigh(v)|}$$

- Local Random Walk — random walk, state of the art.

$$S_{uv}^{LRW}(t) = \frac{d_+(u)}{|E|}((P^T)^t e_u)_v + \frac{d_+(v)}{|E|}((P^T)^t e_v)_u$$

- Superposed Random Walk — random walk, state of the art.

$$S_{uv}^{SRW}(t) = \sum_{i=1}^{t} S_{uv}^{LRW}$$

Time competition

Link prediction methods performance. Time was measured for LiveJournal social network on one core of Intel(R) Xeon(R) E5-2670 2.60GHz CPU.
Time for BLM does not take into account the training effort.

	BLM	Jaccard	LRW (T steps)	SRW (T steps)
Score function evaluation cost	$O(D)$	$O(\frac{\lvert E\rvert}{\lvert V\rvert})$	$O(\lvert E\rvert T)$	$O(\lvert E\rvert T)$
Parameters for one evaluation	$D = 30$		$T = 3$	$T = 3$
Time for one evaluation, sec.	10^{-6}	$4.65 \cdot 10^{-5}$	3.13	3.13

Quality competition

Link prediction, AUC ($n_{total} = 10^5$)

Dataset	BLM(30)	Jaccard	LRW(3)	SRW(3)
soc-LiveJournal	0.975	0.938	**0.986**	0.985
soc-Pocek	**0.978**	0.850	0.966	0.967
web-Google	0.961	0.945	0.977	**0.978**
web-BerkStan	0.979	0.960	**0.996**	**0.996**
cit-HepPh	0.983	0.962	0.988	**0.989**

Future work

- Other models for $p(v|u, \theta)$.
- Properties of the obtained representations investigation (making classifiers on them and so on).
- Cross–cluster hypothesis.

Future work
Cross-cluster hypothesis

ACC — Average Clustering Coefficient

Dataset	BLM(30)	Jaccard	LRW(3)	SRW(3)	ACC
soc-Pocek	**0.978**	0.850	0.966	0.967	0.109
soc-LiveJournal	0.975	0.938	**0.986**	0.985	0.274
cit-HepPh	0.983	0.962	0.988	**0.989**	0.285
web-Google	0.961	0.945	0.977	**0.978**	0.514
web-BerkStan	0.979	0.960	**0.996**	**0.996**	0.597

Questions?

Thanks for listening!