The Peril and Promise of Machine Learning

Benedek Rózemberczki

The University of Edinburgh
Center for Doctoral Training in Data Science

15.05.2018.

About this talk...

This is not a talk about:

1. The General Data Protection Regulation
2. Theoretical machine learning
3. Privacy

This is a talk about:

1. Some intuition
2. Applied machine learning
3. Graph representation learning
4. Raising awareness

A made up threat about machine learning

Technology Intelligence

Facebook shuts down robots after they invent their own language

share 55

Facebook chief executive Mark Zuckerberg has supported AI development CREDIT: REUTERS

FACEBOOK

No, Facebook Did Not Panic and Shut Down an AI Program That Was Getting Dangerously Smart

Tom McKay
7/31/17 8:31pm • Filed to: SKYNET

488.0K 183 14

The real threat about machine learning


Cambridge
Analytica

Graph representation learning

Let $G = (V, E)$ be the graph of interest. The graph representation learning method is a mapping $z : V \rightarrow \mathbb{R}^d$ where d is the dimensionality of the representation space.

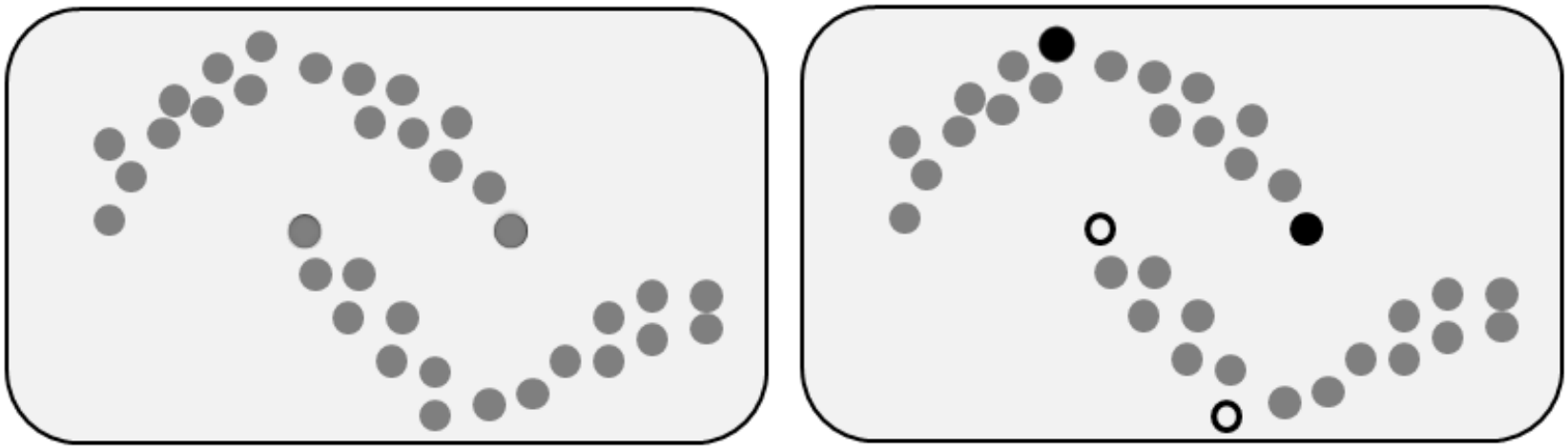
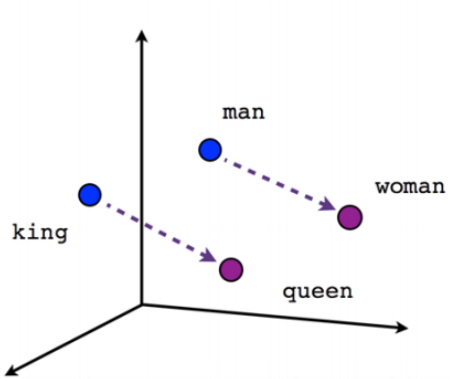


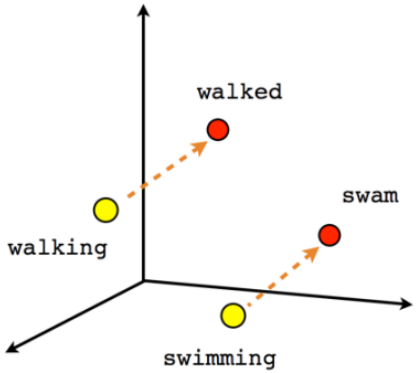
Figure 1: Representation learning happens by doing unsupervised or semi-supervised learning.

Graph embeddings

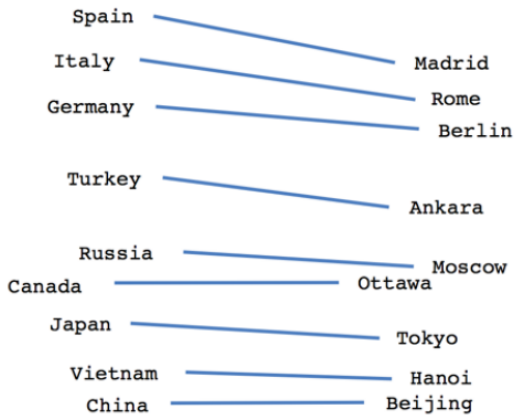
Let us take inspiration from word representation learning. Namely we take a look at Word2Vec.[1]



Male-Female



Verb tense

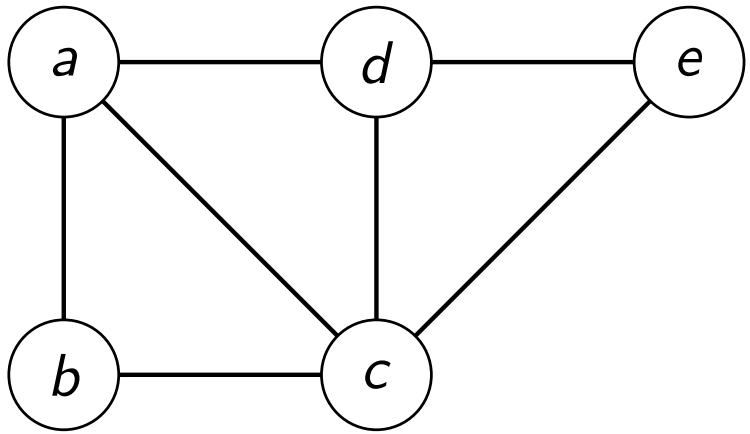


Country-Capital

[1] (Mikolov et al., 2013)

The DeepWalk idea

Nodes that are close in a random walk should have similar representations in the embedding space (Perozzi et al., 2014).



$$a - b - c - d - c - d - e - c - d - c - d$$

$$e - d - e - d - c - d - e$$

$$b - a - c - d - a - b - a - c - b - c - d$$

Figure 2: Example graph with linear vertex sequences.

Setting up an optimization problem

The representation vector specific to node v is $z(v)$. The optimization problem of interest is given by:

$$\min_{z} \quad \sum_{v \in V} -\log P(N_S(v)|z(v)). \tag{1}$$

After a number of transformations and an inner product parametrization we get the following:

$$\min_{z} \quad \sum_{v \in V} \left[\ln \left(\sum_{u \in V} \exp(z(v) \cdot z(u)) \right) - \sum_{n_i \in N_S(v)} z(n_i) \cdot z(v) \right]. \tag{2}$$

On complexity and application

Create an embedding takes:

$$\mathcal{O}(|V| \times \text{Dimension} \times \text{Sequence length} \times \text{Window size} \times \text{Samples per node})$$

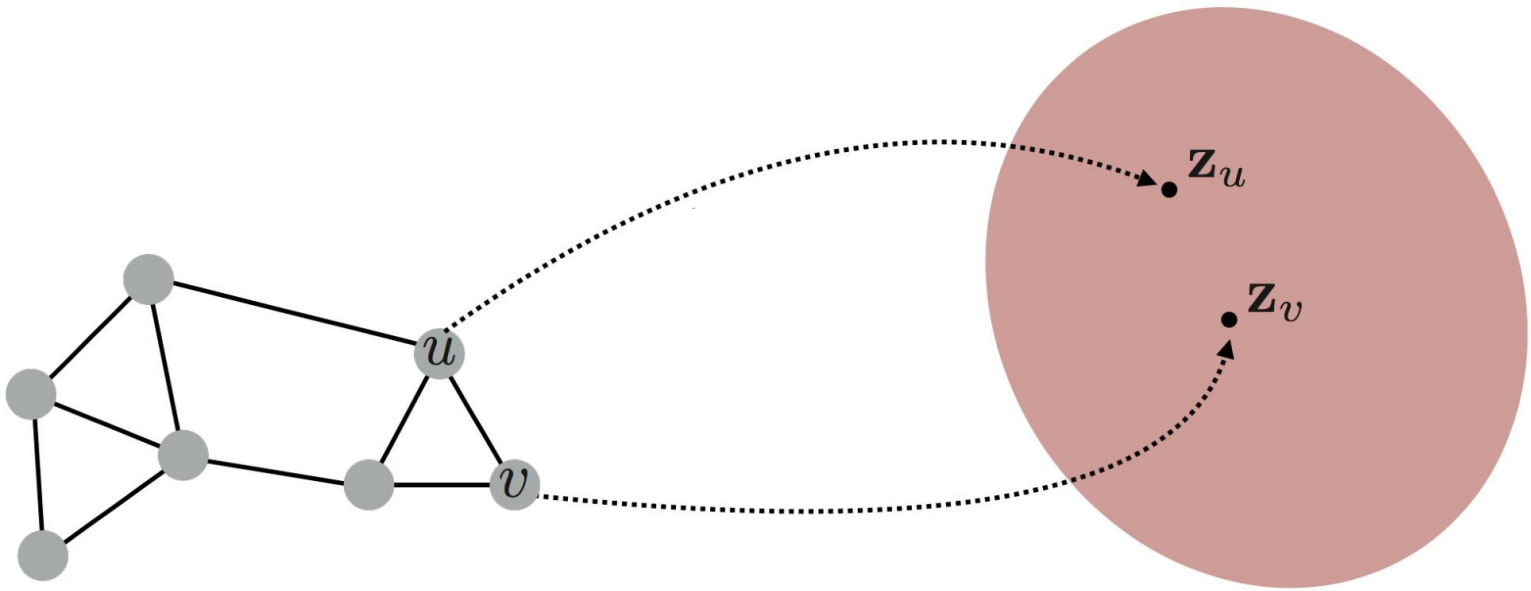


Figure 3: We first create an embedding – this is unsupervised. The features can be used for downstream learning.

Case Study I.: FabSwingers

Performance under sparsity

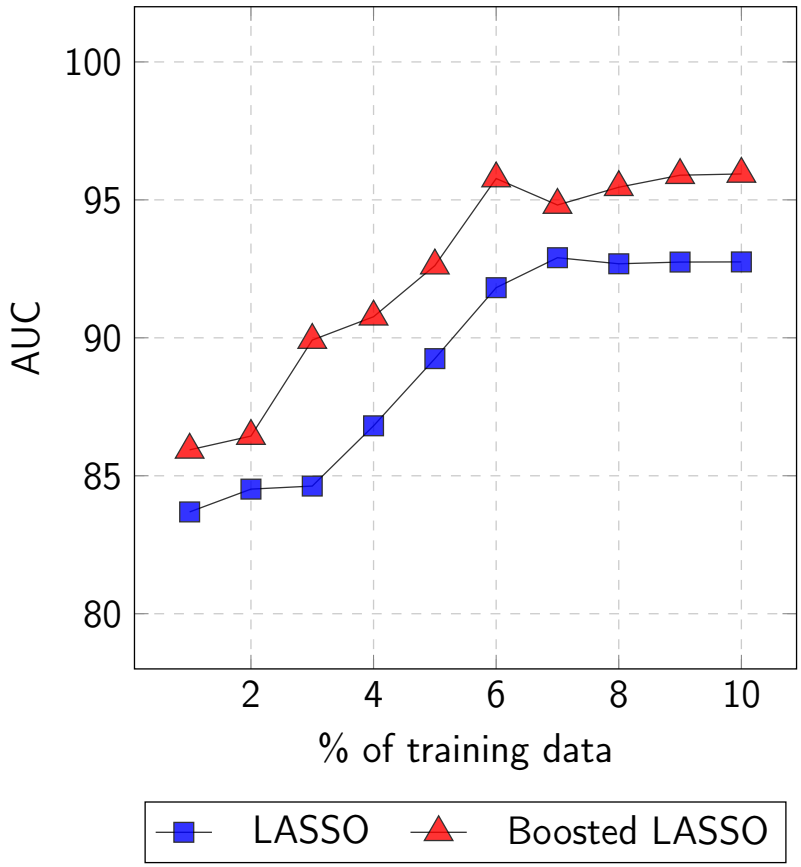


Figure 4: Classification performance on the safe sex preference task measured by area under the ROC curve. Each point was calculated from 10 random train-test splits.

Scalability

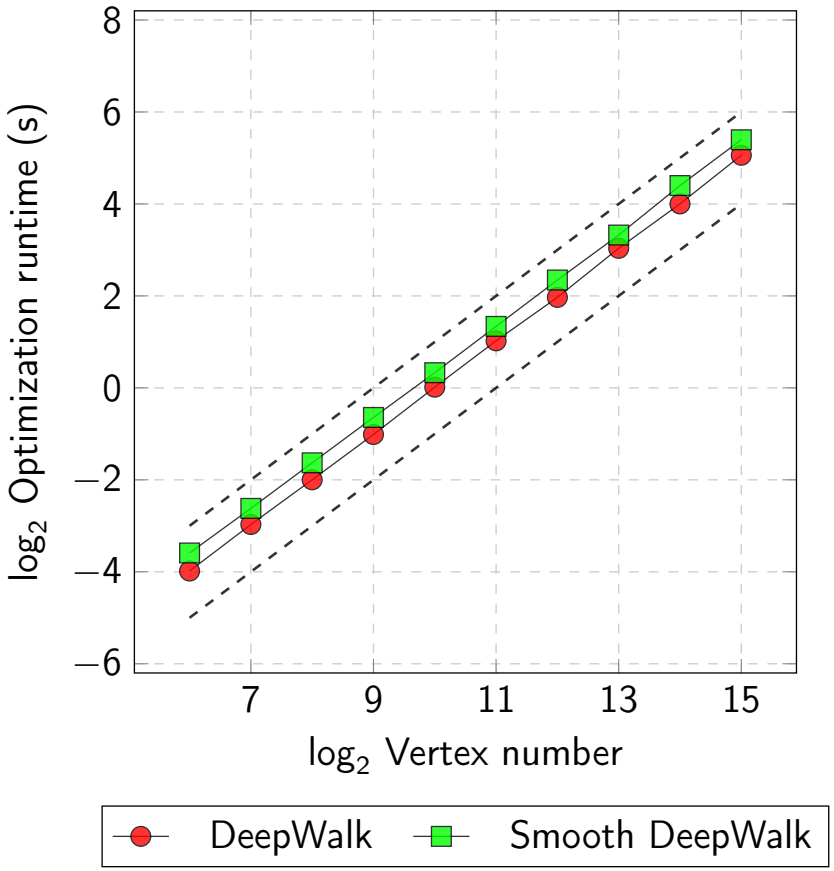


Figure 5: Sensitivity of optimization and random walk sampling runtime to graph size measured by seconds. Training on 2^{15} nodes takes 40 seconds.

Noise tolerance

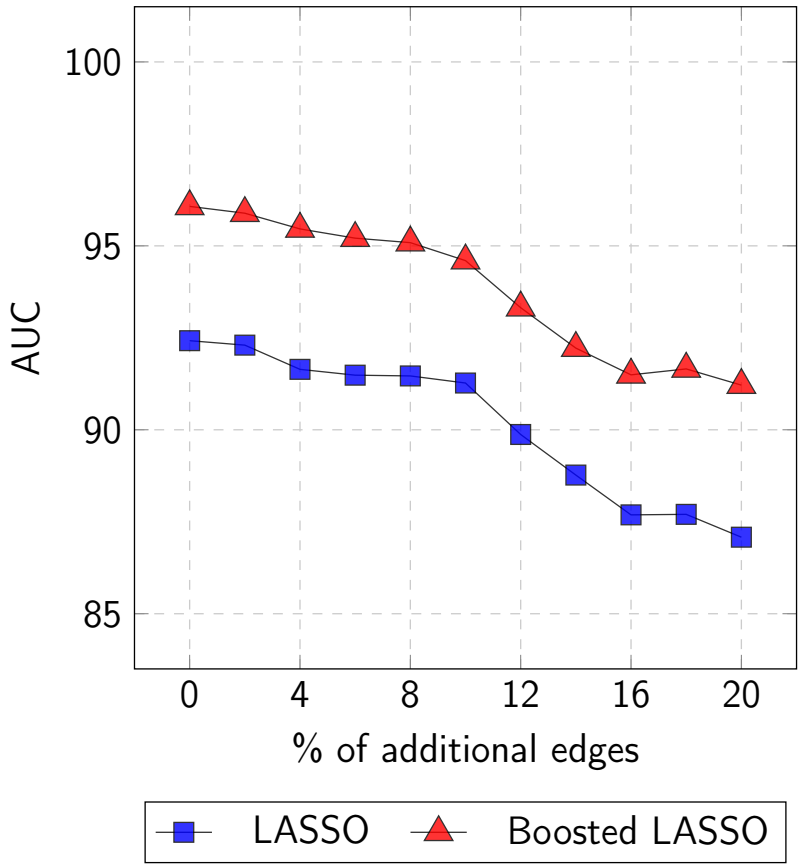


Figure 6: Sensitivity of the classification performance to the presence of noisy relationships on the safe sex task. Each point was calculated from 10 graphs with random edges.

Nice properties and limitations

Nice properties:

1. Totally unsupervised feature extraction.
2. Easy to distribute (sampling and optimization).
3. You can use more advanced tools from neural language understanding (e.g. recurrent neural networks).

Not so nice properties:

1. Not transferable learning.
2. Handling dynamic networks is cumbersome.
3. Uses a lot of memory.

Graph convolutional neural models[2]

Each node in $\mathcal{G}$ filters the signal $\mathbf{X}$ with the weights $\mathbf{W}_1$ and messages the neighbors:

$$\mathbf{Z}_1 = f_1(\mathcal{G}, \mathbf{X}, \mathbf{W}_1)$$

This abstract signal is filtered again:

$$\mathbf{Z}_2 = f_2(\mathcal{G}, \mathbf{Z}_1, \mathbf{W}_2)$$

Finally, we aggregate once again to predict the label of nodes:

$$\widehat{\mathbf{Y}} = \text{Aggregate}(\mathbf{Z}_2, \mathbf{W}_3)$$

We want to minimize the cost of mislabeling nodes:

$$\text{minimize} \quad \mathcal{L}(\mathbf{Y}, \widehat{\mathbf{Y}})$$

[2](Kipf & Welling, 2016)

Scaling it up.

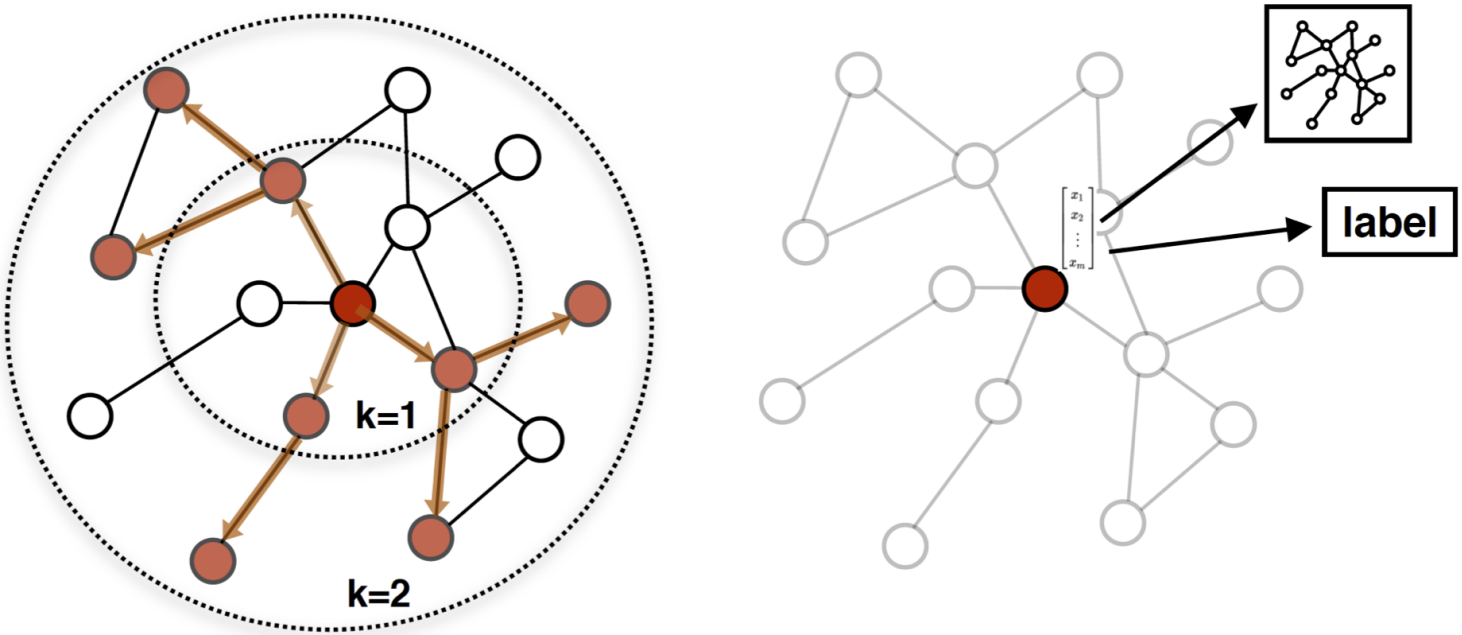
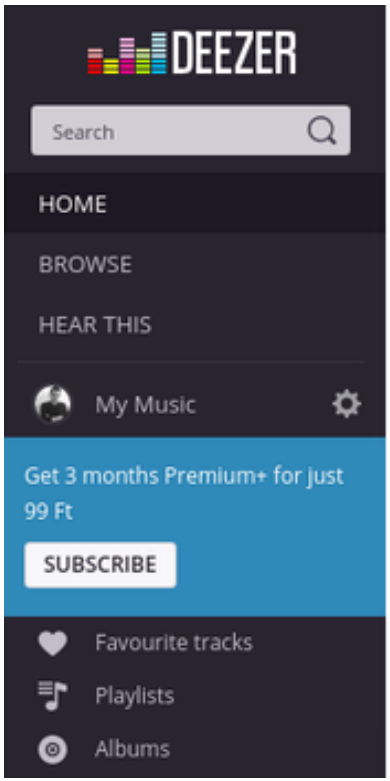
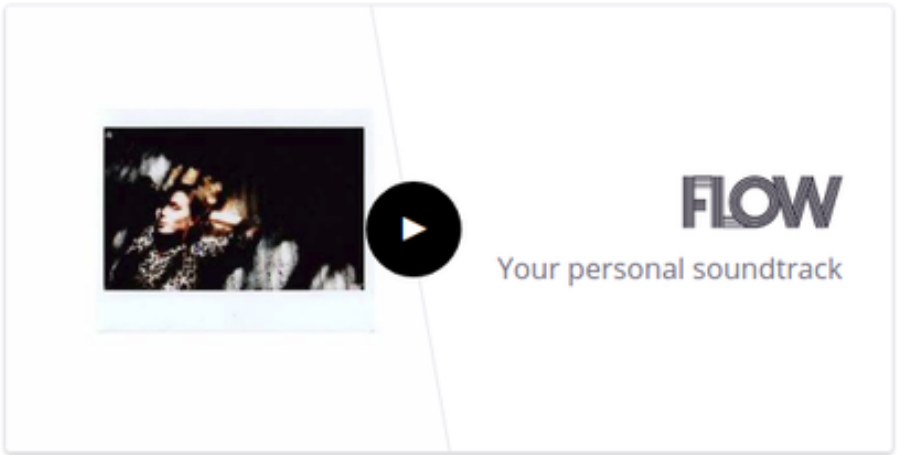


Figure 7: First, let us sample fixed-size neighborhoods from each node. Second, build a model which classifies the neighborhoods. For details see Hamilton et al. (2017).

Case Study II.: Deezer leak





Browse ›
Explore by genre and mood

Predictive performance

	Deezer Social Networks		
	Croatia	**Hungary**	**Romania**
Factorization	0.128 (±0.007)	0.072 (±0.007)	0.051 (±0.004)
DeepWalk	0.173 (±0.006)	0.120 (±0.006)	0.087 (±0.008)
DeepWalk + Factorization	0.215 (±0.006)	0.153 (±0.004)	0.114 (±0.003)
Planetoid	0.230 (±0.006)	0.169 (±0.004)	0.132 (±0.006)
Cold Start GCN	0.288 (±0.005)	0.213 (±0.004)	0.186 (±0.005)
GCN	0.328 (±0.006)	0.244 (±0.004)	0.213 (±0.006)
Resampled GCN	**0.333** (±0.002)	**0.250** (±0.002)	**0.215** (±0.002)

Table 1: Predictive performance of graph convolutional models on the Deezer song datasets measured by Precision@100. Each experiment was repeated 10 times – standard deviations in the parentheses.

Transfer learning

		Target country		
		Croatia	**Hungary**	**Romania**
Source	**Croatia**	–	0.231 (±0.004)	0.198 (±0.007)
	Hungary	0.317 (±0.006)	–	0.196 (±0.006)
	Romania	0.315 (±0.005)	0.232 (±0.004)	–
	GCN	0.328 (±0.006)	0.244 (±0.004)	0.213 (±0.006)

Table 2: Transfer learning performance on the Deezer social networks measured by Precision@100. Each experiment was repeated 10 times – standard deviations in the parentheses.

The double cold start problem

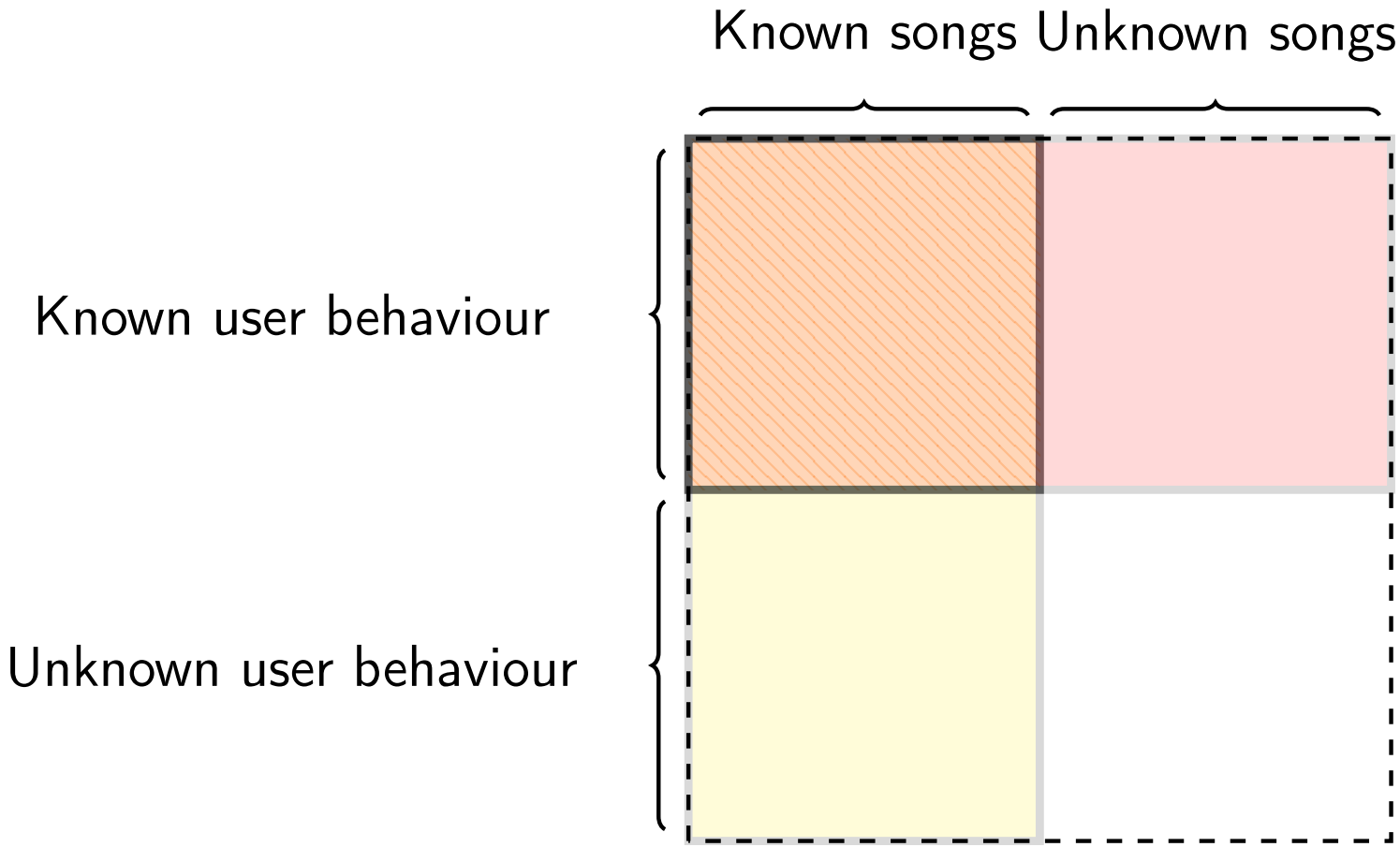


Figure 8: How can we recommend songs that nobody listened to? How can we recommend music to people who never revealed their preferences?

Mapping to song space

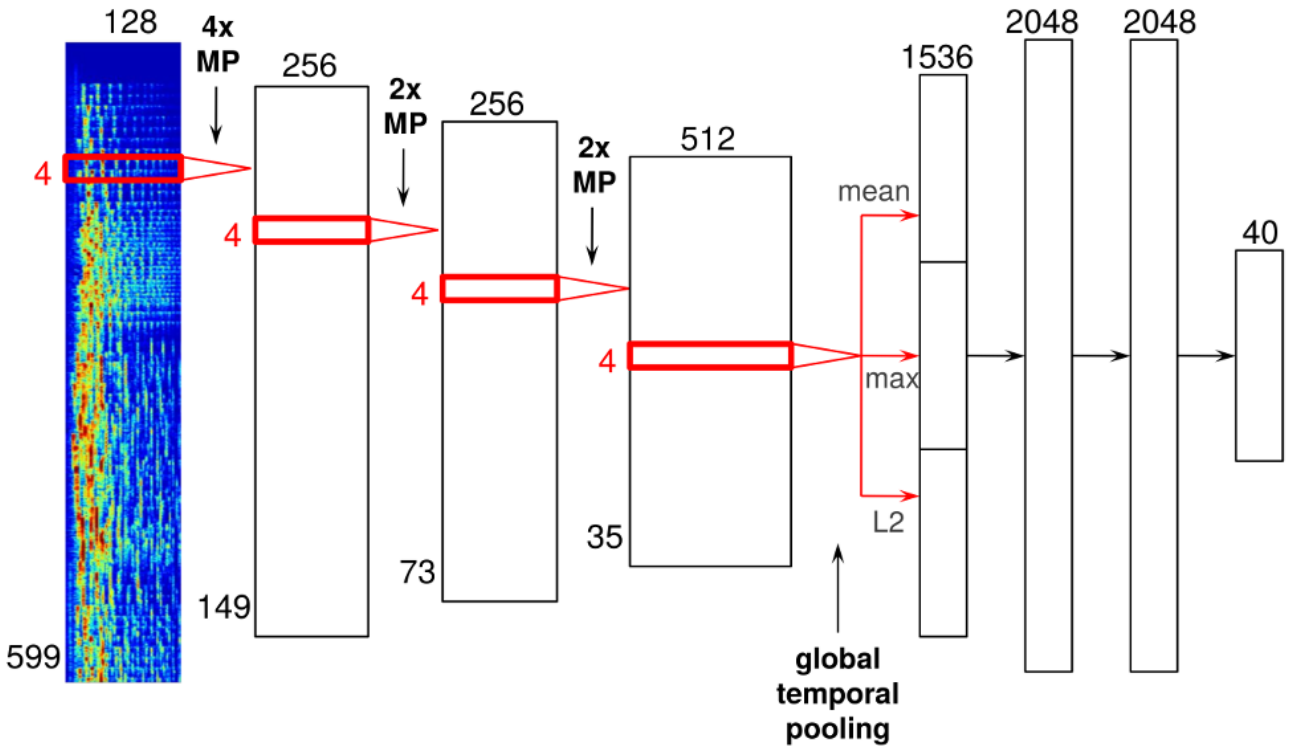


Figure 9: We can learn a mapping from spectral song representations to factors (van den Oord et al., 2013). We assume that we have the first 30 seconds of every song in advance.

Performance under double cold start

	Deezer Social Networks		
	Croatia	**Hungary**	**Romania**
Factorization	0.128 (±0.007)	0.072 (±0.007)	0.051 (±0.004)
Double cold model	0.102 (±0.001)	0.055 (±0.001)	0.027 (±0.001)

Table 3: Predictive performance of the factor mapping model. Each experiment was repeated 10 times – standard deviations in parentheses.

Scalability

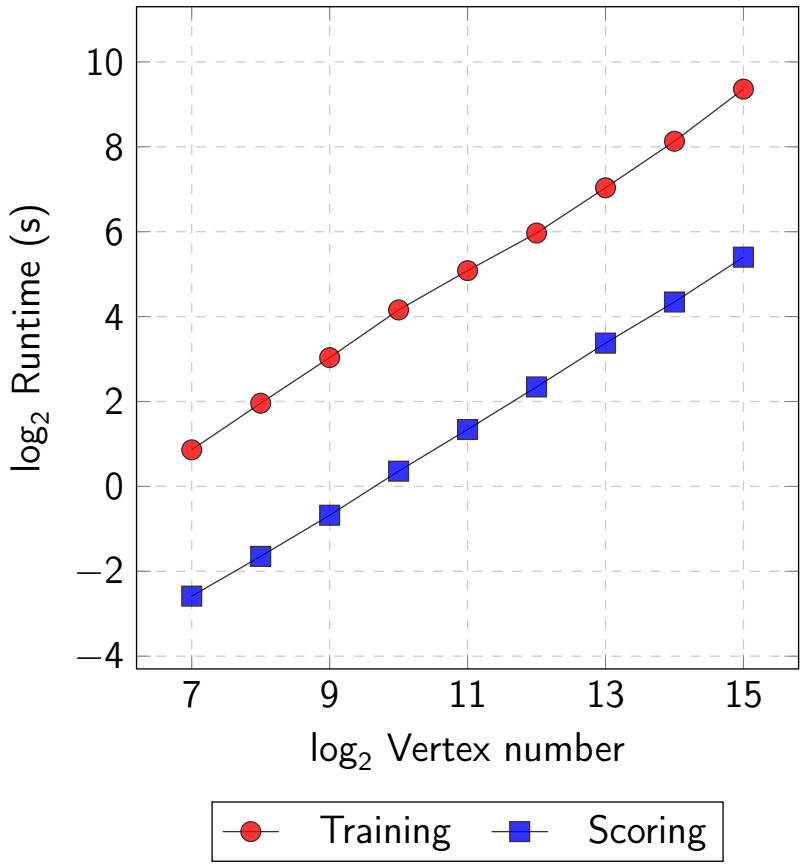


Figure 10: Sensitivity of training and scoring runtime to graph size measured by seconds. Training on 2^{15} nodes takes 10 minutes, and scoring on the same number of nodes takes approximately 1 minute.

Summary

You can learn easily from social context about users and make inference even when labels are sparse.

Specifically:

- Inferring things about You that You never made public is quite easy if I know Your friends.
- Learning on graphs is linear in the number of nodes.
- It even works when you have metadata on a handful of people.
- Features extracted are transferable.
- You can solve cold-start scenarios.

References I

W. L. Hamilton, et al. (2017). 'Inductive Representation Learning on Large Graphs'. In *NIPS*.

T. N. Kipf & M. Welling (2016). 'Semi-Supervised Classification with Graph Convolutional Networks'. *arXiv preprint arXiv:1609.02907* .

T. Mikolov, et al. (2013). 'Distributed Representations of Words and Phrases and their Compositionality'. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. Curran Associates, Inc.

B. Perozzi, et al. (2014). 'DeepWalk: Online Learning of Social Representations'. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pp. 701–710, New York, NY, USA. ACM.

References II

A. van den Oord, et al. (2013). 'Deep content-based music recommendation'. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 26*, pp. 2643–2651. Curran Associates, Inc.