Skoltech

Skolkovo Institute of Science and Technology

Computational and Data Science and Engineering

Combinatorial and Neural Graph Vector Representations

Student: ***Sergey Ivanov***

Scientific Advisor: ***Evgeny Burnaev***

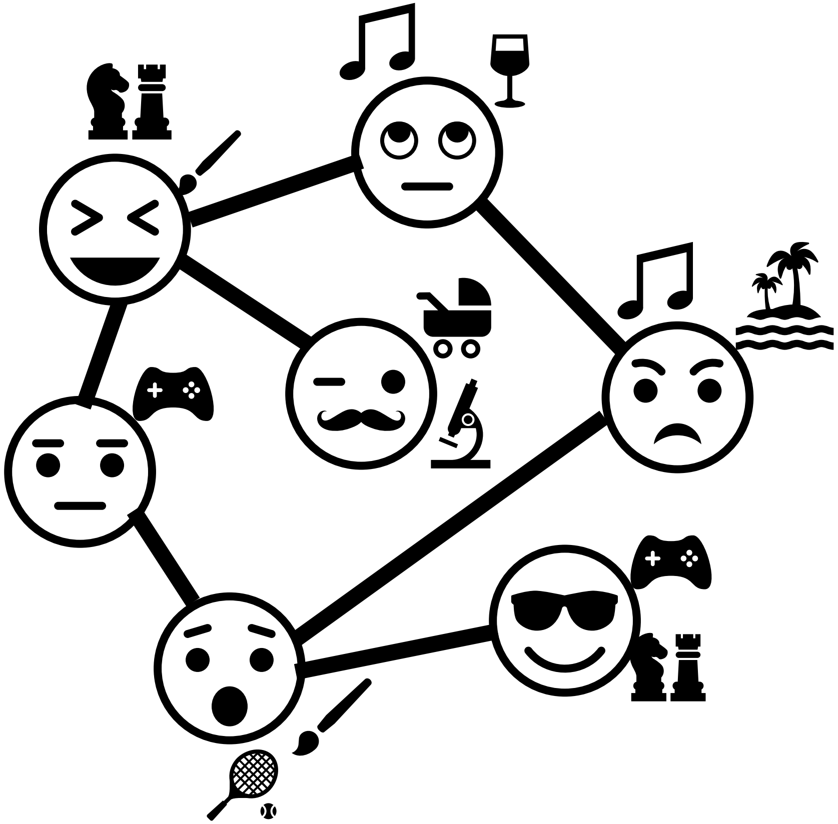
2 December 2019

Product recommendation

Skoltech
Skolkovo Institute of Science and Technology

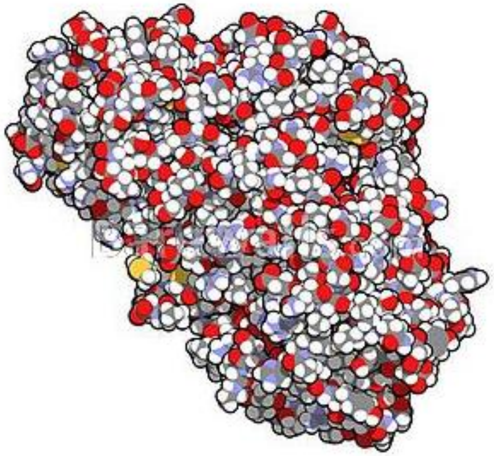
How to find people that maximize product adoption?

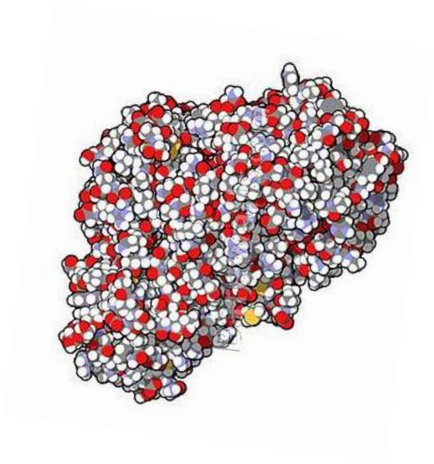
How to scale solutions to billions users and consider user preferences?

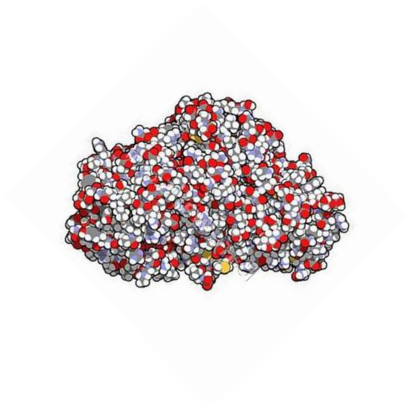


Protein function prediction

How to find similar proteins based on structural, physical, and chemical information?







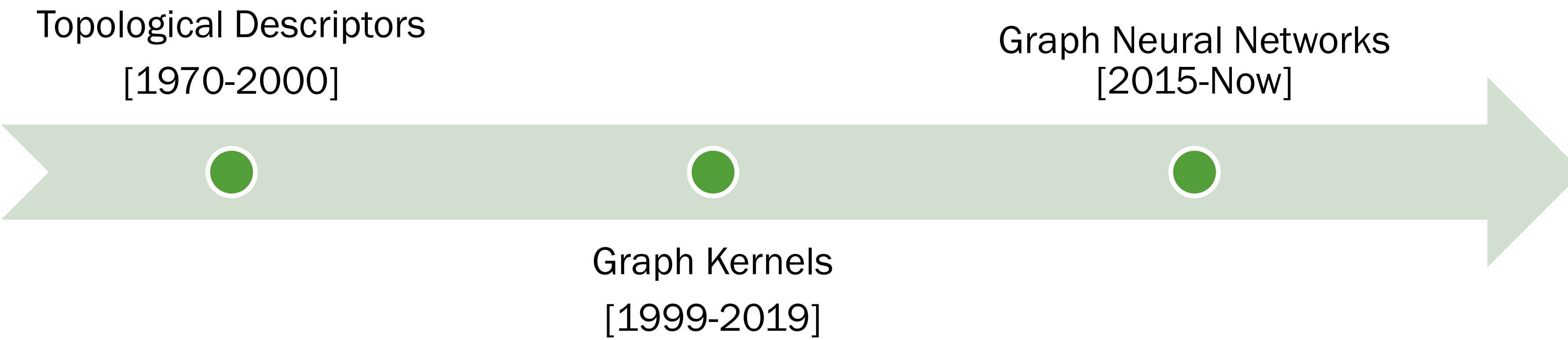
What is common?

- *We can model these problems with graphs;*
- *We can find solution for some instances of graphs.*

We can use ML methods on graphs to solve new instances.

How to represent graphs for ML models?

Representation learning on graphs



Graph vector representation $v \in R^d$ is called *embedding*.

Topological descriptors

Skoltech
Skolkovo Institute of Science and Technology

Simple feature vectors or a scalar number. [1]

Pros:

- Simple and inspired by properties of studied networks

Cons:

- Very limited scope
- Ad-hoc design
- Prediction is not efficient (e.g. via knn)

[1] Handbook of molecular descriptors, Wiley & Sons 2008

Graph kernels

Skoltech
Skolkovo Institute of Science and Technology

Symmetric, positive semidefinite function that maps two graphs to a real number [1]:

$$K(G_1, G_2) \mapsto R$$

Pros:

- More expressive than topological descriptors
- Suitable for kernel machines (e.g. SVM)

Cons:

- Not scalable
- Do not preserve graph isomorphism in feature space

Addressed in this thesis

[1] A survey on graph kernels, Kriege et al. 2019

Isomorphism property

$$K(G_1, G_2) = < \varphi_1, \varphi_2 >,$$

where $\varphi: G \mapsto R^d$

If φ is bijective, then we say graph kernel has *isomorphism property*. Such graph kernel minimizes loss of information.

Graph neural networks

Skoltech
Skolkovo Institute of Science and Technology

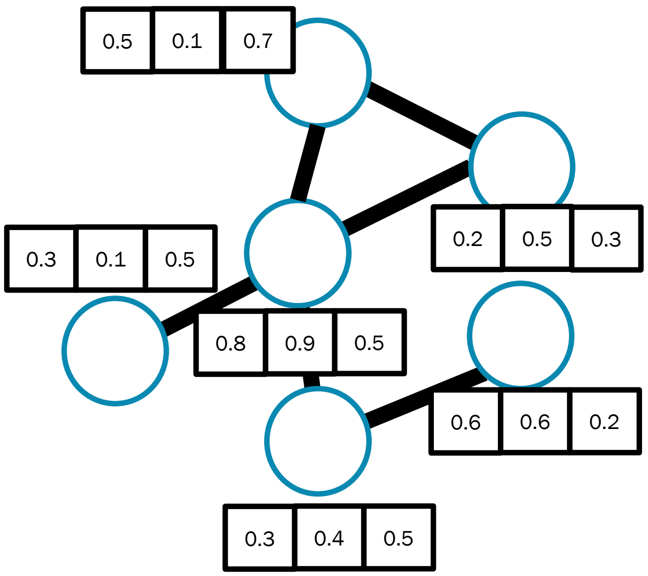
Graph embedding is initialized with random vector, which is updated to fit the given data [1].

Pros:

- Superior empirical performance
- Strong theoretical background

Cons:

- Complex models
- Hardly interpretable



[1] A Comprehensive Survey on Graph Neural Networks, Wu et al. 2019

Main goal of this thesis

Skoltech
Skolkovo Institute of Science and Technology

To develop efficient graph representation that:

- Has isomorphism property
- Inherits strong graph kernel and neural network sides
- is efficient on real-world problems

High-level structure

Skoltech
Skolkovo Institute of Science and Technology

Thesis consists of three major parts:

1. New graph representation framework
2. Graph classification problem
3. Product recommendation on graphs

Skoltech
Skolkovo Institute of Science and Technology

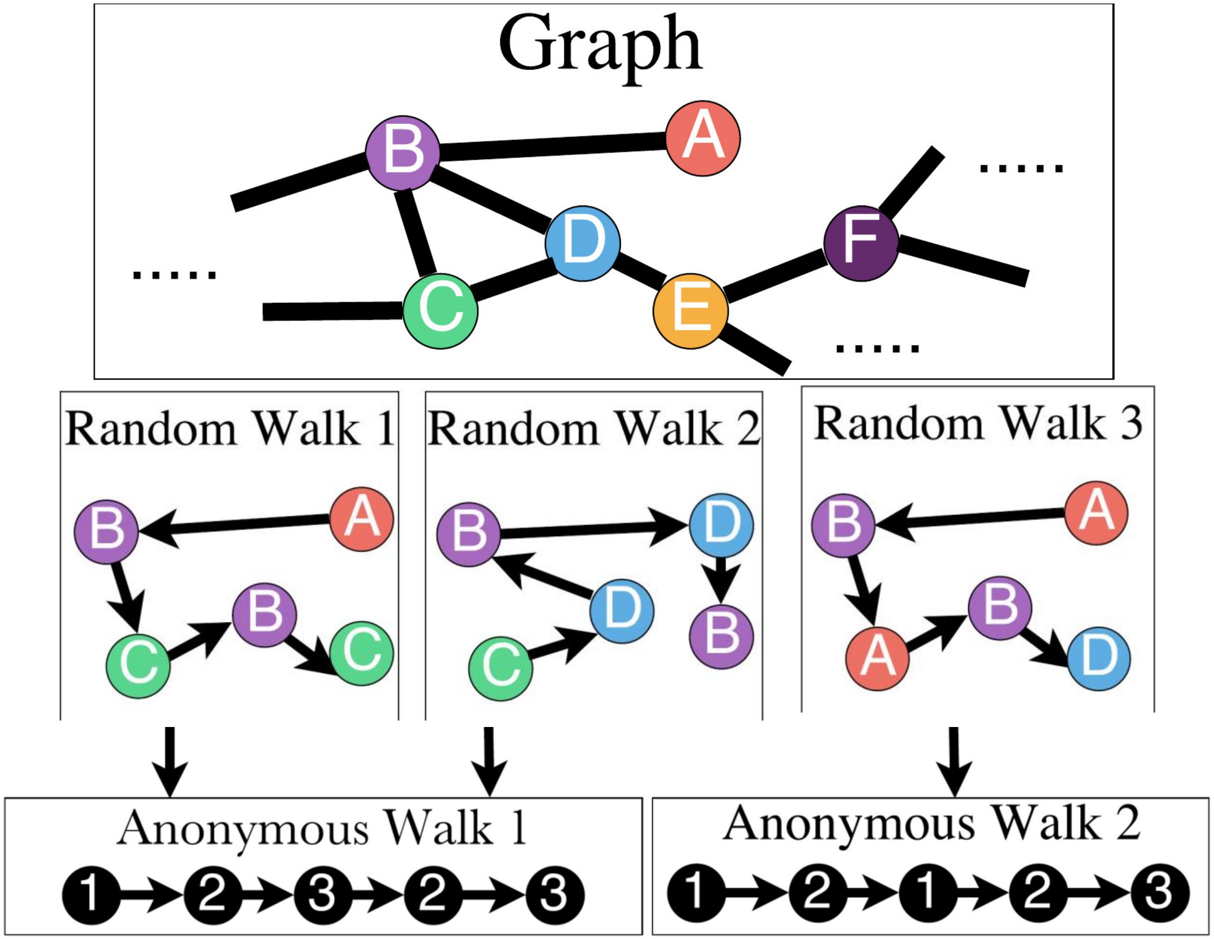
1. Anonymous Walks

Anonymous walks

Definition:

If $w = (v_1, v_2, \ldots, v_l)$ is a random walk then anonymous walk is the sequence $a = (a_1, a_2, \ldots, a_l)$ where a_i is the first position of v_i in w.

Anonymous walks



Reconstruction property

Theorem [Zhu & Micali, 2015]:

Let $B(v,r)$ be the induced graph at node v of radius r containing m edges, and D_l is a set of all possible anonymous walks of length up to $l = 2(m+1)$, that start at node v.

There is an algorithm to reconstruct a graph G that is isomorphic to $B(v,r)$.

Theorem illustration

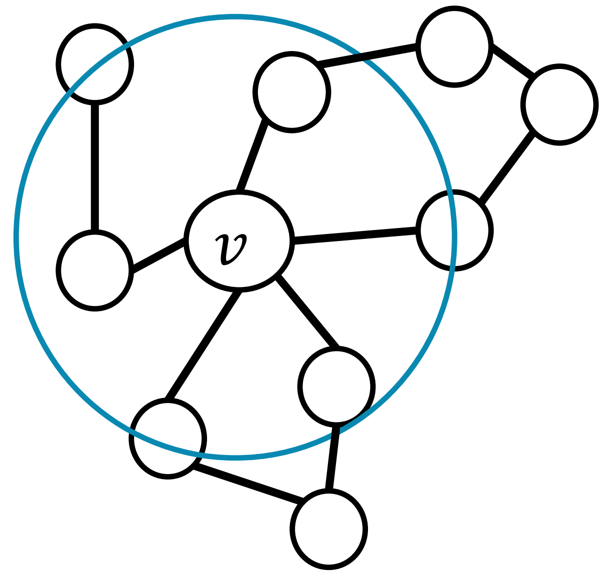
Radius $r = 2$

Edges $\mathrm{m} = 6$

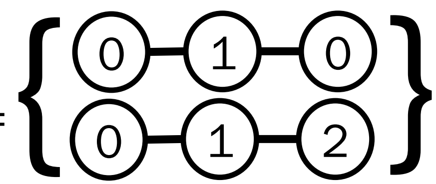
Length $l = 2(m+1) = 17$

Knowing distributions $D_2, D_3, \dots, D_{17}$ we can obtain graph G that is isomorphic to $B(v,r)$.

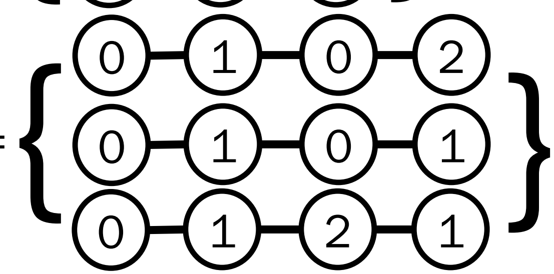
$B(v,r)$



$D_2 =$



$D_3 =$

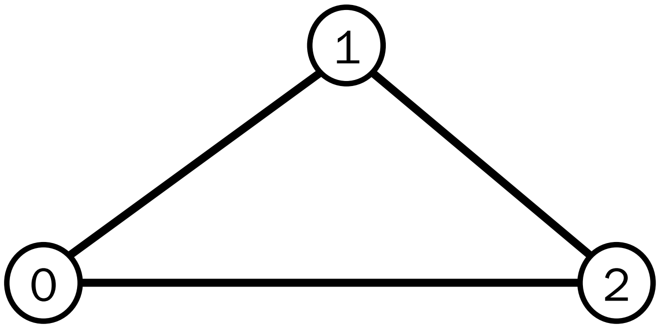


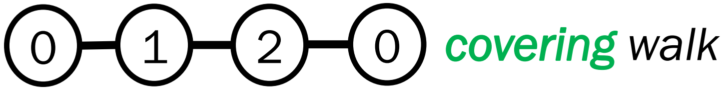
...

Covering walks

Definition:

If anonymous walk traverses each edge of the graph at least once, we call it *covering walk*.





Isomorphism test

Theorem [This thesis]:

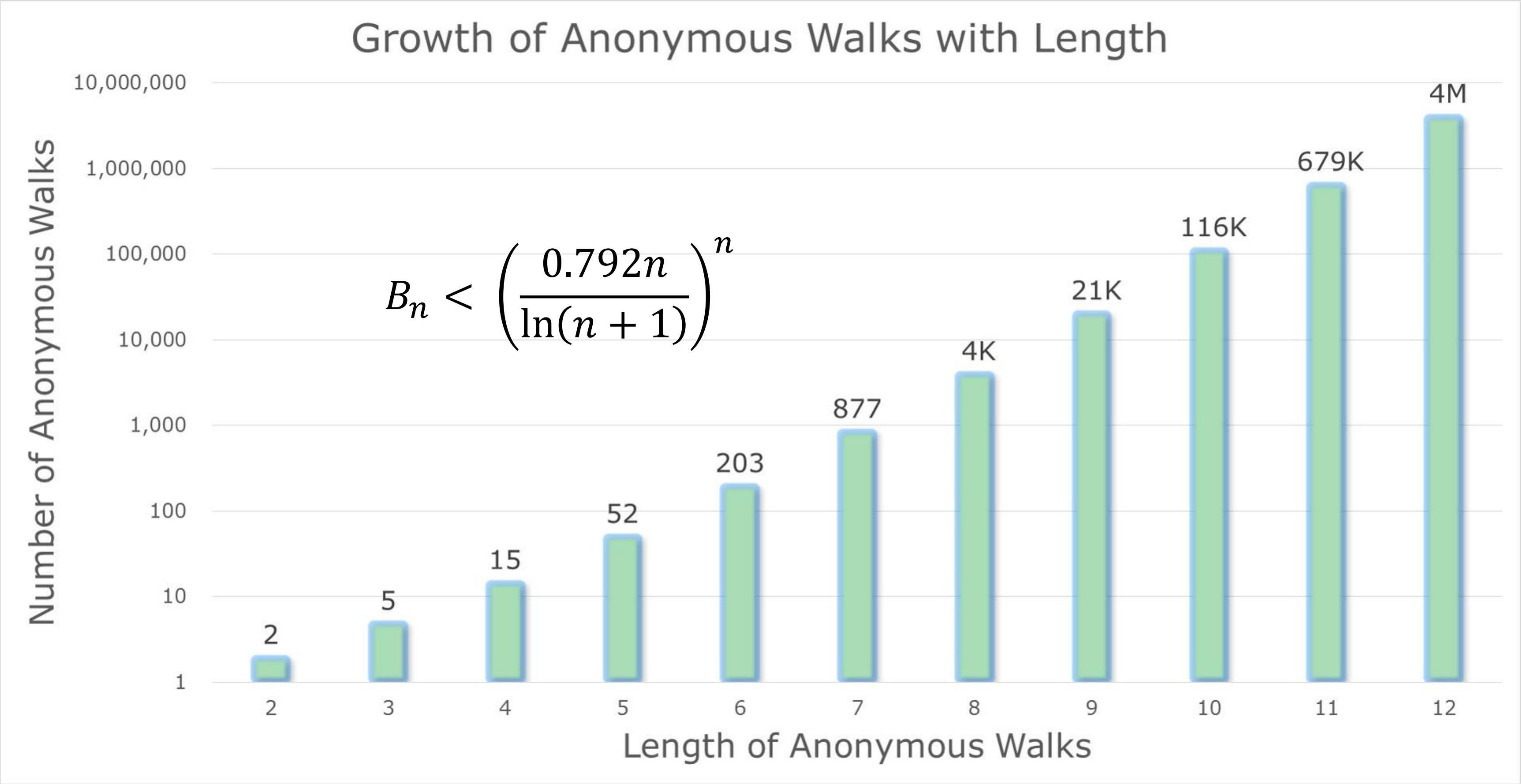
Let $D_l(G_1)$ and$D_l(G_2)$ be the sets of all covering walks of length $l = 2(m + 1)$ for graphs G_1 and G_2 with m edges. Two graphs are isomorphic if and only if $D_l(G_1) \cap D_l(G_2) = \emptyset$.

Running time complexity

Theorem [This thesis]:

The number of possible anonymous walks $|D_l|$ of length l in a graph that start at node v is at most the Bell number B_{l-1}.

$$|D_l| \leq B_{l-1}$$



$$B_n < \left(\frac{0.792n}{\ln(n+1)}\right)^n$$

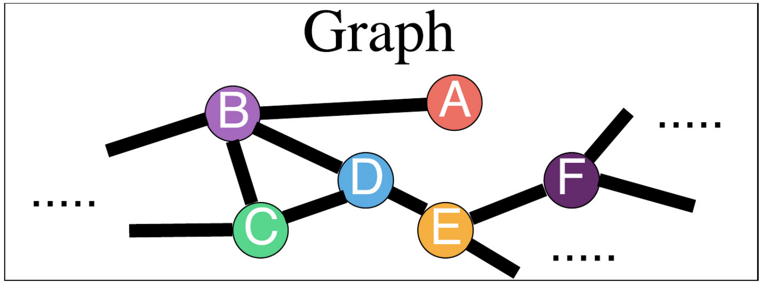
Combinatorial graph embeddings

Let $(a_1, a_2, \ldots, a_\eta)$ be all possible anonymous walks of length l.

<u>Combinatorial Graph Embedding</u>

$$AWE(G) = (p(a_1), p(a_2), \ldots, p(a_\eta))$$

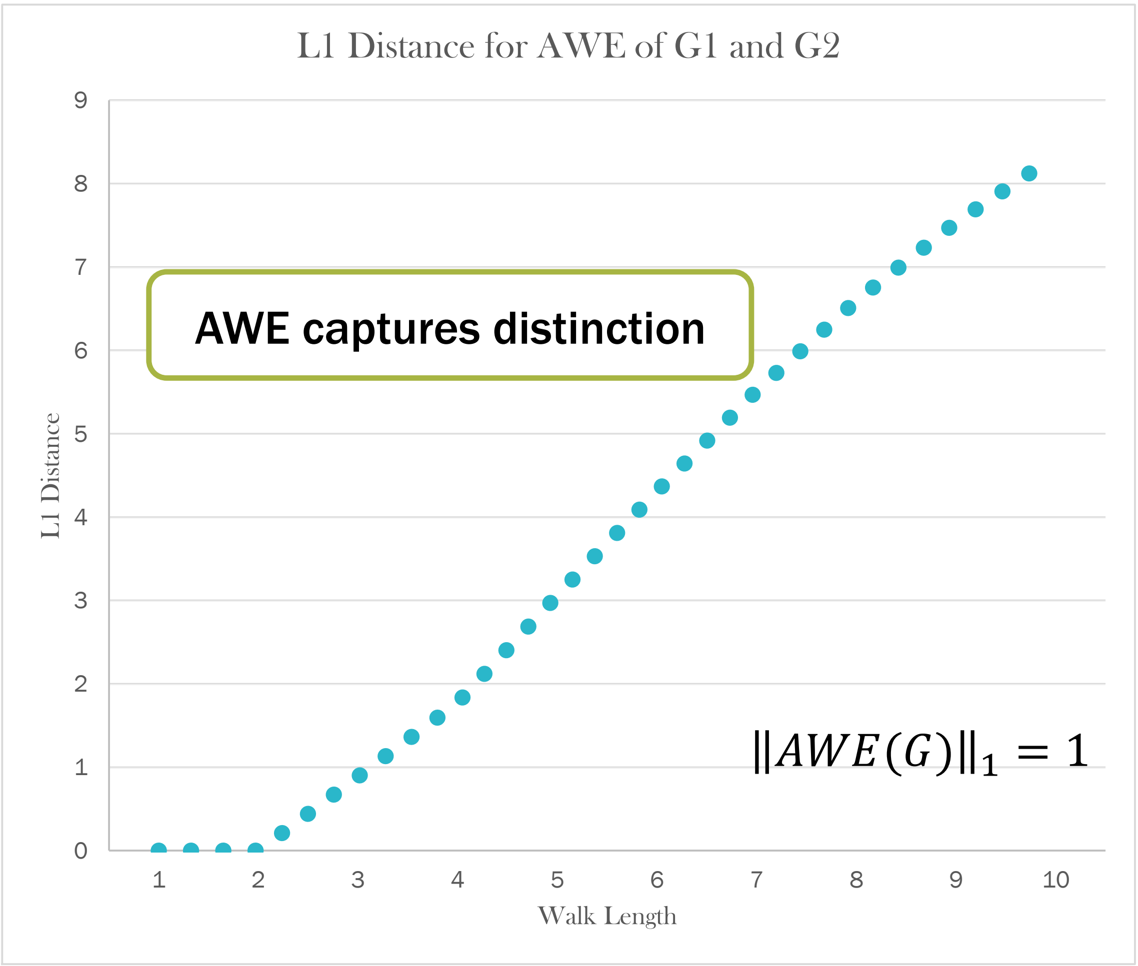
where $p(a_i)$ is frequency of AW a_i, across all nodes in a graph.

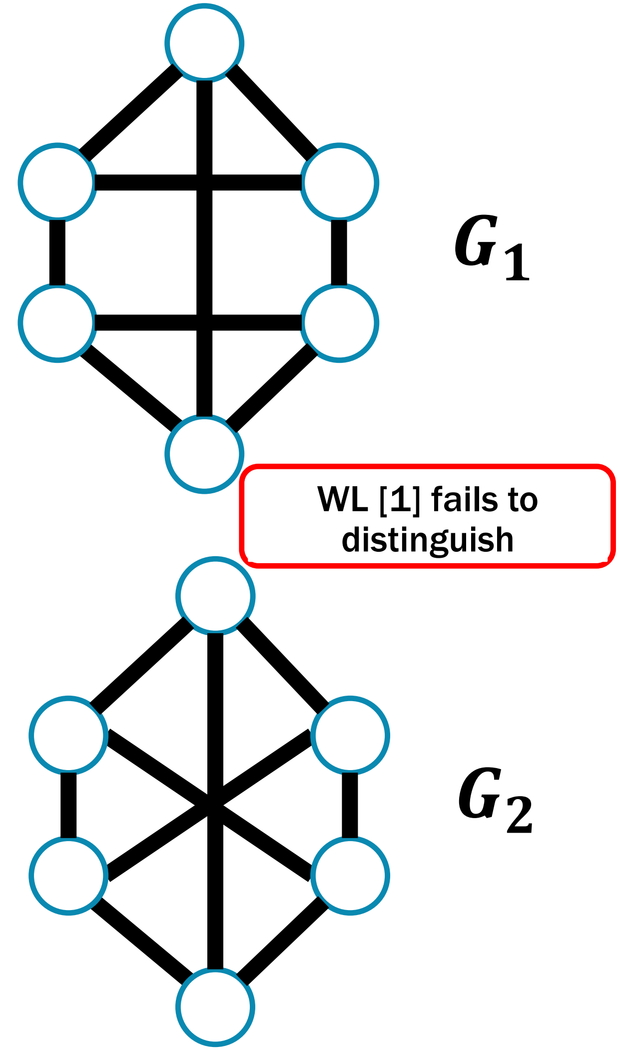


Graph ⇄ **Embedding**

$l = 2(m + 1)$

Example of isomorphism property





[1] Weisfeiler-Lehman Graph Kernels, Shervashidze et al. 2011

Resolving computation complexity

Skoltech
Skolkovo Institute of Science and Technology

Finding all possible anonymous walks $(a_1, a_2, \dots, a_\eta)$ of length l can be expensive.

Instead, we can sample μ anonymous walks and compute embeddings from them.

Can we guarantee the quality of sampled embeddings?

Skoltech

Skolkovo Institute of Science and Technology

Approximation of sampling method

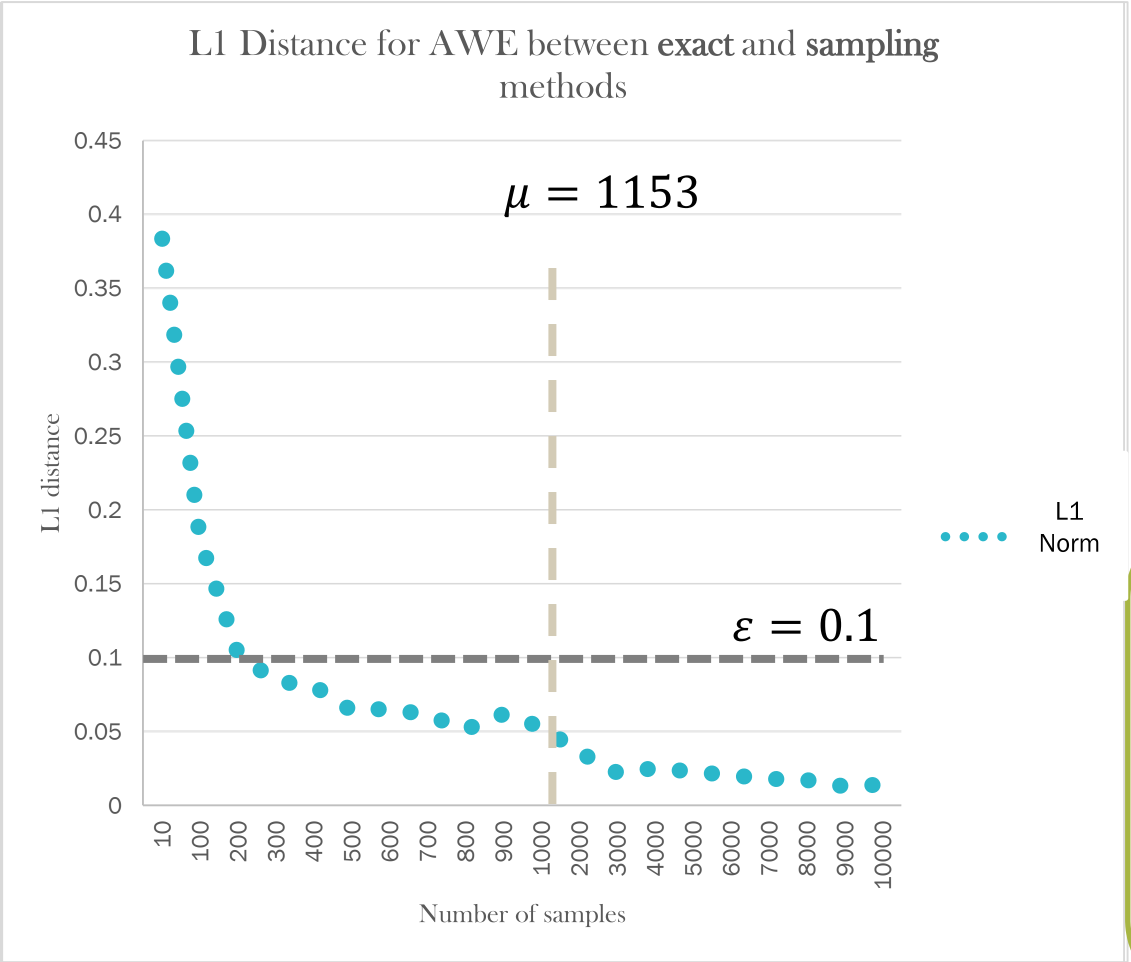
Theorem [This thesis]:

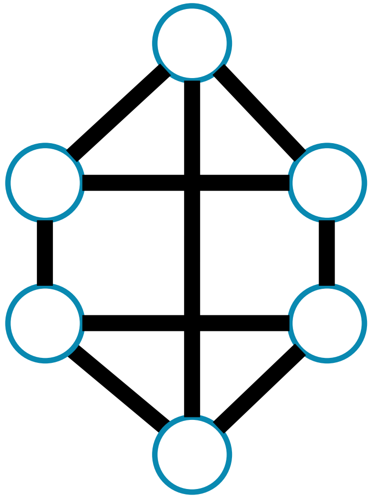
- D_l is *true* distribution of anonymous walks of length l in a graph G
- $\widehat{D}_l$ is *sampled* distribution of μ anonymous walks of length l from graph G.

Let $|D_l| = \eta$. For all $\varepsilon > 0$ and $\delta \in [0,1]$, the number of samples μ to satisfy $P\left(\left\|D_l - \widehat{D}_l\right\|_1 \geq \varepsilon\right) \leq \delta$ equals to:

$$\mu = \left\lceil \frac{2}{\varepsilon^2}\left(\log(2^\eta - 2) - \log(\delta)\right)\right\rceil$$

Example of sampling bound





G_1

$$\varepsilon = 0.1$$

$$\delta = 0.1$$

$$l = 3$$

$$P\left(\left\|D_l - \widehat{D}_l\right\|_1 \geq 0.1\right) \leq 0.1$$

$$\mu = \left\lceil \frac{2}{\varepsilon^2}\left(\log(2^\eta - 2) - \log(\delta)\right) \right\rceil = 1153$$

Neural network embeddings

- Initialize randomly graph embedding d and a matrix of embeddings W for each anonymous walk.
- Sample a corpus of anonymous walks that start from the same node.
- Maximize the average log probability of observing the corpus

Neural network embeddings

Skoltech
Skolkovo Institute of Science and Technology

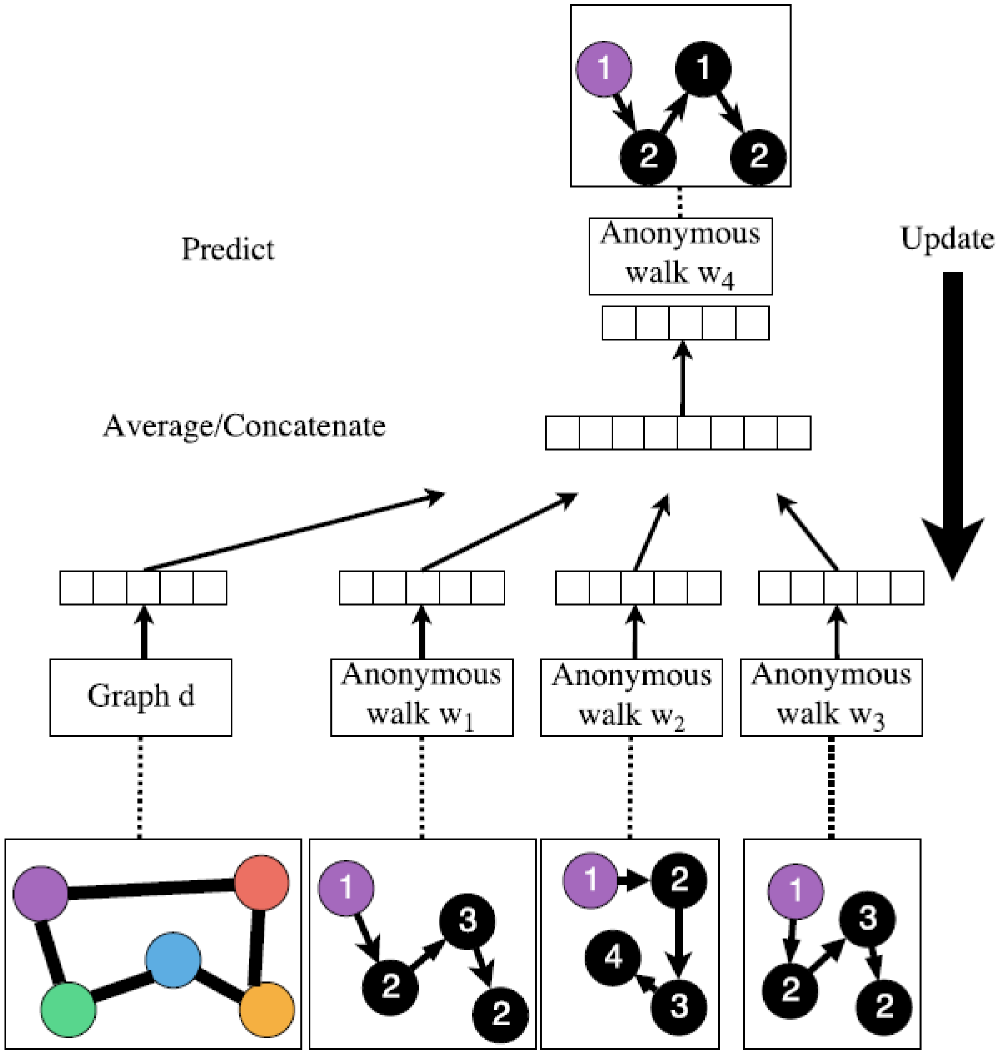
We optimize the objective

$$\frac{1}{T}\sum_{t=\Delta}^{T-\Delta} \log p(w_t|w_{t-\Delta}, \dots w_{t+\Delta}, d) \mapsto_{W,d} max$$

where

$$p(w_t|w_{t-\Delta}, \dots w_{t+\Delta}, d) = \frac{e^{y(w_t)}}{\sum_{i=1}^{\eta} e^{y(w_i)}}$$

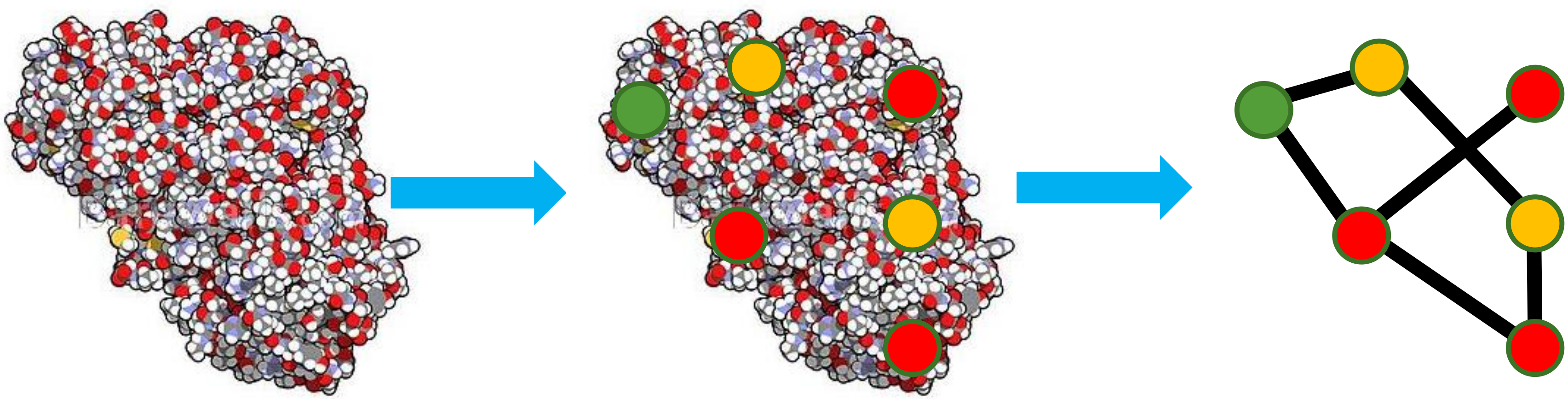
is the softmax probability of seeing anonymous walk in a graph and $y(w_i) = \langle w_i, [\frac{1}{2\Delta}\sum_{j=t+\Delta}^{t+\Delta} w_j \,; d] \rangle$ is similarity between walk w_i and its neighborhood.



Skoltech
Skolkovo Institute of Science and Technology

2. Graph Classification

Protein function prediction



Nodes: SSEs (helices, sheets, turns)

Edges: sequential or structural neighbors

Labels: length between atoms, polarity of SSE, etc. [1]

[1] Protein function prediction via graph kernels, Borgwardt et al. 2005

Graph classification problem

Given

- $T = \{(G_i, y_i)\}_1^N$ - train graph data set
- $Q = \{(G_i, y_i)\}_1^M$ - test graph data set

Using the train set T, find a function $f \in F = \{\phi: G \mapsto Y\}$ such that

$$acc = \frac{1}{M}\sum_1^M [f(G_i) = y_i] \mapsto \max_f acc$$

Classification pipeline

Skoltech
Skolkovo Institute of Science and Technology

Prepare k-fold Cross-Validation splits.

⇓

Train SVM model and choose the best hyperparameters for embeddings and classification models.

⇓

Evaluate the best model on test instances.

<u>Embedding parameters</u>

- Length of a walk
- Window size
- Embedding size

<u>Model parameters</u>

- Penalty term C
- Kernel type (e.g. Gaussian, Polynomial)
- Batch size

Datasets

	Dataset	Source	Graphs	Classes (Max)	Nodes Avg.	Edges Avg.
	COLLAB	Social	5000	3 (2600)	74.49	4914.99
	IMDB-B	Social	1000	2 (500)	19.77	193.06
[1]	IMDB-M	Social	1500	3 (500)	13	131.87
	RE-B	Social	2000	2 (1000)	429.61	995.50
	RE-M5K	Social	4999	5 (1000)	508.5	1189.74
	RE-M12K	Social	12000	11 (2592)	391.4	913.78
[2]	Enzymes	Bio	600	6 (100)	32.6	124.3
[3]	DD	Bio	1178	2 (691)	284.31	715.65
	Mutag	Bio	188	2 (125)	17.93	19.79

[1] Deep Graph Kernels, Yanardag et al. 2012.
[2] Protein function prediction via graph kernels, Borgwardt, 2005.
[3] Distinguishing enzyme structures from non-enzymes without alignments, Dobson & Doig, 2003

Evaluation accuracy

	Algorithm	IMDB-M	IMDB-B	COLLAB
Neural	DGK	44.55 ± 0.52	66.96 ± 0.56	73.09 ± 0.25
Kernel	WL	49.33 ± 4.75	$\mathbf{73.4 \pm 4.63}$	$\mathbf{79.02 \pm 1.77}$
	GK	43.89 ± 0.38	65.87 ± 0.98	72.84 ± 0.28
	ER	OOM	64.00 ± 4.93	OOM
	kR	34.47 ± 2.42	45.8 ± 3.45	OOM
Ours	AWE (NN)	$\mathbf{51.54 \pm 3.61}$	$\mathbf{74.45 \pm 5.83}$	$\mathbf{73.93 \pm 1.94}$
	AWE (GK)	$\mathbf{51.58 \pm 4.66}$	73.13 ± 3.28	70.99 ± 1.49

[1] Deep Graph Kernels, Yanardag et al. 2012
[2] Weisfeiler-Lehman Graph Kernels, Shervashidze et al. 2011
[3] F]Efficient graphlet kernels for large graph comparison, Shervashidze et al. 2009
[4] Graph Kernels, Vishwanathan et al. 2010

Evaluation accuracy

	Algorithm	RE-B	RE-M5K	RE-M12K
Neural	DGK	78.04 ± 0.39	41.27 ± 0.18	32.22 ± 0.10
Kernel	WL	81.1 ± 1.9	49.44 ± 2.36	38.18 ± 1.3
	GK	65.87 ± 0.98	41.01 ± 0.17	31.82 ± 0.08
	ER	OOM	OOM	OOM
	kR	OOM	OOM	OOM
Ours	AWE (NN)	$\mathbf{87.89 \pm 2.53}$	$\mathbf{50.46 \pm 1.91}$	$\mathbf{39.20 \pm 2.09}$
	AWE (GK)	$\mathbf{82.97 \pm 2.86}$	$\mathbf{54.74 \pm 2.93}$	$\mathbf{41.51 \pm 1.98}$

Evaluation accuracy

	Algorithm	Enzymes	DD	Mutag
Neural	DGK	27.08 ± 0.79	–	82.66 ± 1.45
Kernel	WL	$\mathbf{53.15 \pm 1.14}$	$\mathbf{77.95 \pm 0.70}$	80.72 ± 3.00
	GK	32.70 ± 1.20	$\mathbf{78.45 \pm 0.26}$	81.58 ± 2.11
	ER	14.97 ± 0.28	OOM	71.89 ± 0.66
	kR	30.01 ± 1.01	OOM	80.05 ± 1.64
Ours	AWE (GK)	$\mathbf{35.77 \pm 5.93}$	71.51 ± 4.02	$\mathbf{87.87 \pm 9.76}$

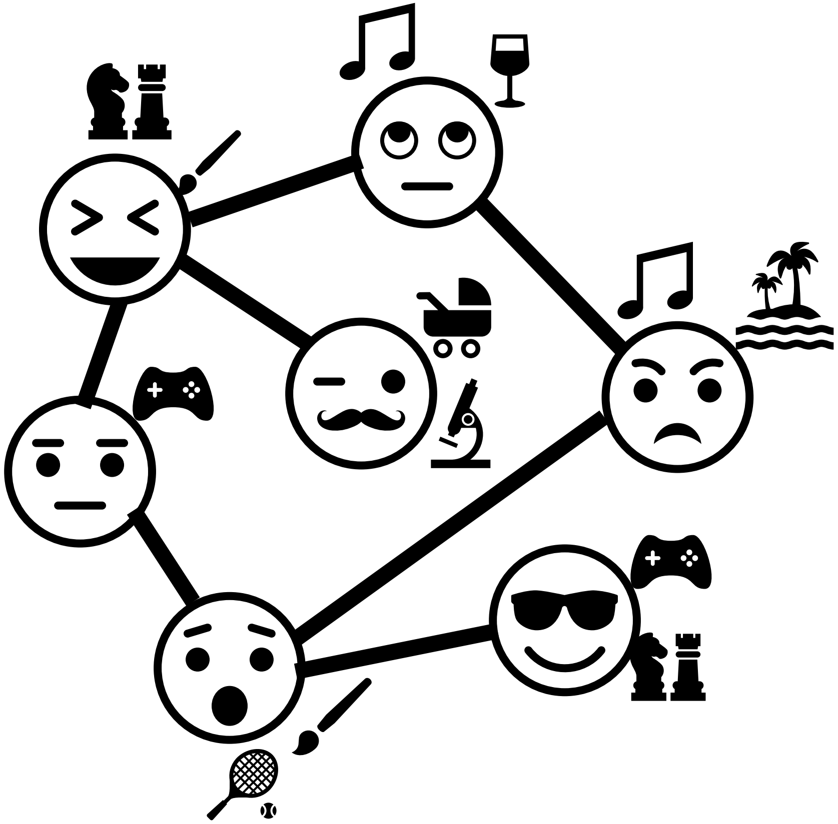
3. Product Recommendation

Skoltech
Skolkovo Institute of Science and Technology

Product recommendation

Online users advertise products through social network.

How to select advertisement products to maximize total adoption?



Propagation model

Skoltech
Skolkovo Institute of Science and Technology

- We aim to find a set of attributes to include in recommendation F.
- Users' interactions are modeled with a directed graph $G = (V, E)$ with set of attributes $\mathrm{F_v}$ for each node $\mathrm{v} \in V$.
- Each edge has a probability of propagation a recommendation: $p_{uv} = b_{uv} + q_{uv}|F_v \cap F|$.
- Propagation of recommendation is a discrete stochastic process according to IC model [1] that goes from a set of active users S to all other users in G.

[1] Maximizing the Spread of Influence through a Social Network, Kempe et al. 2003

Problem formulation

Given:

- Directed graph $G = (V, E)$ with preferences $\mathrm{F_v}$ for each node $\mathrm{v} \in V$ and prior probabilities b_{uv}, q_{uv} for each edge $(\mathrm{u}, \mathrm{v}) \in E$.
- Initial set of active users S and influence function $\sigma(F|S) = E(\#\, activated\ nodes)$

Problem

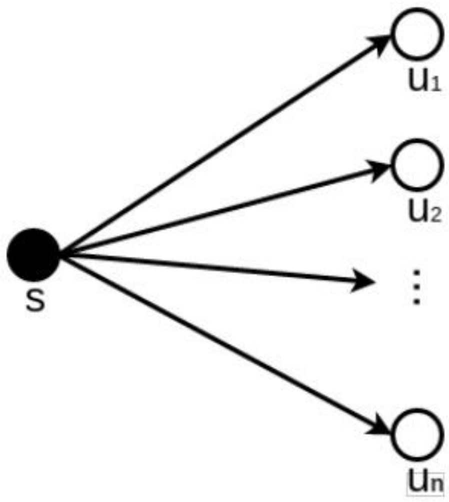
$$\max_F \sigma(F|S) \text{ s.t. } |F| = k$$

Hardness result

Theorem [This thesis]: Product recommendation is NP-hard.

Proof sketch:

- Reduction from Set Cover.
- Each node corresponds to a set element.



Inapproximability result

Theorem [This thesis]: It is NP-hard to approximate optimal solution within a factor of $n^{(1-\varepsilon)}$, $\varepsilon > 0$.

Consequence:

For any polynomial-time algorithm, there are instances of graphs, for which this algorithm performs $\leq \frac{1}{n^{1-\varepsilon}} OPT$.

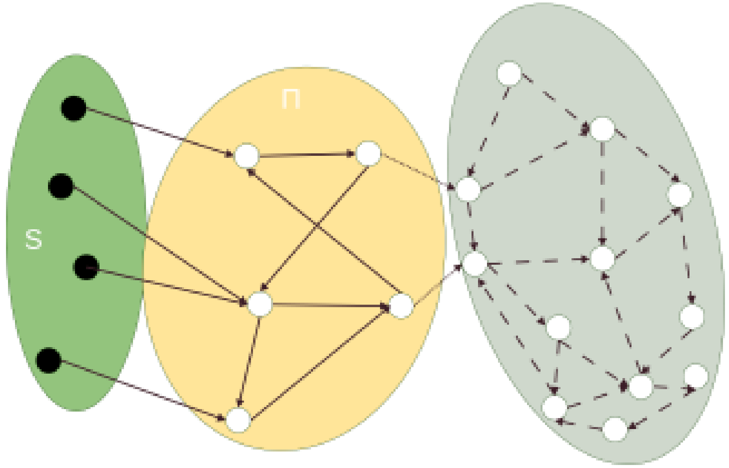
$$\forall \varepsilon > 0\colon \frac{1}{n^{1-\varepsilon}} OPT \leq \frac{1}{n^{1-\varepsilon}} n = n^{\varepsilon}$$

Skoltech

Skolkovo Institute of Science and Technology

Explore-Update algorithm

We propose a new algorithm with a new data structure that is *more efficient* than a greedy algorithm [1].

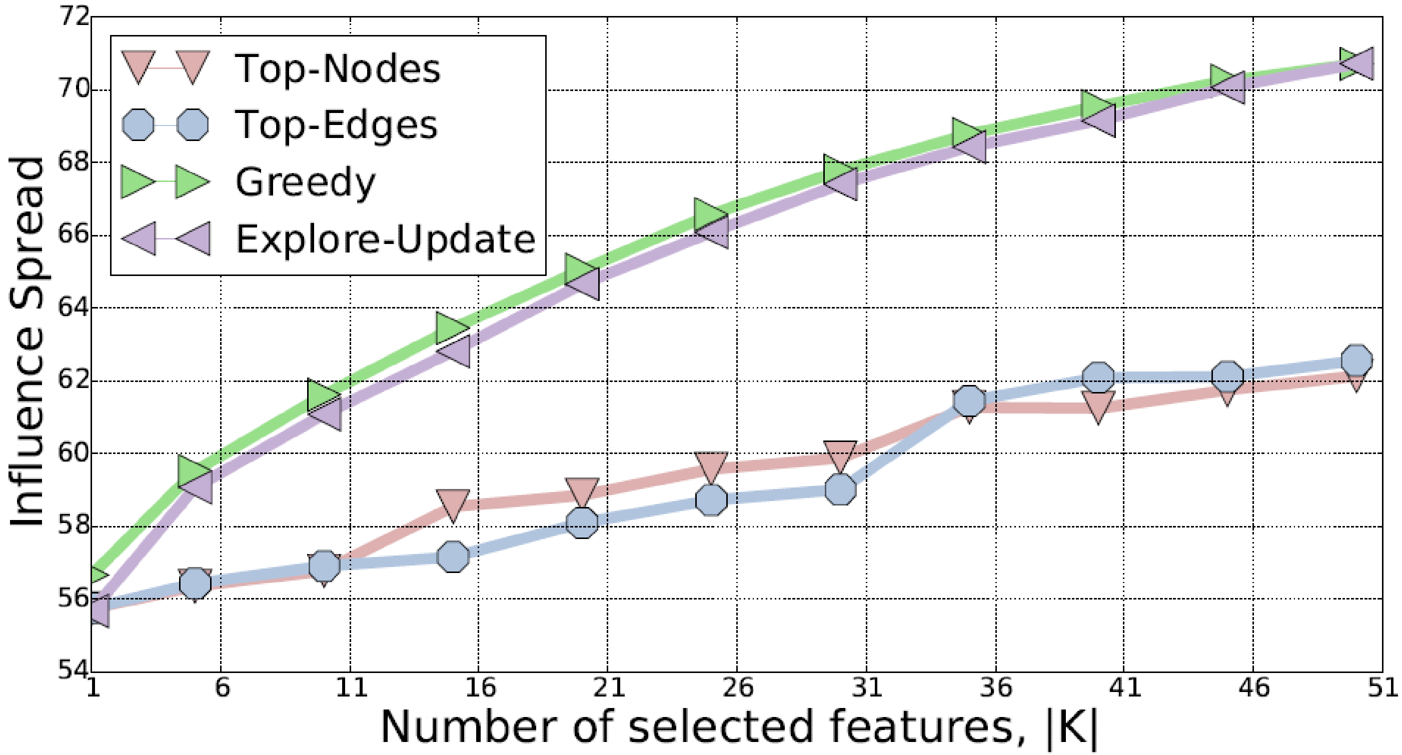


Algorithm Sketch:

- Represent graph as a family of trees;
- Do not compute influence for nodes in grey area;
- Add nodes with highest scores

[1] Maximizing the Spread of Influence through a Social Network, Kempe et al. 2003

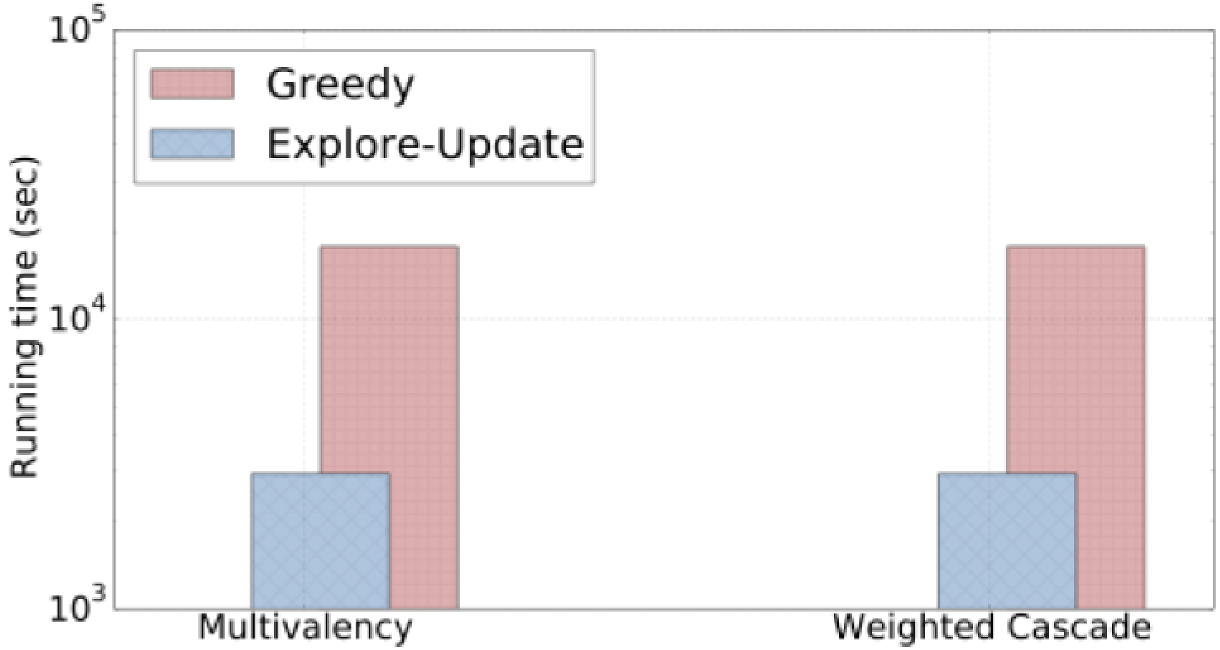
Results: influence function



Gnutella

Results: running time

Skoltech
Skolkovo Institute of Science and Technology



Runtime on Gnutella, $k = 50$.

Influence completion problem

Given:

- Directed graph $G = (V, E)$, a small set S of active users, propagation function $\sigma(S) = E(\#\ activated\ nodes)$, and a recommendation F.

Find:

- A set of k nodes to S so that propagation is maximized.

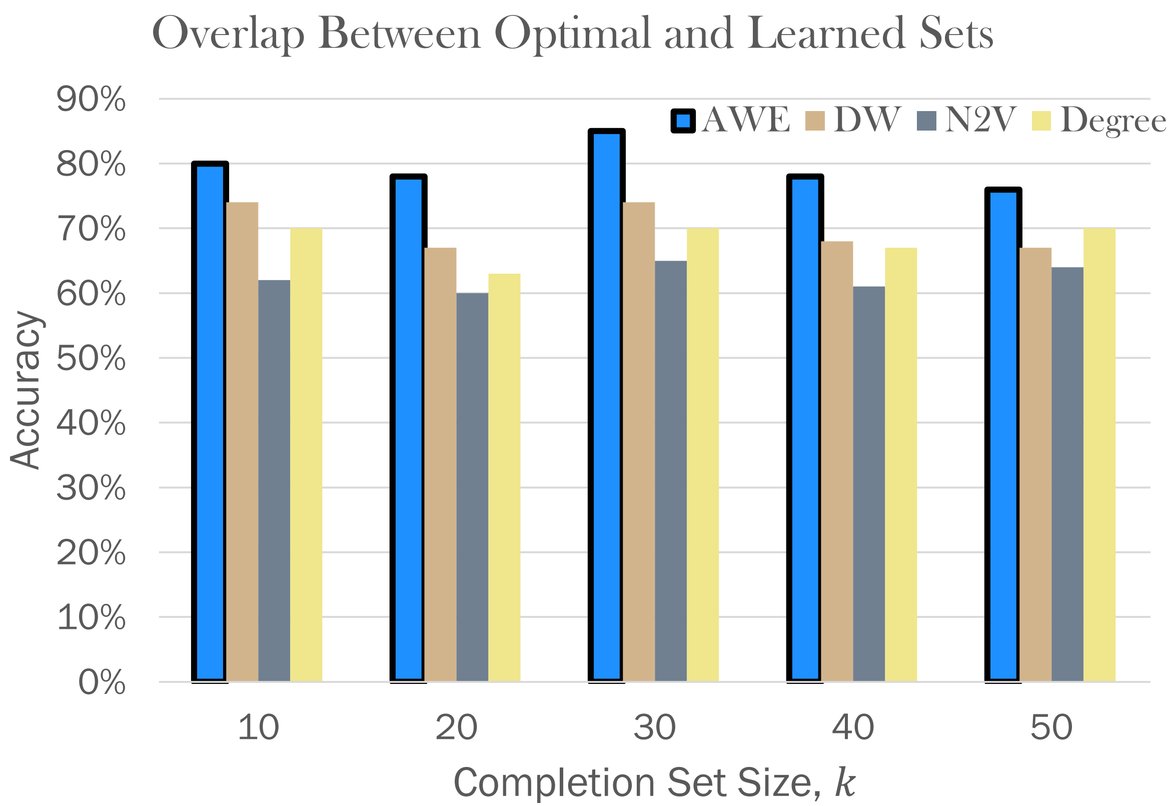
Problem

$$\max_{\cup_1^k v_i} \sigma\left(S + \cup_1^k v_i \middle| F\right)$$

Approach:

- Train a regression model with node embeddings using set S as positive class and non-influential nodes as negative class.

Results: accuracy



Given S and embeddings, we learn a classifier to complete S with influential nodes S_{algo}.

We measure:

$$Acc = \frac{S_{algo}}{S_{opt}}$$

where S_{algo} is learned set by classifier and S_{opt} is optimal set.

$|S| = 10$, GRQC dataset (5K nodes; 15K edges), classifier: SVM/LR

DeepWalk (DW) [Perozzi et al., 2014], Node2Vec (N2V) [Grover et al. 2016],

AWE combinatorial for each node

Main results of this thesis

Skoltech

Skolkovo Institute of Science and Technology

- Proposed and justified a new graph representation that **provides isomorphism property**;
- Designed two approaches for **approximate efficient computation of embeddings**;
- Demonstrated **superior quality of embeddings** in graph classification problem;
- Investigated **product recommendation problem with graph embeddings.**

Published papers

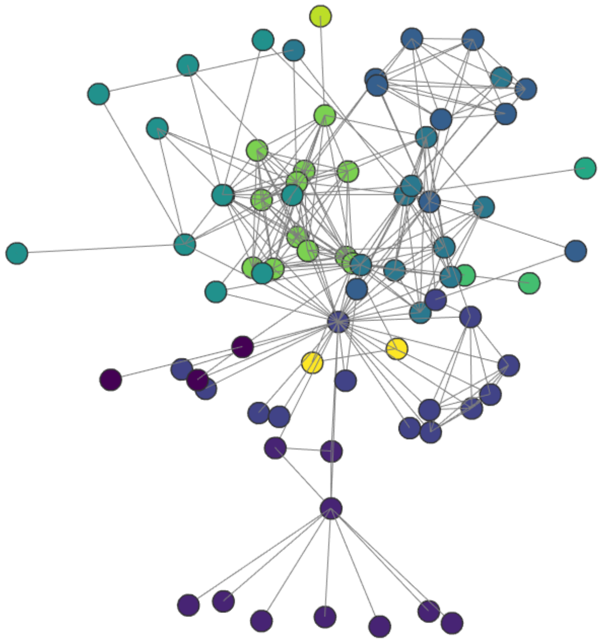
- **S. IVANOV** & P. KARRAS "*HARVESTER: INFLUENCE OPTIMIZATION IN SYMMETRIC INTERACTION NETWORKS*" , **PROCEEDINGS OF IEEE DATA SCIENCE AND ADVANCED ANALYTICS (DSAA)** 2016 SCOPUS.
- **S. IVANOV**, K. THEOCHARIDIS, M. TERROVITIS, P. KARRAS *"CONTENT RECOMMENDATION FOR VIRAL SOCIAL INFLUENCE"* **PROCEEDINGS OF SIG INFORMATION RETRIEVAL (SIGIR)** 2017.
- **S. IVANOV**, E. BURNAEV *"ANONYMOUS WALK EMBEDDINGS"* **PROCEEDINGS OF INTERNATIONAL CONFERENCE ON MACHINE LEARNING (ICML)** 2018.
- **S. IVANOV**, N. DURASOV, E. BURNAEV *"LEARNING NODE EMBEDDINGS FOR INFLUENCE SET COMPLETION"* **IEEE INTERNATIONAL CONFERENCE IN DATA MINING (ICDM) 2018 WORKSHOPS PROCEEDINGS** 2018.
- SHARAEV, ARTEMOV, BERNSTEIN, KONDRATYEVA, SUSHCHINSKAYA, BURNAEV, **IVANOV** "LEARNING CONNECTIVITY PATTERNS VIA GRAPH KERNELS FOR FMRI-BASED DEPRESSION DIAGNOSTICS" **IEEE INTERNATIONAL CONFERENCE IN DATA MINING (ICDM) 2018 WORKSHOPS PROCEEDINGS** 2018.

Acknowledgements

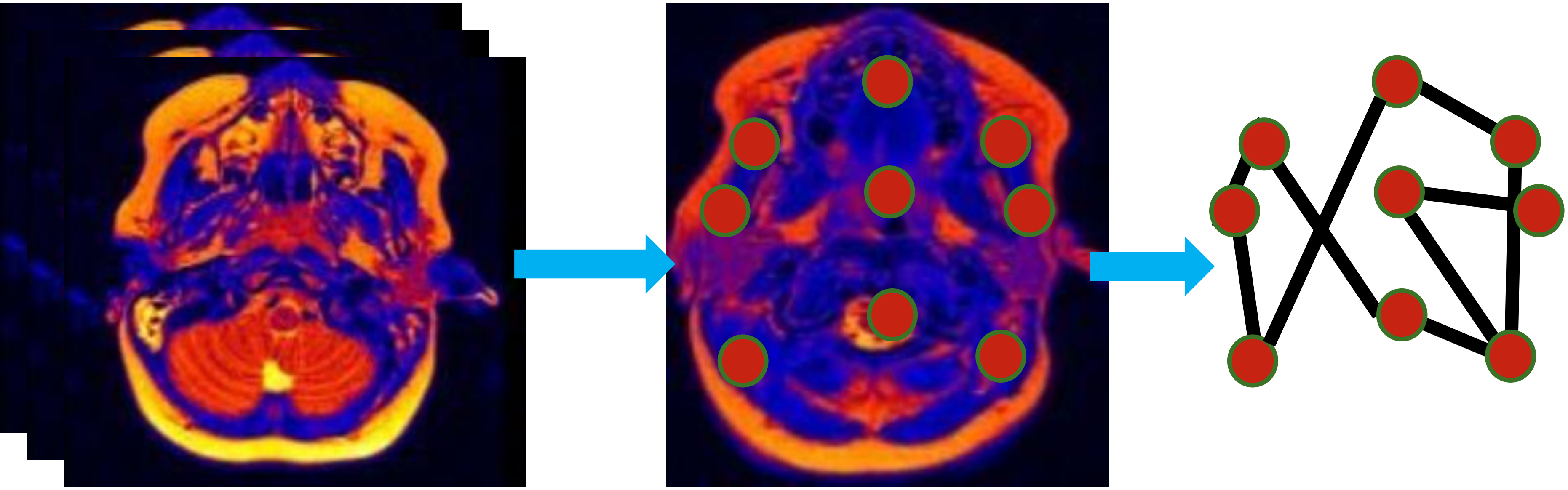
Skoltech
Skolkovo Institute of Science and Technology

- Supervisor: Evgeny Burnaev.
- Reviewers and Committee: Matthew Blaschko, Michael Bronstein, Andrzej Cichocki, Maxim Panov, Alexander Panchenko, Ivan Oseledets, Alexander Bernstein, and others.
- Skoltech educational office: Anna, Dmitry, Matthew, Natasha, and many others.
- Friends and family.

THANK YOU!



Medical Diagnostics



Nodes: AAL brain regions

Edges: correlation of changes in fMRI

Evaluation

- 4 groups of patients: healthy (H), depression (D), epilepsy (E), depression + epilepsy (DE)
- Each group has 25 graphs
- Two classification tasks: DvsH and DvsDE

Task	Naïve	WL	AWE
DvsH	$73 \pm 15\%$	$78 \pm 15\%$	$\mathbf{80 \pm 12\%}$
EvsDE	$67 \pm 15\%$	$75 \pm 14\%$	$\mathbf{76 \pm 16\%}$