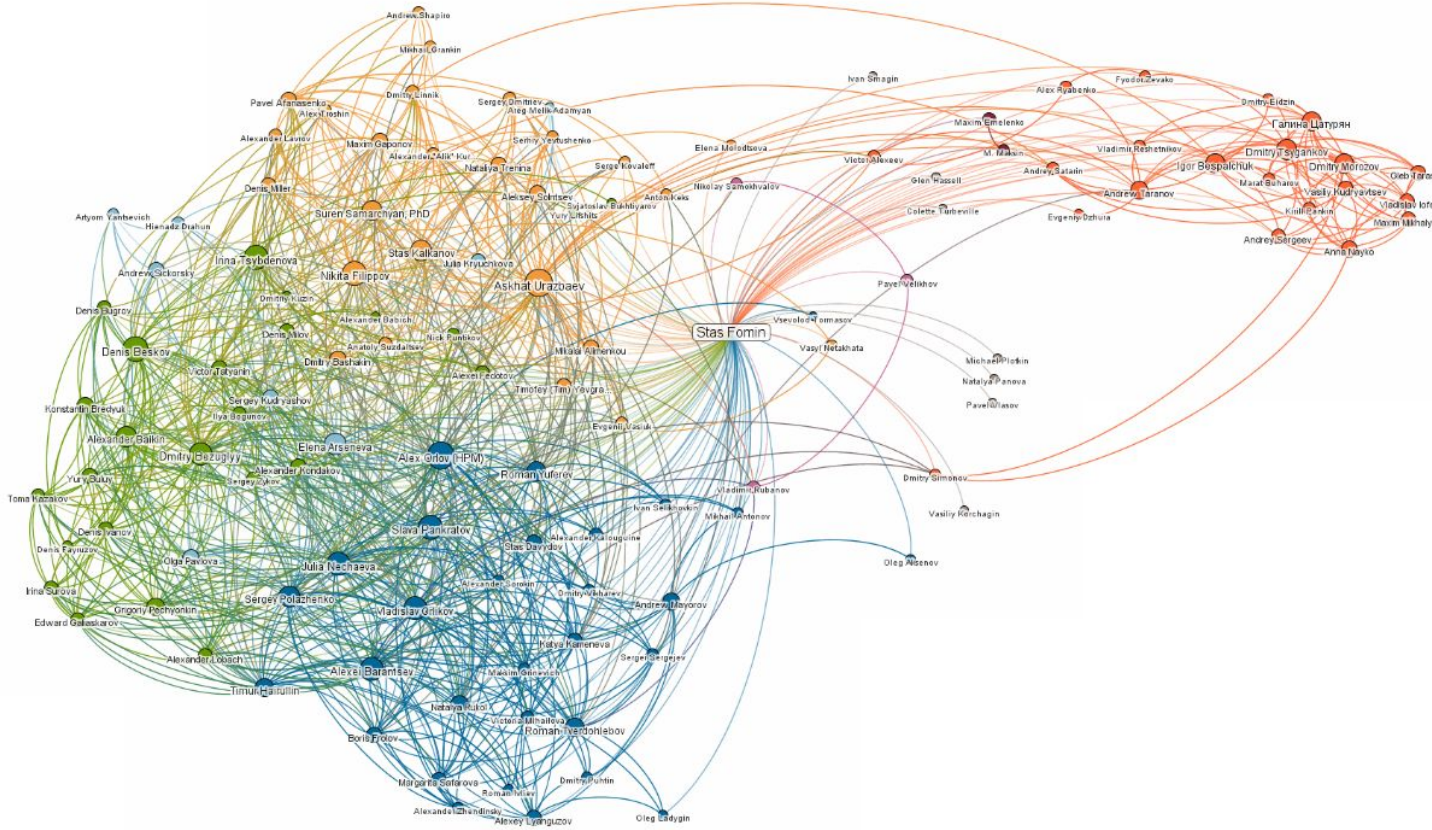
Graph Classification: A Comparison Study

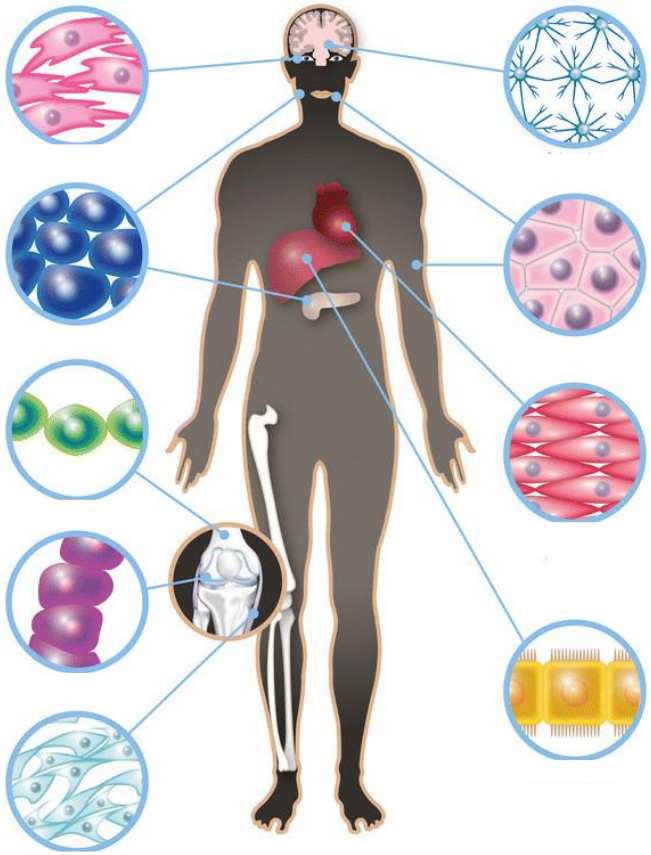
02/04/19

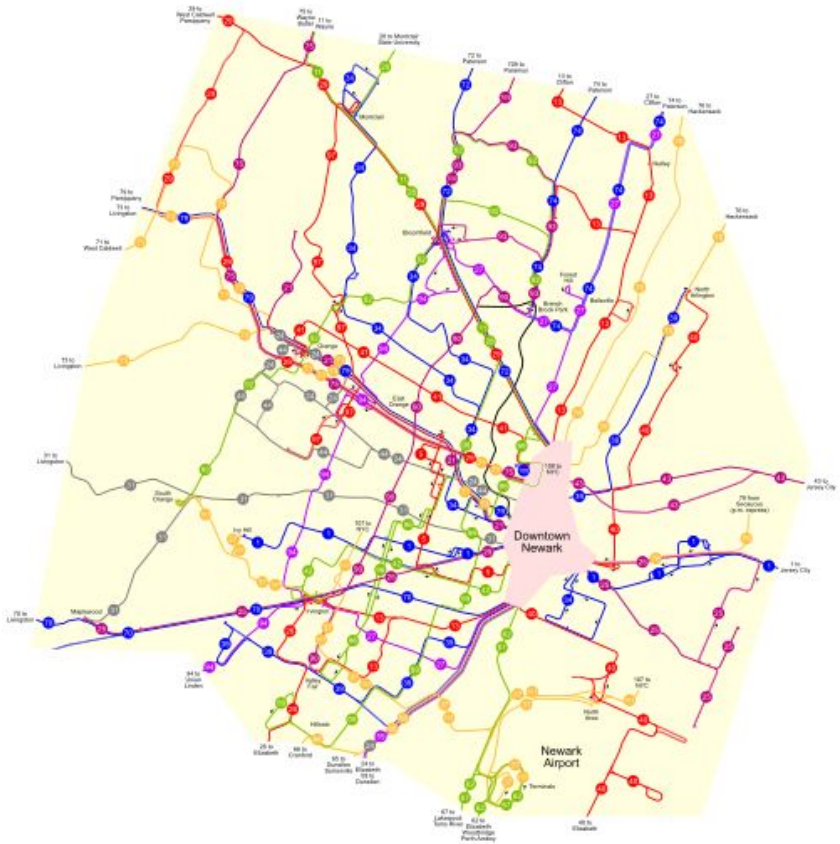
Presented by: Camilo Muñoz
Juan Carrillo

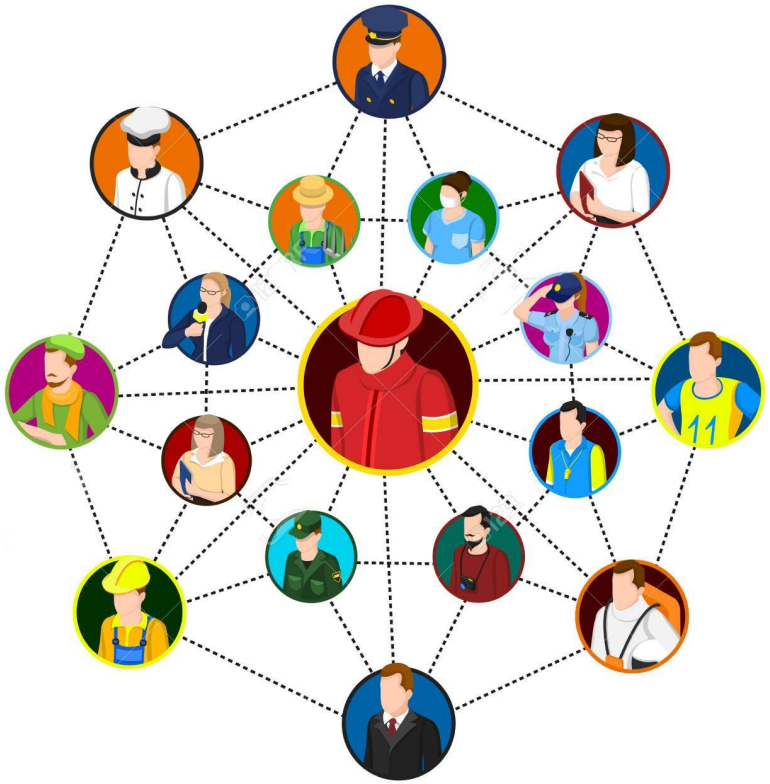


UNIVERSITY OF WATERLOO

Graph Classification



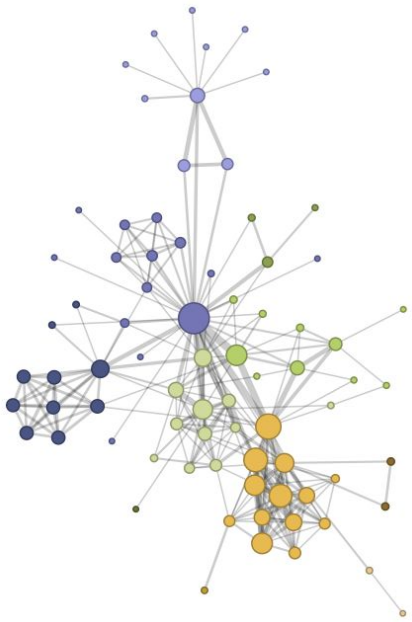




Graph Classification

Induce a mapping $f(x) : X \rightarrow \{\pm 1\}$
given a set of training samples

Input graph **Classifier** **Label**



Action
Comedy
Romance
Sci-fi

Algorithms for Graph Similarity
and Subgraph Matching

An Application of Boosting to Graph Classification

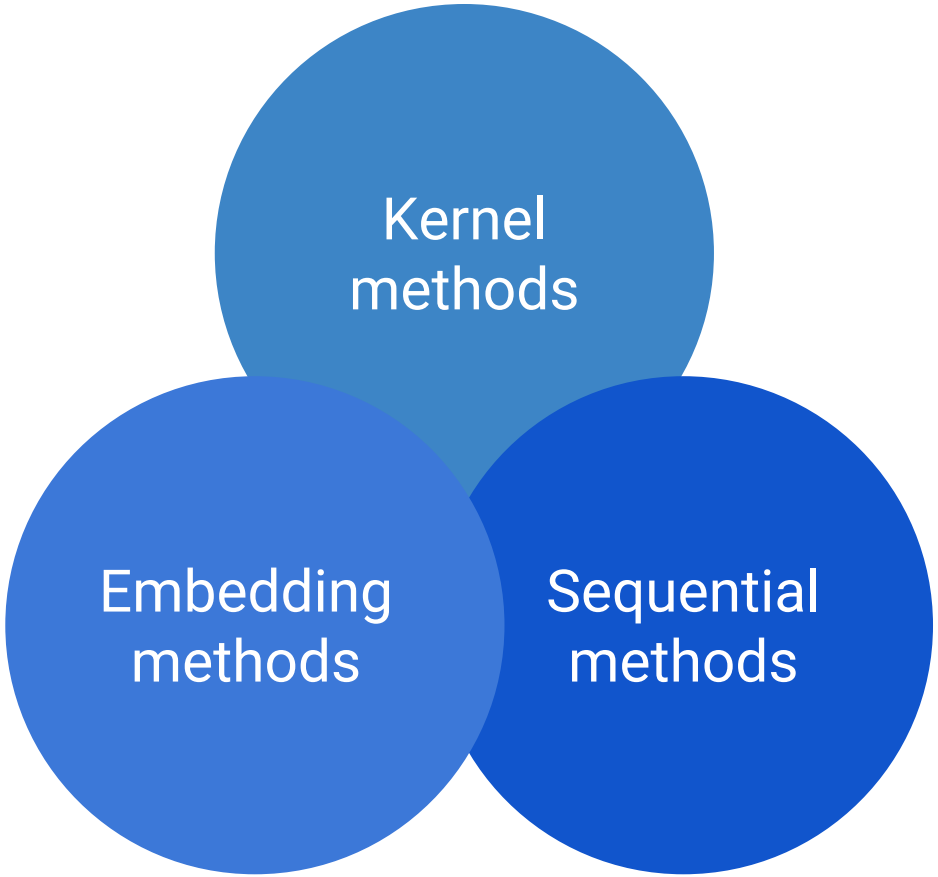
Taku Kudo, Eisaku Maeda
NTT Communication Science Laboratories.
2-4 Hikaridai, Seika-cho, Soraku, Kyoto, Japan
{taku,maeda}@cslab.kecl.ntt.co.jp

Yuji Matsumoto
Nara Institute of Science and Technology.
8916-5 Takayama-cho, Ikoma, Nara, Japan
matsu@is.naist.jp

Abstract

This paper presents an application of Boosting for classifying labeled graphs, general structures for modeling a number of real-world data, such as chemical compounds, natural language texts, and bio sequences. The proposal consists of i) decision stumps that use subgraph as features, and ii) a Boosting algorithm in which subgraph-based decision stumps are used as weak learners. We also discuss the relation between our algorithm and SVMs with convolution kernels. Two experiments using natural language data and chemical compounds show that our method achieves comparable or even better performance than SVMs with convolution kernels as well as improves the testing efficiency.

Graph Classification



Learning Graph-Level Representations with Recurrent Neural Networks

Yu Jin, Joseph F. JaJa
Department of Electrical and Computer Engineering
Institute for Advanced Computer Studies
University of Maryland, College Park
Email: yuj@umd.edu, joseph@umiacs.umd.edu

A Simple Baseline Algorithm for Graph Classification

Nathan de Lara
Telecom ParisTech
ndelara@enst.fr

Edouard Pineau
Telecom ParisTech - Safran
edouard.pineau@safrangroup.com

Abstract

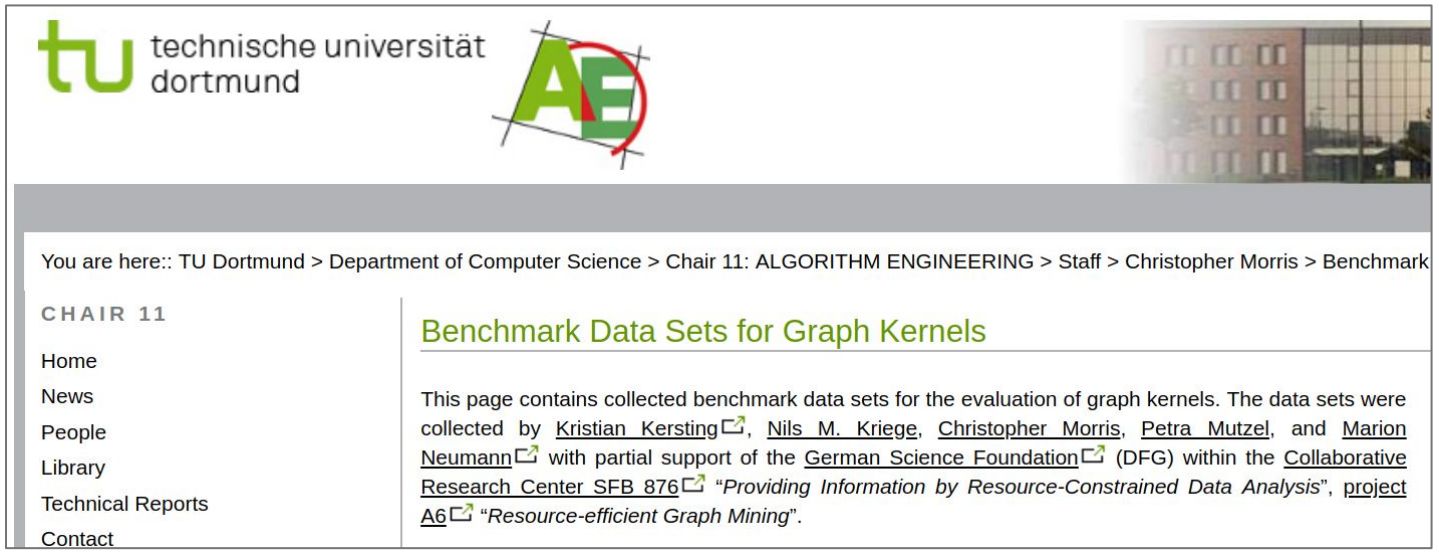
Graph classification has recently received a lot of attention from various fields of machine learning e.g. kernel methods, sequential modeling or graph embedding. All these approaches offer promising results with different respective strengths and weaknesses. However, most of them rely on complex mathematics and require heavy computational power to achieve their best performance. We propose a simple and fast algorithm based on the spectral decomposition of graph Laplacian to perform graph classification and get a first reference score for a dataset. We show that this method obtains competitive results compared to state-of-the-art algorithms.

The Comparison Study

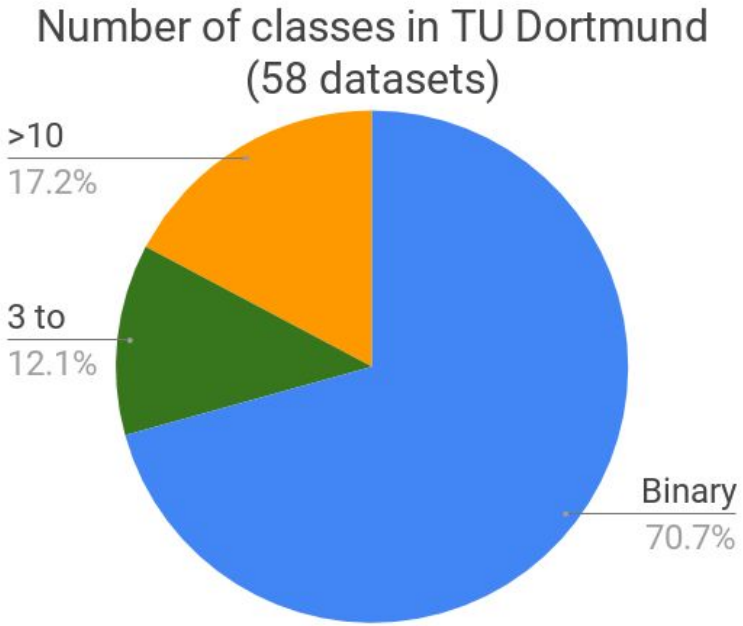
- Kernel CNN (KCNN)
- Deep Graph Kernels (DGK)
- graph2vec >> Embedding
- Multi-hop Assortativity (MHA)

Motivation

- Lack of evaluation of graph similarity techniques across categories
- Lack of experimental evaluation regarding **multiclass** classification



https://ls11-www.cs.tu-dortmund.de/staff/morris/graphkerneldatasets



Kernel CNN (KCNN)

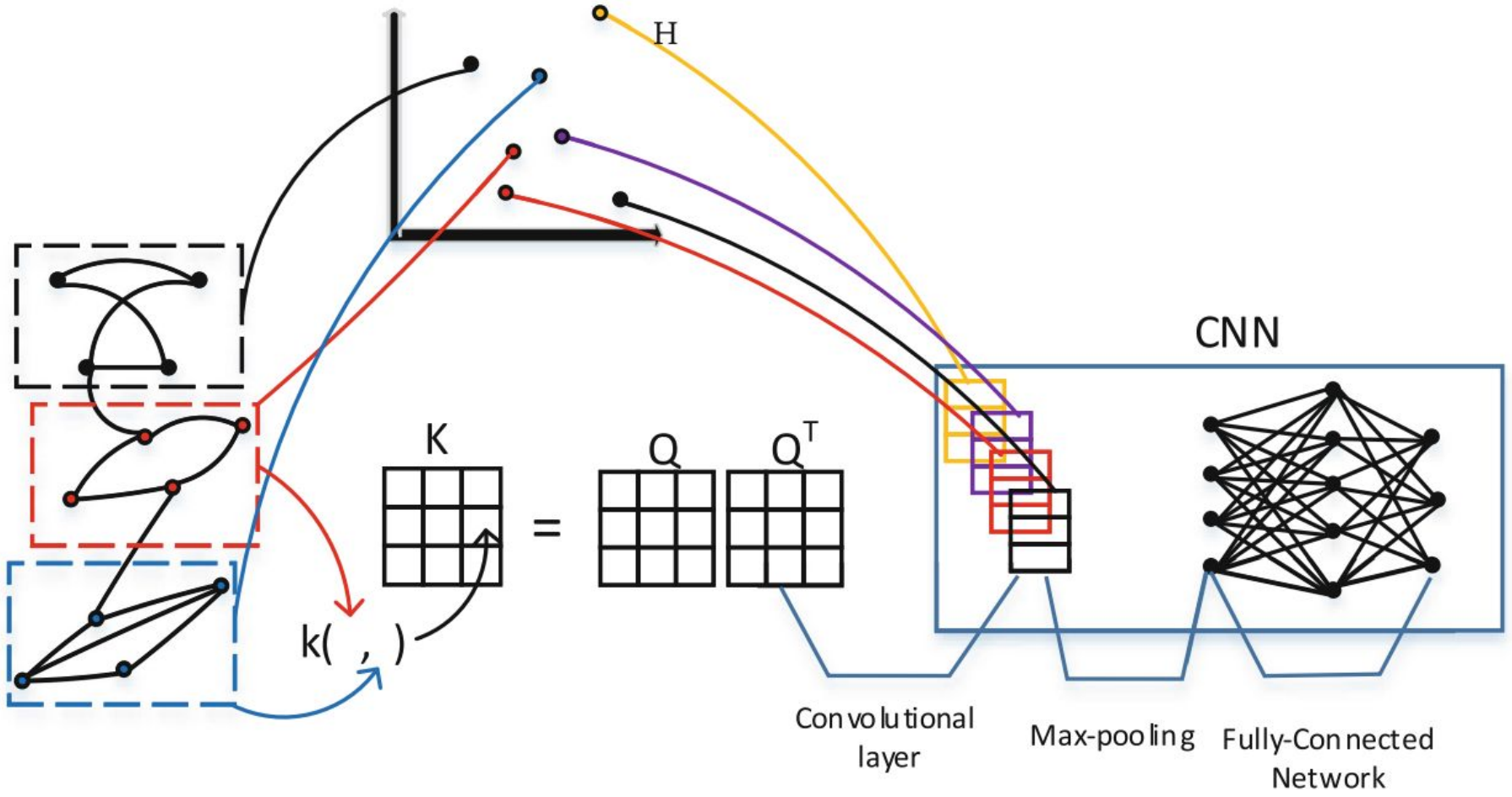
Kernel Graph Convolutional Neural Networks

Giannis Nikolentzos[1(✉)], Polykarpos Meladianos[2], Antoine Jean-Pierre Tixier[1], Konstantinos Skianis[1], and Michalis Vazirgiannis[1,2]

[1] École Polytechnique, Palaiseau, France
{nikolentzos,anti5662,kskianis,mvazirg}@lix.polytechnique.fr
[2] Athens University of Economics and Business, Athens, Greece
pmeladianos@aueb.gr

Abstract. Graph kernels have been successfully applied to many graph classification problems. Typically, a kernel is first designed, and then an SVM classifier is trained based on the features defined implicitly by this kernel. This two-stage approach decouples data representation from learning, which is suboptimal. On the other hand, Convolutional Neural Networks (CNNs) have the capability to learn their own features directly from the raw data during training. Unfortunately, they cannot handle irregular data such as graphs. We address this challenge by using graph kernels to embed meaningful local neighborhoods of the graphs in a continuous vector space. A set of filters is then convolved with these patches, pooled, and the output is then passed to a feedforward network. With limited parameter tuning, our approach outperforms strong base-

Kernel CNN (KCNN)



Deep Graph Kernels

Deep Graph Kernels

Pinar Yanardag
Department of Computer Science
Purdue University
West Lafayette, IN, 47906, USA
ypinar@purdue.edu

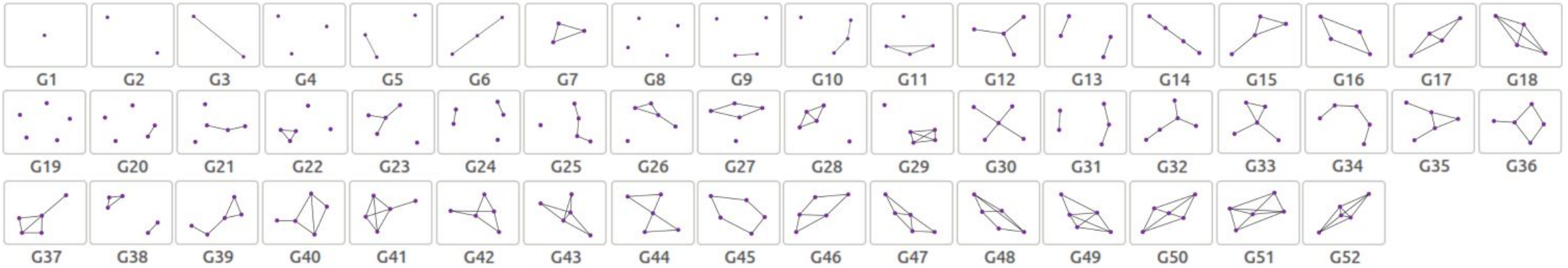
S.V.N. Vishwanathan
Department of Computer Science
University of California
Santa Cruz, CA, 95064, USA
vishy@ucsc.edu

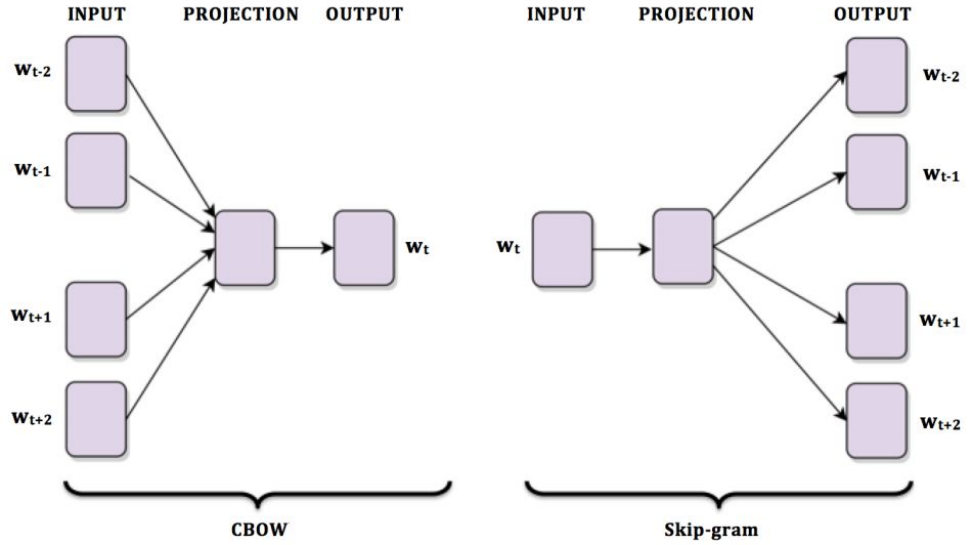
ABSTRACT

In this paper, we present Deep Graph Kernels, a unified framework to learn latent representations of sub-structures for graphs, inspired by latest advancements in language modeling and deep learning. Our framework leverages the dependency information between sub-structures by *learning* their latent representations. We demonstrate instances of our framework on three popular graph kernels, namely Graphlet kernels, Weisfeiler-Lehman subtree kernels, and Shortest-Path graph kernels. Our experiments on several benchmark datasets show that Deep Graph Kernels achieve significant improvements in classification accuracy over state-of-the-art graph kernels.

Then, the task is to predict which sub-community a discussion thread belongs to based on its communication graph. Similarly, in bioinformatics, one might be interested in the problem of identifying whether a given protein is an enzyme or not. In this case, the secondary structure of a protein is represented as a graph where nodes correspond to atoms and edges represent the chemical bonds between atoms. If the graph structure of the protein is similar to known enzymes, one can conclude that the given graph is also an enzyme [33]. Therefore, computing semantically meaningful similarities between graphs is an important problem in various domains.

Deep Graph Kernels





UNIVERSITY OF WATERLOO

graph2vec

graph2vec: Learning Distributed Representations of Graphs

Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu and Shantanu Jaiswal
Nanyang Technological University, Singapore
annamala002@e.ntu.edu.sg,
{mahinthan,rajasekarv,elhchen,yangliu}@ntu.edu.sg,shantanu004@e.ntu.edu.sg

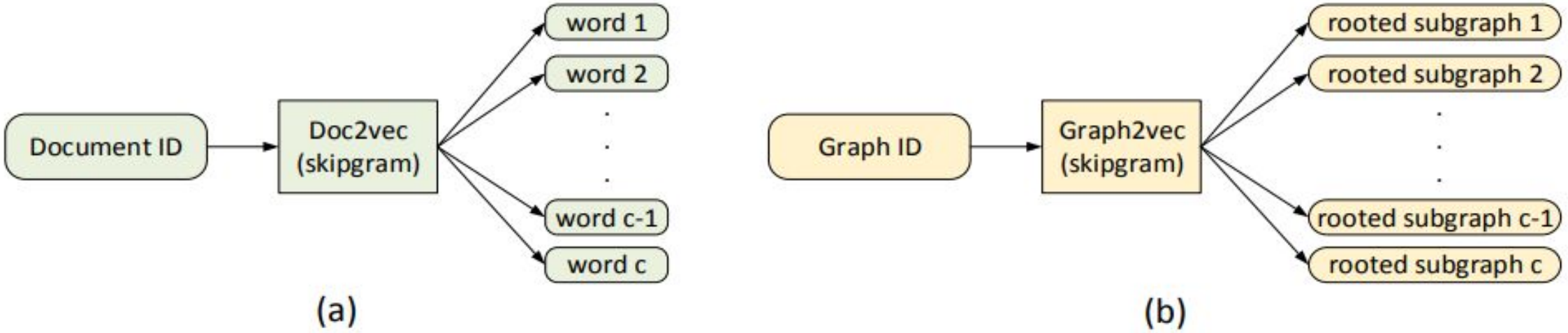
ABSTRACT

Recent works on representation learning for graph structured data predominantly focus on learning distributed representations of graph substructures such as nodes and subgraphs. However, many graph analytics tasks such as graph classification and clustering require representing entire graphs as fixed length feature vectors. While the aforementioned approaches are naturally unequipped to learn such representations, graph kernels remain as the most effective way of obtaining them. However, these graph kernels use handcrafted features (e.g., shortest paths, graphlets, etc.) and hence are hampered by problems such as poor generalization. To address this limitation, in this work, we propose a neural embedding framework named `graph2vec` to learn data-driven distributed representations of arbitrary sized graphs. `graph2vec`'s embeddings are learnt in an unsupervised manner and are task agnostic. Hence, they could be used for any downstream task such as graph classification,

malware [6] and those of chemical compounds could be used to predict their properties such as solubility and anti-cancer activity [7].

Graph Kernels and handcrafted features. Graph kernels are one of the most prominent ways of catering the aforementioned graph analytics tasks. Graph kernels evaluate the similarity (*aka* kernel value) between a pair of graphs G and G' by recursively decomposing them into atomic substructures (e.g., random walks, shortest paths, graphlets etc.) and defining a similarity (*aka* kernel) function over the substructures (e.g., counting the number of common substructures across G and G'). Subsequently, kernel methods (e.g., Support Vector Machines (SVMs)) could be used for performing classification/clustering. However, these kernels exhibit two critical limitations: (1) Many of them do not provide explicit graph embeddings. This renders using general purpose ML algorithms which operate on vector embeddings (e.g., Random Forests (RFs), Neural

graph2vec


Document ID
Doc2vec
(skipgram)
word 1
word 2
word c-1
word c
(a)
Graph ID
Graph2vec
(skipgram)
rooted subgraph 1
rooted subgraph 2
rooted subgraph c-1
rooted subgraph c
(b)

Multi-hop Assortativity (MHA)

Multi-hop assortativities for network classification

LEONARDO GUTIÉRREZ-GÓMEZ*,

Institute for Information and Communication Technologies, Electronics and Applied Mathematics (ICTEAM), Université catholique de Louvain, Avenue Georges Lemaître, 4, 1348 Louvain-la-Neuve, Belgium

*Corresponding author: leonardo.gutierrez@uclouvain.be

AND

JEAN-CHARLES DELVENNE

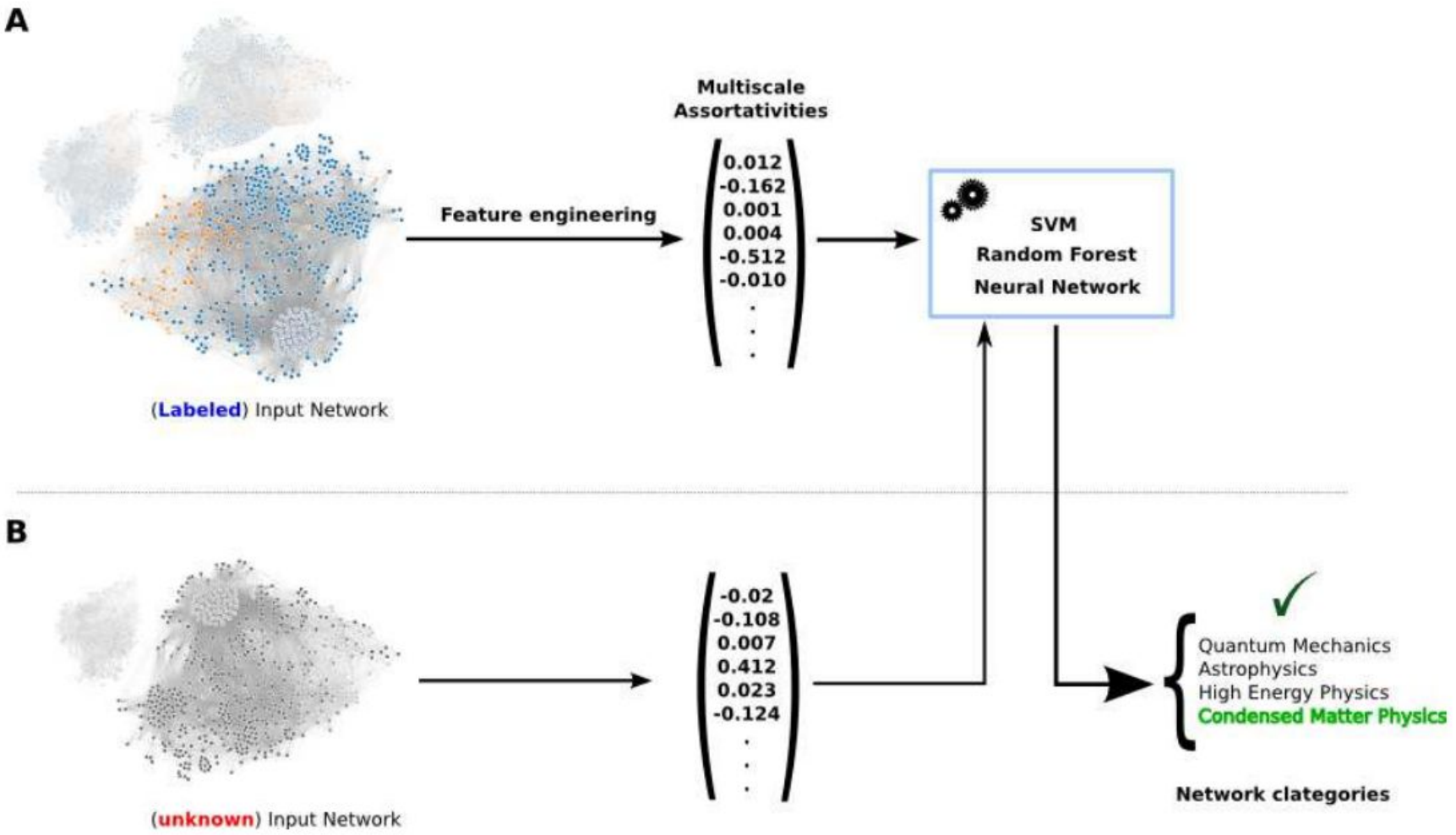
Institute for Information and Communication Technologies, Electronics and Applied Mathematics (ICTEAM) and Center for Operations Research and Econometrics (CORE), Université catholique de Louvain, Avenue Georges Lemaître, 4, 1348 Louvain-la-Neuve, Belgium

jean-charles.delvenne@uclouvain.be

[Dated 19 November 2018]

Several social, medical, engineering and biological challenges rely on discovering the functionality of networks from their structure and node metadata, when it is available. For example, in chemoinformatics one might want to detect whether a molecule is toxic based on structure and atomic types, or discover the research field of a scientific collaboration network.

Multi-hop Assortativity (MHA)

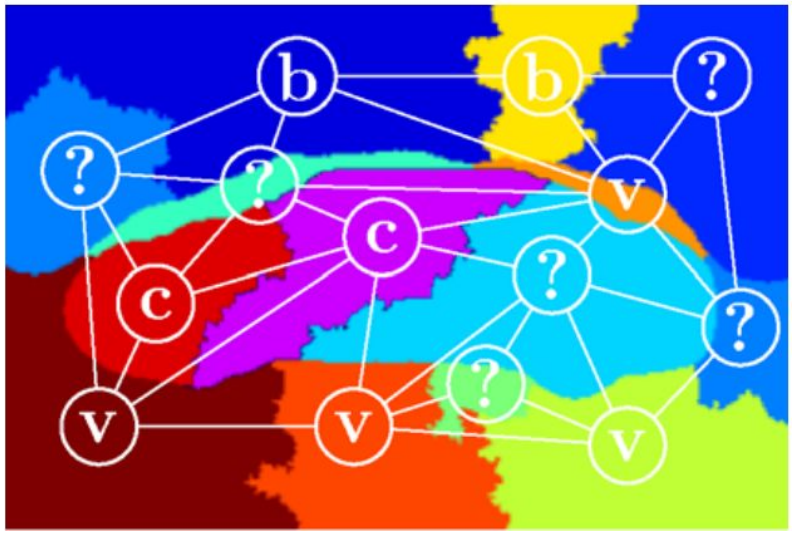


Experimental Evaluation

- RQ1: How can the nature of the graph data (e.g. number of nodes, average number of edges per node) impact the performance of the techniques?
- RQ2: Is there a clear difference in performance when using binary classification datasets versus using multiclass graph data?
- RQ3: Is there a technique that clearly outperforms the others in terms of performance?

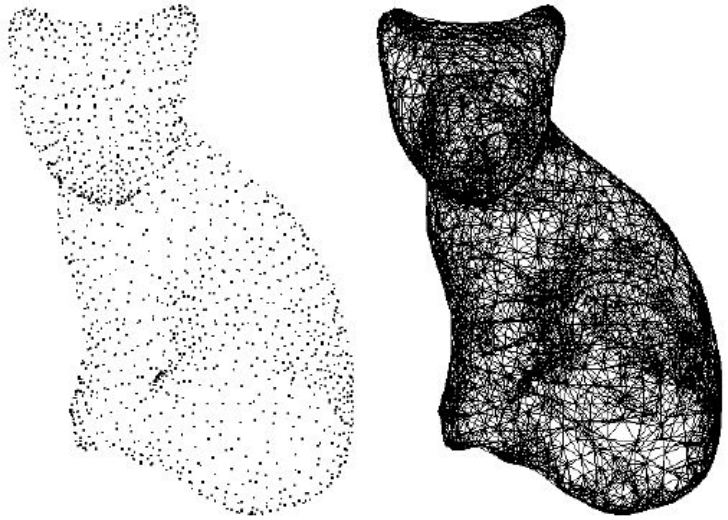
Microsoft datasets

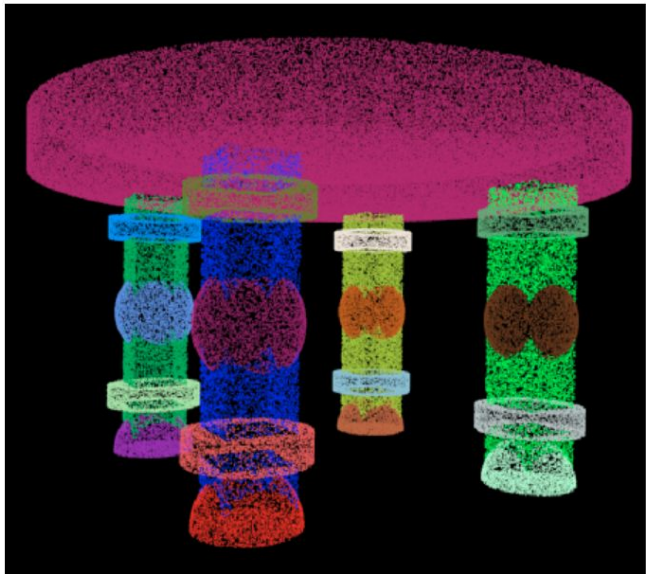
- Semantic image processing
- The class for each graph corresponds to its semantic meaning. For example *building*, *grass*, *tree*, *face*, *car*, *bicycle*



First-MM dataset

- First-MM stands for *Flexible Skill Acquisition and Intuitive Robot Tasking for Mobile Manipulation in the Real World*
- The graphs represent 3d point clouds of household objects





IMDb datasets

- Movie collaboration graphs
- The class correspond to the genre of the movie such as *Action*, *Romance*, *Comedy*, and *Sci-Fi*

Reddit datasets

- User discussion datasets
- Binary dataset contains posts from 4 popular subreddits: *IAmA*, *AskReddit*, *TrollXChromosomes*, and *atheism*
- Multiclass dataset contains posts from 5 subreddits: *worldnews*, *videos*, *AdviceAnimals*, *aww*, and *mildlyinteresting*

Experimental Evaluation

Name	Number of Graphs	Number of Classes	Average Number of Nodes	Average Number of Edges	Node Labels	Edge Labels
FIRSTMM_DB	41	11	1377.27	3074.1	yes	no
IMDB-BINARY	1000	2	19.77	96.53	no	no
IMDB-MULTI	1500	3	13	65.94	no	no
MSRC_9	221	8	40.58	97.94	yes	no
MSRC_21	563	20	77.52	198.32	yes	no
REDDIT-BINARY	2000	2	429.63	497.75	no	no
REDDIT-MULTI-5k	4999	5	508.52	594.87	no	no

Experimental Setup

- Dataset preprocessing
- Code customization
- Selection of initialization parameters
- Graph transformation technique
- Graph Classification

Evaluation Metrics

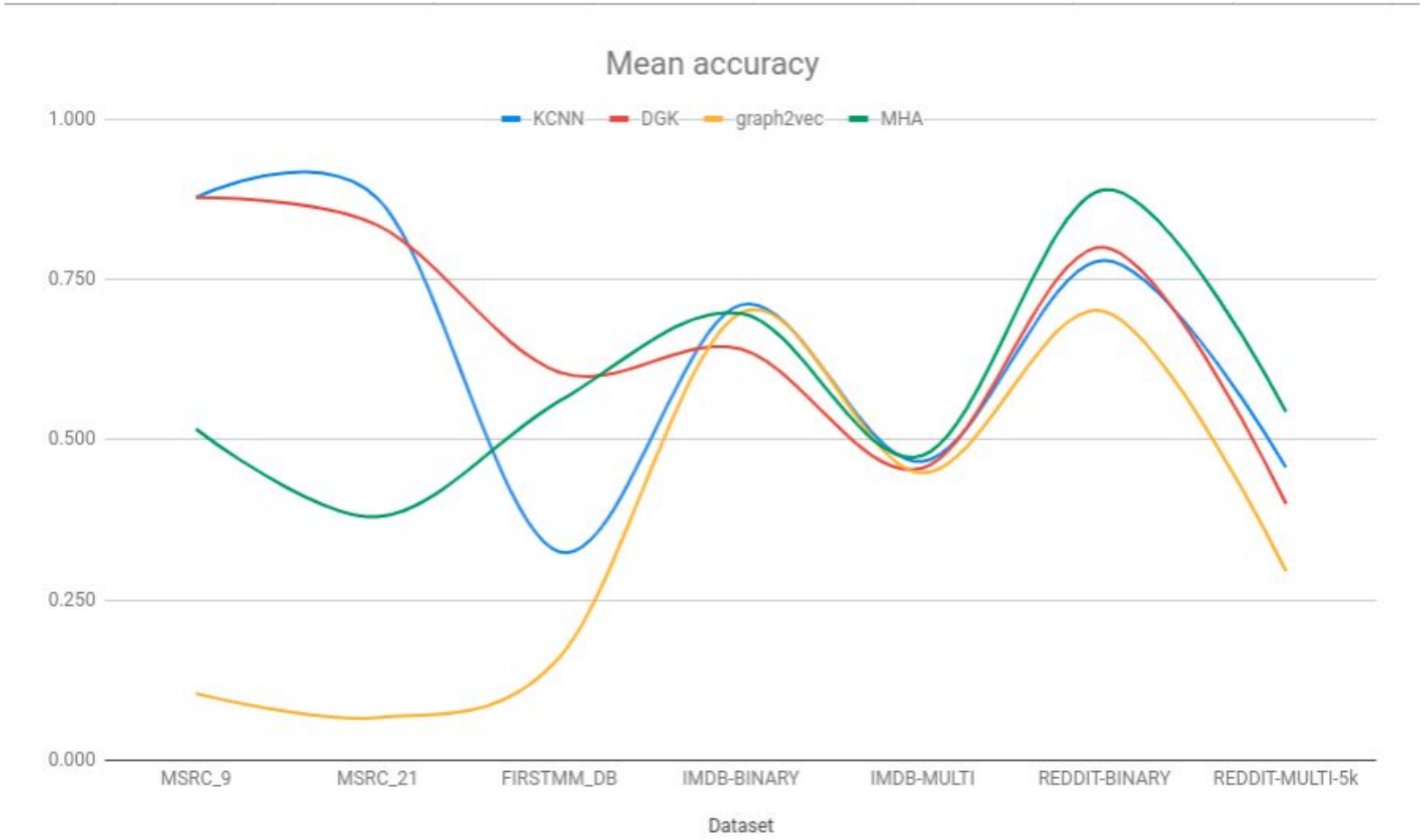
- Mean prediction accuracies and standard deviations
- Graph transformation runtime
- Additional storage required for transformed data

Evaluation Results

- Mean prediction accuracies and standard deviations

Dataset	KCNN	DGK	graph2vec	MHA
MSRC_9	87.77 ± 0.08	87.73 ± 0.01	**10.4 ± 6.59**	51.63 ± 0.05
MSRC_21	87.58 ± 0.03	83.41 ± 0.01	**6.67 ± 4**	**37.95 ± 0.04**
FIRSTMM_DB	**32.5 ± 0.23**	60.5 ± 0.04	**16 ± 8.94**	56 ± 0.01
IMDB-BINARY	71 ± 0.06	64.11 ± 0	69.8 ± 6.06	69.7 ± 0.04
IMDB-MULTI	**46.6 ± 0.03**	**45.55 ± 0**	**44.8 ± 2.88**	**47.53 ± 0.03**
REDDIT-BINARY	77.95 ± 0.05	80 ± 0.17	70 ± 5.5	89 ± 0.02
REDDIT-MULTI-5k	**45.67 ± 0.02**	**40 ± 0.08**	**29.5 ± 3.88**	54.37 ± 0.02

Evaluation Results



Evaluation Results

- Graph transformation runtime

Dataset	KCNN	DGK	graph2vec	MHA
MSRC_9	0.161 min	**1.09 hr**	1.09 min	0.039 min
MSRC_21	0.638 min	**12.72 hr**	5.09 min	0.168 min
FIRSTMM_DB	6.023 min	12.397 min	7.957 min	6.177 min
IMDB-BINARY	0.445 min	1.559 min	2.267 min	0.105 min
IMDB-MULTI	0.407 min	3.906 min	2.232 min	0.11 min
REDDIT-BINARY	**1.86 hr**	**5.44 hr**	**1.83 hr**	*Memory error
REDDIT-MULTI-5k	**4.66 hr**	**21.68 hr**	**6.14 hr**	*Memory error

Evaluation Results

- Additional storage required for transformed data

Dataset	KCNN	DGK	graph2vec	MHA
MSRC_9	3.36 MB	0.39 MB	6.7 MB	0.03 MB
MSRC_21	10.92 MB	2.54 MB	17.1 MB	0.15 MB
FIRSTMM_DB	2.63 MB	0.01 MB	1.2 MB	0 MB
IMDB-BINARY	8.81 MB	8 MB	30.4 MB	0.39 MB
IMDB-MULTI	8.29 MB	18 MB	45.6 MB	0.57 MB
REDDIT-BINARY	140.85 MB	32 MB	60.8 MB	0.79 MB
REDDIT-MULTI-5k	**365.29 MB**	**199.92 MB**	152.1 MB	2 MB

Potential Extensions

apk2vec: Semi-supervised multi-view representation learning for profiling Android applications

Annamalai Narayanan*, Charlie Soh*, Lihui Chen, Yang Liu and Lipo Wang
[annamala002,csoh004]@e.ntu.edu.sg,[elhchen,yangliu,elpwang]@ntu.edu.sg
Nanyang Technological University, Singapore

ABSTRACT

Building behavior profiles of Android applications (apps) with holistic, rich and multi-view information (e.g., incorporating several semantic views of an app such as API sequences, system calls, etc.) would help catering downstream analytics tasks such as app categorization, recommendation and malware analysis significantly better. Towards this goal, we design a semi-supervised Representation Learning (RL) framework named apk2vec to automatically generate a compact representation (*aka* profile/embedding) for a given app. More specifically, apk2vec has the three following unique characteristics which make it an excellent choice for large-scale app profiling: (1) it encompasses information from multiple semantic views such as API sequences, permissions, etc., (2) being a semi-supervised embedding technique, it can make use of labels associated with apps (e.g., malware family or app category labels) to build high quality app profiles, and (3) it combines RL and feature hashing which allows it to efficiently build profiles of apps that stream over time (i.e., online learning).

The resulting semi-supervised multi-view hash embeddings of apps could then be used for a wide variety of downstream tasks such as the ones mentioned above. Our extensive evaluations with more than 42,000 apps demonstrate that apk2vec's app profiles could significantly outperform state-of-the-art techniques in four app analytics tasks namely, malware detection, familial clustering, app clone detection and app recommendation.

KEYWORDS

Representation Learning, Graph Embedding, Skipgram, Malware Detection, App Recommendation

1 INTRODUCTION

becoming increasingly tough for markets to recommend up-to-date and meaningful apps that matches users' search queries, and (iii) with a significant number of plagiarists and malware authors hidden among app developers, these markets have been plagued with app clones and malicious apps.

One could observe that a systematic and deep understanding of apps' behaviors is essential to solve the aforementioned issues. Building high-quality behavior profiles of apps could help in determining the semantic similarity among the apps, which is pivotal to addressing these issues. Recent research [23, 24, 29, 32–38] reveals that compared to primitive representations of programs (e.g., counts of system-calls, Application Programming Interfaces (APIs) used etc.) graph representations (e.g., Control Flow Graphs (CFGs), call graphs, etc.) are ideally suited for app profiling, as the latter retain program semantics well, even when the apps are obfuscated. Reinforcing this fact, many recent works achieved excellent results using graph representations along with Machine Learning (ML) techniques on a plethora of program analytics tasks such as malware detection [23, 24, 32, 33, 38], familial classification [37], clone detection [29, 42], library detection [39] etc. In effect, these works cast their respective program analytics task as a graph analytics task and apply existing graph mining techniques [33] to solve them. Typically, these ML algorithms work on vectorial representations (*aka* embeddings) of graphs. Hence, arguably, one of the most important factors that determines the efficacy of these downstream analytics tasks is the quality of such embeddings.

Besides the choice of graph representations, another pivotal factor that influences the aforementioned tasks are the features that could be extracted from them. In the case of app analytics, the most prominent features in recent literature include API/system-call sequences observed [23], permissions [27] and information

Shortest-Path Graph Kernels for Document Similarity

Giannis Nikolentzos
École Polytechnique and AUEB
nikolentzos@aueb.gr

Polykarpos Meladianos
École Polytechnique and AUEB
pmeladianos@aueb.gr

François Rousseau
École Polytechnique
rousseau@lix.polytechnique.fr

Michalis Vazirgiannis
École Polytechnique and AUEB
mvazirg@aueb.gr

Yannis Stavrakas
IMIS / RC ATHENA
yannis@imis.athena-innovation.gr

Abstract

In this paper, we present a novel document similarity measure based on the definition of a graph kernel between pairs of documents. The proposed measure takes into account both the terms contained in the documents and the relationships between them. By representing each document as a graph-of-words, we are able to model these relationships and then determine how similar two documents are by using a modified shortest-path graph kernel. We evaluate our approach on two tasks and compare it against several baseline approaches using various performance metrics such as DET curves and macro-average F1-score. Experimental results on a range of datasets showed that our proposed approach outperforms traditional techniques and is capable of measuring more accurately the similarity between two documents.

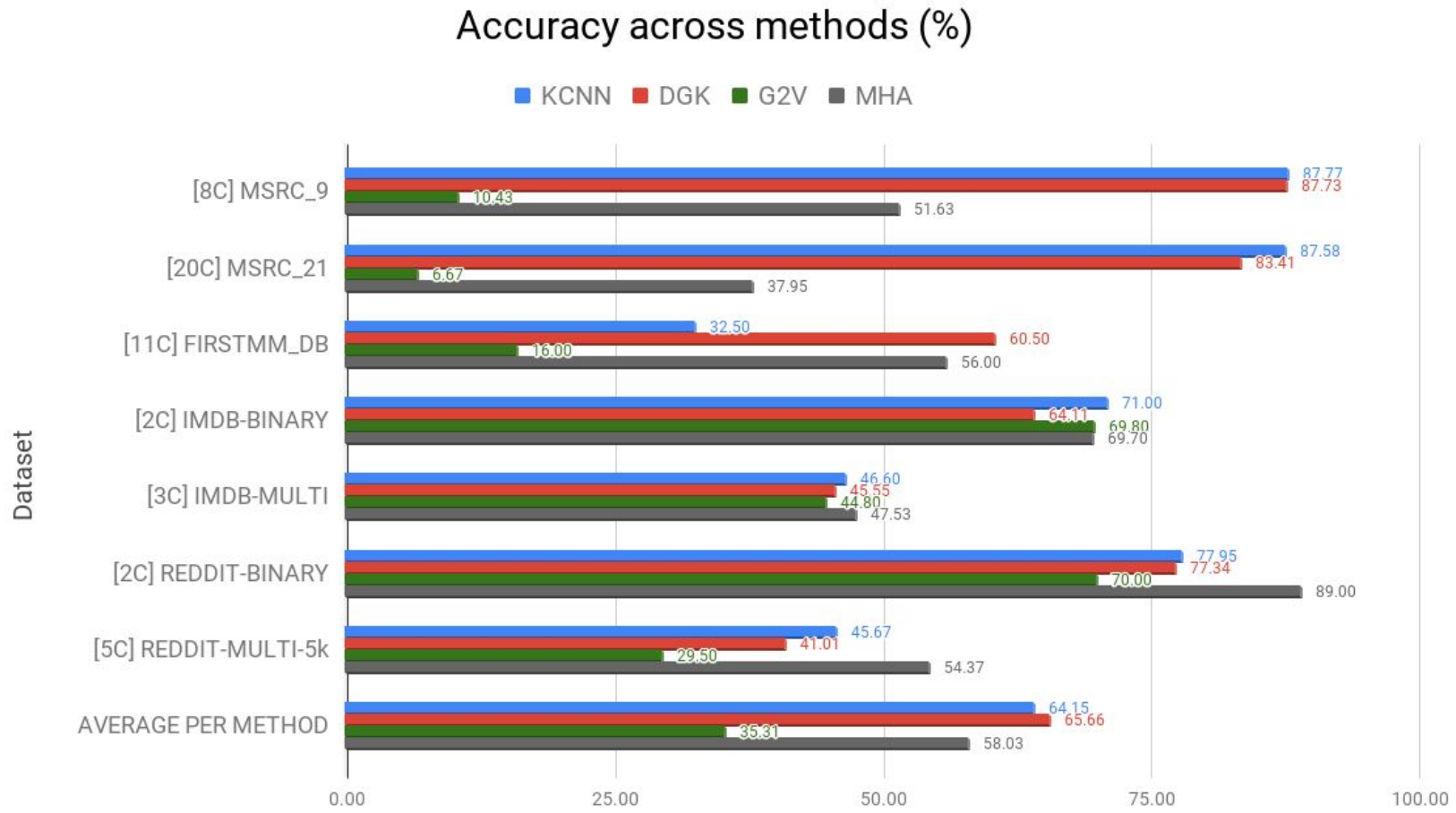
information shared by two objects (in our case documents). Determining the similarity between two documents is not a trivial task. Whether two documents are similar or different is not always clear and may vary from application to application.

Similarity measures that make use of the vector-space model (Salton et al., 1975) treat words in a document as if they were independent of one another, which is not realistic. In fact, words relate to one another to form meaningful phrases and to develop ideas. It is known that the human brain utilizes these relations between words to facilitate understanding (Altmann and Steedman, 1988). In general, we assume that two terms are related if they co-occur together in a small context, typically a phrase or a window of specific size, which resulted in n-gram features in many text mining tasks (an n-gram is a sequence of n terms in this paper). But n-grams correspond to sequences of words and thus fail to capture word inversion and subset matching (e. g., "article about news" vs.

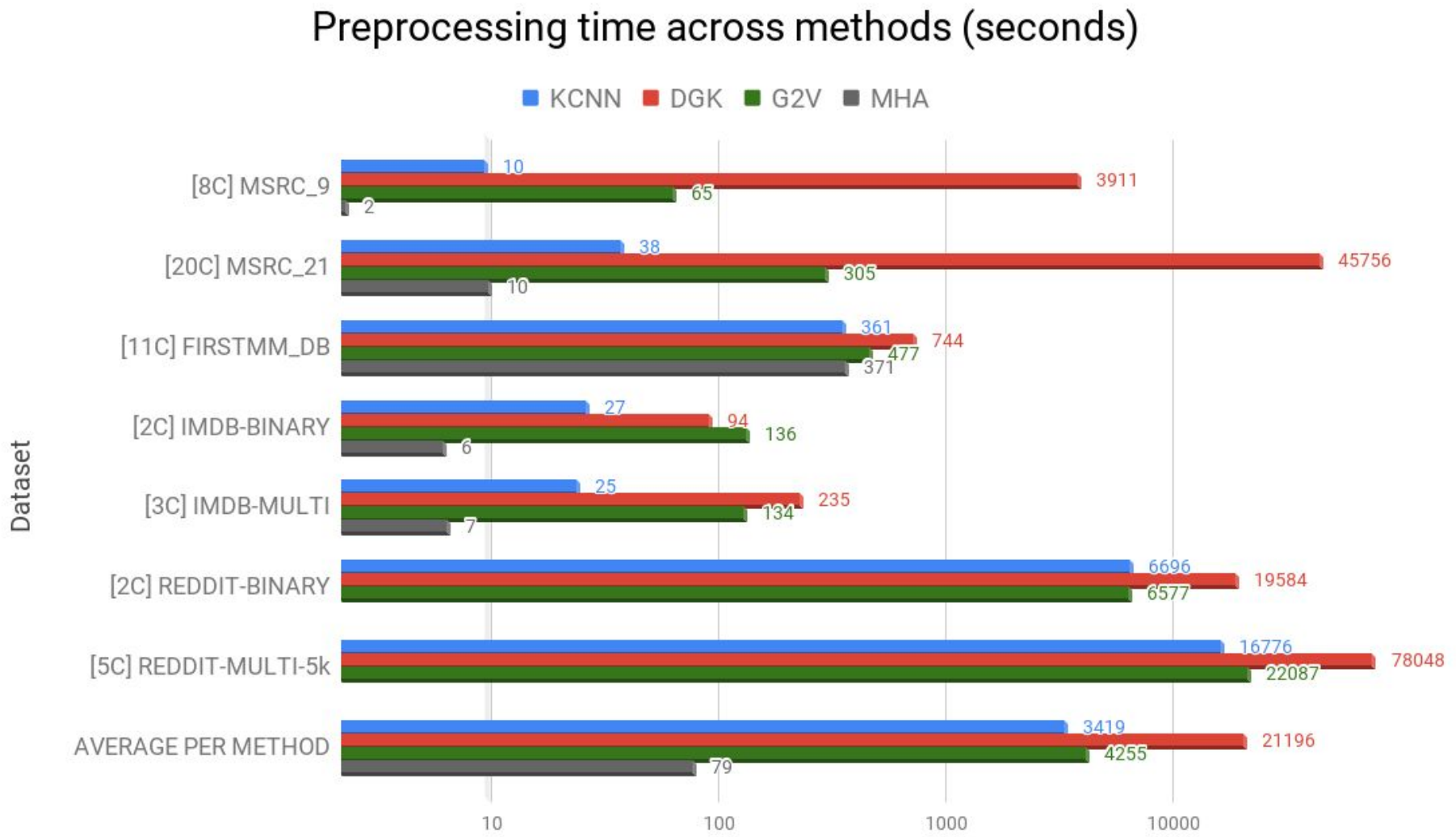
UNIVERSITY OF WATERLOO

Discussion

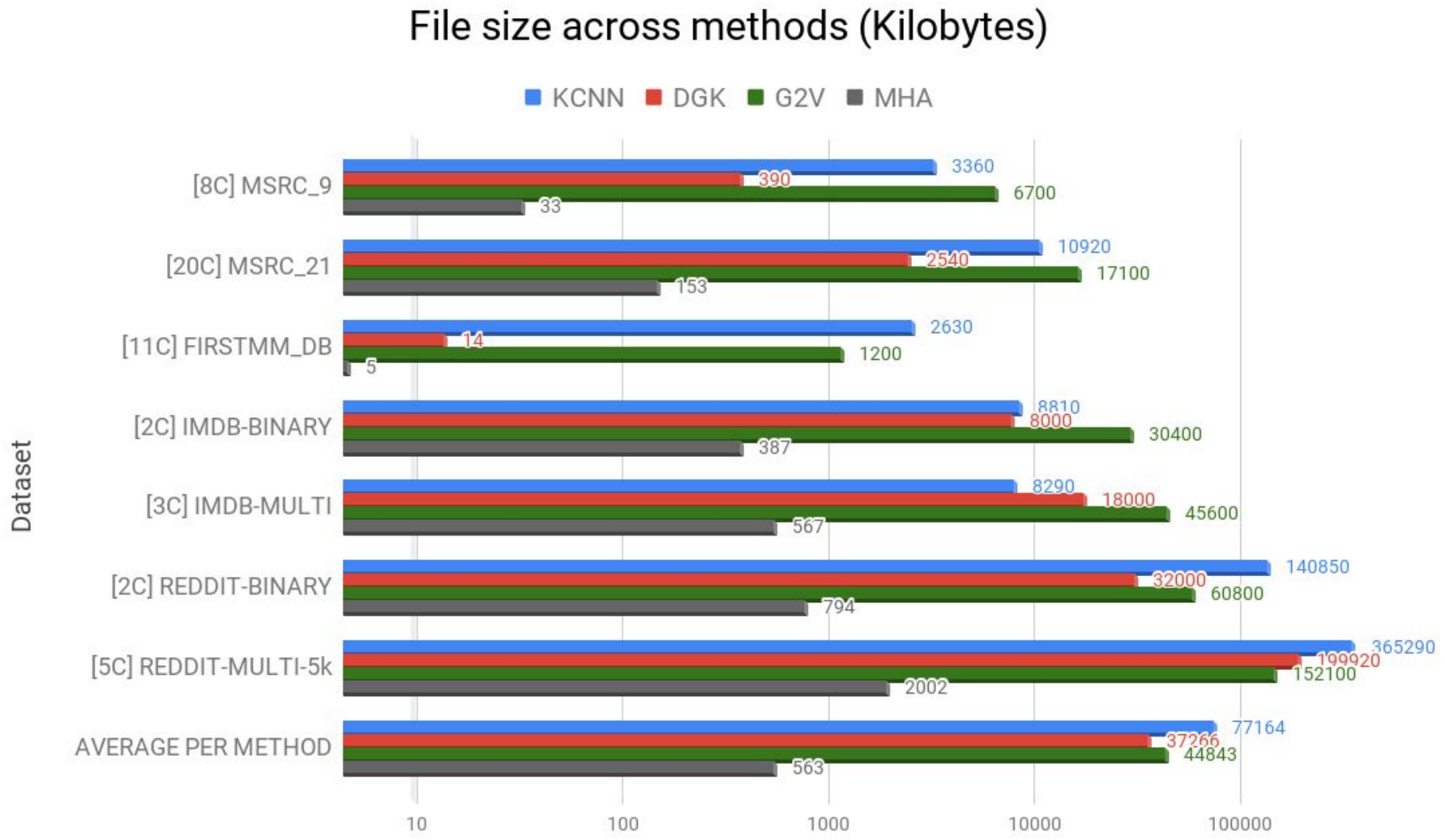
Appendix



UNIVERSITY OF WATERLOO



UNIVERSITY OF WATERLOO



UNIVERSITY OF WATERLOO