Rep the Set: Neural Networks for Learning Set representations

K. Skianis, G. Nikolentzos, S. Limnios, **M. Vazirgiannis**

Data Science and Mining group (DaSciM),
Laboratoire d'Informatique (LIX), École Polytechnique, France
http://www.lix.polytechnique.fr/dascim/ École Polytechnique, France
Preprint available at: https://arxiv.org/abs/1904.01962

April 26, 2019

Data Science and Mining @ Ecole Polytechnique, France

ÉCOLE POLYTECHNIQUE
UNIVERSITÉ PARIS-SACLAY

DaScim

Data Science and Mining Team
École Polytechnique

Home News Members Research Publications Teaching External services Contact

Data Science & Mining group

LIX @ Ecole Polytechnique

AI methods for large scale Graph and Text data

M. Vazirgiannis

http://www.lix.polytechnique.fr/dascim/

Data Science and Mining @ Ecole Polytechnique, France

Research Topics

- Machine Learning and AI
  - AI and Data Science methods (degeneracy, similarity, deep learning, multi-label classification)
  - Applications to: Text Mining/NLP, Social nets, Web marketing/advertising, Time Series

  J. Read, M. Vazirgiannis

- Operations Research and Mathematical programming
  - Optimization for Energy apps
  - Distance Geometry, protein conformation

  C. d'Ambrosio, L. Liberti

Graph of Words: graph based text/NLP

- *bag-of-words* vs. ***graph-of-words**:*

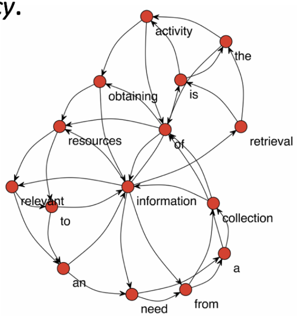
graph captures *word order* and *dependency*.

information retrieval is the activity of obtaining

information resources relevant to an information need

from a collection of information resources

Bag of words: ((activity,1), (collection,1)
(information,4), (relevant,1),
(resources, 2), (retrieval, 1)..)

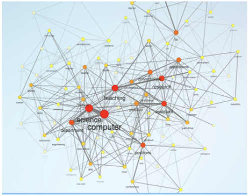


"Graph of word approach for ad-hoc information retrieval", F. Rousseau, M. Vazirgiannis,
Best paper mention award ACM CIKM 2013

Graph of Words: graph based text/NLP

Graph of Words approach with applications to

- Ad Hoc Information Retrieval (tw-idf) [CIKM2013]
- Keyword Extraction [ECIR2015, EMNLP2016]
- Extractive/Abstractive summarization of text streams [EACL2017, ACL 2018]
- Event Detection in Textual Streams (twitter, banking,…) [ICWSM2015, ECIR2018]
- Text Categorization/opinion mining/sentiment analysis [ACL2015, EMNLP2015, EMNLP2016, EMNLP2017]
- *Document visualization and summarization* [ACL2016, ACL2018]
  - GoWis prototype software



Other production

- Software protection - 2013
- Tech Transfer: Startup creation – NELPER@ incubator (automated text generation for web marketing/ads)

Machine/Deep Learning methods for Graphs

- Novel metrics for node /community importance
  - Extensions of k-core to weighted, directed (D-core) and signed graphs [ASONAM2011, ICDM2011, KAIS2013 , SIAMDM2013]
- Scalable Degeneracy-based graph clustering
  - Acceleration of high complexity clustering algorithms based on the k-core structure [AAAI2014]
  - 10^9 node graph clustering and community detection for fraud detection
- Identification of influential spreaders
  - Identification of influential spreaders [Scientific Reports/Nature 2016]
  - Novel influence metrics (citation and social networks) [PLOS2018]
  - RCG: Novel metric for academic paper influence [Infometrics2019]

Machine/Deep Learning methods for Graphs

Deep learning for graph and node embeddings
- Kernel Graph CNN [ICANN 2018]
- Learning Structural Node Representations on Directed Graphs [COMPLEX NETS 2018]
- Graph Classification with 2D Convolutional Neural Networks [https://arxiv.org/abs/1708.02218]

Deep Learning for Sets
- RepSet: Neural Networks for Learning Set Representations [https://arxiv.org/abs/1904.01962]

Graph kernels for graph similarity
- Message Passing GKs [arxiv]
- Matching Node Embeddings for Graph Similarity [AAAI 2017]
- Degeneracy framework for graph similarity [IJCAI 2018 - best paper award]
- Enhancing graph kernels via successive embeddings [CIKM 2018]
- Shortest-path graph kernels for document similarity [ENMLP 2017]

Grakel: open source *graph similarity* python library: - https://github.com>ysig>Grakel

Industrial Collaborations and Projects

- BNP (2016 – 2019) - CIFRE Ph.D.
  - Entity & event detection in online streaming documents
- Linagora (BPI – 2015 – 2021)
  - Automated summarization for online meetings
- AXA Industrial chair (2015-18)
  - Data science on insurance data
- AIRBUS (2014 - 17)
  - Data Analytics & Predictions of critical events
  - CIFRE PhD funding: Predictive Maintenance in Aviation: Failure Prediction for predictive maintenance [IEEE-ICDE 2018]
- HUAWEI (2018 - 21) - CIFRE Ph.D.
  - Deep Learning for Graphs
- Google
  - Graph mining for citation and social networks with degeneracy (2012-15 – Ph.D. fellowship)
- Tradelab (2017-20)
  - COM4U: Machine Learning for web marketing and advertising
  - CIFRE Ph.D. funding
- Microsoft
  - Azure grant
  - Open academic data initiative
- Tencent
  - Fraud detection in graphs

Machine Learing on Sets

- Typical ML algorithms (i.e. regression or classification) designed for fixed dimensionality objects.

<these words are in sequence for pedagogical purposes>

<pedagogical purposes are in sequence for these words>

...

vs.

{are, in, for, pedagogical, purposes, these, sequence, words}

- *s*imalrity learning between sets should be *invariant to permutation*: challenging task
  - **supervised tasks:** set output label invariant or equivariant to the permutationi its elements.
  - population statistics estimation, giga-scale cosmology, nano-scale quantum chemistry.
  - **unsupervised tasks**, "set" representation needs to be learned.
    - *set expansion* - assume a set of similar objects - find similar to the set extensions, i.e. extend the set {*lion, tiger, leopard*} with *cheetah*
    - *web marketing* extend a set high-value customers with similar people.
    - *astrophysics*: assuming set of interesting celestial objects, find similar ones in sky surveys.

Background and State of the art

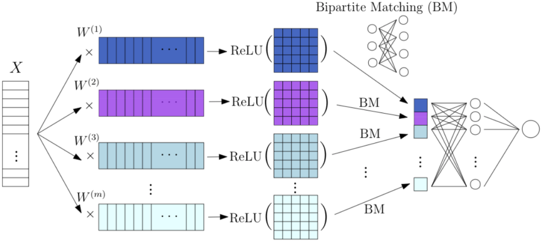
- NNs for sets became very popular isnpired by computer vision problems such as the automated classification of point clouds. Proposed architectures have achieved state-of-the-art results on many different tasks.
- Base approaches: PointNet [Qi et.al., CVPR2017] and DeepSets [Zaheer et.al., NIPS2017]
  - transform sets' elements vectors using several NN layers into new representations
  - apply some permutation-invariant function to the emerging vectors to generate representations for the sets.
  - Pointnet: max pooling, DeepSets: vector sum
  - representation of the set is then passed on to a standard architecture (e.g., fully connected layers,nonlinearities, etc).
  - Other efforts: PointNet++, SO-Net

Motivation and Contribution

- Data objects decomposed into sets of simpler objects: natural to represent each object as the *set* of its components or parts.
- Conventional ML algorithms operate on vectors / sequences. Thus unable to process *sets* as
  - sets may vary in cardinality
  - set elements lack a meaningful ordering
- **Challenge**: Sets as input to Neural Network Architectures
- **Contribution**: RepSet: a new neural network architecture, handling examples as *sets of vectors*.
  - computes the correspondences between an input set and some hidden sets by solving a series of network flow problems.
  - resulting representation fed to a NN architecture to produce the output.
  - allows end-to-end gradient-based learning.
  - Experimental evaluation: favorable on classification (text, graph) tasks outperforming satet of the art

Repset architecture - Permutation Invariance

- Assume an example X represented as a set $X = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ of d-dimensional vectors, $\mathbf{v}_i \in \mathbb{R}^d$. (i.e the embeddings of $X's$ elements)
- Objective: design architecture whose output is invariant for all n! permutations of X elements $=>$ permutation invariant function.

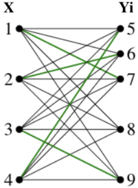


- propose a novel permutation invariant layer
- contains m "hidden sets" $Y_1, Y_2, \ldots, Y_m$ of d-dimensional vectors (same dim as X elements)
- based on bipartite graph matching
- its components are trainable,
- elements of a hidden set Y_i correspond to the columns of a trainable matrix $\mathbf{W}^{(i)}$.

Repset architecture - Similarity via graph matching

- to measure the similarity between X and each one of the hidden sets Y_i: comparing their components.
- capitalize on network flow algorithms - specifically **bipartite matching**: compute optimal mapping between the elements of X and the elements of each hidden set Y_i.

1	1.76	0.40	0.97
2	2.24	1.86	−0.97
3	0.95	−0.15	−0.10
4	0.41	0.14	1.45



5	0.52	0.08	1.62
6	2.14	1.72	−1.05
7	1.55	−0.45	0.88
8	−0.34	−1.26	0.24
9	1.08	−0.21	−0.09

- Each edge e connects a vertex in X to one in Y_i.
- Matching M: subset of edges - each node in X connects to one in Y_i.
- *optimal solution* is interpreted as similarity between node sets X and Y_i.
- The bipartite graph is a matrix $|X| \times |Y|$, cell values from $\{0,1\}$

Repset architecture - bipartite matching Optimization

- Assume a set of vectors, $X = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{|X|}\}$ and a hidden set $Y = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{|Y|}\}$, the bipartite matching between the elements of the two sets is solving the optimization problem:

$$
\begin{aligned}
& \sum_{i=1}^{|X|} \sum_{j=1}^{|Y|} x_{ij} f(\mathbf{v}_i, \mathbf{u}_j) \\
& \text{subject to:} \\
& \sum_{i=1}^{|X|} x_{ij} \leq 1 \quad \forall j \in \{1, \ldots, |Y|\} \\
& \sum_{j=1}^{|Y|} x_{ij} \leq 1 \quad \forall i \in \{1, \ldots, |X|\} \\
& x_{ij} \geq 0 \quad \forall i \in \{1, \ldots, |X|\}, \forall j \in \{1, \ldots, |Y|\}
\end{aligned}
\tag{1}
$$

- $f(\mathbf{v}_i, \mathbf{u}_j)$ differentiable function, and $x_{ij} = 1$ if component i of X assigned to component j of Y_i, 0 otherwise.
- we defined $f(\mathbf{v}_i, \mathbf{u}_j) = \text{ReLU}(\mathbf{v}_i^\top \mathbf{u}_j)$.

Repset architecture - Learning and output

- Given input set X and the m hidden sets $Y_1, Y_2, \ldots, Y_m$, formulate m bipartite matching problems,
- solving we end up with an m-dimensional vector $\mathbf{v}_X$: hidden representation of set X.
- This m-dimensional vector can be used as features for different machine learning tasks such as *set regression* or *set classification*. For instance, in the case of a set classification problem with $|\mathcal{C}|$ classes, the output is computed as follows:

$$\mathbf{p}_X = \text{softmax}(\mathbf{W}^{(c)}\,\mathbf{v}_X + \mathbf{b}^{(c)}) \tag{2}$$

where $\mathbf{W}^{(c)} \in \mathbb{R}^{m \times |\mathcal{C}|}$ is a matrix of trainable parameters and $\mathbf{b}^{(c)} \in \mathbb{R}^{|\mathcal{C}|}$ is the bias term. We use the negative log likelihood of the correct labels as training loss:

$$L = -\sum_{X} \log \mathbf{p}_{X_i} \tag{3}$$

where i is the class label of set X. Note that we can create a deeper architecture by adding more fully-connected layers.

Repset architecture - Learning and output

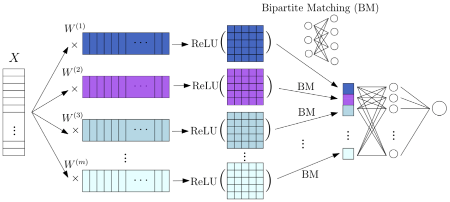
- Given input set X and the m hidden sets $Y_1, Y_2, \ldots, Y_m$, formulate m bipartite matching problems,
- end up with an m-dimensional vector $\mathbf{v}_X$: hidden representation of set X. Can be used as features for different machine learning tasks such as *set regression* or *set classification*. For set classification with $|\mathcal{C}|$ classes, the output is computed as:

$$\mathbf{p}_X = \text{softmax}(\mathbf{W}^{(c)}\,\mathbf{v}_X + \mathbf{b}^{(c)}) \tag{4}$$

We use the negative log likelihood of the correct labels as training loss:

$$L = -\sum_X \log \mathbf{p}_{X_i} \tag{5}$$

where i is the class label of set X.
* The architecture supports permutation invariance (proof in the paper)



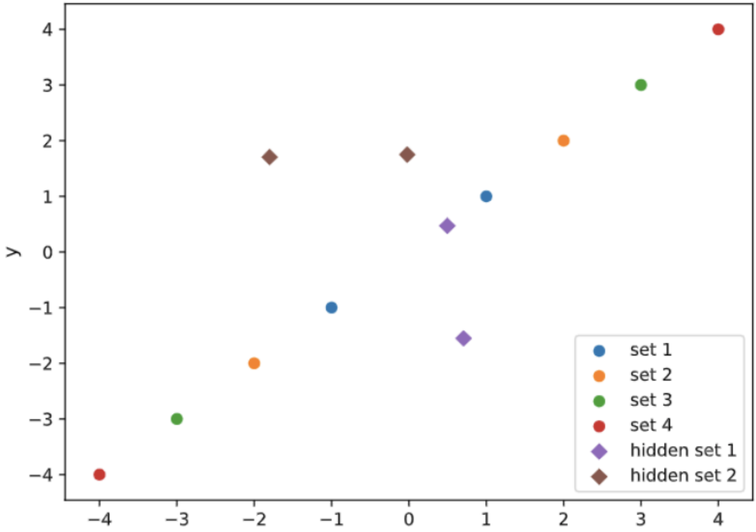
Repset architecture - Tackling the complexity of the bipartite matching

- major weakness the computational complexity: maximum cardinality matching in a weighted bipartite graph with n vertices and m edges takes time $\mathcal{O}(mn + n^2 \log n)$, with the classical Hungarian algorithm.
- Prohibitive for very large datasets.
  ApproxRepSet: approximation of bipartite matching problem involving operations that can be performed on a GPU
- Assuming an input set of vectors, $X = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{|X|}\}$ and a hidden set $Y = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{|Y|}\}$. Assume $|X| \geq |Y|$, optimization becomes:

$$
\begin{aligned}
&\max \sum_{i=1}^{|X|} \sum_{j=1}^{|Y|} x_{ij} f(\mathbf{v}_i, \mathbf{u}_j) \\
&\text{subject to:} \\
&\sum_{i=1}^{|X|} x_{ij} \leq 1 \quad \forall j \in \{1, \ldots, |Y|\} \\
&x_{ij} \geq 0 \quad \forall i \in \{1, \ldots, |X|\}, \forall j \in \{1, \ldots, |Y|\}
\end{aligned} \tag{6}
$$

- relaxed formulation of the problem - constraint has been removed.

Repset - Experimental Evaluation - Synthetic Data



Repset - Experimental Evaluation - Text Categorization

Dataset	n	Voc	Unique Words(avg)	y
BBCSPORT	517	13243	117	5
TWITTER	2176	6344	9.9	3
RECIPE	3059	5708	48.5	15
OHSUMED	3999	31789	59.2	10
CLASSIC	4965	24277	38.6	4
REUTERS	5485	22425	37.1	8
AMAZON	5600	42063	45.0	4
20NG	11293	29671	72	20

Repset - Experimental Evaluation - Text Categorization

	BBCSPORT	TWITTER	RECIPE	OHSUMED	CLASSIC	REUTERS	AMAZON	20NG
WMD	4.60 ± 0.70	28.70 ± 0.60	42.60 ± 0.30	44.50	$\mathbf{2.88} \pm 0.10$	3.50	7.40 ± 0.30	26.80
S-WMD	2.10 ± 0.50	27.50 ± 0.50	39.20 ± 0.30	34.30	3.20 ± 0.20	3.20	5.80 ± 0.10	26.80
DeepSets	25.45 ± 20.1	29.66 ± 1.62	70.25 ± 0.00	71.53	5.95 ± 1.50	10.00	8.58 ± 0.67	38.88
NN-mean	10.09 ± 2.62	31.56 ± 1.53	64.30 ± 7.30	45.37	5.35 ± 0.75	11.37	13.66 ± 3.16	38.40
NN-max	2.18 ± 1.75	30.27 ± 1.26	43.47 ± 1.05	35.88	4.21 ± 0.11	4.33	7.55 ± 0.63	32.15
NN-attention	4.72 ± 0.97	29.09 ± 0.62	43.18 ± 1.22	**31.36**	4.42 ± 0.73	3.97	6.92 ± 0.51	28.73
RepSet	$\mathbf{2.00} \pm 0.89$	$\mathbf{25.42} \pm 1.10$	$\mathbf{38.57} \pm 0.83$	33.88	3.38 ± 0.50	3.15	$\mathbf{5.29} \pm 0.28$	**22.98**
ApproxRepSet	4.27 ± 1.73	27.40 ± 1.95	40.94 ± 0.40	35.94	3.76 ± 0.45	**2.83**	5.69 ± 0.40	23.82

Classification test error of the proposed architecture and the baselines on the 8 text categorization datasets.

Repset - Experimental Evaluation - Text Set Extension

Hidden set	Terms similar to elements of hidden sets	Terms similar to centroids of hidden sets
1	chelsea, football, striker, club, champions	footballing
2	qualify, madrid, arsenal, striker, united, france	ARSENAL_Wenger
3	olympic, athlete, olympics, sport, pentathlon	Olympic_Medalist
4	penalty, cup, rugby, coach, goal	rugby
5	match, playing, batsman, batting, striker	batsman

Terms of the employed pre-trained model that are most similar to the elements and centroids of elements of5 hidden sets

Repset - Experimental Evaluation - Graph classification

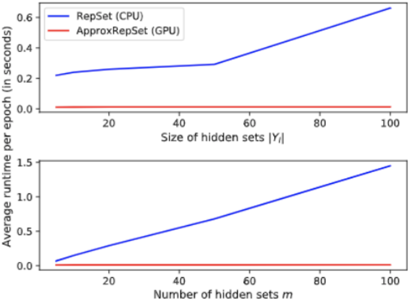
Dataset	#Graphs	y	Nodes(avg)	Edges(avg)
MUTAG	188	2	17.93	19.79
PROTEINS	1113	2	39.06	72.82
IMDB BINARY	1000	2	19.77	96.53
IMDB MULTI	1500	3	13.00	65.94
REDDIT BINARY	2000	2	429.63	497.75

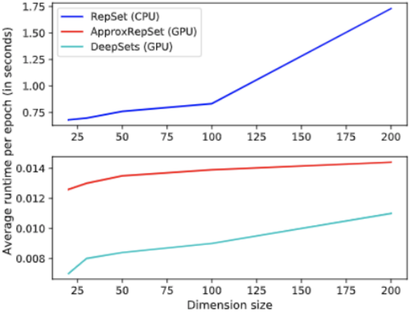
Repset - Experimental Evaluation - Graph Classification

	MUTAG	PROTEINS	IMDB BINARY	IMDB MULTI	REDDIT BINARY
PSCN $k = 10$	88.95 ($\pm$ 4.37)	75.00 ($\pm$ 2.51)	71.00 ($\pm$ 2.29)	45.23 ($\pm$ 2.84)	86.30 ($\pm$ 1.58)
Deep GR	82.66 ($\pm$ 1.45)	71.68 ($\pm$ 0.50)	66.96 ($\pm$ 0.56)	44.55 ($\pm$ 0.52)	78.04 ($\pm$ 0.39)
EMD	86.11 ($\pm$ 0.84)	-	-	-	-
DGCNN	85.80 ($\pm$ 1.70)	75.50 ($\pm$ 0.90)	70.03 ($\pm$ 0.86)	47.83 ($\pm$ 0.85)	-
SAEN	84.99 ($\pm$ 1.82)	75.31 ($\pm$ 0.70)	71.59 ($\pm$ 1.20)	48.53 ($\pm$ 0.76)	87.22 ($\pm$ 0.80)
RetGK	**90.30** ($\pm$ 1.10)	76.20 ($\pm$ 0.50)	72.30 ($\pm$ 0.60)	48.70 ($\pm$ 0.60)	**92.60** ($\pm$ 0.30)
DiffPool	-	**76.25**	-	-	-
DeepSets	86.26 ($\pm$ 1.09)	60.82 ($\pm$ 0.79)	69.84 ($\pm$ 0.64)	47.62 ($\pm$ 1.18)	52.01 ($\pm$ 1.47)
NN-mean	87.55 ($\pm$ 0.98)	73.00 ($\pm$ 1.21)	71.48 ($\pm$ 0.48)	49.92 ($\pm$ 0.82)	84.57 ($\pm$ 0.84)
NN-max	85.84 ($\pm$ 0.99)	71.05 ($\pm$ 0.54)	69.56 ($\pm$ 0.91)	48.28 ($\pm$ 0.43)	80.98 ($\pm$ 0.79)
NN-attention	85.92 ($\pm$ 1.16)	74.48 ($\pm$ 0.22)	**72.40** ($\pm$ 0.45)	49.56 ($\pm$ 0.47)	88.74 ($\pm$ 0.53)
RepSet	88.63 ($\pm$ 0.86)	73.04 ($\pm$ 0.42)	**72.40** ($\pm$ 0.73)	**49.93** ($\pm$ 0.60)	87.45 ($\pm$ 0.86)
ApproxRepSet	86.33 ($\pm$ 1.48)	70.74 ($\pm$ 0.85)	71.46 ($\pm$ 0.91)	48.92 ($\pm$ 0.28)	80.30 ($\pm$ 0.56)

Classification accuracy ($\pm$ standard deviation) of proposed architecture(s) and the baselines. For MU-TAG, PROTEINS (bioinformatics datasets) the node embeddings that we generated do not incorporateinformation about them.

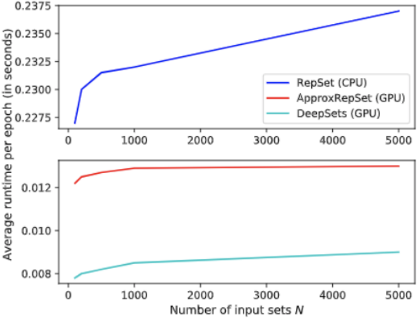
Repset - Experimental Evaluation - Runtimes

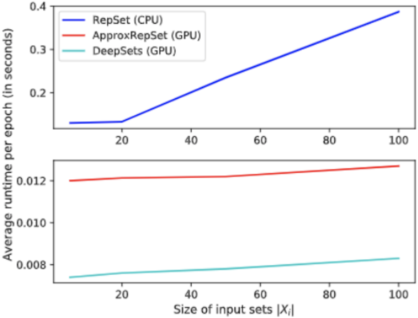




Runtimes with respect to the number of hidden setsm, the size of the hidden sets—Yi—(left)and embeddings with different dimensions (right).

Repset - Experimental Evaluation - Runtimes





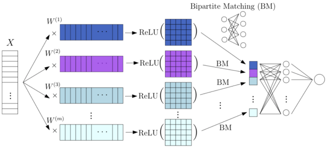
Runtimes with respect to the number of input setsN(left) and the size of the input sets—Xi—(right).

Repset - Conclusion

- Machine learning with sets is increasingly important
- Sets may vary in cardinality and their elements lack a meaningful ordering: standard machine learning algorithms fail to learn high-quality representations.

We proposed **RepSet**, a neural network approach for *learning set representations*.

- exhibits powerful permutation invariance properties.
- computes mappings between input sets and some hidden sets by solving a graph matching/network flow problems.
- Since matching/network flow algorithms are differentiable, we can use standard backpropagation for learning the parameters of the hidden sets.
- for large sets we introduced a relaxedversion (ApproxRepSet) - fast matrix operations and scales to very large datasets.
- Repsets performs favorably on text/ graph classification.



- **Future Work**

  : apply Repset on Group Recommentation (i.e. gaming)

THANK YOU !

Acknowledgements
Dr. I. Nikolentzos, Dr. K. Skianis, Dr. P. Meladianos

http://www.lix.polytechnique.fr/dascim/

Software and data sets:
http://www.lix.polytechnique.fr/dascim/software_datasets/
Repset preprint available at: https://arxiv.org/abs/1904.01962