Rep the Set: Neural Networks for Learning Set representations

K. Skianis, G. Nikolentzos, S. Limnios, M. Vaziriannis

Data Science and Mining group (DaSciM),
Laboratoire d’Informatique (LIX), École Polytechnique, France
http://www.lix.polytechnique.fr/dascim/ École Polytechnique, France

April 26, 2019
Data Science & Mining group
LIX @ Ecole Polytechnique

AI methods for large scale
Graph and Text data

M. Vazirgiannis
http://www.lix.polytechnique.fr/dascim/
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. Vaziriannis, 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]
- *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

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]

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
- **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 Learning 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}

- Similarity learning between sets should be *invariant to permutation*: challenging task
  
  - **supervised tasks**: set output label invariant or equivariant to the permutation of 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.
NNs for sets became very popular inspired 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 state of the art
Repset architecture - Permutation Invariance

- Assume an example $X$ represented as a set $X = \{v_1, v_2, \ldots, v_n\}$ of $d$-dimensional vectors, $v_i \in \mathbb{R}^d$. (i.e the embeddings of $X$’s elements)
- Objective: design architecture whose output is invariant for all $n!$ permutations of $X$ elements $\Rightarrow$ 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 $W^{(i)}$. 

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 $Y_i$.

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, $X = \{v_1, v_2, \ldots, v_{|X|}\}$ and a hidden set $Y = \{u_1, u_2, \ldots, u_{|Y|}\}$, the bipartite matching between the elements of the two sets is solving the optimization problem:

$$\sum_{i=1}^{|X|} \sum_{j=1}^{|Y|} x_{ij} f(v_i, u_j)$$

subject to:

1. $|X| \sum_{i=1}^{|X|} x_{ij} \leq 1 \quad \forall j \in \{1, \ldots, |Y|\}$

2. $|Y| \sum_{j=1}^{|Y|} x_{ij} \leq 1 \quad \forall i \in \{1, \ldots, |X|\}$

3. $x_{ij} \geq 0 \quad \forall i \in \{1, \ldots, |X|\}, \forall j \in \{1, \ldots, |Y|\}$

- $f(v_i, u_j)$ differentiable function, and $x_{ij} = 1$ if component $i$ of $X$ assigned to component $j$ of $Y$, 0 otherwise.
- we defined $f(v_i, u_j) = \text{ReLU}(v_i^\top u_j)$. 

K. Skianis, G. Nikolentzos, S. Limnios, M. Vazirgiannis Data Science and M

Rep the Set: Neural Networks for Learning Set representations
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 $v_X$: 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 $|C|$ classes, the output is computed as follows:

$$p_X = \text{softmax}(W^{(c)} v_X + b^{(c)})$$

(2)

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

$$L = -\sum_X \log p_{X_i}$$

(3)

where $i$ is the class label of set $X$. Note that we can create a deeper architecture by adding more fully-connected layers.
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 $v_X$: 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 $|C|$ classes, the output is computed as:

$$p_X = \text{softmax}(W^{(c)} v_X + b^{(c)})$$  \hspace{1cm} (4)

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

$$L = - \sum_X \log p_{X_i}$$  \hspace{1cm} (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 $O(mn + n^2 \log n)$, 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, $X = \{v_1, v_2, \ldots, v_{|X|}\}$ and a hidden set $Y = \{u_1, u_2, \ldots, u_{|Y|}\}$. Assume $|X| \geq |Y|$, optimization becomes:

$$
\max_{i=1}^{X} \sum_{j=1}^{Y} x_{ij} f(v_i, u_j)
$$

subject to:

$$
\sum_{i=1}^{X} x_{ij} \leq 1 \quad \forall j \in \{1, \ldots, |Y|\}
$$

$$
x_{ij} \geq 0 \quad \forall i \in \{1, \ldots, |X|\}, \forall j \in \{1, \ldots, |Y|\}
$$

- relaxed formulation of the problem - constraint has been removed.
Repset - Experimental Evaluation - Synthetic Data

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>n</th>
<th>Voc</th>
<th>Unique Words(avg)</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBCSPORT</td>
<td>517</td>
<td>13243</td>
<td>117</td>
<td>5</td>
</tr>
<tr>
<td>TWITTER</td>
<td>2176</td>
<td>6344</td>
<td>9.9</td>
<td>3</td>
</tr>
<tr>
<td>RECIPE</td>
<td>3059</td>
<td>5708</td>
<td>48.5</td>
<td>15</td>
</tr>
<tr>
<td>OHSUMED</td>
<td>3999</td>
<td>31789</td>
<td>59.2</td>
<td>10</td>
</tr>
<tr>
<td>CLASSIC</td>
<td>4965</td>
<td>24277</td>
<td>38.6</td>
<td>4</td>
</tr>
<tr>
<td>REUTERS</td>
<td>5485</td>
<td>22425</td>
<td>37.1</td>
<td>8</td>
</tr>
<tr>
<td>AMAZON</td>
<td>5600</td>
<td>42063</td>
<td>45.0</td>
<td>4</td>
</tr>
<tr>
<td>20NG</td>
<td>11293</td>
<td>29671</td>
<td>72</td>
<td>20</td>
</tr>
</tbody>
</table>
Classification test error of the proposed architecture and the baselines on the 8 text categorization datasets.
<table>
<thead>
<tr>
<th>Hidden set</th>
<th>Terms similar to elements of hidden sets</th>
<th>Terms similar to centroids of hidden sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>chelsea, football, striker, club, champions</td>
<td>footballing</td>
</tr>
<tr>
<td>2</td>
<td>qualify, madrid, arsenal, striker, united, france</td>
<td>ARSENAL_Wenger</td>
</tr>
<tr>
<td>3</td>
<td>olympic, athlete, olympics, sport, pentathlon</td>
<td>Olympic_Medalist</td>
</tr>
<tr>
<td>4</td>
<td>penalty, cup, rugby, coach, goal</td>
<td>rugby</td>
</tr>
<tr>
<td>5</td>
<td>match, playing, batsman, batting, striker</td>
<td>batsman</td>
</tr>
</tbody>
</table>

Terms of the employed pre-trained model that are most similar to the elements and centroids of elements of 5 hidden sets.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Graphs</th>
<th>$y$</th>
<th>Nodes(avg)</th>
<th>Edges(avg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUTAG</td>
<td>188</td>
<td>2</td>
<td>17.93</td>
<td>19.79</td>
</tr>
<tr>
<td>PROTEINS</td>
<td>1113</td>
<td>2</td>
<td>39.06</td>
<td>72.82</td>
</tr>
<tr>
<td>IMDB BINARY</td>
<td>1000</td>
<td>2</td>
<td>19.77</td>
<td>96.53</td>
</tr>
<tr>
<td>IMDB MULTI</td>
<td>1500</td>
<td>3</td>
<td>13.00</td>
<td>65.94</td>
</tr>
<tr>
<td>REDDIT BINARY</td>
<td>2000</td>
<td>2</td>
<td>429.63</td>
<td>497.75</td>
</tr>
</tbody>
</table>
Classification accuracy (± standard deviation) of proposed architecture(s) and the baselines. For MU-TAG, PROTEINS (bioinformatics datasets) the node embeddings that we generated do not incorporate information about them.
Runtimes with respect to the number of hidden sets $m$, the size of the hidden sets $|Y_i|$ (left) and embeddings with different dimensions (right).
Runtimes with respect to the number of input sets $N$ (left) and the size of the input sets $|X_i|$ (right).
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 relaxed version (ApproxRepSet) - fast matrix operations and scales to very large datasets.
- Repsets performs favorably on text/ graph classification.

**Future Work**

* apply Repset on Group Recommendation (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/