The multiscale Laplacian graph kernel
Risi Kondor and Horace Pan
The University of Chicago
Graph kernel $k(G_1, G_2)$
Basic requirements:

• Positive semi-definiteness
• Invariance to permuting the vertices
• Should capture a sensible notion of similarity
1. Local graph kernels

e.g., Graphlet kernels [Shervashidze et al., 2009]
1. Local graph kernels

e.g., Graphlet kernels [Shervashidze et al., 2009]
1. Local graph kernels

e.g., Graphlet kernels [Shervashidze et al., 2009]
2. Spectral graph kernels

\[ L = \begin{pmatrix} 
1 & -1 \\ 
-1 & 2 & -1 \\ 
-1 & 4 & -1 & -1 \\ 
-1 & 1 \\ 
-1 & 2 & -1 \\ 
-1 & -1 & 2 
\end{pmatrix} \]

[Gärtner, 2002] [Vishwanathan et al, 2010]
3. Propagation kernels

Weisfeiler—Lehmann kernels [Shervashidze et al., 2011]
Propagation kernels [Neumann et al., 2016]
3. Propagation kernels

Weisfeiler—Lehmann kernels [Shervashidze et al., 2011]
Propagation kernels [Neumann et al., 2016]
3. Propagation kernels

Weisfeiler—Lehmann kernels [Shervashidze et al., 2011]
Propagation kernels [Neumann et al., 2016]
3. Propagation kernels

Weisfeiler—Lehmann kernels [Shervashidze et al., 2011]
Propagation kernels [Neumann et al., 2016]
3. Propagation kernels

Weisfeiler—Lehmann kernels [Shervashidze et al., 2011]
Propagation kernels [Neumann et al., 2016]
3. Propagation kernels

Weisfeiler—Lehmann kernels [Shervashidze et al., 2011]
Propagation kernels [Neumann et al., 2016]
3. Propagation kernels

Weisfeiler—Lehmann kernels [Shervashidze et al., 2011]
Propagation kernels [Neumann et al., 2016]
3. Propagation kernels

Weisfeiler—Lehmann kernels [Shervashidze et al., 2011]
Propagation kernels [Neumann et al., 2016]
The Laplacian Graph Kernel
The graph ellipsoid
\[ k(G_1, G_2) \]
\[ k_{\text{LG}}(G_1, G_2) = \frac{\left| \left( \frac{1}{2} L_1 + \frac{1}{2} L_2 \right)^{-1} \right|^{1/2}}{\left| L_1^{-1} \right|^{1/4} \left| L_2^{-1} \right|^{1/4}} \]
\[ p(y) \propto e^{-y^\top (L + \eta I)y/2} \]
Bhattacharyya kernel:

$$k(p_1, p_2) = \int \sqrt{p_1(x)} \sqrt{p_2(x)} \, dx,$$

$$k_{LG}(\mathcal{G}_1, \mathcal{G}_2) = k(p_1, p_2) = \frac{|(\frac{1}{2} S_1^{-1} + \frac{1}{2} S_2^{-1})^{-1}|^{1/2}}{|S_1|^{1/4} |S_2|^{1/4}}$$

[K. and Jebara, 2003]
But the LG kernel is not relabeling invariant!
Transformation of variables:

\[ z := U y \quad \quad \Sigma_z = U \Sigma_y U^\top \]

\[ U_{*,i} \] are the "features" of node \( i \).
The Feature space Laplacian graph kernel (FLG kernel) is defined

$$k_{\text{FLG}}(G_1, G_2) = \left| \left( \frac{1}{2} S_1^{-1} + \frac{1}{2} S_2^{-1} \right)^{-1} \right|^{1/2},$$

where $S_1 = U_1 L_1^{-1} U_1^\top + \gamma I$ and $S_2 = U_2 L_2^{-1} U_2^\top + \gamma I$. 
The ellipsoid is now in feature space and combines information about the graph structure with the features. Don’t even need explicit features — can be induced from another kernel!
The Multiscale Laplacian Graph Kernel
1. Base features

Start with something simple like node degrees.
2. Small subgraphs...
3. Larger subgraphs...

and iterate...
Level 1 subgraphs
Level 2 subgraphs
Entire graph

Low rank approximation reduces overall complexity from $O(MLn^5)$ to $O(ML\tilde{n}^2\tilde{p}^3)$. 
Let $G$ be a graph with vertex set $V$, and $\kappa$ a positive semi-definite kernel on $V$. Assume that for each $v \in V$ we have a nested sequence of $L$ neighborhoods

$$v \in N_1(v) \subseteq N_2(v) \subseteq \ldots \subseteq N_L(v) \subseteq V,$$

and for each $N_\ell(v)$, let $G_\ell(v)$ be the corresponding induced subgraph of $G$.

We define the **Multiscale Laplacian Subgraph Kernels (MLS kernels)**, $\mathcal{K}_1, \ldots, \mathcal{K}_L : V \times V \to \mathbb{R}$ as follows:

1. $\mathcal{K}_1$ is just the FLG kernel $k_{FLG}^\kappa$ induced from the base kernel $\kappa$ between the lowest level subgraphs:

$$\mathcal{K}_1(v, v') = k_{FLG}^\kappa(G_1(v), G_1(v')).$$

2. For $\ell = 2, 3, \ldots, L$, $\mathcal{K}_\ell$ is the FLG kernel induced from $\mathcal{K}_{\ell-1}$ between $G_\ell(v)$ and $G_\ell(v')$:

$$\mathcal{K}_\ell(v, v') = k_{FLG}^{\mathcal{K}_{\ell-1}}(G_\ell(v), G_\ell(v')).$$
We define the Multiscale Laplacian Graph Kernel (MLG kernel) between any two graphs $G_1, G_2 \in \mathcal{G}$ as

$$\mathcal{K}(G_1, G_2) = k_{FLG}^{FLG}(G_1, G_2).$$
1. True multiscale/multiresolution graph kernel.
2. Combines information from subgraphs with relative position of subgraphs.
3. Invariant to relabeling.
4. Can compare graphs of different sizes.
5. “Smooth” w.r.t. perturbations.
<table>
<thead>
<tr>
<th>Method</th>
<th>MUTAG</th>
<th>PTC</th>
<th>ENZYMES</th>
<th>PROTEINS</th>
<th>NCI1</th>
<th>NCI109</th>
</tr>
</thead>
<tbody>
<tr>
<td>WL</td>
<td>84.50(±2.16)</td>
<td>59.97(±1.60)</td>
<td>53.75(±1.37)</td>
<td>75.43(±1.95)</td>
<td><strong>84.76(±0.32)</strong></td>
<td><strong>85.12(±0.29)</strong></td>
</tr>
<tr>
<td>WL-Edge</td>
<td>82.94(±2.33)</td>
<td>60.18(±2.19)</td>
<td>52.00(±0.72)</td>
<td>73.63(±2.12)</td>
<td><strong>84.65(±0.25)</strong></td>
<td><strong>85.32(±0.34)</strong></td>
</tr>
<tr>
<td>SP</td>
<td><strong>85.50(±2.50)</strong></td>
<td>59.53(±1.71)</td>
<td>42.31(±1.37)</td>
<td>75.61(±0.45)</td>
<td>73.61(±0.36)</td>
<td>73.23(±0.26)</td>
</tr>
<tr>
<td>Graphlet</td>
<td>82.44(±1.29)</td>
<td>55.88(±0.31)</td>
<td>30.95(±0.73)</td>
<td>71.63(±0.33)</td>
<td>62.40(±0.27)</td>
<td>62.35(±0.28)</td>
</tr>
<tr>
<td>p–RW</td>
<td>80.33(±1.35)</td>
<td>59.85(±0.95)</td>
<td>28.17(±0.76)</td>
<td>71.67(±0.78)</td>
<td>TIMED OUT</td>
<td>TIMED OUT</td>
</tr>
<tr>
<td>MLG</td>
<td>84.21(±2.61)</td>
<td><strong>63.62(±4.69)</strong></td>
<td><strong>57.92(±5.39)</strong></td>
<td><strong>76.14(±1.95)</strong></td>
<td>80.83(±1.29)</td>
<td>81.30(±0.80)</td>
</tr>
</tbody>
</table>

Code at [github.com/horacepan/MLGkernel](http://github.com/horacepan/MLGkernel)
Conclusions

- Truly multiscale kernel: subgraphs compared by comparing their constituent sub-subgraphs
- Ideas extend beyond just kernels world, e.g., hierarchical deep learning architectures; structure2vec [Dai, Dai & Song, 2016].

Support: DARPA D16AP00112 YFA “Multiresolution Machine Learning for Molecular Modeling”