From k-mers to gap-allowed k-mers modeling

Convolutional kernel networks [1] that model k-mers:
\[
K_{CKN}(x, x') = \sum_{k=1}^{\lfloor |x|/k \rfloor} \sum_{j=1}^{\lfloor |x'|/k \rfloor} K_0(x[i : i+k], x'[j : j+k])
\]

- \( K_0 \) is a Gaussian kernel over one-hot representations of k-mers.
- A natural feature map of \( x \) is \( \sum_{j=1}^{\lfloor |x'|/k \rfloor} \phi_0(x[j : j+k]) \) with \( \phi_0 \) the kernel mapping associated to \( K_0 \).
- Scalable and data or task-adaptive with Nyström approximation. Interpretable using end-to-end training with few filters.
- Unable to capture gappy motifs.

Recurrent kernel networks that generalize k-mers with gaps:
\[
K_{RKN}(x, x') = \sum_{k=1}^{\lfloor |x|/k \rfloor} \sum_{j=1}^{\lfloor |x'|/k \rfloor} \lambda_{k, j} \lambda_{k, j} K_0(x[i : i+k], x'[j : j+k])
\]

- Take gapped k-mers into account. \( \lambda_{k, j} \) penalizes the gaps, e.g. \( \lambda_{1, 1} = \lambda_{\text{gap}(0)} \).
- A nature feature map is \( \sum_{k=1}^{\lfloor |x|/k \rfloor} \lambda_{k, 1} \phi_0(x[i]) \).
- Computationally fast using dynamic programming.
- The gate components in RNNs play the same role as gap penalization in substring kernels.

Definition of gap-allowed k-mers

- For \( 1 \leq k \leq n \in \mathbb{N} \), denote by \( I(k, n) \) the set of indices of k-mers elements \( i = (i_1, \ldots, i_k) \), with \( 1 \leq i_1 < \cdots < i_k \leq n \).
- For a sequence \( x = x_1 \ldots x_n \in \mathcal{X} \) of length \( n \), for a sequence of indices \( i \in I(k, n) \), we define a k-substring as:
  \[
x[i] = x_{i_k}, \ldots, x_{i_1}.
  \]
- The length of the gaps in the substring is \( \text{gap}(i) = i_k - i_{k-1} - 1 \).

Nyström approximation and RNNs

Nyström approximation:

Fast computation with dynamic programming:

For any \( j \in \{1, \ldots, k\} \) and \( t \in \{1, \ldots, |x|\} \),
\[
\psi(x) = \sum_{k=1}^{\lfloor |x|/k \rfloor} \lambda_{k, t} K_0(x[i]) = K_{\text{RKN}} \sum_{k=1}^{\lfloor |x|/k \rfloor} \lambda_{k, t} K_0(x[i]).
\]

Protein fold recognition on SCOP 1.67

Max pooling in RKHS and extensions

- The sum can be replaced by a max, the corresponding recursive equations can be obtained by replacing all the sum with max.
- Generalized max pooling (GMP): build a representation \( \phi_{\text{GMP}} \) such that \( \phi_{\text{GMP}}(x) = 1 \) for a set of features \( \{\phi_1, \ldots, \phi_q\} \) in \( \mathbb{R}^q \).
- Multilayer extension and link with string kernels can be found in [1].

Recurrent Kernel Networks

Dexiong Chen † Laurent Jacob † Julien Mairal †

† Inria - firstname.lastname@inria.fr † CNRS - firstname.lastname@univ-lyon1.fr

Overview

Kernel supervised learning for sequence objects
\[
\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) + \frac{\mu}{2} \|f\|_2^2
\]

- \( x_1, \ldots, x_n \in \mathcal{X} \) are sequences (biological sequences or texts).
- Goal: learning a predictive and interpretable function \( f \).

A feature map of RKN

- A feature vector of RKN for \( x \) is a mixture of Gaussians centered at \( x[i] \), weighted by the corresponding \( \lambda_{k, j} \).

Learning strategies

The supervised learning problem becomes
\[
\min_{\psi \in \mathbb{R}^q} \sum_{i=1}^{n} L(\psi_k(x_i), w^y_i) + \frac{\mu}{2} \|w\|^2,
\]
where \( \psi_k \) depends on \( Z \). The model can be trained in 2 ways:
- Unsupervised: learning \( Z \) with K-means using (subsampled) k-mers (eventually with gaps). Then train a linear classifier.
- Supervised: jointly learning \( Z \) and \( w \) with SGD.

Experiments

Protein fold recognition on SCOP 1.67

Method pooling on one-hot BLOSUM62

<table>
<thead>
<tr>
<th>Method</th>
<th>auroC50</th>
<th>auroC50</th>
<th>auroC50</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA-kernel</td>
<td>--</td>
<td>--</td>
<td>0.834</td>
</tr>
<tr>
<td>LSTM</td>
<td>0.830</td>
<td>0.566</td>
<td>--</td>
</tr>
<tr>
<td>CKN [1]</td>
<td>0.837</td>
<td>0.572</td>
<td>0.866</td>
</tr>
<tr>
<td>RKN</td>
<td>mean</td>
<td>0.829</td>
<td>0.541</td>
</tr>
<tr>
<td>RKN</td>
<td>max</td>
<td>0.844</td>
<td>0.587</td>
</tr>
<tr>
<td>RKN</td>
<td>GMP</td>
<td>0.848</td>
<td>0.570</td>
</tr>
<tr>
<td>RKN (unemp)</td>
<td>mean</td>
<td>0.805</td>
<td>0.504</td>
</tr>
</tbody>
</table>

Protein fold classification on SCOP 2.06

Method params Accuracy Level-stratified accuracy (top1/top5) family superfamily fold

<table>
<thead>
<tr>
<th>Method</th>
<th>Params</th>
<th>Accuracy</th>
<th>Level-stratified accuracy (top1/top5)</th>
<th>family</th>
<th>superfamily</th>
<th>fold</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSI-BLAST</td>
<td>-</td>
<td>84.53</td>
<td>86.84</td>
<td>82.20/84.50</td>
<td>86.90/86.40</td>
<td>18.90/33.10</td>
</tr>
<tr>
<td>DeepSF</td>
<td>920k</td>
<td>73.00</td>
<td>920.95</td>
<td>72.23/90.08</td>
<td>51.35/67.57</td>
<td></td>
</tr>
<tr>
<td>CKN (512 filters)</td>
<td>843k</td>
<td>82.33</td>
<td>94.20</td>
<td>45.41/69.19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RKN (512 filters)</td>
<td>843k</td>
<td>85.29</td>
<td>94.95</td>
<td>85.99/92.52</td>
<td>71.35/84.86</td>
<td></td>
</tr>
</tbody>
</table>

Relevant reference