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

$$\min_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)) + \frac{\mu}{2}\|f\|_{\mathcal{H}}^2$$

- $\mathbf{x}_1, \dots \mathbf{x}_n \in \mathcal{X}$ are sequences (biological sequences or texts).

Goal: learning a predictive and interpretable function $f : \mathcal{X} \to \mathcal{Y}$.

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

Convolutional kernel networks [1] that model k-mers:

$$K_{\text{CKN}}(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{|\mathbf{x}|}\sum_{j=1}^{|\mathbf{x}'|} K_0\left(\mathbf{x}[i:i+k], \mathbf{x}'[j:j+k]\right)$$

- K_0 is a Gaussian kernel over one-hot representations of k-mers.
- A continuous relaxation of the mismatch kernel.
- $\varphi(\mathbf{x}) := \sum_{j=1}^{|\mathbf{x}|} \varphi_0(\mathbf{x}[j:j+k])$ with φ_0 the kernel mapping associated to K_0.
- Scalable and task-adaptive with Nyström approximation. Interpretable using end-to-end training with few filters.
- Limitation: unable to capture gapped motifs.

Recurrent kernel networks that generalize k-mers with gaps:

$$K_{\text{RKN}}(\mathbf{x}, \mathbf{x}') = \sum_{\mathbf{i}\in\mathcal{I}(k,|\mathbf{x}|)}\sum_{\mathbf{j}\in\mathcal{I}(k,|\mathbf{x}'|)} \lambda^{\text{gaps}(\mathbf{i})}\lambda^{\text{gaps}(\mathbf{j})} K_0(\mathbf{x}[\mathbf{i}], \mathbf{x}'[\mathbf{j}])$$

- Take gapped k-mers into account. $\lambda^{\text{gaps}(\mathbf{i})}$ penalizes the gaps.
- $\varphi(\mathbf{x}) = \sum_{\mathbf{i}\in\mathcal{I}(k,|\mathbf{x}|)} \lambda^{\text{gaps}(\mathbf{i})}\varphi_0(\mathbf{x}[\mathbf{i}])$.
- Computationally fast using dynamic programming.
- Leads to a particular RNN with a kernel interpretation.

Definition of gap-allowed k-mers

- For $1 \le k \le n \in \mathbb{N}$, we denote by $\mathcal{I}(k,n)$ the set of sequences of indices with k elements $\mathbf{i} = (i_1, \dots, i_k)$, with $1 \le i_1 < \cdots < i_k \le n$.
- For a sequence $\mathbf{x} = x_1 \dots x_n \in \mathcal{X}$ of length n, for a sequence of indices $\mathbf{i} \in \mathcal{I}(k,n)$, we define a k-substring as:

$$\mathbf{x}[\mathbf{i}] = x_{i_1}x_{i_2}\dots x_{i_k}.$$

- The length of the gaps in the substring is

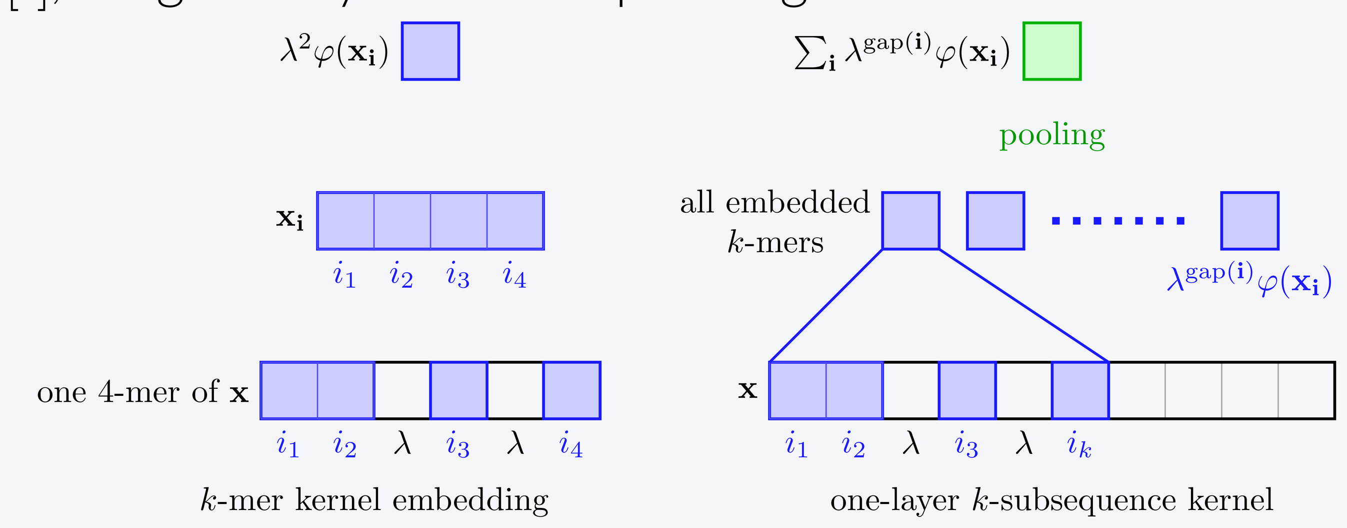
$$\text{gaps}(\mathbf{i}) = \text{number of gaps in the substring indices.}$$

- Example: $\mathbf{x} =$ BAARACADACRB

$$\mathbf{i} = (4,5,8,9,11) \quad \mathbf{x}[\mathbf{i}] = \text{RADAR} \quad \text{gaps}(\mathbf{i}) = 3$$

A feature map of RKN

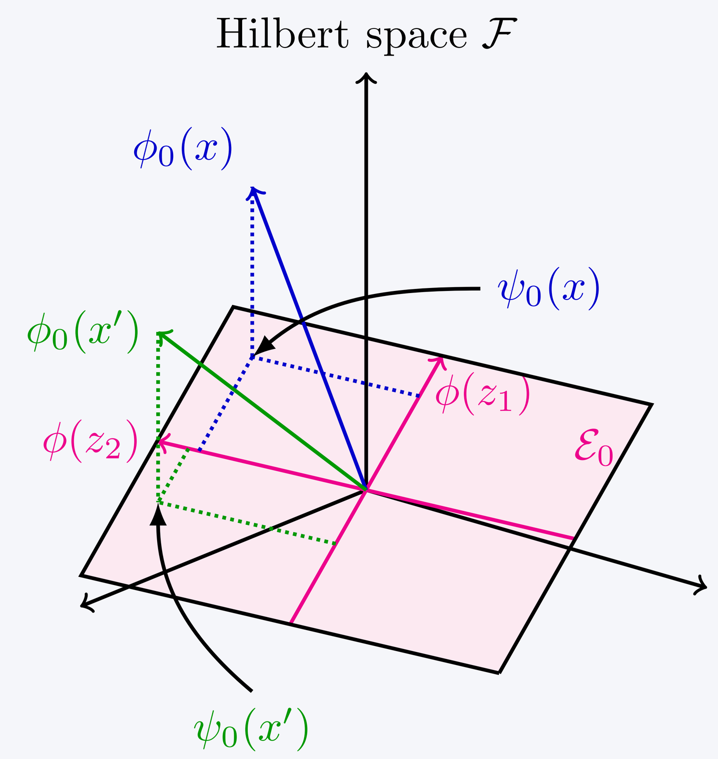
- A feature vector of RKN for x is a mixture of Gaussians centered at $x[\mathbf{i}]$, weighted by the corresponding $\lambda^{\text{gaps}(\mathbf{i})}$.



Nyström approximation and RNNs

Nyström approximation:

$\mathcal{E}_0 = \text{span}(\varphi_0(z_1), \dots, \varphi_0(z_q))$
$\psi_0(x) := K_{ZZ}^{-\frac{1}{2}}K_Z(x)$
where $[K_{ZZ}]_{ij} = K_0(z_i, z_j)$
and $[K_Z(x)]_i = K_0(z_i, x)$.



Finite-dimensional projection of the kernel map: given a set of anchor points $Z = (z_1, \dots, z_q)$ with $z_i \in \mathbb{R}^{k\times d}$, we project $\varphi_0(x)$ orthogonally onto $\mathcal{E}_0$ such that $K_0(x,x') \approx \langle\psi_0(x), \psi_0(x')\rangle_{\mathbb{R}^q}$. An approximate feature map for K_{RKN} is

$$\psi_k(\mathbf{x}) = \sum_{\mathbf{i}\in\mathcal{I}(k,|\mathbf{x}|)} \lambda^{\text{gaps}(\mathbf{i})}\psi_0(\mathbf{x}[\mathbf{i}]) = K_{ZZ}^{-\frac{1}{2}} \sum_{\mathbf{i}\in\mathcal{I}(k,|\mathbf{x}|)} \lambda^{\text{gaps}(\mathbf{i})} K_Z(\mathbf{x}[\mathbf{i}]) \in \mathbb{R}^q.$$

Fast computation with dynamic programming:

For any $j \in \{1, \dots, k\}$ and $t \in \{1, \dots, |\mathbf{x}|\}$,

$$\psi_j(\mathbf{x}_{1:t}) = K_{Z_jZ_j}^{-\frac{1}{2}}\mathbf{h}_j[t]$$

where $\mathbf{c}_j[t]$ and $\mathbf{h}_j[t]$ in $\mathbb{R}^q$ obeying the recursion

$$\begin{aligned}
\mathbf{c}_j[1] &= \mathbf{h}_j[1] = 0 & 1 \le j \le k,\\
\mathbf{c}_0[t] &= 1 & 1 \le t \le |\mathbf{x}|,\\
\mathbf{c}_j[t] &= \lambda\mathbf{c}_j[t-1] + \mathbf{c}_{j-1}[t-1] \odot \kappa(Z_j\mathbf{x}_t) & 1 \le j \le k,\\
\mathbf{h}_j[t] &= \mathbf{h}_j[t-1] + \mathbf{c}_{j-1}[t-1] \odot \kappa(Z_j\mathbf{x}_t) & 1 \le j \le k,
\end{aligned} \qquad (1)$$

where κ is a non-linear function $\kappa(x) = e^{\alpha(x-1)}$ and Z_j is a matrix in $\mathbb{R}^{qd}$ whose i-th row is the j-th row of z_i.

Learning strategies

The supervised learning problem becomes

$$\min_{\mathbf{w}\in\mathbb{R}^q} \frac{1}{n}\sum_{i=1}^{n} L\left(\langle\psi_k(\mathbf{x}_i), \mathbf{w}\rangle, y_i\right) + \frac{\mu}{2}\|\mathbf{w}\|^2, \qquad (2)$$

where ψ_k depends on Z. The model can be trained in 2 ways:

- **Unsupervised:** learning Z with K-means over (subsampled) k-mers (eventually with gaps). Then train a linear classifier.
- **Supervised:** jointly learning Z and $\mathbf{w}$ with SGD.

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 φ_{gmp} such that $\langle\varphi_{\text{gmp}}, \varphi_i\rangle_{\mathcal{H}} = 1$ for a set of features $(\varphi_1, \dots, \varphi_N)$ in $\mathcal{H}^N$.
- Multilayer extension and link with string kernels in [1].

Experiments

Protein fold recognition on SCOP 1.67

Method	pooling	one-hot auROC	one-hot auROC50	BLOSUM62 auROC	BLOSUM62 auROC50
LA-kernel		--	--	0.834	0.504
LSTM		0.830	0.566	--	--
CKN [1]		0.837	0.572	0.866	0.621
RKN	mean	0.829	0.541	0.840	0.571
RKN	max	0.844	**0.587**	**0.871**	**0.629**
RKN	GMP	**0.848**	0.570	0.852	0.609
RKN (unsup)	mean	0.805	0.504	0.833	0.570

Protein fold classification on SCOP 2.06

Method	♯Params	Accuracy top 1	Accuracy top 5	Level-stratified accuracy (top1/top5) family	superfamily	fold
PSI-BLAST	-	84.53	86.48	82.20/84.50	86.90/88.40	18.90/35.100
DeepSF	920k	73.00	90.25	75.87/91.77	72.23/90.08	51.35/67.57
CKN (512 filters)	843k	84.11	94.29	**90.24**/**95.77**	82.33/94.20	45.41/69.19
RKN (512 filters)	843k	**85.29**	**94.95**	84.31/94.80	**85.99**/**95.22**	**71.35**/**84.86**

Relevant reference

[1] D. Chen, L. Jacob, and J. Mairal. Biological sequence modeling with convolutional kernel networks. *Bioinformatics*, 35(18):3294--3302, 02 2019.