Deep Sketched Output Kernel Regression for Structured Prediction

Tamim El Ahmad* [1] Junjie Yang* [1] Pierre Laforgue[2] Florence d'Alché-Buc [1]

[1]LTCI, Télécom Paris, Institut Polytechnique de Paris, France [2]Department of Computer Science, University of Milan, Italy

Motivation

Problem: learn a mapping $f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y}$ is a structured space (e.g. graphs, rankings)

Existing works: $\hat{f} = d \circ \hat{h}$: 2-step surrogate method based on input/output kernels [3, 1, 2]

Advantage: versatility (i.e., able to handle different output types within a unified framework)

Drawback: lack of expressiveness (i.e., not able to handle complex inputs such as texts)

Goal: Build a versatile and expressive estimator

Some Notations

- $k : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ p.d. kernel, $\mathcal{H}$ its RKHS, $\psi(y) := k(\cdot, y) \in \mathcal{H}$ relevant representation of the outputs
- $m \ll n$, $R \in \mathbb{R}^{m \times n}$ sketching matrix, i.e. randomly drawn matrix (e.g. sub-sampling, Gaussian)
- $K = (k(y_i, y_j))_{1 \leq i,j \leq n} \in \mathbb{R}^{n \times n}$, $\widetilde{K} = RKR^\top \in \mathbb{R}^{m \times m}$ and $\left\{\left(\sigma_i(\widetilde{K}), \tilde{\mathbf{v}}_i\right), i \in [m]\right\}$ its eigenpairs
- $\widehat{\mathcal{H}} = \operatorname{span}((\psi(y_i))_{i=1}^n)$, $\widehat{\mathrm{C}} = (1/n)\sum_{i=1}^n \psi(y_i) \otimes \psi(y_i) \in \widehat{\mathcal{H}}^{\mathcal{H}}$ empirical covariance operator
- $\widetilde{\mathcal{H}} = \operatorname{span}((\sum_{j=1}^n \mathrm{R_{ij}}\, \psi(y_j))_{i=1}^{\mathrm{m}})$, $\widetilde{\mathrm{C}} = \frac{1}{n}\sum_{l=1}^m \left(\sum_{i=1}^n R_{li}\psi(y_i)\right) \otimes \left(\sum_{j=1}^n R_{lj}\psi(y_j)\right) \in \widetilde{\mathcal{H}}^{\mathcal{H}}$

Output Kernel Regression: a surrogate approach

Kernel-induced loss: $\Delta(y, y') := \|\psi(y) - \psi(y')\|_{\mathcal{H}}^2 = \mathrm{k}(y,y) - 2\,\mathrm{k}(y,y') + \mathrm{k}(y',y')$

Goal: for $f_\theta : \mathcal{X} \to \mathcal{Y}$ a Deep Neural Network ($\theta \in \Theta$ denotes its weights), solve

$$\min_{\theta \in \Theta} \frac{1}{n}\sum_{i=1}^n \|\psi(f_\theta(x_i)) - \psi(y_i)\|_{\mathcal{H}}^2 \tag{1}$$

How: let $h_\theta : \mathcal{X} \to \mathcal{H}$ be a DNN, 2-step surrogate method:

1. $\hat{\theta} = \underset{\theta \in \Theta}{\arg\min}\ \frac{1}{n}\sum_{i=1}^n \|h_\theta(x_i) - \psi(y_i)\|_{\mathcal{H}}^2$ (training step)
2. $f_{\hat{\theta}}(x) = d \circ h_{\hat{\theta}}(x) = \underset{y \in \mathcal{Y}}{\arg\min}\ \|h_{\hat{\theta}}(x) - \psi(y)\|_{\mathcal{H}}^2$ (inference step)

Problem: what if $\psi(y)$ is infinite-dimensional or implicit?

Deep Sketched Output Kernel Regression

Solution: consider an orthonormal basis $\widetilde{E} = ((\tilde{e}_i)_{i=1}^p)$ of a p-dimensional subspace of $\mathcal{H}$, where $p \in \mathbb{N}$ is small, and for a DNN $g_W : \mathcal{X} \to \mathbb{R}^p$ (W its weights),

$$h_\theta(x) := g_{\widetilde{E}} \circ g_W(x) = \sum_{j=1}^p g_W(x)_j \tilde{e}_j$$

How to build the basis $\widetilde{E}$?

Let $p = \operatorname{rank}\left(\widetilde{K}\right)$, and $\forall\, 1 \leq i \leq p$, $\tilde{e}_i = \sqrt{\frac{n}{\sigma_i(\widetilde{K})}}\sum_{j=1}^n [R^\top \tilde{\mathbf{v}}_i]_j \psi(y_j) \in \mathcal{H}$.

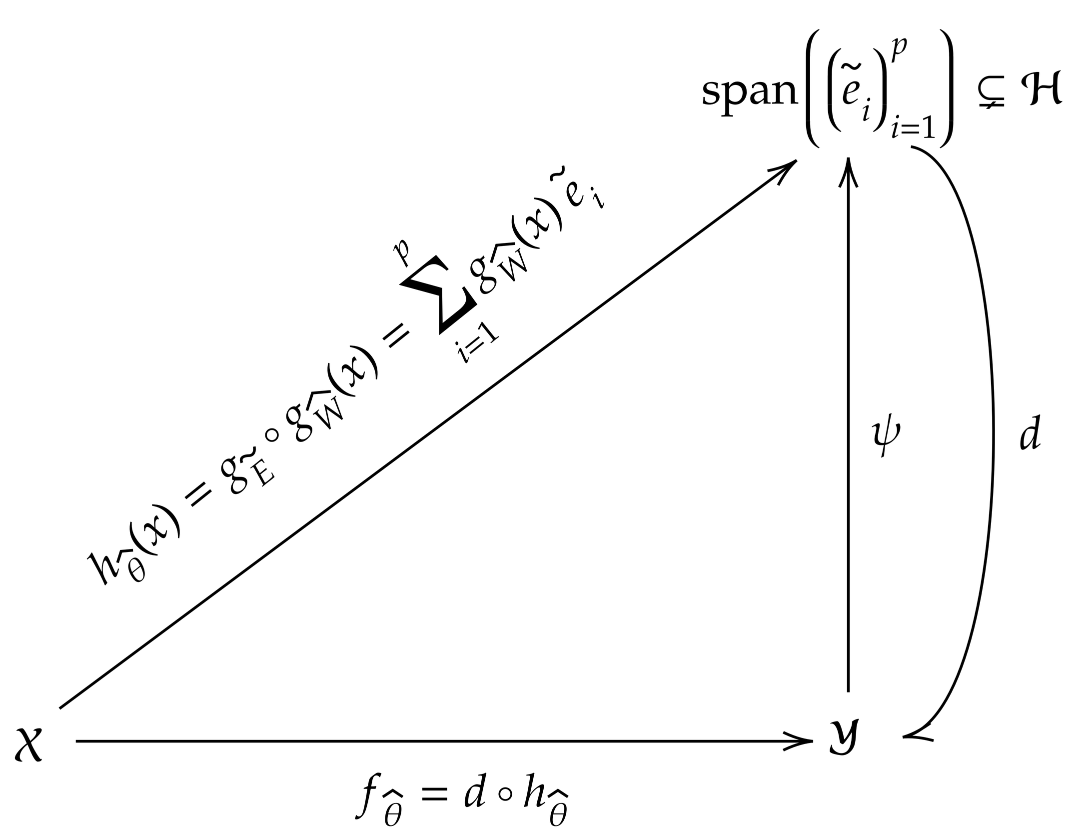
Proposition. The $\tilde{e}_i$s are the eigenfunctions, associated to the eigenvalues $\sigma_i(\widetilde{K})/n$, of $\widetilde{C}$ whose range is $\widetilde{\mathcal{H}}$. Then, $\widetilde{E} = ((\tilde{e}_i)_{i=1}^p)$ is an orthonormal basis of $\widetilde{\mathcal{H}}$.

How to solve the surrogate problem and learn the weights W?

Proposition. Let $\widetilde{E} = ((\tilde{e}_i)_{i=1}^p)$ and $h_\theta = g_{\widetilde{E}} \circ g_W$. Then

$$\frac{1}{n}\sum_{i=1}^n \|h_\theta(x_i) - \psi(y_i)\|_{\mathcal{H}}^2 = \frac{1}{n}\sum_{i=1}^n \left\|g_W(x_i) - \tilde{\psi}(y_i)\right\|_2^2,$$

where $\tilde{\psi}(y) = (\tilde{e}_1(y), \ldots, \tilde{e}_{\mathrm{p}}(y))^\top = \widetilde{D}_{\mathrm{p}}^{-1/2}\widetilde{V}_{\mathrm{p}}^\top \mathrm{R}\,\mathrm{k}^y \in \mathbb{R}^{\mathrm{p}}$, $\widetilde{D}_{\mathrm{p}} \in \mathbb{R}^{\mathrm{p} \times \mathrm{p}}$ and $\widetilde{V}_{\mathrm{p}} \in \mathbb{R}^{\mathrm{m} \times \mathrm{p}}$ are such that $\widetilde{V}_{\mathrm{p}}\widetilde{D}_{\mathrm{p}}\widetilde{V}_{\mathrm{p}}^\top = \widetilde{\mathrm{K}}$ (SVD of $\widetilde{\mathrm{K}}$), and $\mathrm{k}^y = (\mathrm{k}(y, y_1), \ldots, \mathrm{k}(y, y_n))$.



Algorithm

1. Training. a. Computations for the basis $\widetilde{E}$.
   - Construct $\widetilde{D}_{\mathrm{p}} \in \mathbb{R}^{\mathrm{p} \times \mathrm{p}}$, $\widetilde{V}_{\mathrm{p}} \in \mathbb{R}^{\mathrm{m} \times \mathrm{p}}$ such that $\widetilde{V}_{\mathrm{p}}\widetilde{D}_{\mathrm{p}}\widetilde{V}_{\mathrm{p}}^\top = \widetilde{\mathrm{K}}$ (SVD of $\widetilde{\mathrm{K}}$)
   - $\widetilde{\Omega} = \widetilde{D}_{\mathrm{p}}^{-1/2}\widetilde{V}_{\mathrm{p}}^\top \in \mathbb{R}^{\mathrm{p} \times \mathrm{m}}$

1. Training. b. Solving the surrogate problem.
   - $\tilde{\psi}(y_i) = \widetilde{\Omega}\,\mathrm{R}\,k^{y_i} \in \mathbb{R}^{\mathrm{p}}, \forall\, 1 \leq i \leq n$, $\tilde{\psi}(y_i^{\mathrm{val}}) = \widetilde{\Omega}\,\mathrm{R}\,k^{y_i^{\mathrm{val}}} \in \mathbb{R}^{\mathrm{p}}, \forall\, 1 \leq i \leq n_{\mathrm{val}}$
   - $\hat{W} = \underset{W \in \mathcal{W}}{\arg\min}\ \frac{1}{n}\sum_{i=1}^n \left\|g_W(x_i) - \tilde{\psi}(y_i)\right\|_2^2$ (training of g_W with training $\{(x_i, \tilde{\psi}(y_i))\}_{i=1}^n$ and validation $\{(x_i^{\mathrm{val}}, \tilde{\psi}(y_i^{\mathrm{val}}))\}_{i=1}^{n_{\mathrm{val}}}$ pairs and Mean Squared Error loss)

2. Inference.
   - $\tilde{\psi}(y_i^{\mathrm{c}}) = \widetilde{\Omega}\,\mathrm{R}\,k^{y_i^{\mathrm{c}}} \in \mathbb{R}^{\mathrm{p}}, \forall\, 1 \leq i \leq n_{\mathrm{c}}$
   - $f_{\hat{\theta}}(x_i^{\mathrm{te}}) = y_j^{\mathrm{c}}$ where $j = \underset{1 \leq j \leq n_{\mathrm{c}}}{\arg\max}\ g_{\hat{W}}(x_i^{\mathrm{te}})^\top \tilde{\psi}(y_j^{\mathrm{c}}), \forall\, 1 \leq i \leq n_{\mathrm{te}}$

Sketching Size Selection Strategy

Goal: Set the minimal value of m s.t. it captures the information contained in $\widehat{\mathrm{C}}$

Solutions:

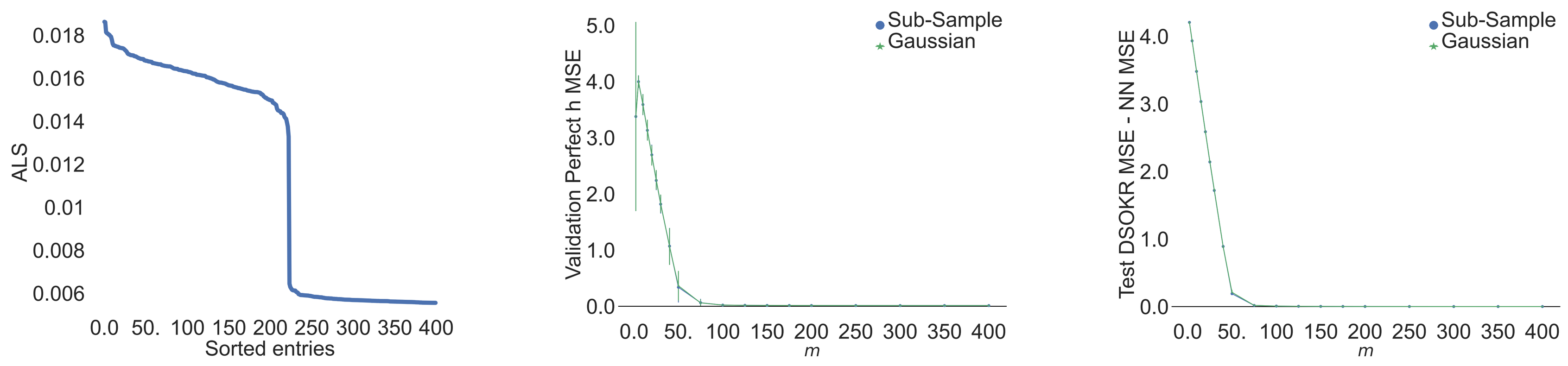
- Approximate leverage scores of $\widehat{\mathrm{C}}$
- Set the optimal m according to the performance of the perfect h estimator on the validation set,

$$h : (x, y) \mapsto \sum_{j=1}^{\mathrm{p}} \langle \tilde{e}_j, \psi(y) \rangle_{\mathcal{H}}\, \tilde{e}_j = \sum_{j=1}^{\mathrm{p}} \tilde{\psi}(y)_j\, \tilde{e}_j$$

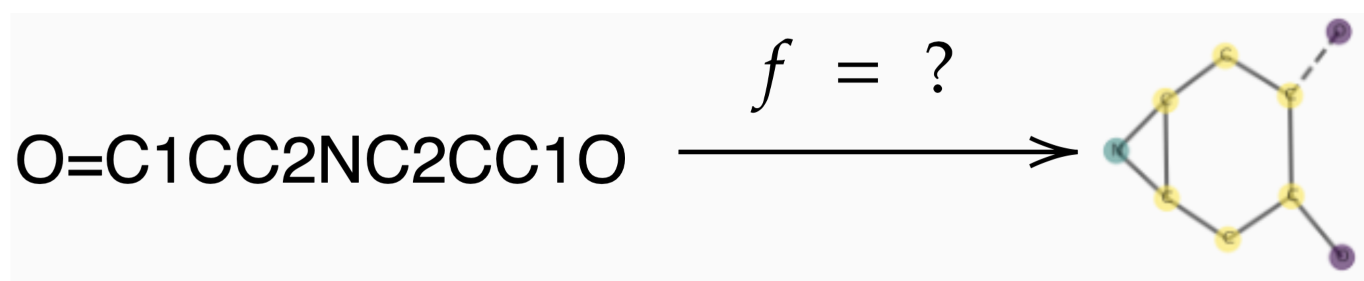
Experiment: Synthetic Least Squares Regression

Setting: $n = 50{,}000$ training data points, $\mathcal{X} = \mathbb{R}^{2{,}000}$, $\mathcal{Y} = \mathbb{R}^{1{,}000}$, k linear kernel so that $\mathcal{H} = \mathcal{Y} = \mathbb{R}^{1{,}000}$.

Goal: build a dataset such that the outputs lie in a subspace of $\mathcal{Y}$ of dimension $d = 50 < 1{,}000$.

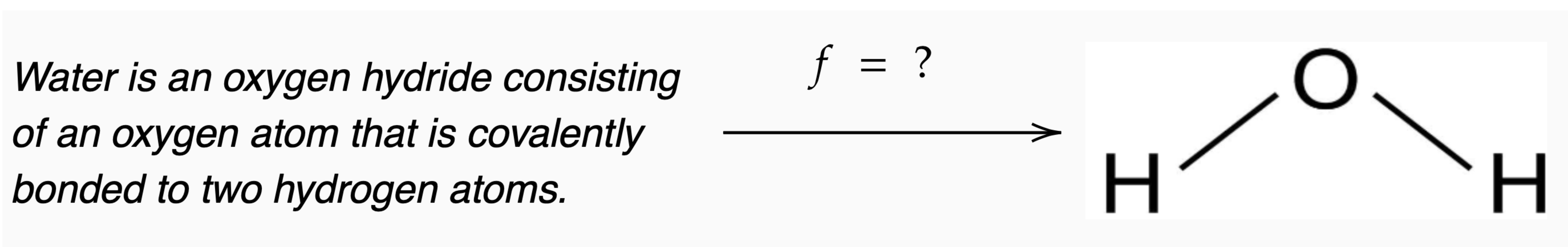


Experiment: SMILES to Molecule on the QM9 dataset



	GED w/o edge feature ↓	GED w/ edge feature ↓
SISOKR	3.330 ± 0.080	4.192 ± 0.109
NNBary-FGW	5.115 ± 0.129	-
Sketched ILE-FGW	2.998 ± 0.253	-
DSOKR	**1.951** $\pm$ **0.074**	**2.960** $\pm$ **0.079**

Experiment: Text to Molecule on the ChEBI-20 dataset



	Hits@1 ↑	Hits@10 ↑	MRR ↑
SISOKR	0.4%	2.8%	0.015
SciBERT Regression	16.8%	56.9%	0.298
CMAM - MLP	34.9%	84.2%	0.513
CMAM - GCN	33.2%	82.5%	0.495
CMAM - Ensemble (MLP×3 + GCN×3)	44.2%	**88.7%**	0.597
DSOKR - SubSample Sketch	48.2%	87.4%	0.624
DSOKR - Gaussian Sketch	49.0%	87.5%	0.630
DSOKR - Ensemble (SubSample×3)	**51.0%**	88.2%	**0.642**
DSOKR - Ensemble (Gaussian×3)	50.5%	87.9%	**0.642**
DSOKR - Ensemble (SubSample×3 + Gaussian×3)	50.0%	88.3%	0.640

References

[1] C. Brouard, M. Szafranski, and F. d'Alché Buc. Input output kernel regression: supervised and semi-supervised structured output prediction with operator-valued kernels. *JMLR*, 17(1):6105–6152, 2016.

[2] C. Ciliberto, L. Rosasco, and A. Rudi. A general framework for consistent structured prediction with implicit loss embeddings. *JMLR*, 21(98):1–67, 2020.

[3] J. Weston, O. Chapelle, V. Vapnik, A. Elisseeff, and B. Schölkopf. Kernel dependency estimation. In *NeurIPS*, pages 897–904. MIT Press, 2003.

Check-out our code!

Acknowledgements

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement 101120237 (ELIAS), the Télécom Paris research chair on Data Science and Artificial Intelligence for Digitalized Industry and Services (DSAIDIS) and the PEPR-IA through the project FOUNDRY.