Structured Data: Learning and Prediction Differentiable Ranks and Sorting using Optimal Transport

Benjamin Cohen, Vladimir Kondratyev, Jean Pachebat

IP Paris Master Data Science

28/03/2022

Difficult problem for differentiable pipelines

Sorting $\Leftrightarrow$ find R, S $\rightarrow$ not differentiable and R piecewise constant (i.e. Jacobian $\frac{\partial R}{\partial x} = 0$ a.e).

- $S(x) := x_\sigma$ - vectors of sorted values
- $R(x) := \sigma^{-1}$ - the rank of each entry of x.

Article's goal: implement a differentiable proxy.

$\Longrightarrow$ Optimal assignment problem: sort n values by matching them to a probability measure supported on m increasing values on any increasing family of the n target values.

$\Longrightarrow$ Optimal Transport (OT) $\rightarrow$ relaxation of the basic problem allowing us to extend rank and sort operators using probability measures.

$\Longrightarrow$ Regularization with entropic penalty then apply Sinkhorn iterations to gain back differentiable operators.

$\Longrightarrow$ Smooth approximation of rank and sort for classification 0/1 loss and the quantile regression loss.

OT between 1D measures using sorting

We introduce a translation invariant, non-negative ground metric:

$$(x, y) \to h(y - x) \text{ with } h : \mathbb{R} \to \mathbb{R}_+$$

the OT problem between ξ and ν is the linear program:

$$OT_h(\xi, \nu) = \min_{P \in U(a,b)} \langle P, C_{xy} \rangle \tag{1}$$

where $U(a, b) := \{P \in {R_+}^{n \times m} \mid P\mathbf{1}_m = a,\ P^\top \mathbf{1}_m = b\}$.
Write $C_{xy} = [h(y_j - x_i]_{ij}$ When h is supposed convex, we get a closed form solution using quantile functions:

$$OT_h(\xi, \nu) = \int_0^1 h(Q_\nu(u) - Q_\xi(u))\, du \tag{2}$$

Complexity of $O(nm(n + m))$ and non differentiable aspect of the solutions of (1). $\implies$ Entropic Regularization of the OT problem.

Entropic regularization of the OT problem

$$P_\star^\varepsilon := \underset{P \in U(\mathbf{a},\mathbf{b})}{\operatorname{argmin}} \langle P, C_{\mathbf{xy}} \rangle - \varepsilon H(P), \quad H(P) = -\sum_{i,j} P_{ij} (\log P_{ij} - 1)$$

$$P_\star^\epsilon = \mathbf{D}(\mathbf{u}) K \mathbf{D}(\mathbf{v}) \quad \widetilde{R}_\varepsilon(\mathbf{a}, \mathbf{x}; \mathbf{b}, \mathbf{y}) := n\mathbf{a}^{-1} \circ \mathbf{u} \circ K(\mathbf{v} \circ \overline{\mathbf{b}}) \in [0, n]^n$$

$$K = exp(-C_{xy}/\epsilon) \quad \widetilde{S}_\varepsilon(\mathbf{a}, \mathbf{x}; \mathbf{b}, \mathbf{y}) := \mathbf{b}^{-1} \circ \mathbf{v} \circ K^T(\mathbf{u} \circ \mathbf{x}) \in \mathbb{R}^m$$

Inputs: $\mathbf{a}, \mathbf{b}, \mathbf{x}, \mathbf{y}, \epsilon, h, \eta$
$C_{\mathbf{xy}} \leftarrow [h(y_j - x_i)]_{i,j}$
$K \leftarrow e^{-C_{\mathbf{xy}}/\epsilon}$, $\mathbf{u} = \mathbf{1}_n$;
while $\Delta(\mathbf{v} \circ K^T\mathbf{u}, \mathbf{b}) < \eta$ **do**
    $\mathbf{v} \leftarrow \mathbf{v}/K^T\mathbf{u}, \mathbf{u} \leftarrow \mathbf{a}/K^T\mathbf{v}$
end whilereturn $\mathbf{u}$, $\mathbf{v}$, K

Use case: learning with Smoothed Ranks and Sorts

Labels $1, ..., L$, Set of points Ω. $f_\theta : \Omega \to \mathbb{R}^L$. The function selects the class attributed to ω by taking $l^\star = \text{argmax}_l [f_\theta(\omega)]_l$.
To train this model, classical approach:

$$\min_\theta \ \Sigma_i \ \mathbf{1}_L^T log f_\theta(\omega_i) - [f_\theta(\omega_i)]_{l_i}$$

which writes:

$$\mathcal{L}_{0/1}\left(f_\theta(\omega), l\right) = H\left(L - [R\left(f_\theta(\omega)\right]_l\right)$$

with: $H(u) = \mathbf{1}_{\{u<0\}}$
Differentiable approximation:

$$\widetilde{\mathcal{L}}_{k,\varepsilon}\left(f_\theta(\omega), l\right) = J_k\left(L - \left[\widetilde{R}_\varepsilon\left(\frac{\mathbf{1}_L}{L}, f_\theta(\omega); \frac{\mathbf{1}_L}{L}, \frac{\overline{\mathbf{1}}_L}{L}, h\right)\right]_l\right)$$

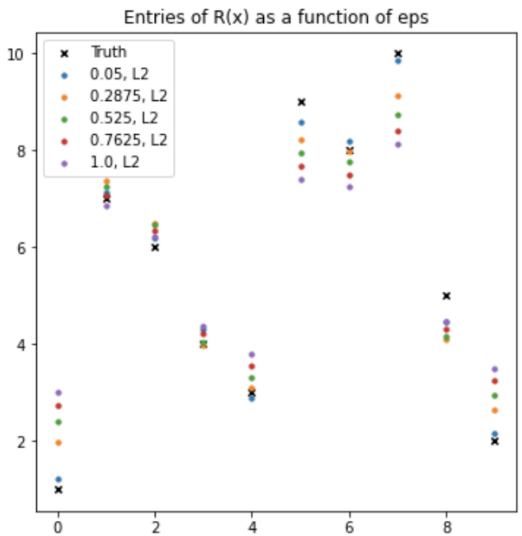
Ranking And Sorting Numpy

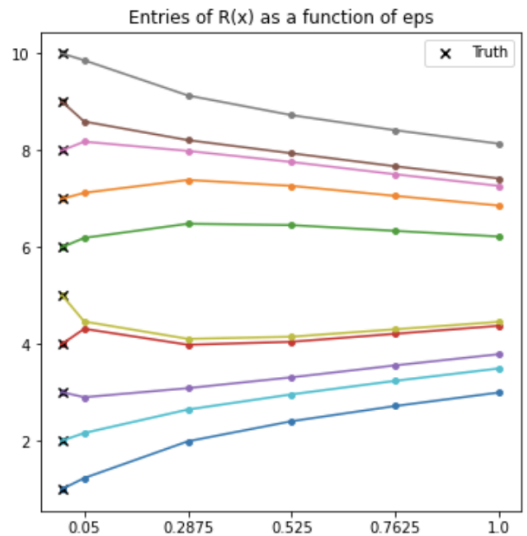
We have conducted the experiments to check influence of the parameter of regularisation ϵ, introduced in

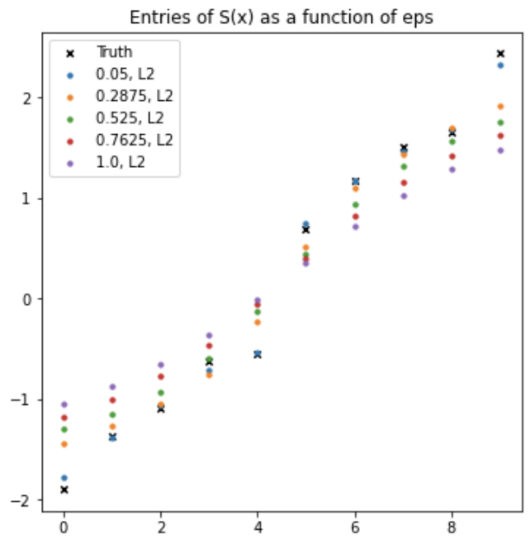
$$K \leftarrow e^{-C_{\mathbf{xy}}/\epsilon}, \mathbf{u} = \mathbf{1}_n$$

$$\text{While } \Delta(\mathbf{v} \circ K^T \mathbf{u}, \mathbf{b}) < \eta :$$

$$\mathbf{v} \leftarrow \mathbf{v}/K^T\mathbf{u}, \mathbf{u} \leftarrow \mathbf{a}/K^T\mathbf{v}$$





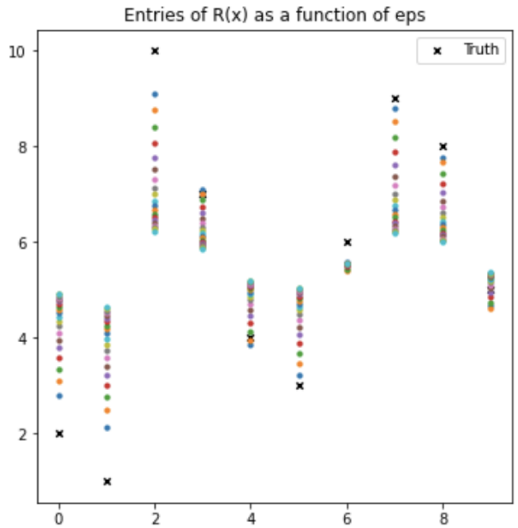


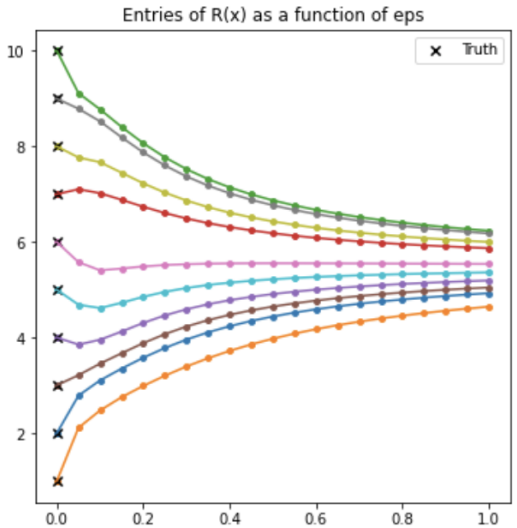
More numerically stable version

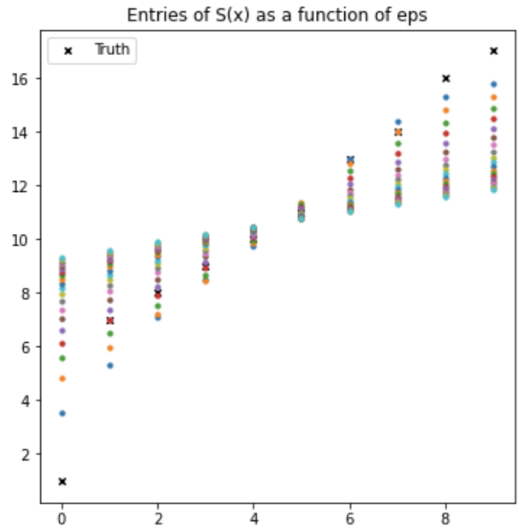
Sinkhorn:

$$\forall s, \boldsymbol{\beta}_s \leftarrow \varepsilon \log \mathbf{b}_s + \min_c \left(C_s^T - \mathbf{1}_m \boldsymbol{\alpha}_s^T - \boldsymbol{\beta}_s \mathbf{1}_n^T \right) + \boldsymbol{\beta}_s$$

$$\forall s, \boldsymbol{\alpha}_s \leftarrow \varepsilon \log \mathbf{a}_s + \min_e \left(C_s - \boldsymbol{\alpha}_s \mathbf{1}_m^T - \mathbf{1}_n \boldsymbol{\beta}_s^T \right) + \boldsymbol{\alpha}_s$$







Even though it may look like, its optimal to take value of epsilon as small as possible, to approximate the exact solution. We noted, that with lover values of epsilon, the Sinkhorn approximation takes longer to converge.