A Functional Extension of Multi-Output Learning

Alex Lambert [*1] **Romain Brault** [*2] **Zoltan Szabo** [3] **Florence d'Alché-Buc** [1]

Abstract

When considering simultaneously a finite number of tasks, multi-output learning enables one to account for the similarities of the tasks via appropriate regularizers. We propose a generalization of the classical setting to a continuum of tasks by using Vector-Valued Reproducing Kernel Hilbert Spaces.

1. Introduction

Several fundamental problems in machine learning and statistics can be phrased as the minimization of a loss function described by a hyperparameter. The hyperparameter might capture numerous aspects of the problem: (i) the tolerance w. r. t. outliers as the ϵ-insensitivity in Support Vector Regression (Vapnik et al., 1997), (ii) importance of smoothness or sparsity such as the weight of the l_2-norm in Tikhonov regularization (Tikhonov & Arsenin, 1977), l_1-norm in LASSO (Tibshirani, 1996), or more general structured-sparsity inducing norms (Bach et al., 2012), (iii) Density Level-Set Estimation (DLSE), see for example one-class support vector machines One-Class Support Vector Machine (OCSVM, Schölkopf et al., 2000), (iv) confidence as exemplified by Quantile Regression (QR, Koenker & Bassett Jr, 1978), or (v) importance of different decisions as implemented by Cost-Sensitive Classification (CSC, Zadrozny & Elkan, 2001). In various cases including QR, CSC or DLSE, one is interested in solving the parameterized task for several hyperparameter values. Multi-Task Learning (Evgeniou & Pontil, 2004) provides a principled way of benefiting from the relationship between similar tasks while preserving local properties of the algorithms: ν-property in DLSE (Glazer et al., 2013) or quantile property in QR (Takeuchi et al., 2006).

A natural extension from the traditional multi-task setting is to provide a prediction tool being able to deal with *any* value of the hyperparameter. In their seminal work, (Takeuchi et al., 2013) extended multi-task learning by considering an infinite number of parametrized tasks in a framework called Parametric Task Learning (PTL). Assuming that the loss is piecewise affine in the hyperparameter, the authors are able to get the whole solution path through parametric programming, relying on techniques developed by Hastie et al. (2004).

In this paper[1], we relax the affine model assumption on the tasks as well as the piecewise-linear assumption on the loss, and take a different angle. We propose Infinite Task Learning (ITL) within the framework of function-valued function learning to handle a continuum number of parameterized tasks using Vector-Valued Reproducing Kernel Hilbert Space (vv-RKHS, Pedrick, 1957).

2. Problem Formulation

After introducing a few notations, we gradually define our goal by moving from single parameterized tasks to ITL through multi-output learning.

A *supervised parametrized task* is defined as follows. Let $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ be a random variable with joint distribution $\mathbf{P}_{X,Y}$ which is assumed to be fixed but unknown; we also assume that $\mathcal{Y} \subset \mathbb{R}$. We have access to n independent identically distributed (i. i. d.) observations called training samples: $\mathcal{S} := ((x_i, y_i))_{i=1}^n \sim \mathbf{P}_{X,Y}^{\otimes n}$. Let Θ be the domain of hyperparameters, and $v_\theta \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a loss function associated to $\theta \in \Theta$. Let $\mathcal{H} \subset \mathcal{F}(\mathcal{X}; \mathcal{Y})$ denote our hypothesis class; throughout the paper $\mathcal{H}$ is assumed to be a Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. For a given θ, the goal is to estimate the minimizer of the expected risk

$$R^\theta(h) := \mathbf{E}_{X,Y}[v_\theta(Y, h(X))] \tag{1}$$

over $\mathcal{H}$, using the training sample $\mathcal{S}$. This task can be addressed by solving the regularized empirical risk minimization problem

$$\min_{h \in \mathcal{H}} R_{\mathcal{S}}{}^\theta(h) + \Omega(h), \tag{2}$$

where $R_{\mathcal{S}}^\theta(h) := \frac{1}{n} \sum_{i=1}^n v_\theta(y_i, h(x_i))$ is the empirical risk and $\Omega : \mathcal{H} \to \mathbb{R}$ is a regularizer. Below we give two examples.

[*]Equal contribution [1]Télécom Paris, France [2]Thales [3]Ecole Polytechnique, France. Correspondence to: Alex Lambert <alex.lambert@telecom-paristech.fr>.

Proceedings of the 1st Adaptive & Multitask Learning Workshop, Long Beach, California, 2019.

[1]This paper is a short version of (Brault et al., 2019)

Quantile Regression: In this setting $\theta \in (0, 1)$. For a given hyperparameter θ, in Quantile Regression the goal is to predict the θ-quantile of the real-valued output conditional distribution $\mathbf{P}_{Y|X}$. The task can be tackled using the pinball loss (Koenker & Bassett Jr, 1978) defined in Eq. (3).

$$v_\theta(y, h(x)) = \big|\theta - \mathbb{1}_{\mathbb{R}_-}(y - h(x))\big||y - h(x)|, \qquad (3)$$
$$\Omega(h) = \tfrac{\lambda}{2}\|h\|^2_{\mathcal{H}}, \quad \lambda > 0.$$

Density Level-Set Estimation: Examples of parameterized tasks can also be found in the unsupervised setting. For instance in outlier detection, the goal is to separate outliers from inliers. A classical technique to tackle this task is OCSVM (Schölkopf et al., 2000). OCSVM has a free parameter $\theta \in (0, 1]$, which can be proven to be an upper bound on the fraction of outliers. This unsupervised learning problem can be empirically described by the minimization of a regularized empirical risk ${R_{\mathcal{S}}}^{\theta}(h, t) + \Omega(h)$, solved *jointly* over $h \in \mathcal{H}$ *and* $t \in \mathbb{R}$ with

$$v_\theta(t, h(x)) = -t + \frac{1}{\theta}|t - h(x)|_+, \quad \Omega(h) = \tfrac{1}{2}\|h\|^2_{\mathcal{H}}.$$

In the aforementioned problems, one is rarely interested in the choice of a single hyperparameter value (θ) and associated risk $\left({R_{\mathcal{S}}}^{\theta}\right)$, but rather in the joint solution of multiple tasks. The naive approach of solving the different tasks independently can easily lead to inconsistencies. A principled way of solving many parameterized tasks has been cast as a MTL problem (Evgeniou et al., 2005) which takes into account the similarities between tasks and helps providing consistent solutions. For example it is possible to encode the similarities of the different tasks in MTL through an explicit constraint function (Ciliberto et al., 2017). In the current work, the similarity between tasks is designed in an implicit way through the loss function and the use of a kernel on the hyperparameters. Moreover, in contrast to MTL, in our case the input space and the training samples are the same for each task; a task is specified by a value of the hyperparameter. This setting is sometimes refered to as multi-output learning (Álvarez et al., 2012).

Formally, assume that we have p tasks described by parameters $(\theta_j)_{j=1}^p$. The idea of multi-task learning is to minimize the sum of the local loss functions ${R_{\mathcal{S}}}^{\theta_j}$, i. e.

$$\arg\min_h \sum\nolimits_{j=1}^{p} {R_{\mathcal{S}}}^{\theta_j}(h_j) + \Omega(h),$$

where the individual tasks are modelled by the real-valued h_j functions, the overall $\mathbb{R}^p$-valued model is the vector-valued function $h\colon x \mapsto (h_1(x), \dots, h_p(x))$, and Ω is a regularization term encoding similarities between tasks. Such approaches have been developed in (Sangnier et al., 2016) for QR and in (Glazer et al., 2013) for DLSE.

Learning a continuum of tasks: In the following, we propose a novel framework called Infinite Task Learning in which we learn a function-valued function $h \in \mathcal{F}(\mathcal{X}; \mathcal{F}(\Theta; \mathcal{Y}))$. Our goal is to be able to handle new tasks after the learning phase and thus, not to be limited to given predefined values of the hyperparameter. Regarding this goal, our framework generalizes the Parametric Task Learning approach introduced by Takeuchi et al. (2013), by allowing a wider class of models and relaxing the hypothesis of piece-wise linearity of the loss function. Moreover a nice byproduct of this vv-RKHS based approach is that one can benefit from the functional point of view, design new regularizers and impose various constraints on the whole continuum of tasks, e. g.,

- The continuity of the $\theta \mapsto h(x)(\theta)$ function is a natural desirable property: for a given input x, the predictions on similar tasks should also be similar.
- Another example is to impose a shape constraint in QR: the conditional quantile should be increasing w. r. t. the hyperparameter θ. This requirement can be imposed through the functional view of the problem.
- In DLSE, to get nested level sets, one would want that for all $x \in \mathcal{X}$, the decision function $\theta \mapsto \mathbb{1}_{\mathbb{R}_+}(h(x)(\theta) - t(\theta))$ changes its sign only once.

To keep the presentation simple, in the sequel we are going to focus on ITL in the supervised setting; unsupervised tasks can be handled similarly.

Assume that h belongs to some space $\mathcal{H} \subseteq \mathcal{F}(\mathcal{X}; \mathcal{F}(\Theta; \mathcal{Y}))$ and introduce an integrated loss function

$$V(y, h(x)) := \int_\Theta v(\theta, y, h(x)(\theta))\mathrm{d}\mu(\theta), \qquad (4)$$

where the local loss $v\colon \Theta \times \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ denotes v_θ seen as a function of three variables including the hyperparameter and μ is a probability measure on Θ which encodes the importance of the prediction at different hyperparameter values. Without prior information and for compact Θ, one may consider μ to be uniform. The true risk reads then

$$R(h) := \mathbf{E}_{X,Y}\left[V(Y, h(X))\right]. \qquad (5)$$

Intuitively, minimizing the expectation of the integral over θ in a rich enough space corresponds to searching for a pointwise minimizer $x \mapsto h^*(x)(\theta)$ of the parametrized tasks introduced in Eq. (1) with, for instance, the implicit space constraint that $\theta \mapsto h^*(x)(\theta)$ is a continuous function for each input x. We show in Proposition S.4.1 that this is precisely the case in QR.

Interestingly, the empirical counterpart of the true risk minimization can now be considered with a much richer family

of penalty terms:

$$\min_{h\in\mathcal{H}} R_{\mathcal{S}}(h)+\Omega(h), \quad R_{\mathcal{S}}(h):=\frac{1}{n}\sum_{i=1}^{n} V(y_i,h(x_i)). \quad (6)$$

Here, $\Omega(h)$ can be a weighted sum of various penalties as seen in Section 3. Many different models ($\mathcal{H}$) could be applied to solve this problem. In our work we consider Reproducing Kernel Hilbert Spaces as they offer a simple and principled way to define regularizers by the appropriate choice of kernels and exhibit a significant flexibility.

3. Using RKHSs to Solve this Problem

This section is dedicated to solving the ITL problem defined in Eq. (6). We first focus on the objective ($\widetilde{V}$), then detail the applied vv-RKHS model family with various penalty examples, followed by representer theorems which give rise to computational tractability.

Sampled Empirical Risk: In practice solving Eq. (6) can be rather challenging due to the integral over θ. One might consider different numerical integration techniques to handle this issue. We focus here on Quasi Monte Carlo (QMC) methods as they allow (i) efficient optimization over vv-RKHSs which we will use for modelling $\mathcal{H}$ (Proposition 3.1), and (ii) enable us to derive generalization guarantees (Proposition 3.3). Indeed, let

$$\widetilde{V}(y,h(x)):=\sum_{j=1}^{m} w_j v(\theta_j,y,h(x)(\theta_j)) \quad (7)$$

be the QMC approximation of Eq. (4). Let $w_j = m^{-1}F^{-1}(\theta_j)$, and $(\theta_j)_{j=1}^m$ be a sequence with values in $[0,1]^d$ such as the Sobol or Halton sequence where μ is assumed to be absolutely continuous w. r. t. the Lebesgue measure and F is the associated cdf. Using this notation and the training samples $\mathcal{S}=((x_i,y_i))_{i=1}^n$, the empirical risk takes the form

$$\widetilde{R}_{\mathcal{S}}(h):=\frac{1}{n}\sum_{i=1}^{n}\widetilde{V}(y_i,h(x_i)) \quad (8)$$

and the problem to solve is

$$\min_{h\in\mathcal{H}} \widetilde{R}_{\mathcal{S}}(h)+\Omega(h). \quad (9)$$

Hypothesis class ($\mathcal{H}$): Recall that $\mathcal{H} \subseteq \mathcal{F}(\mathcal{X};\mathcal{F}(\Theta;\mathcal{Y}))$, in other words $h(x)$ is a $\Theta\mapsto\mathcal{Y}$ function for all $x\in\mathcal{X}$. In this work we assume that the $\Theta\mapsto\mathcal{Y}$ mapping can be described by an RKHS $\mathcal{H}_{k_\Theta}$ associated to a $k_\Theta\colon\Theta\times\Theta\to\mathbb{R}$ scalar-valued kernel defined on the hyperparameters. Let $k_{\mathcal{X}}\colon\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ be a scalar-valued kernel on the input space. The $x\mapsto$ (hyperparameter $\mapsto$ output) relation, i. e. $h\colon\mathcal{X}\to\mathcal{H}_{k_\Theta}$ is then modelled by the Vector-Valued Reproducing Kernel Hilbert Space $\mathcal{H}_K = \overline{\text{span}}\{K(\cdot,x)f \mid x\in\mathcal{X},\ f\in\mathcal{H}_{k_\Theta}\}$, where the operator-valued kernel K is defined as $K(x,z) = k_{\mathcal{X}}(x,z)I$, and $I = I_{\mathcal{H}_{k_\Theta}}$ is the identity operator on $\mathcal{H}_{k_\Theta}$.

This so-called decomposable Operator-Valued Kernel has several benefits and gives rise to a function space with a well-known structure. One can consider elements $h\in\mathcal{H}_K$ as mappings from $\mathcal{X}$ to $\mathcal{H}_{k_\Theta}$, and also as functions from $(\mathcal{X}\times\Theta)$ to $\mathbb{R}$. It is indeed known that there is an isometry between $\mathcal{H}_K$ and $\mathcal{H}_{k_\mathcal{X}}\otimes\mathcal{H}_{k_\Theta}$, the RKHS associated to the product kernel $k_\mathcal{X}\otimes k_\Theta$. The equivalence between these views allows a great flexibility and enables one to follow a functional point of view (to analyse statistical aspects) or to leverage the tensor product point of view (to design new kind of penalization schemes). Below we detail various regularizers before focusing on the representer theorems.

- **Ridge Penalty**: For QR, a natural regularization is the squared vv-RKHS norm

$$\Omega^{\text{RIDGE}}(h)=\tfrac{\lambda}{2}\|h\|^2_{\mathcal{H}_K}, \quad \lambda>0. \quad (10)$$

  This choice is amenable to excess risk analysis (see Proposition 3.3). It can be also seen as the counterpart of the classical (multi-task regularization term introduced by Sangnier et al. (2016), compatible with an infinite number of tasks. $\|\cdot\|^2_{\mathcal{H}_K}$ acts by constraining the solution to a ball of a finite radius within the vv-RKHS, whose shape is controlled by both $k_\mathcal{X}$ and k_Θ.

- **$L^{2,1}$-penalty**: For DLSE, it is more adequate to apply an $L^{2,1}$-RKHS mixed regularizer:

$$\Omega^{\text{DLSE}}(h)=\frac{1}{2}\int_\Theta \|h(\cdot)(\theta)\|^2_{\mathcal{H}_{k_\mathcal{X}}}\mathrm{d}\mu(\theta) \quad (11)$$

  which is an example of a Θ-integrated penalty. This Ω choice allows the preservation of the θ-property (see Fig. S.3), i. e. that the proportion of the outliers is θ.

- **Shape Constraints**: Taking the example of QR it is advantageous to ensure the monotonicity of the estimated quantile function Let $\partial_\Theta h$ denotes the derivative of $h(x)(\theta)$ with respect to θ. Then one should solve

$$\begin{aligned}&\operatorname*{arg\,min}_{h\in\mathcal{H}_K} \ \widetilde{R}_{\mathcal{S}}(h)+\Omega^{\text{RIDGE}}(h)\\ &\text{s.t.} \quad \forall(x,\theta)\in\mathcal{X}\times\Theta, (\partial_\Theta h)(x)(\theta)\geq 0.\end{aligned}$$

  However, the functional constraint prevents a tractable optimization scheme. To mitigate this bottleneck, we penalize if the derivative of h w. r. t. θ is negative:

$$\Omega_{\text{nc}}(h):=\lambda_{\text{nc}}\int_\mathcal{X}\int_\Theta |-(\partial_\Theta h)(x)(\theta)|_+ d\mu(\theta)d\mathbf{P}(x). \quad (12)$$

  When $\mathbf{P}:=\mathbf{P}_X$ this penalization can rely on the same anchors and weights as the ones used to approximate the integrated loss function:

$$\widetilde{\Omega}_{\text{nc}}(h)=\lambda_{\text{nc}}\sum_{i,j=1}^{n,m} w_j|-(\partial_\mathcal{X} h)(x_i)(\theta_j)|_+. \quad (13)$$

Thus, one can modify the overall regularizer in QR to be

$$\Omega(h) := \Omega^{\text{RIDGE}}(h) + \widetilde{\Omega}_{\text{nc}}(h). \tag{14}$$

Representer Theorems: Apart from the flexibility of regularizer design, the other advantage of applying vv-RKHS as hypothesis class is that it gives rise to finite-dimensional representation of the ITL solution under mild conditions.

Proposition 3.1 (Representer)**.** *Assume that for $\forall \theta \in \Theta$, v_θ is a proper lower semicontinuous convex function with respect to its second argument. Then*

$$\arg\min_{h \in \mathcal{H}_K} \widetilde{R}_{\mathcal{S}}(h) + \Omega(h), \quad \lambda > 0$$

with $\Omega(h)$ defined as in Eq. (14), has a unique solution h^*, and $\exists\, (\alpha_{ij})_{i,j=1}^{n,m}, (\beta_{ij})_{i,j=1}^{n,m} \in \mathbb{R}^{2nm}$ such that $\forall x \in \mathcal{X}$

$$h^*(x) = \sum_{i=1}^{n} k_{\mathcal{X}}(x, x_i) \left(\sum_{j=1}^{m} \alpha_{ij} k_\Theta(\cdot, \theta_j) + \beta_{ij} (\partial_2 k_\Theta)(\cdot, \theta_j) \right).$$

For DLSE, we similarly get a representer theorem with the following modelling choice. Let $k_b : \Theta \times \Theta \to \mathbb{R}$ be a scalar-valued kernel (possibly different from k_θ), $\mathcal{H}_{k_b}$ the associated RKHS and $t \in \mathcal{H}_{k_b}$. Assume also that $\Theta \subseteq [\epsilon,\, 1]$ where $\epsilon > 0$.[2] Then, learning a continuum of level sets boils down to the minimization problem

$$\arg\min_{h \in \mathcal{H}_K, t \in \mathcal{H}_{k_b}} \widetilde{R}_{\mathcal{S}}(h, t) + \widetilde{\Omega}(h, t), \quad \lambda > 0, \tag{15}$$

where $\widetilde{\Omega}(h,t) = \frac{1}{2} \sum_{j=1}^{m} w_j \|h(\cdot)(\theta_j)\|^2_{\mathcal{H}_{k_{\mathcal{X}}}} + \frac{\lambda}{2} \|t\|^2_{\mathcal{H}_{k_b}}$,
$\widetilde{R}_{\mathcal{S}}(h,t) = \frac{1}{n} \sum_{i,j=1}^{n,m} \frac{w_j}{\theta_j} \Big(|t(\theta_j) - h(x_i)(\theta_j)|_+ - t(\theta_j) \Big)$.

Proposition 3.2 (Representer)**.** *Assume that k_Θ is bounded: $\sup_{\theta \in \Theta} k_\Theta(\theta, \theta) < +\infty$. Then the minimization problem described in Eq. (15) has a unique solution (h^*, t^*) and there exist $(\alpha_{ij})_{i,j=1}^{n,m} \in \mathbb{R}^{n \times m}$ and $(\beta_j)_{j=1}^{m} \in \mathbb{R}^m$ such that for $\forall (x, \theta) \in \mathcal{X} \times [\epsilon,\, 1]$,*

$$h^*(x)(\theta) = \sum\nolimits_{i,j=1}^{n,m} \alpha_{ij} k_{\mathcal{X}}(x, x_i) k_\Theta(\theta, \theta_j),$$
$$t^*(\theta) = \sum\nolimits_{j=1}^{m} \beta_j k_b(\theta, \theta_j).$$

Remarks:

- Relation to Joint Quantile Regression (JQR): In Infinite Quantile Regression (∞-QR), by choosing k_Θ to be the Gaussian kernel, $k_b(x, z) = \mathbb{1}_{\{x\}}(z)$, $\mu = \frac{1}{m} \sum_{j=1}^{m} \delta_{\theta_j}$, where δ_θ is the Dirac measure concentrated on θ, one gets back Sangnier et al. (2016)'s Joint Quantile Regression (JQR) framework as a special case of our approach. In contrast to the JQR, however, in ∞-QR one can predict the quantile value at any $\theta \in (0, 1)$, even outside the $(\theta_j)_{j=1}^m$ used for learning.
- Relation to q-OCSVM: In DLSE, by choosing $k_\Theta(\theta, \theta') = 1$ (for all $\theta\,, \theta' \in \Theta$) to be the constant kernel, $k_b(\theta, \theta') = \mathbb{1}_{\{\theta\}}(\theta')$, $\mu = \frac{1}{m} \sum_{j=1}^{m} \delta_{\theta_j}$, our approach specializes to q-OCSVM (Glazer et al., 2013).
- Relation to Kadri et al. (2016): Note that Operator-Valued Kernels for functional outputs have also been used in (Kadri et al., 2016), under the form of integral operators acting on L^2 spaces. Both kernels give rise to the same space of functions; the benefit of our approach being to provide an *exact* finite representation of the solution (see Proposition 3.1).
- Efficiency of the decomposable kernel: this kernel choice allows to rewrite the expansions in Propositions 3.1 and 3.2 as a Kronecker products and the complexity of the prediction of n' points for m' quantile becomes $\mathcal{O}(m'mn + n'nm)$ instead of $\mathcal{O}(m'mn'n)$.

Excess Risk Bounds: Below we provide a generalization error analysis to the solution of Eq. (9) for QR (with Ridge regularization and without shape constraints) by stability argument (Bousquet & Elisseeff, 2002), extending the work of Audiffren & Kadri (2013) to Infinite-Task Learning. The proposition (finite sample bounds are given in Corollary S.5.6) instantiates the guarantee for the QMC scheme.

Proposition 3.3 (Generalization)**.** *Let $h^* \in \mathcal{H}_K$ be the solution of Eq. (9) for the* QR *problem with* QMC *approximation. Under mild conditions on the kernels $k_{\mathcal{X}}, k_\Theta$ and $\mathbf{P}_{X,Y}$, stated in the supplement, one has*

$$R(h^*) \leq \widetilde{R}_{\mathcal{S}}(h^*) + \mathcal{O}_{\mathbf{P}_{X,Y}} \left(\frac{1}{\sqrt{\lambda n}} \right) + \mathcal{O} \left(\frac{\log(m)}{\sqrt{\lambda} m} \right). \tag{16}$$

$(\mathbf{n}, \mathbf{m})$ Trade-off: The proposition reveals the interplay between the two approximations, n (the number of training samples) and m (the number of locations taken in the integral approximation), and allows to identify the regime in $\lambda = \lambda(n, m)$ driving the excess risk to zero. Indeed by choosing $m = \sqrt{n}$ and discarding logarithmic factors for simplicity, $\lambda \gg n^{-1}$ is sufficient. The mild assumptions imposed are: boundedness on both kernels and the random variable Y, as well as some smoothness of the kernels.

Numerical Experiments: The efficiency of the ITL scheme for QR has been tested on several benchmarks; the results are summarized in Table S.1 for 20 real datasets from the UCI repository. An additional experiment concerning the non-crossing property on a synthetic dataset can be found in Fig. S.2.

[2] We choose $\Theta \subseteq [\epsilon,\, 1]$, $\epsilon > 0$ rather than $\Theta \subseteq [0,\, 1]$ because the loss might not be integrable on $[0,\, 1]$.

In the DLSE case, one can refer to Fig. S.3 for an experiment on the θ-property (proportion of inliers/outliers).

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. In *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 265–283, 2016.

Álvarez, M. A., Rosasco, L., and Lawrence, N. D. Kernels for vector-valued functions: a review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012.

Audiffren, J. and Kadri, H. Stability of multi-task kernel regression algorithms. In *Asian Conference on Machine Learning (ACML)*, volume 29, pp. 1–16. PMLR, 2013.

Bach, F., Jenatton, R., Mairal, J., and Obozinski, G. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

Bousquet, O. and Elisseeff, A. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

Brault, R., Lambert, A., Szabo, Z., Sangnier, M., and d'Alche Buc, F. Infinite task learning in rkhss. In Chaudhuri, K. and Sugiyama, M. (eds.), *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pp. 1294–1302. PMLR, 16–18 Apr 2019. URL http://proceedings.mlr.press/v89/brault19a.html.

Choquet, G. *Cours d'analyse: Tome II. Topologie*. Masson et Cie., 1969.

Ciliberto, C., Rudi, A., Rosasco, L., and Pontil, M. Consistent multitask learning with nonlinear output relations. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1986–1996, 2017.

Evgeniou, T. and Pontil, M. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 109–117. ACM, 2004.

Evgeniou, T., Micchelli, C. A., and Pontil, M. Learning multiple tasks with kernel methods. *JMLR*, 6:615–637, 2005.

Fei, Y., Rong, G., Wang, B., and Wang, W. Parallel l-bfgs-b algorithm on GPU. *Computers & Graphics*, 40:1–9, 2014.

Glazer, A., Lindenbaum, M., and Markovitch, S. q-ocsvm: A q-quantile estimator for high-dimensional distributions. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 503–511, 2013.

Hastie, T., Rosset, S., Tibshirani, R., and Zhu, J. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5(Oct):1391–1415, 2004.

Jones, E., Oliphant, T., Peterson, P., et al. SciPy: Open source scientific tools for Python, 2001.

Kadri, H., Duflos, E., Preux, P., Canu, S., Rakotomamonjy, A., and Audiffren, J. Operator-valued kernels for learning from functional response data. *Journal of Machine Learning Research*, 17:1–54, 2016.

Keskar, N. and Wächter, A. A limited-memory quasi-newton algorithm for bound-constrained non-smooth optimization. *Optimization Methods and Software*, pp. 1–22, 2017.

Koenker, R. and Bassett Jr, G. Regression quantiles. *Econometrica: journal of the Econometric Society*, pp. 33–50, 1978.

Li, Y., Liu, Y., and Zhu, J. Quantile regression in reproducing kernel hilbert spaces. *Journal of the American Statistical Association*, 102(477):255–268, 2007.

Pedrick, G. *Theory of reproducing kernels for Hilbert spaces of vector-valued functions*. PhD thesis, University of Kansas, 1957.

Sangnier, M., Fercoq, O., and d'Alché Buc, F. Joint quantile regression in vector-valued rkhss. *Advances in Neural Information Processing Systems (NIPS)*, pp. 3693–3701, 2016.

Schölkopf, B., Smola, A. J., Williamson, R. C., and Bartlett, P. L. New support vector algorithms. *Neural computation*, 12(5):1207–1245, 2000.

Skajaa, A. Limited memory bfgs for nonsmooth optimization. *Master's thesis*, 2010.

Takeuchi, I., Le, Q. V., Sears, T. D., and Smola, A. J. Nonparametric quantile estimation. *Journal of Machine Learning Research*, 7:1231–1264, 2006.

Takeuchi, I., Hongo, T., Sugiyama, M., and Nakajima, S. Parametric task learning. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1358–1366, 2013.

Tibshirani, R. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

Tikhonov, A. N. and Arsenin, V. Y. *Solution of Ill-posed Problems*. Winston & Sons, 1977.

Vapnik, V., Golowich, S. E., and Smola, A. J. Support vector method for function approximation, regression estimation and signal processing. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 281–287, 1997.

Zadrozny, B. and Elkan, C. Learning and making decisions when costs and probabilities are both unknown. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 204–213, 2001.

Zhu, C., Byrd, R. H., Lu, P., and Nocedal, J. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.

SUPPLEMENTARY MATERIAL

S.4. Quantile Regression

S.4.1. Theoretical aspects

Let us recall the expression of the pinball loss:

$$v_\theta : (y,\, y') \in \mathbb{R}^2 \mapsto \max\left(\theta(y-y'), (\theta-1)(y-y')\right) \in \mathbb{R}. \tag{17}$$

Proposition S.4.1. *Let X, Y be two random variables (r. v.s) respectively taking values in $\mathcal{X}$ and $\mathbb{R}$, and q: $\mathcal{X} \to \mathcal{F}([0,1], \mathbb{R})$ the associated conditional quantile function. Let μ be a positive measure on $[0,1]$ such that $\int_0^1 \mathbf{E}\left[v_\theta\left(Y, q(X)(\theta)\right)\right] \mathrm{d}\mu(\theta) < \infty$. Then for $\forall h \in \mathcal{F}\left(\mathcal{X};\, \mathcal{F}\left([0,\, 1];\, \mathbb{R}\right)\right)$*

$$R(h) - R(q) \geq 0,$$

where R is the risk defined in Eq. (5).

Proof. The proof is based on the one given in (Li et al., 2007) for a single quantile. Let $f \in \mathcal{F}\left(\mathcal{X};\, \mathcal{F}\left([0,\, 1];\, \mathbb{R}\right)\right)$, $\theta \in (0,1)$ and $(x,y) \in \mathcal{X} \times \mathbb{R}$. Let also

$$s = \begin{cases} 1 \text{ if } y \leq f(x)(\theta) \\ 0 \text{ otherwise} \end{cases}, \qquad t = \begin{cases} 1 \text{ if } y \leq q(x)(\theta) \\ 0 \text{ otherwise} \end{cases}.$$

It holds that

$$\begin{aligned} v_\theta(y, h(x)(\theta)) - v_\theta(y, q(x)(\theta)) &= \theta(1-s)(y - h(x)(\theta)) + (\theta-1)s(y-h(x)(\theta)) \\ &\quad - \theta(1-t)(y-q(x)(\theta)) - (\theta-1)t(y-q(x)(\theta)) \\ &= \theta(1-t)(q(x)(\theta) - h(x)(\theta)) + \theta((1-t)-(1-s))h(x)(\theta) \\ &\quad + (\theta-1)t(q(x)(\theta - h(x)(\theta))) + (\theta-1)(t-s)h(x)(\theta) + (t-s)y \\ &= (\theta-t)(q(x)(\theta) - h(x)(\theta)) + (t-s)(y - h(x)(\theta)). \end{aligned}$$

Then, notice that

$$\mathbf{E}[(\theta-t)(q(X)(\theta) - h(X)(\theta))] = \mathbf{E}[\mathbf{E}[(\theta-t)(q(X)(\theta) - h(X)(\theta))]|X] = \mathbf{E}[\mathbf{E}[(\theta-t)|X](q(X)(\theta) - h(X)(\theta))]$$

and since q is the true quantile function,

$$\mathbf{E}[t|X] = \mathbf{E}[\mathbf{1}_{\{Y \leq q(X)(\theta)\}}|X] = \mathbf{P}[Y \leq q(X)(\theta)|X] = \theta,$$

so

$$\mathbf{E}[(\theta-t)(q(X)(\theta) - h(X)(\theta))] = 0.$$

Moreover, $(t-s)$ is negative when $q(x)(\theta) \leq y \leq h(x)(\theta)$, positive when $h(x)(\theta) \leq y \leq q(x)(\theta)$ and 0 otherwise, thus the quantity $(t-s)(y - h(x)(\theta))$ is always positive. As a consequence,

$$R(h) - R(q) = \int_{[0,1]} \mathbf{E}[v_\theta(Y, h(X)(\theta)) - v_\theta(Y, q(X)(\theta))] \mathrm{d}\mu(\theta) \geq 0$$

which concludes the proof. □

The Proposition S.4.1 allows us to derive conditions under which the minimization of the risk above yields the true quantile function. Under the assumption that (i) q is continuous (as seen as a function of two variables), (ii) $\mathrm{Supp}(\mu) = [0,1]$, then the minimization of the integrated pinball loss performed in the space of continuous functions yields the true quantile function on the support of $\mathbf{P}_{X,Y}$.

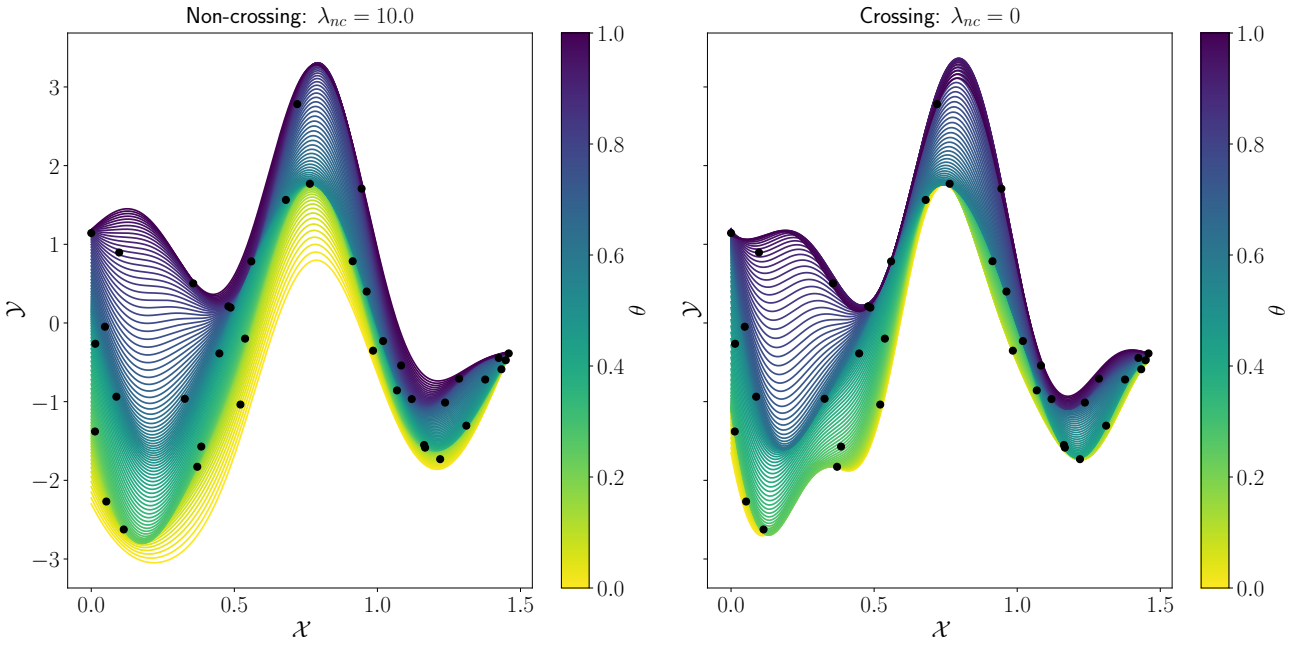

Figure S.2. Impact of crossing penalty on toy data. Left plot: strong non-crossing penalty ($\lambda_{\text{nc}} = 10$). Right plot: no non-crossing penalty ($\lambda_{\text{nc}} = 0$). The plots show 100 quantiles of the continuum learned, linearly spaced between 0 (blue) and 1 (red). Notice that the non-crossing penalty does not provide crossings to occur in the regions where there is no points to enforce the penalty (e. g. $x \in [0.13,\ 0.35]$). This phenomenon is alleviated by the regularity of the model.

S.4.2. Experiments

There are several ways to solve the non-smooth optimization problems associated to the QR, DLSE and CSC tasks. One could proceed for example by duality—as it was done in JQR Sangnier et al. (2016)—, or apply sub-gradient descent techniques (which often converge quite slowly). In order to allow unified treatment and efficient solution in our experiments we used the L-BFGS-B (Zhu et al., 1997) optimization scheme which is widely popular in large-scale learning, with non-smooth extensions (Skajaa, 2010; Keskar & Wächter, 2017). The technique requires only evaluation of objective function along with its gradient, which can be computed automatically using reverse mode automatic differentiation (as in Abadi et al. (2016)). To benefit from from the available fast smooth implementations (Jones et al., 2001; Fei et al., 2014), we applied an infimal convolution on the non-differentiable terms of the objective. Under the assumption that $m = \mathcal{O}(\sqrt{n})$ (see Proposition 3.3), the complexity per L-BFGS-B iteration is $\mathcal{O}(n^2\sqrt{n})$.

The efficiency of the non-crossing penalty is illustrated in Fig. S.2 on a synthetic sine wave dataset where $n = 40$ and $m = 20$ points have been generated. Many crossings are visible on the right plot, while they are almost not noticible on the left plot, using the non-crossing penalty. Concerning our real-world examples (20 UCI datasets), to study the efficiency of the proposed scheme in quantile regression the following experimental protocol was applied. Each dataset was splitted randomly into a training set (70%) and a test set (30%). We optimized the hyperparameters by minimizing a 5-folds cross validation with a Bayesian optimizer[3]. Once the hyperparameters were obtained, a new regressor was learned on the whole training set using the optimized hyperparameters. We report the value of the pinball loss and the crossing loss on the test set for three methods: our technique is called ∞-QR, we refer to Sangnier et al. (2016)'s approach as JQR, and independent learning (abbreviated as IND-QR) represents a further baseline.

We repeated 20 simulations (different random training-test splits); the results are also compared using a Mann-Whitney-Wilcoxon test. A summary is provided in Table S.1.

Notice that while JQR is tailored to predict finite many quantiles, our ∞-QR method estimates the *whole quantile function*

[3]We used a Gaussian Process model and minimized the Expected improvement. The optimizer was initialized using 27 samples from a Sobol sequence and ran for 50 iterations.

Table S.1. Quantile Regression on 20 UCI datasets. Reported: $100\times$value of the pinball loss, $100\times$crossing loss (smaller is better). p.-val.: outcome of the Mann-Whitney-Wilcoxon test of JQR vs. ∞-QR and Independent vs. ∞-QR. Boldface: significant values w. r. t. ∞-QR.

DATASET	JQR				IND-QR				∞-QR	
	(PINBALL	P.-VAL.)	(CROSS	P.-VAL.)	(PINBALL	P.-VAL.)	(CROSS	P.-VAL.)	PINBALL	CROSS
COBARORE	159 ± 24	$9 \cdot 10^{-01}$	0.1 ± 0.4	$6 \cdot 10^{-01}$	150 ± 21	$2 \cdot 10^{-01}$	0.3 ± 0.8	$7 \cdot 10^{-01}$	165 ± 36	2.0 ± 6.0
ENGEL	175 ± 555	$6 \cdot 10^{-01}$	0.0 ± 0.2	$1 \cdot 10^{+00}$	63 ± 53	$8 \cdot 10^{-01}$	4.0 ± 12.8	$8 \cdot 10^{-01}$	47 ± 6	0.0 ± 0.1
BOSTONHOUSING	49 ± 4	$8 \cdot 10^{-01}$	0.7 ± 0.7	$2 \cdot 10^{-01}$	49 ± 4	$8 \cdot 10^{-01}$	$\mathbf{1.3 \pm 1.2}$	$1 \cdot 10^{-05}$	49 ± 4	0.3 ± 0.5
CAUTION	88 ± 17	$6 \cdot 10^{-01}$	0.1 ± 0.2	$6 \cdot 10^{-01}$	89 ± 19	$4 \cdot 10^{-01}$	$\mathbf{0.3 \pm 0.4}$	$2 \cdot 10^{-04}$	85 ± 16	0.0 ± 0.1
FTCOLLINSSNOW	154 ± 16	$8 \cdot 10^{-01}$	0.0 ± 0.0	$6 \cdot 10^{-01}$	155 ± 13	$9 \cdot 10^{-01}$	0.2 ± 0.9	$8 \cdot 10^{-01}$	156 ± 17	0.1 ± 0.6
HIGHWAY	103 ± 19	$4 \cdot 10^{-01}$	0.8 ± 1.4	$2 \cdot 10^{-02}$	99 ± 20	$9 \cdot 10^{-01}$	$\mathbf{6.2 \pm 4.1}$	$1 \cdot 10^{-07}$	105 ± 36	0.1 ± 0.4
HEIGHTS	127 ± 3	$1 \cdot 10^{+00}$	0.0 ± 0.0	$1 \cdot 10^{+00}$	127 ± 3	$9 \cdot 10^{-01}$	0.0 ± 0.0	$1 \cdot 10^{+00}$	127 ± 3	0.0 ± 0.0
SNIFFER	43 ± 6	$8 \cdot 10^{-01}$	0.1 ± 0.3	$2 \cdot 10^{-01}$	44 ± 5	$7 \cdot 10^{-01}$	$\mathbf{1.4 \pm 1.2}$	$6 \cdot 10^{-07}$	44 ± 7	0.1 ± 0.1
SNOWGEESE	55 ± 20	$7 \cdot 10^{-01}$	0.3 ± 0.8	$3 \cdot 10^{-01}$	53 ± 18	$6 \cdot 10^{-01}$	0.4 ± 1.0	$5 \cdot 10^{-02}$	57 ± 20	0.2 ± 0.6
UFC	81 ± 5	$6 \cdot 10^{-01}$	$\mathbf{0.0 \pm 0.0}$	$4 \cdot 10^{-04}$	82 ± 5	$7 \cdot 10^{-01}$	$\mathbf{1.0 \pm 1.4}$	$2 \cdot 10^{-04}$	82 ± 4	0.1 ± 0.3
BIGMAC2003	80 ± 21	$7 \cdot 10^{-01}$	$\mathbf{1.4 \pm 2.1}$	$4 \cdot 10^{-04}$	74 ± 24	$9 \cdot 10^{-02}$	$\mathbf{0.9 \pm 1.1}$	$7 \cdot 10^{-05}$	84 ± 24	0.2 ± 0.4
UN3	98 ± 9	$8 \cdot 10^{-01}$	0.0 ± 0.0	$1 \cdot 10^{-01}$	99 ± 9	$1 \cdot 10^{+00}$	$\mathbf{1.2 \pm 1.0}$	$1 \cdot 10^{-05}$	99 ± 10	0.1 ± 0.4
BIRTHWT	141 ± 13	$1 \cdot 10^{+00}$	0.0 ± 0.0	$6 \cdot 10^{-01}$	140 ± 12	$9 \cdot 10^{-01}$	0.1 ± 0.2	$7 \cdot 10^{-02}$	141 ± 12	0.0 ± 0.0
CRABS	$\mathbf{11 \pm 1}$	$4 \cdot 10^{-05}$	0.0 ± 0.0	$8 \cdot 10^{-01}$	$\mathbf{11 \pm 1}$	$2 \cdot 10^{-04}$	$\mathbf{0.0 \pm 0.0}$	$2 \cdot 10^{-05}$	13 ± 3	0.0 ± 0.0
GAGURINE	61 ± 7	$4 \cdot 10^{-01}$	0.0 ± 0.1	$3 \cdot 10^{-03}$	62 ± 7	$5 \cdot 10^{-01}$	$\mathbf{0.1 \pm 0.2}$	$4 \cdot 10^{-04}$	62 ± 7	0.0 ± 0.0
GEYSER	105 ± 7	$9 \cdot 10^{-01}$	0.1 ± 0.3	$9 \cdot 10^{-01}$	105 ± 6	$9 \cdot 10^{-01}$	0.2 ± 0.3	$6 \cdot 10^{-01}$	104 ± 6	0.1 ± 0.2
GILGAIS	51 ± 6	$5 \cdot 10^{-01}$	0.1 ± 0.1	$1 \cdot 10^{-01}$	49 ± 6	$6 \cdot 10^{-01}$	$\mathbf{1.1 \pm 0.7}$	$2 \cdot 10^{-05}$	49 ± 7	0.3 ± 0.3
TOPO	69 ± 18	$1 \cdot 10^{+00}$	0.1 ± 0.5	$1 \cdot 10^{+00}$	71 ± 20	$1 \cdot 10^{+00}$	$\mathbf{1.7 \pm 1.4}$	$3 \cdot 10^{-07}$	70 ± 17	0.0 ± 0.0
MCYCLE	66 ± 9	$9 \cdot 10^{-01}$	0.2 ± 0.3	$7 \cdot 10^{-03}$	66 ± 8	$9 \cdot 10^{-01}$	$\mathbf{0.3 \pm 0.3}$	$7 \cdot 10^{-06}$	65 ± 9	0.0 ± 0.1
CPUS	$\mathbf{7 \pm 4}$	$2 \cdot 10^{-04}$	$\mathbf{0.7 \pm 1.0}$	$5 \cdot 10^{-04}$	$\mathbf{7 \pm 5}$	$3 \cdot 10^{-04}$	$\mathbf{1.2 \pm 0.8}$	$6 \cdot 10^{-08}$	16 ± 10	0.0 ± 0.0

hence solves a more challenging task. Despite the more difficult problem solved, as Table S.1 suggest that the performance in terms of pinball loss of ∞-QR is comparable to that of the state-of-the-art JQR on all the twenty studied benchmarks, except for the 'crabs' and 'cpus' datasets (p.-val. $< 0.25\%$). In addition, when considering the non-crossing penalty one can observe that ∞-QR outperforms the IND-QR baseline on eleven datasets (p.-val. $< 0.25\%$) and JQR on two datasets. This illustrates the efficiency of the constraint based on the continuum scheme.

S.5. Generalization Properties in the Context of Stability

The analysis of the generalization error will be performed using the notion of uniform stability introduced in (Bousquet & Elisseeff, 2002). For a derivation of generalization bounds in vv-RKHS, we refer to (Kadri et al., 2016). In their framework, the goal is to minimize a risk which can be expressed as

$$R_{\mathcal{S},\lambda}(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, h, x_i) + \lambda \|h\|^2_{\mathcal{H}_K}, \tag{18}$$

where $\mathcal{S} = ((x_1, y_1), \ldots, (x_n, y_n))$ are i. i. d. inputs and $\lambda > 0$. We almost recover their setting by using losses defined as

$$\ell\colon \begin{cases} \mathbb{R} \times \mathcal{H}_K \times \mathcal{X} & \to \mathbb{R} \\ (y, h, x) & \mapsto \widetilde{V}(y, f(x)), \end{cases}$$

where $\widetilde{V}$ is a loss associated to some local cost defined in Eq. (7). Then, they study the stability of the algorithm which, given a dataset $\mathcal{S}$, returns

$$h^*{}_{\mathcal{S}} = \operatorname*{arg\,min}_{h \in \mathcal{H}_K} R_{\mathcal{S},\lambda}(h). \tag{19}$$

There is a slight difference between their setting and ours, since they use losses defined for some y in the output space of the vv-RKHS, but this difference has no impact on the validity of the proofs in our case. The use of their theorem requires

some assumption that are listed below. We recall the shape of the OVK we use : $K : (x, z) \in \mathcal{X} \times \mathcal{X} \mapsto k_{\mathcal{X}}(x, z) I_{\mathcal{H}_{k_\Theta}} \in \mathcal{L}(\mathcal{H}_{k_\Theta})$, where $k_{\mathcal{X}}$ and k_Θ are both bounded scalar-valued kernels, in other words there exist $(\kappa_{\mathcal{X}}, \kappa_\Theta) \in \mathbb{R}^2$ such that $\sup_{x \in X} k_{\mathcal{X}}(x, x) < \kappa_{\mathcal{X}}^2$ and $\sup_{\theta \in \Theta} k_\Theta(\theta, \theta) < \kappa_\Theta^2$.

Assumption 1. *$\exists \kappa > 0$ such that $\forall x \in \mathcal{X}$, $\|K(x, x)\|_{\mathcal{L}(\mathcal{H}_{k_\Theta})} \leq \kappa^2$.*

Assumption 2. *$\forall h_1, h_2 \in \mathcal{H}_{k_\Theta}$, the function $(x_1, x_2) \in \mathcal{X} \times \mathcal{X} \mapsto \langle K(x_1, x_2)h_1, h_2 \rangle_{\mathcal{H}_{k_\Theta}} \in \mathbb{R}$, is measurable.*

Remark 1. *Assumptions 1, 2 are satisfied for our choice of kernel.*

Assumption 3. *The application $(y, h, x) \mapsto \ell(y, h, x)$ is σ-admissible, i. e. convex with respect to f and Lipschitz continuous with respect to $f(x)$, with σ as its Lipschitz constant.*

Assumption 4. *$\exists \xi \geq 0$ such that $\forall (x, y) \in \mathcal{X} \times \mathcal{Y}$ and $\forall \mathcal{S}$ training set, $\ell(y, h^*_{\mathcal{S}}, x) \leq \xi$.*

Definition S.5.1. *Let $\mathcal{S} = ((x_i, y_i))_{i=1}^n$ be the training data. We call $\mathcal{S}^i$ the training data $\mathcal{S}^i = ((x_1, y_1), \ldots, (x_{i-1}, y_{i-1}), (x_{i+1}, y_{i+1}), \ldots, (x_n, y_n))$, $1 \leq i \leq n$.*

Definition S.5.2. *A learning algorithm mapping a dataset $\mathcal{S}$ to a function $h^*_{\mathcal{S}}$ is said to be β-uniformly stable with respect to the loss function ℓ if $\forall n \geq 1$, $\forall 1 \leq i \leq n$, $\forall \mathcal{S}$ training set, $||\ell(\cdot, h^*_{\mathcal{S}}, \cdot) - \ell(\cdot, h^*_{\mathcal{S}^i}, \cdot)||_\infty \leq \beta$.*

Proposition S.5.1. *(Bousquet & Elisseeff, 2002) Let $\mathcal{S} \mapsto h^*_{\mathcal{S}}$ be a learning algorithm with uniform stability β with respect to a loss ℓ satisfying Assumption 4. Then $\forall n \geq 1$, $\forall \delta \in (0, 1)$, with probability at least $1 - \delta$ on the drawing of the samples, it holds that*

$$R(h^*_{\mathcal{S}}) \leq R_{\mathcal{S}}(h^*_{\mathcal{S}}) + 2\beta + (4\beta + \xi)\sqrt{\frac{\log(1/\delta)}{n}}.$$

Proposition S.5.2. *(Kadri et al., 2016) Under assumptions 1, 2, 3, a learning algorithm that maps a training set $\mathcal{S}$ to the function $h^*_{\mathcal{S}}$ defined in Eq. (19) is β-stable with $\beta = \frac{\sigma^2 \kappa^2}{2\lambda n}$.*

Quantile Regression: We recall that in this setting, $v(\theta, y, h(x)(\theta)) = \max(\theta(y - h(x)(\theta)), (1 - \theta)(y - h(x)(\theta)))$ and the loss is

$$\ell : \begin{cases} \mathbb{R} \times \mathcal{H}_K \times \mathcal{X} & \to \mathbb{R} \\ (y, h, x) & \mapsto \frac{1}{m} \sum_{j=1}^m \max(\theta_j(y - h(x)(\theta_j)), (\theta_j - 1)(y - h(x)(\theta_j))). \end{cases} \tag{20}$$

Moreover, we will assume that $|Y|$ is bounded by $B \in \mathbb{R}$ as a r. v.. We will therefore verify the hypothesis for $y \in [-B, B]$ and not $y \in \mathbb{R}$.

Lemma S.5.3. *In the case of the* QR*, the loss ℓ is σ-admissible with $\sigma = 2\kappa_\Theta$.*

Proof. Let $h_1, h_2 \in \mathcal{H}_K$ and $\theta \in [0, 1]$. $\forall x, y \in \mathcal{X} \times \mathbb{R}$, it holds that

$$v(\theta, y, h_1(x)(\theta)) - v(\theta, y, h_2(x)(\theta)) = (\theta - t)(h_2(x)(\theta) - h_1(x)(\theta)) + (t - s)(y - h_1(x)(\theta)),$$

where $s = \mathbf{1}_{y \leq h_1(x)(\theta)}$ and $t = \mathbf{1}_{y \leq h_2(x)(\theta)}$. We consider all possible cases for t and s :

- $t = s = 0 : |(t - s)(y - h_1(x)(\theta))| \leq |h_2(x)(\theta) - h_1(x)(\theta)|$
- $t = s = 1 : |(t - s)(y - h_1(x)(\theta))| \leq |h_2(x)(\theta) - h_1(x)(\theta)|$
- $s = 1, t = 0 : |(t - s)(y - h_1(x)(\theta))| = |h_1(x)(\theta) - y| \leq |h_1(x)(\theta) - h_2(x)(\theta)|$
- $s = 0, t = 1 : |(t - s)(y - h_1(x)(\theta))| = |y - h_1(x)(\theta)| \leq |h_1(x)(\theta) - h_2(x)(\theta)|$ because of the conditions on t, s.

Thus $|v(\theta, y, h_1(x)(\theta)) - v(\theta, y, h_2(x)(\theta))| \leq (\theta + 1)|h_1(x)(\theta) - h_2(x)(\theta)| \leq (\theta + 1)\kappa_\Theta ||h_1(x) - h_2(x)||_{\mathcal{H}_{k_\Theta}}$. By summing this expression over the $(\theta_j)_{j=1}^m$, we get that

$$|\ell(x, h_1, y) - \ell(x, h_2, y)| \leq \frac{1}{m} \sum_{j=1}^m (\theta_j + 1)\kappa_\Theta ||h_1(x) - h_2(x)||_{\mathcal{H}_{k_\Theta}} \leq 2\kappa_\Theta ||h_1(x) - h_2(x)||_{\mathcal{H}_{k_\Theta}}$$

and ℓ is σ-admissible with $\sigma = 2\kappa_\Theta$. □

Lemma S.5.4. *Let $\mathcal{S} = ((x_1, y_1), \dots, (x_n, y_n))$ be a training set and $\lambda > 0$. Then $\forall x, \theta \in \mathcal{X} \times (0,1)$, it holds that* $|h^*{}_{\mathcal{S}}(x)(\theta)| \leq \kappa_{\mathcal{X}} \kappa_{\Theta} \sqrt{\frac{B}{\lambda}}$.

Proof. Since $h^*{}_{\mathcal{S}}$ is the output of our algorithm and $0 \in \mathcal{H}_K$, it holds that

$$\lambda ||h^*{}_{\mathcal{S}}||^2 \leq \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} v(\theta_j, y_i, 0) \leq \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \max(\theta_j, 1-\theta_j)|y_i| \leq B.$$

Thus $||h^*{}_{\mathcal{S}}|| \leq \sqrt{\frac{B}{\lambda}}$. Moreover, $\forall x, \theta \in \mathcal{X} \times (0,1)$, $|h^*{}_{\mathcal{S}}(x)(\theta)| = |\langle h^*{}_{\mathcal{S}}(x), k_{\Theta}(\theta, \cdot)\rangle_{\mathcal{H}_{k_\Theta}}| \leq ||h^*{}_{\mathcal{S}}(x)||_{\mathcal{H}_{k_\Theta}} \kappa_{\Theta} \leq ||h^*{}_{\mathcal{S}}||_{\mathcal{H}_{k_\Theta}} \kappa_{\mathcal{X}} \kappa_{\Theta}$ which concludes the proof. □

Lemma S.5.5. *Assumption 4 is satisfied for $\xi = 2\left(B + \kappa_{\mathcal{X}} \kappa_{\Theta} \sqrt{\frac{B}{\lambda}}\right)$.*

Proof. Let $\mathcal{S} = ((x_1, y_1), \dots, (x_n, y_n))$ be a training set and $h^*{}_{\mathcal{S}}$ be the output of our algorithm. $\forall (x, y) \in \mathcal{X} \times [-B, B]$, it holds that

$$\begin{aligned}\ell(y, h^*{}_{\mathcal{S}}, x) &= \frac{1}{m} \sum_{j=1}^{m} \max(\theta_j(y - h^*{}_{\mathcal{S}}(x)(\theta_j)), (\theta_j - 1)(y - h^*{}_{\mathcal{S}}(x)(\theta_j))) \leq \frac{2}{m} \sum_{j=1}^{m} |y - h^*{}_{\mathcal{S}}(x)(\theta_j)| \\ &\leq \frac{2}{m} \sum_{j=1}^{m} |y| + |h^*{}_{\mathcal{S}}(x)(\theta_j)| \leq 2\left(B + \kappa_{\mathcal{X}} \kappa_{\Theta} \sqrt{\frac{B}{\lambda}}\right).\end{aligned}$$

□

Corollary S.5.6. *The* QR *learning algorithm defined in Eq. (9) is such that $\forall n \geq 1$, $\forall \delta \in (0,1)$, with probability at least $1 - \delta$ on the drawing of the samples, it holds that*

$$\widetilde{R}(h^*{}_{\mathcal{S}}) \leq \widetilde{R}_{\mathcal{S}}(h^*{}_{\mathcal{S}}) + \frac{4\kappa_{\mathcal{X}}^2 \kappa_{\Theta}^2}{\lambda n} + \left[\frac{8\kappa_{\mathcal{X}}^2 \kappa_{\Theta}^2}{\lambda n} + 2\left(B + \kappa_{\mathcal{X}} \kappa_{\Theta} \sqrt{\frac{B}{\lambda}}\right)\right] \sqrt{\frac{\log(1/\delta)}{n}}. \tag{21}$$

Proof. This is a direct consequence of Proposition S.5.2, Proposition S.5.1, Lemma S.5.3 and Lemma S.5.5. □

Definition S.5.3 (Hardy-Krause variation)**.** *Let Π be the set of subdivisions of the interval $\Theta = [0,1]$. A subdivision will be denoted $\sigma = (\theta_1, \theta_2, \dots, \theta_p)$ and $f : \Theta \to \mathbb{R}$ be a function. We call Hardy-Krause variation of the function f the quantity* $\sup_{\sigma \in \Pi} \sum_{i=1}^{p-1} |f(\theta_{i+1}) - f(\theta_i)|$.

Remark 2. *If f is continuous, $V(f)$ is also the limit as the mesh of σ goes to zero of the above quantity.*

In the following, let $f : \theta \mapsto \mathbf{E}_{X,Y}[v(\theta, Y, h^*{}_{\mathcal{S}}(X)(\theta))]$. This function is of primary importance for our analysis, since in the Quasi Monte-Carlo setting, the bound of Proposition 3.3 makes sense only if the function f has finite Hardy-Krause variation, which is the focus of the following lemma.

Lemma S.5.7. *Assume the boundeness of both scalar kernels $k_{\mathcal{X}}$and k_{Θ}. Assume moreover that k_{Θ} is $\mathcal{C}^1$ and that its partial derivatives are uniformly bounded by some constant C. Then*

$$V(f) \leq B + \kappa_{\mathcal{X}} \kappa_{\Theta} \sqrt{\frac{B}{\lambda}} + 2\kappa_{\mathcal{X}} \sqrt{\frac{2BC}{\lambda}}. \tag{22}$$

Proof. It holds that

$$\begin{aligned}
\sup_{\sigma\in\Pi}\sum_{i=1}^{p-1}|f(\theta_{i+1})-f(\theta_i)| &= \sup_{\sigma\in\Pi}\sum_{i=1}^{p-1}\left|\int v(\theta_{i+1},y,{h^*}_{\mathcal{S}}(x)(\theta_{i+1}))\mathrm{d}\mathbf{P}_{X,Y}-\int v(\theta_i,y,{h^*}_{\mathcal{S}}(x)(\theta_i))\mathrm{d}\mathbf{P}_{X,Y}\right| \\
&= \sup_{\sigma\in\Pi}\sum_{i=1}^{p-1}\left|\int v(\theta_{i+1},y,{h^*}_{\mathcal{S}}(x)(\theta_{i+1}))-v(\theta_i,y,{h^*}_{\mathcal{S}}(x)(\theta_i))\mathrm{d}\mathbf{P}_{X,Y}\right| \\
&\leq \sup_{\sigma\in\Pi}\sum_{i=1}^{p-1}\int|v(\theta_{i+1},y,{h^*}_{\mathcal{S}}(x)(\theta_{i+1}))-v(\theta_i,y,{h^*}_{\mathcal{S}}(x)(\theta_i))|\mathrm{d}\mathbf{P}_{X,Y} \\
&\leq \sup_{\sigma\in\Pi}\int\sum_{i=1}^{p-1}|v(\theta_{i+1},y,{h^*}_{\mathcal{S}}(x)(\theta_{i+1}))-v(\theta_i,y,{h^*}_{\mathcal{S}}(x)(\theta_i))|\mathrm{d}\mathbf{P}_{X,Y}.
\end{aligned}$$

The supremum of the integral is smaller than the integral of the supremum, as such

$$V(f)\leq\int V(f_{x,y})\mathrm{d}\mathbf{P}_{X,Y}, \tag{23}$$

where $f_{x,y}\colon\theta\mapsto v(\theta,y,{h^*}_{\mathcal{S}}(x)(\theta))$ is the counterpart of the function f at point (x,y). To bound this quantity, let us first bound locally $V(f_{x,y})$. To that extent, we fix some (x,y) in the following. Since $f_{x,y}$ is continuous (because k_Θ is $\mathcal{C}^1$), then using Choquet (1969, Theorem 24.6), it holds that

$$V(f_{x,y})=\lim_{|\sigma|\to 0}\sum_{i=1}^{p-1}|f_{x,y}(\theta_{i+1})-f_{x,y}(\theta_i)|.$$

Moreover since $k\in\mathcal{C}^1$ and $\partial k_\theta=(\partial_1 k)(\cdot,\theta)$ has a finite number of zeros for all $\theta\in\times$, one can assume that in the subdivision considered afterhand all the zeros (in θ) of the residuals $y-{h^*}_{\mathcal{S}}(x)(\theta)$ are present, so that $y-{h^*}_{\mathcal{S}}(x)(\theta_{i+1})$ and $y-{h^*}_{\mathcal{S}}(x)(\theta_i)$ are always of the same sign. Indeed, if not, create a new, finer subdivision with this property and work with this one. Let us begin the proper calculation: let $\sigma=(\theta_1,\theta_2,\dots,\theta_p)$ be a subdivision of Θ, it holds that $\forall i\in\{1,\dots,p-1\}$:

$$\begin{aligned}
|f_{x,y}(\theta_{i+1})-f_{x,y}(\theta_i)| = |&\max\left(\theta_{i+1}(y-{h^*}_{\mathcal{S}}(x)(\theta_{i+1})),(1-\theta_{i+1})(y-{h^*}_{\mathcal{S}}(x)(\theta_{i+1}))\right) \\
&-\max\left(\theta_i(y-{h^*}_{\mathcal{S}}(x)(\theta_i)),(1-\theta_{i+1})(y-{h^*}_{\mathcal{S}}(x)(\theta_i))\right)|.
\end{aligned}$$

We now study the two possible outcomes for the residuals:

- If $y-h(x)(\theta_{i+1})\geq 0$ and $y-h(x)(\theta_i)\geq 0$ then

$$\begin{aligned}
|f_{x,y}(\theta_{i+1})-f_{x,y}(\theta_i)| &= |\theta_{i+1}(y-{h^*}_{\mathcal{S}}(x)(\theta_{i+1}))-\theta_i(y-{h^*}_{\mathcal{S}}(x)(\theta_i))| \\
&= |(\theta_{i+1}-\theta_i)y+(\theta_i-\theta_{i+1}){h^*}_{\mathcal{S}}(x)(\theta_{i+1})+\theta_i({h^*}_{\mathcal{S}}(x)(\theta_i)-{h^*}_{\mathcal{S}}(x)(\theta_{i+1}))| \\
&\leq |(\theta_{i+1}-\theta_i)y|+|(\theta_i-\theta_{i+1}){h^*}_{\mathcal{S}}(x)(\theta_{i+1})|+|\theta_i({h^*}_{\mathcal{S}}(x)(\theta_i)-{h^*}_{\mathcal{S}}(x)(\theta_{i+1}))|.
\end{aligned}$$

From Lemma S.5.4, it holds that ${h^*}_{\mathcal{S}}(x)(\theta_{i+1})\leq\kappa_{\mathcal{X}}\kappa_\Theta\sqrt{\frac{B}{\lambda}}$. Moreover,

$$\begin{aligned}
|{h^*}_{\mathcal{S}}(x)(\theta_i)-{h^*}_{\mathcal{S}}(x)(\theta_{i+1})| &= \left|\langle h(x),k_\Theta(\theta_i,\cdot)-k_\Theta(\theta_{i+1},\cdot)\rangle_{\mathcal{H}_{k_\Theta}}\right| \\
&\leq \|h(x)\|_{\mathcal{H}_{k_\Theta}}\|k_\Theta(\theta_i,\cdot)-k_\Theta(\theta_{i+1},\cdot)\|_{\mathcal{H}_{k_\Theta}} \\
&\leq \kappa_{\mathcal{X}}\sqrt{\frac{B}{\lambda}}\sqrt{|k_\Theta(\theta_i,\theta_i)+k_\Theta(\theta_{i+1},\theta_{i+1})-2k_\Theta(\theta_{i+1},\theta_i)|} \\
&\leq \kappa_{\mathcal{X}}\sqrt{\frac{B}{\lambda}}\left(\sqrt{|k_\Theta(\theta_{i+1},\theta_{i+1})-k_\Theta(\theta_{i+1},\theta_i)|}+\sqrt{|k_\Theta(\theta_i,\theta_i)-k_\Theta(\theta_{i+1},\theta_i)|}\right).
\end{aligned}$$

Since k_Θ is $\mathcal{C}^1$, with partial derivatives uniformly bounded by C, $|k_\Theta(\theta_{i+1},\theta_{i+1}) - k_\Theta(\theta_{i+1},\theta_i)| \leq C(\theta_{i+1}-\theta_i)$ and $|k_\Theta(\theta_i,\theta_i) - k_\Theta(\theta_{i+1},\theta_i)| \leq C(\theta_{i+1}-\theta_i)$ so that $|h^*{}_\mathcal{S}(x)(\theta_i) - h^*{}_\mathcal{S}(x)(\theta_{i+1})| \leq \kappa_\mathcal{X}\sqrt{\frac{2BC}{\lambda}}\sqrt{\theta_{i+1}-\theta_i}$ and overall

$$|f_{x,y}(\theta_{i+1}) - f_{x,y}(\theta_i)| \leq \left(B + \kappa_\mathcal{X}\kappa_\Theta\sqrt{\frac{B}{\lambda}}\right)(\theta_{i+1}-\theta_i) + \kappa_\mathcal{X}\sqrt{\frac{2BC}{\lambda}}\sqrt{\theta_{i+1}-\theta_i}.$$

- If $y - h(x)(\theta_{i+1}) \leq 0$ and $y - h(x)(\theta_i) \leq 0$ then $|f_{x,y}(\theta_{i+1}) - f_{x,y}(\theta_i)| = |(1-\theta_{i+1})(y - h^*{}_\mathcal{S}(x)(\theta_{i+1})) - (1-\theta_i)(y - h^*{}_\mathcal{S}(x)(\theta_i))| \leq |h^*{}_\mathcal{S}(x)(\theta_i) - h^*{}_\mathcal{S}(x)(\theta_{i+1})| + |(\theta_{i+1}-\theta_i)y| + |(\theta_i - \theta_{i+1})h^*{}_\mathcal{S}(x)(\theta_{i+1})| + |\theta_i(h^*{}_\mathcal{S}(x)(\theta_i) - h^*{}_\mathcal{S}(x)(\theta_{i+1}))|$ so that with similar arguments one gets

$$|f_{x,y}(\theta_{i+1}) - f_{x,y}(\theta_i)| \leq \left(B + \kappa_\mathcal{X}\kappa_\Theta\sqrt{\frac{B}{\lambda}}\right)(\theta_{i+1}-\theta_i) + 2\kappa_\mathcal{X}\sqrt{\frac{2BC}{\lambda}}\sqrt{\theta_{i+1}-\theta_i}. \tag{24}$$

Therefore, regardless of the sign of the residuals $y - h(x)(\theta_{i+1})$ and $y - h(x)(\theta_i)$, one gets Eq. (24). Since the square root function has Hardy-Kraus variation of 1 on the interval $\Theta = [0,1]$, it holds that

$$\sup_{\sigma\in\Pi}\sum_{i=1}^{p-1}|f_{x,y}(\theta_{i+1}) - f_{x,y}(\theta_i)| \leq B + \kappa_\mathcal{X}\kappa_\Theta\sqrt{\frac{B}{\lambda}} + 2\kappa_\mathcal{X}\sqrt{\frac{2BC}{\lambda}}.$$

Combining this with Eq. (23) finally gives

$$V(f) \leq B + \kappa_\mathcal{X}\kappa_\Theta\sqrt{\frac{B}{\lambda}} + 2\kappa_\mathcal{X}\sqrt{\frac{2BC}{\lambda}}.$$

□

Lemma S.5.8. *Let R be the risk defined in Eq. (5) for the quantile regression problem. Assume that $(\theta)_{j=1}^m$ have been generated via the Sobol sequence and that k_Θ is $\mathcal{C}^1$ and that its partial derivatives are uniformly bounded by some constant C. Then*

$$|R(h^*{}_\mathcal{S}) - \widetilde{R}(h^*{}_\mathcal{S})| \leq \left(B + \kappa_\mathcal{X}\kappa_\Theta\sqrt{\frac{B}{\lambda}} + 2\kappa_\mathcal{X}\sqrt{\frac{2BC}{\lambda}}\right)\frac{\log(m)}{m}. \tag{25}$$

Proof. Let $f\colon\theta \mapsto \mathbf{E}_{X,Y}[v(\theta,Y,h^*{}_\mathcal{S}(X)(\theta))]$. It holds that $|R(h^*{}_\mathcal{S}) - \widetilde{R}(h^*{}_\mathcal{S})| \leq V(f)\frac{\log(m)}{m}$ according to classical Quasi-Monte Carlo approximation results, where $V(f)$ is the Hardy-Krause variation of f. Lemma S.5.7 allows then to conclude. □

Proof of Proposition 3.3. Combine Lemma S.5.8 and Corollary S.5.6 to get an asymptotic behaviour as $n, m \to \infty$. □

S.6. DLSE

To assess the quality of the estimated model by ∞-OCSVM, we illustrate the θ-property (Schölkopf et al., 2000): the proportion of inliers has to be approximately $1-\theta$ ($\forall\theta \in (0,1)$). For the studied datasets (Wilt, Spambase) we used the raw inputs without applying any preprocessing. Our input kernel was the exponentiated χ^2 kernel $k_\mathcal{X}(x,z) := \exp\left(-\gamma_\mathcal{X}\sum_{k=1}^d (x_k - z_k)^2/(x_k+z_k)\right)$ with bandwidth $\gamma_\mathcal{X} = 0.25$. A Gauss-Legendre quadrature rule provided the integral approximation in Eq. (7), with $m = 100$ samples. We chose the Gaussian kernel for k_Θ; its bandwidth parameter γ_Θ was the $0.2-$quantile of the pairwise Euclidean distances between the θ_j's obtained via the quadrature rule. The margin (bias) kernel was $k_b = k_\Theta$. As it can be seen in Fig. S.3, the θ-property holds for the estimate which illustrates the efficiency of the proposed continuum approach for density level-set estimation.

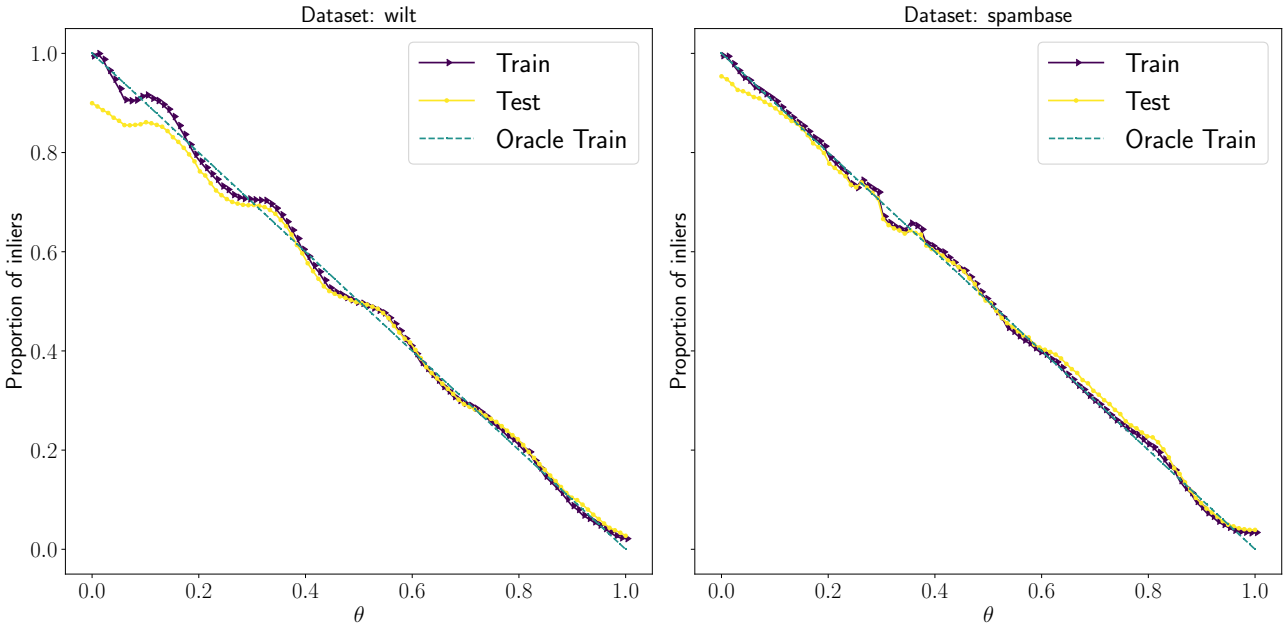

Figure S.3. Density Level-Set Estimation: the θ-property is approximately satisfied.