
Joint quantile regression in vector-valued RKHSs

Maxime Sangnier Olivier Fercoq Florence d’Alché-Buc
LTCI, CNRS, Télécom ParisTech
Université Paris-Saclay
75013, Paris, France
{maxime.sangnier, olivier.fercoq, florence.dalche}
@telecom-paristech.fr

Abstract

Addressing the will to give a more complete picture than an average relationship provided by standard regression, a novel framework for estimating and predicting simultaneously several conditional quantiles is introduced. The proposed methodology leverages kernel-based multi-task learning to curb the embarrassing phenomenon of quantile crossing, with a one-step estimation procedure and no post-processing. Moreover, this framework comes along with theoretical guarantees and an efficient coordinate descent learning algorithm. Numerical experiments on benchmark and real datasets highlight the enhancements of our approach regarding the prediction error, the crossing occurrences and the training time.

1 Introduction

Given a couple (X, Y) of random variables, where Y takes scalar values, a common aim in statistics and machine learning is to estimate the conditional expectation $\mathbb{E}[Y | X = x]$ as a function of x . In the previous setting, called regression, one assumes that the main information in Y is a scalar value corrupted by a centered noise. However, in some applications such as medicine, economics, social sciences and ecology, a more complete picture than an average relationship is required to deepen the analysis. Expectiles and quantiles are different quantities able to achieve this goal.

This paper deals with this last setting, called (conditional) quantile regression. This topic has been championed by Koenker and Bassett [16] as the minimization of the pinball loss (see [15] for an extensive presentation) and brought to the attention of the machine learning community by Takeuchi et al. [26]. Ever since then, several studies have built upon this framework and the most recent ones include regressing a single quantile of a random vector [12]. On the contrary, we are interested in estimating and predicting simultaneously several quantiles of a scalar-valued random variable $Y|X$ (see Figure 1), thus called *joint quantile regression*. For this purpose, we focus on non-parametric hypotheses from a vector-valued Reproducing Kernel Hilbert Space (RKHS).

Since quantiles of a distribution are closely related, joint quantile regression is subsumed under the field of multi-task learning [3]. As a consequence, vector-valued kernel methods are appropriate for such a task. They have already been used for various applications, such as structured classification [10] and prediction [7], manifold regularization [21, 6] and functional regression [14]. Quantile regression is a new opportunity for vector-valued RKHSs to perform in a multi-task problem, along with a loss that is different from the ℓ_2 cost predominantly used in the previous references.

In addition, such a framework offers a novel way to curb the phenomenon of quantile curve crossing, while preserving the so called *quantile property* (which may not be true for current approaches). This one guarantees that the ratio of observations lying below a predicted quantile is close to the quantile level of interest.

In a nutshell, the contributions of this work are (following the outline of the paper): **i)** a novel methodology for joint quantile regression, that is based on vector-valued RKHSs; **ii)** enhanced predictions thanks to a multi-task approach along with limited appearance of crossing curves; **iii)** theoretical guarantees regarding the generalization of the model; **iv)** an efficient coordinate descent algorithm, that is able to handle the intercept of the model in a manner that is simple and different from Sequential Minimal Optimization (SMO). Besides these novelties, the enhancements of the proposed method and the efficiency of our learning algorithm are supported by numerical experiments on benchmark and real datasets.

2 Problem definition

2.1 Quantile regression

Let $\mathcal{Y} \subset \mathbb{R}$ be a compact set, \mathcal{X} be an arbitrary input space and $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ a pair of random variables following an unknown joint distribution. For a given probability $\tau \in (0, 1)$, the conditional τ -quantile of (X, Y) is the function $\mu_\tau: \mathcal{X} \rightarrow \mathbb{R}$ such that $\mu_\tau(\mathbf{x}) = \inf\{\mu \in \mathbb{R} : \mathbb{P}(Y \leq \mu | X = \mathbf{x}) \geq \tau\}$. Thus, given a training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$, the quantile regression problem aims at estimating this conditional τ -quantile function μ_τ . Following Koenker [15], this can be achieved by minimization of the *pinball loss*: $\ell_\tau(r) = \max(\tau r, (\tau - 1)r)$, where $r \in \mathbb{R}$ is a residual. Using such a loss first arose from the observation that the location parameter μ that minimizes the ℓ_1 -loss $\sum_{i=1}^n |y_i - \mu|$ is an estimator of the unconditional median [16].

Now focusing on the estimation of a conditional quantile, one can show that the target function μ_τ is a minimizer of the τ -quantile risk $R_\tau(h) = \mathbb{E}[\ell_\tau(Y - h(X))]$ [17]. However, since the joint probability of (X, Y) is unknown but we are provided with an independent and identically distributed (*iid*) sample of observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, we resort to minimizing the empirical risk: $R_\tau^{\text{emp}}(h) = \frac{1}{n} \sum_{i=1}^n \ell_\tau(y_i - h(\mathbf{x}_i))$, within a class $\mathcal{H} \subset (\mathbb{R})^{\mathcal{X}}$ of functions, calibrated in order to overcome the shift from the true risk to the empirical one. In particular, when \mathcal{H} has the form: $\mathcal{H} = \{h = f + b : b \in \mathbb{R}, f \in (\mathbb{R})^{\mathcal{X}}, \psi(f) \leq c\}$, with $\psi: (\mathbb{R})^{\mathcal{X}} \rightarrow \mathbb{R}$ being a convex function and $c > 0$ a constant, Takeuchi et al. [26] proved that (similarly to the unconditional case) the *quantile property* is satisfied: for any estimator \hat{h} , obtained by minimizing R_τ^{emp} in \mathcal{H} , the ratio of observations lying below \hat{h} (*i.e.* $y_i < \hat{h}(\mathbf{x}_i)$) equals τ to a small error (the ration of observations exactly equal to $\hat{h}(\mathbf{x}_i)$). Moreover, under some regularity assumptions, this quantity converges to τ when the sample grows. Note that these properties are true since the intercept b is unconstrained.

2.2 Multiple quantile regression

In many real problems (such as medical reference charts), one is not only interested by estimating a single quantile curve but a few of them. Thus, denoting \mathbb{N}_p the range of integers between 1 and p , for several quantile levels τ_j ($j \in \mathbb{N}_p$) and functions $h_j \in \mathcal{H}$, the empirical loss to be minimized can be written as the following separable function: $R_\tau^{\text{emp}}(h_1, \dots, h_p) = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^p \ell_{\tau_j}(y_i - h_j(\mathbf{x}_i))$, where τ denotes the p dimensional vector of quantile levels.

A nice feature of multiple quantile regression is thus to extract slices of the conditional distribution of $Y|X$. However, when quantiles are estimated independently, an embarrassing phenomenon often appears: quantile functions cross, thus violating the basic principle that the cumulative distribution function should be monotonically non-decreasing. We refer to that pitfall as the *crossing problem*.

In this paper, we propose to prevent curve crossing by considering the problem of multiple quantile regression as a vector-valued regression problem where outputs are not independent. An interesting feature of our method is to preserve the quantile property while most other approaches lose it when struggling to the crossing problem.

2.3 Related work

Going beyond linear and spline-based models, quantile regression in RKHSs has been introduced a decade ago [26, 17]. In [26], the authors proposed to minimize the pinball loss in a scalar-valued RKHS and to add hard constraints on the training points in order to prevent the crossing problem. Our work can be legitimately seen as an extension of [26] to multiple quantile regression using

a vector-valued RKHS and structural constraints against curve crossing thanks to an appropriate matrix-valued kernel.

Another related work is [27], which first introduced the idea of multi-task learning for quantile regression. In [27], linear quantile curves are estimated jointly with a common feature subspace shared across the tasks, based on multi-task feature learning [3]. In addition, the authors showed that for such linear regressors, a common representation shared across infinitely many tasks can be computed, thus estimating simultaneously conditional quantiles for all possible quantile levels. Both previous approaches will be considered in the numerical experiments.

Quantile regression has been investigated from many perspectives, including different losses leading to an approximate quantile property (ϵ -insensitive [25], re-weighted least squares [22]) along with models and estimation procedures to curb the crossing problem: location-scale model with a multi-step strategy [13], tensor product spline surface [22], non-negative valued kernels [18], hard non-crossing constraints [26, 28, 5], inversion and monotonicization of a conditional distribution estimation [9] and rearrangement of quantile estimations [8], to cite only a few references. Let us remark that some solutions such as non-crossing constraints [26] lose theoretically the quantile property because of constraining the intercept.

In comparison to the literature, we propose a novel methodology, based on vector-valued RKHSs, with a one-step estimation, no post-processing, and keeping the quantile property while dealing with curve crossing. We also provide an efficient learning algorithm and theoretical guarantees.

3 Vector-valued RKHS for joint quantile regression

3.1 Joint estimation

Given a vector $\boldsymbol{\tau} \in (0, 1)^p$ of quantile levels, multiple quantile regression is now considered as a joint estimation in $(\mathbb{R}^p)^{\mathcal{X}}$ of the target function $\mathbf{x} \in \mathcal{X} \mapsto (\mu_{\tau_1}(\mathbf{x}), \dots, \mu_{\tau_p}(\mathbf{x})) \in \mathbb{R}^p$ of conditional quantiles. Thus, let now ψ be a convex regularizer on $(\mathbb{R}^p)^{\mathcal{X}}$ and $\mathcal{H} = \{h = f + \mathbf{b} : \mathbf{b} \in \mathbb{R}^p, f \in (\mathbb{R}^p)^{\mathcal{X}}, \psi(f) \leq c\}$ be the hypothesis set. Similarly to previously, joint quantile regression aims at minimizing $R_{\boldsymbol{\tau}}^{\text{emp}}(h) = \frac{1}{n} \sum_{i=1}^n \ell_{\boldsymbol{\tau}}(y_i \mathbf{1} - h(\mathbf{x}_i))$, where $\mathbf{1}$ stands for the all-ones vector, $\ell_{\boldsymbol{\tau}}(\mathbf{r}) = \sum_{j=1}^p \ell_{\tau_j}(r_j)$ and h is in \mathcal{H} , which is to be appropriately chosen in order to estimate the p conditional quantiles while enhancing predictions and avoiding curve crossing. It is worthwhile remarking that, independently of the choice of ψ , the quantile property is still verified for a vector-valued estimator since the loss is separable and the intercept is unconstrained. Similarly, the vector-valued function whose components are the conditional τ_j -quantiles is still a minimizer of the $\boldsymbol{\tau}$ -quantile risk $R_{\boldsymbol{\tau}}(h) = \mathbb{E}[\ell_{\boldsymbol{\tau}}(Y\mathbf{1} - h(X))]$.

In this context, the constraint ψ does not necessarily apply independently on each coordinate function h_j but can impose dependency between them. The theory of vector-valued RKHS seems especially well suited for this purpose when considering ψ as the norm associated to it. In this situation, the choice of the kernel does not only influence the nature of the hypotheses (linear, non-linear, universal approximators) but also the way the estimation procedure is regularized. In particular, the kernel critically operates on the output space by encoding structural constraints on the outputs.

3.2 Matrix-valued kernel

Let us denote \cdot^\top the transpose operator and $\mathcal{L}(\mathbb{R}^p)$ the set of linear and bounded operators from \mathbb{R}^p to itself. In our (finite) case, $\mathcal{L}(\mathbb{R}^p)$ comes down to the set of $p \times p$ real-valued matrices. A matrix-valued kernel is a function $K: \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{L}(\mathbb{R}^p)$, that is symmetric and positive [20]: $\forall(\mathbf{x}, \mathbf{x}') \in \mathcal{X} \times \mathcal{X}, K(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}', \mathbf{x})^\top$ and $\forall m \in \mathbb{N}, \forall\{\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i\}_{1 \leq i \leq m} \in (\mathcal{X} \times \mathbb{R}^p)^m, \sum_{1 \leq i, j \leq m} \langle \boldsymbol{\beta}_i | K(\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j) \boldsymbol{\beta}_j \rangle_{\ell_2} \geq 0$.

Let K be such a kernel and for any $\mathbf{x} \in \mathcal{X}$, let $K_{\mathbf{x}}: \mathbf{y} \in \mathbb{R}^p \mapsto K_{\mathbf{x}}\mathbf{y} \in (\mathbb{R}^p)^{\mathcal{X}}$ be the linear operator such that: $\forall \mathbf{x}' \in \mathcal{X}, (K_{\mathbf{x}}\mathbf{y})(\mathbf{x}') = K(\mathbf{x}', \mathbf{x})\mathbf{y}$. There exists a unique Hilbert space of functions $\mathcal{K}_K \subset (\mathbb{R}^p)^{\mathcal{X}}$ (with an inner product and a norm respectively denoted $\langle \cdot | \cdot \rangle_{\mathcal{K}}$ and $\|\cdot\|_{\mathcal{K}}$), called the RKHS associated to K , such that $\forall \mathbf{x} \in \mathcal{X}$ [20]: $K_{\mathbf{x}}$ spans the space \mathcal{K}_K ($\forall \mathbf{y} \in \mathbb{R}^p: K_{\mathbf{x}}\mathbf{y} \in \mathcal{K}$), $K_{\mathbf{x}}$ is bounded for the uniform norm ($\sup_{\mathbf{y} \in \mathbb{R}^p} \|K_{\mathbf{x}}\mathbf{y}\|_{\mathcal{K}} < \infty$) and $\forall f \in \mathcal{K}: f(\mathbf{x}) = K_{\mathbf{x}}^*f$ (reproducing property), where \cdot^* is the adjoint operator.

From now on, we assume that we are provided with a matrix-valued kernel K and we limit the hypothesis space to: $\mathcal{H} = \{f + \mathbf{b} : \mathbf{b} \in \mathbb{R}^p, f \in \mathcal{K}_K, \|f\|_{\mathcal{K}} \leq c\}$ (i.e. $\psi = \|\cdot\|_{\mathcal{K}}$). Though several candidates are available [1], we focus on one of the simplest and most efficiently computable kernels, called *decomposable kernel*: $K : (\mathbf{x}, \mathbf{x}') \mapsto k(\mathbf{x}, \mathbf{x}')\mathbf{B}$, where $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a scalar-valued kernel and \mathbf{B} is a $p \times p$ symmetric Positive Semi-Definite (PSD) matrix. In this particular case, the matrix \mathbf{B} encodes the relationship between the components f_j and thus, the link between the different conditional quantile estimators. A rational choice is to consider $\mathbf{B} = (\exp(-\gamma(\tau_i - \tau_j)^2))_{1 \leq i, j \leq p}$. To explain it, let us consider two extreme cases (see also Figure 1).

First, when $\gamma = 0$, \mathbf{B} is the all-ones matrix. Since \mathcal{K}_K is the closure of the space $\text{span}\{K_{\mathbf{x}}\mathbf{y} : (\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathbb{R}^p\}$, any $f \in \mathcal{K}_K$ has all its components equal. Consequently, the quantile estimators $h_j = f_j + b_j$ are parallel (and non-crossing) curves. In this case, the regressor is said *homoscedastic*. Second, when $\gamma \rightarrow +\infty$, then $\mathbf{B} \rightarrow \mathbf{I}$ (identity matrix). In this situation, it is easy to show that the components of $f \in \mathcal{K}_K$ are independent from each other and that $\|f\|_{\mathcal{K}}^2 = \sum_{j=1}^p \|f_j\|_{\mathcal{K}'}^2$ (where $\|\cdot\|_{\mathcal{K}'}$ is the norm coming with the RKHS associated to k) is separable. Thus, each quantile function is learned independently from the others. Regressors are said *heteroscedastic*. It appears clearly that between these two extreme cases, there is a room for learning a non-homoscedastic and non-crossing quantile regressor (while preserving the quantile property).

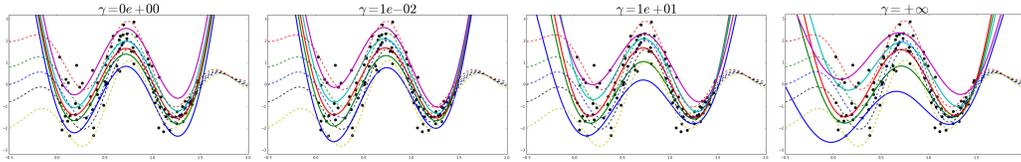


Figure 1: Estimated (plain lines) and true (dashed lines) conditional quantiles of $Y|X$ (synthetic dataset) from homoscedastic regressors ($\gamma = 0$) to heteroscedastic ones ($\gamma \rightarrow +\infty$).

4 Theoretical analysis

This section is intended to give a few theoretical insights about the expected behavior of our hypotheses. Here, we do assume working in an RKHS but not specifically with a decomposable kernel. First, we aim at providing a uniform generalization bound. For this purpose, let $\mathcal{F} = \{f \in \mathcal{K}_K, \|f\|_{\mathcal{K}} \leq c\}$, $\text{tr}(\cdot)$ be the trace operator, $((X_i, Y_i))_{1 \leq i \leq n} \in (\mathcal{X} \times \mathcal{Y})^n$ be an *iid* sample and denote $\hat{R}_n(h) = \frac{1}{n} \sum_{i=1}^n \ell_\tau(Y_i \mathbb{1} - h(X_i))$, the random variable associated to the empirical risk of a hypothesis h .

Theorem 4.1 (Generalization). *Let $a \in \mathbb{R}_+$ such that $\sup_{y \in \mathcal{Y}} |y| \leq a$, $\mathbf{b} \in \mathcal{Y}^p$ and $\mathcal{H} = \{f + \mathbf{b} : f \in \mathcal{F}\}$ be the class of hypotheses. Moreover, assume that there exists $\kappa \geq 0$ such that: $\sup_{\mathbf{x} \in \mathcal{X}} \text{tr}(K(\mathbf{x}, \mathbf{x})) \leq \kappa$. Then with probability at least $1 - \delta$ (for $\delta \in (0, 1]$):*

$$\forall h \in \mathcal{H}, \quad R(h) \leq \hat{R}_n(h) + 2\sqrt{2}c\sqrt{\frac{p\kappa}{n}} + (2pa + c\sqrt{p\kappa})\sqrt{\frac{\log(1/\delta)}{2n}}.$$

Sketch of proof (full derivation in Appendix A.1). We start with a concentration inequality for scalar-valued functions [4] and we use a vector-contraction property [19]. The bound on the Rademacher complexity of [24, Theorem 3.1] concludes the proof. \square

The uniform bound in Theorem 4.1 states that, with high probability, all the hypotheses of interest have a true risk which is less than an empirical risk to an additive bias in $O(1/\sqrt{n})$. Let us remark that it makes use of the output dimension p . However, there exist non-uniform generalization bounds for operator-valued kernel-based hypotheses, which do not depend on the output dimension [14], being thus well-suited for infinite-dimensional output spaces. Yet those results, only hold for optimal solutions \hat{h} of the learning problem, which we never obtain in practice.

As a second theoretical insight, Theorem 4.2 gives a bound on the quantile property, which is similar to the one provided in [26] for scalar-valued functions. This one states that $\mathbb{E}[\mathbb{P}(Y \leq h_j(X) | X)]$ does not deviate to much from τ_j .

Theorem 4.2 (Quantile deviation). *Let us consider that the assumptions of Theorem 4.1 hold. Moreover, let $\epsilon > 0$ be an artificial margin, $\Gamma_\epsilon^+ : r \in \mathbb{R} \mapsto \text{proj}_{[0,1]}(1 - \frac{r}{\epsilon})$ and $\Gamma_\epsilon^- : r \in \mathbb{R} \mapsto \text{proj}_{[0,1]}(-\frac{r}{\epsilon})$, two ramp functions, $j \in \mathbb{N}_p$ and $\delta \in (0, 1]$. Then with probability at least $1 - \delta$:*

$$\forall h \in \mathcal{H}, \quad \frac{1}{n} \sum_{i=1}^n \Gamma_\epsilon^-(Y_i - h_j(X_i)) - \Delta \leq \mathbb{E}[\mathbb{P}(Y \leq h_j(X) | X)] \leq \underbrace{\frac{1}{n} \sum_{i=1}^n \Gamma_\epsilon^+(Y_i - h_j(X_i))}_{\approx \tau_j} + \Delta,$$

where $\Delta = \frac{2c}{\epsilon} \sqrt{\frac{\kappa}{n}} + \sqrt{\frac{\log(2/\delta)}{2n}}$.

Sketch of proof (full derivation in Appendix A.2). The proof is similar to the one of Theorem 4.1, when remarking that Γ_ϵ^+ and Γ_ϵ^- are $1/\epsilon$ -Lipschitz continuous. \square

5 Optimization algorithm

In order to finalize the M-estimation of a non-parametric function, we need a way to jointly solve the optimization problem of interest and compute the estimator. For ridge regression in vector-valued RKHSs, representer theorems enable to reformulate the hypothesis f and to derive algorithms based on matrix inversion [20, 6] or Sylvester equation [10]. Since the optimization problem we are tackling is quite different, those methods do not apply. Yet, deriving a dual optimization problem makes it possible to hit the mark.

Quantile estimation, as presented in this paper, comes down to minimizing a regularized empirical risk, defined by the pinball loss ℓ_τ . Since this loss function is non-differentiable, we introduce slack variables ξ and ξ^* to get the following primal formulation. We also consider a regularization parameter C to be tuned:

$$\underset{\substack{f \in \mathcal{K}_K, \mathbf{b} \in \mathbb{R}^p, \\ \xi, \xi^* \in (\mathbb{R}^p)^n}}{\text{minimize}} \quad \frac{1}{2} \|f\|_{\mathcal{K}}^2 + C \sum_{i=1}^n (\langle \tau | \xi_i \rangle_{\ell_2} + \langle \mathbb{1} - \tau | \xi_i^* \rangle_{\ell_2}) \quad \text{s. t.} \quad \begin{cases} \forall i \in \mathbb{N}_n: \xi_i \succcurlyeq 0, \xi_i^* \succcurlyeq 0, \\ y_i - f(\mathbf{x}_i) - \mathbf{b} = \xi_i - \xi_i^*, \end{cases} \quad (1)$$

where \succcurlyeq is a pointwise inequality. A dual formulation of Problem (1) is (see Appendix B):

$$\underset{\alpha \in (\mathbb{R}^p)^n}{\text{minimize}} \quad \frac{1}{2} \sum_{i,j=1}^n \langle \alpha_i | K(\mathbf{x}_i, \mathbf{x}_j) \alpha_j \rangle_{\ell_2} - \sum_{i=1}^n y_i \langle \alpha_i | \mathbb{1} \rangle_{\ell_2} \quad \text{s. t.} \quad \begin{cases} \sum_{i=1}^n \alpha_i = 0_{\mathbb{R}^p}, \forall i \in \mathbb{N}_n: \\ C(\tau - \mathbb{1}) \preccurlyeq \alpha_i \preccurlyeq C\tau, \end{cases} \quad (2)$$

where the linear constraints come from considering an intercept \mathbf{b} . The Karush-Kuhn-Tucker (KKT) conditions of Problem (1) indicate that a minimizer \hat{f} of (1) can be recovered from a solution $\hat{\alpha}$ of (2) with the formula $\hat{f} = \sum_{i=1}^n K_{\mathbf{x}_i} \hat{\alpha}_i$. Moreover, $\hat{\mathbf{b}}$ can also be obtained thanks to KKT conditions. However, as we deal with a numerical approximate solution α , in practice \mathbf{b} is computed by solving Problem (1) with f fixed. This boils down to taking b_j as the τ_j -quantile of $(y_i - f_j(\mathbf{x}_i))_{1 \leq i \leq n}$.

Problem (2) is a common quadratic program that can be solved with off-the-shelf solvers. However, since we are essentially interested in decomposable kernels $K(\cdot, \cdot) = k(\cdot, \cdot)\mathbf{B}$, it appears that the quadratic part of the objective function would be defined by the $np \times np$ matrix $\mathbf{K} \otimes \mathbf{B}$, where \otimes is the Kronecker product and $\mathbf{K} = (k(\mathbf{x}_i, \mathbf{x}_j))_{1 \leq i, j \leq n}$. Storing this matrix explicitly is likely to be time and memory expensive. In order to improve the estimation procedure, ad hoc algorithms can be derived. For instance, regression with a decomposable kernel boils down to solving a Sylvester equation (which can be done efficiently) [10] and vector-valued Support Vector Machine (SVM) without intercept can be learned with a coordinate descent algorithm [21]. However, these methods can not be used in our setting since the loss function is different and considering the intercept is necessary for the quantile property. Yet, coordinate descent could theoretically be extended in an SMO technique, able to handle the linear constraints introduced by the intercept. However, SMO works usually with a single linear constraint and needs heuristics to run efficiently, which are quite difficult to find (even though an implementation exists for two linear constraints [25]).

Therefore, for the sake of efficiency, we propose to use a Primal-Dual Coordinate Descent (PDCD) technique, recently introduced in [11]. This algorithm (which is proved to converge) is able to deal with the linear constraints coming from the intercept and is thus utterly workable for the problem at hand. Moreover, PDCD has been proved favorably competitive with SMO for SVMs.

Table 1: Empirical pinball loss and crossing loss $\times 100$ (the less, the better). Bullets (resp. circles) indicate statistically significant (resp non-significant) differences. The proposed method is JQR.

Data set	Pinball loss				Crossing loss			
	IND.	IND. (NC)	MTFL	JQR	IND.	IND. (NC)	MTFL	JQR
caution	- 102.6 \pm 17.3	103.2 \pm 17.2	102.9 \pm 19.0	ooo 102.6 \pm 19.0	- 0.53 \pm 0.67	0.31 \pm 0.70	0.69 \pm 0.54	ooo 0.09 \pm 0.14
fccollinsnow	- 151.1 \pm 8.2	150.8 \pm 8.0	152.4 \pm 8.9	ooo 153.7 \pm 12.1	- 0.00 \pm 0.00	0.00 \pm 0.00	0.00 \pm 0.00	ooo 0.00 \pm 0.00
highway	- 102.9 \pm 39.1	102.8 \pm 38.9	102.0 \pm 34.5	ooo 103.7 \pm 35.7	- 9.08 \pm 7.38	9.00 \pm 7.39	3.48 \pm 4.49	ooo 8.81 \pm 7.46
heights	- 128.2 \pm 2.4	128.2 \pm 2.4	128.6 \pm 2.2	ooo 127.9 \pm 1.8	- 0.04 \pm 0.05	0.04 \pm 0.05	0.07 \pm 0.14	ooo 0.00 \pm 0.00
sniffer	- 44.8 \pm 6.7	44.6 \pm 6.8	46.9 \pm 7.6	ooo 45.2 \pm 6.9	- 1.01 \pm 0.75	0.52 \pm 0.48	1.23 \pm 0.77	ooo 0.15 \pm 0.22
snowgeese	- 68.4 \pm 35.3	68.4 \pm 35.3	75.3 \pm 38.2	ooo 76.0 \pm 31.5	- 3.24 \pm 5.10	2.60 \pm 4.28	8.93 \pm 19.52	ooo 0.94 \pm 3.46
ufc	- 81.8 \pm 4.6	81.6 \pm 4.6	84.9 \pm 4.7	ooo 80.6 \pm 4.1	- 0.24 \pm 0.22	0.27 \pm 0.42	0.82 \pm 1.47	ooo 0.05 \pm 0.15
birthwt	- 139.0 \pm 9.9	139.0 \pm 9.9	142.6 \pm 11.6	ooo 139.8 \pm 11.7	- 0.00 \pm 0.00	0.00 \pm 0.00	0.31 \pm 0.88	ooo 0.00 \pm 0.00
crabs	- 12.3 \pm 1.0	12.3 \pm 1.0	12.6 \pm 1.0	ooo 11.9 \pm 0.9	- 0.46 \pm 0.33	0.35 \pm 0.24	0.30 \pm 0.22	ooo 0.06 \pm 0.20
GAGurine	- 62.6 \pm 8.2	62.6 \pm 8.2	64.5 \pm 7.5	ooo 62.6 \pm 8.1	- 0.05 \pm 0.08	0.04 \pm 0.07	0.05 \pm 0.09	ooo 0.03 \pm 0.08
geyser	- 110.2 \pm 7.8	110.1 \pm 7.8	109.4 \pm 7.1	ooo 111.3 \pm 8.2	- 0.87 \pm 1.60	0.92 \pm 2.02	0.80 \pm 1.18	ooo 0.72 \pm 1.51
gilgais	- 47.4 \pm 4.4	47.2 \pm 4.4	49.9 \pm 3.6	ooo 46.9 \pm 4.6	- 1.23 \pm 0.96	0.95 \pm 0.85	0.71 \pm 0.96	ooo 0.81 \pm 0.43
topo	- 71.1 \pm 13.0	70.1 \pm 13.7	73.1 \pm 11.8	ooo 69.6 \pm 13.4	- 2.72 \pm 3.26	1.52 \pm 2.47	2.75 \pm 2.93	ooo 1.14 \pm 2.02
BostonHousing	- 48.5 \pm 5.0	48.5 \pm 5.0	49.7 \pm 4.7	ooo 47.4 \pm 4.7	- 0.64 \pm 0.32	0.48 \pm 0.27	1.11 \pm 0.33	ooo 0.58 \pm 0.34
CobarOre	- 0.5 \pm 0.5	0.5 \pm 0.5	5.0 \pm 4.9	ooo 0.6 \pm 0.5	- 0.10 \pm 0.13	0.10 \pm 0.13	0.29 \pm 0.35	ooo 0.02 \pm 0.05
engel	- 61.3 \pm 18.3	61.2 \pm 19.0	58.7 \pm 17.9	ooo 64.4 \pm 23.2	- 1.50 \pm 4.94	1.25 \pm 4.53	1.65 \pm 5.97	ooo 0.06 \pm 0.14
mcycle	- 89.2 \pm 8.5	88.9 \pm 8.4	102.0 \pm 11.7	ooo 84.3 \pm 10.3	- 2.10 \pm 1.83	0.92 \pm 1.25	1.13 \pm 1.10	ooo 0.14 \pm 0.37
BigMac2003	- 71.0 \pm 21.0	70.9 \pm 21.1	68.7 \pm 18.1	ooo 67.6 \pm 20.9	- 2.50 \pm 2.12	1.87 \pm 1.68	0.73 \pm 0.92	ooo 1.55 \pm 1.75
UN3	- 99.5 \pm 7.0	99.4 \pm 7.0	101.8 \pm 7.1	ooo 98.8 \pm 7.6	- 1.06 \pm 0.85	0.85 \pm 0.70	0.65 \pm 0.62	ooo 0.09 \pm 0.31
cpus	- 20.0 \pm 13.7	19.9 \pm 13.6	23.8 \pm 16.0	ooo 19.7 \pm 13.7	- 1.29 \pm 1.13	1.17 \pm 1.15	0.46 \pm 0.28	ooo 0.09 \pm 0.13

PDCD is described in Algorithm 1, where, for $\alpha = (\alpha_i)_{1 \leq i \leq n} \in (\mathbb{R}^p)^n$, $\alpha^j \in \mathbb{R}^n$ denotes its j^{th} row vector and α_i^j its i^{th} component, diag is the operator mapping a vector to a diagonal matrix and $\text{proj}_{\mathbb{1}}$ and $\text{proj}_{[C(\tau_l - 1), C\tau_l]}$ are respectively the projectors onto the vector $\mathbb{1}$ and the compact set $[C(\tau_l - 1), C\tau_l]$. PDCD uses dual variables $\theta \in (\mathbb{R}^p)^n$ (which are updated during the descent) and has two sets of parameters $\nu \in (\mathbb{R}^p)^n$ and $\mu \in (\mathbb{R}^p)^n$, that verify $(\forall (i, l) \in \mathbb{N}_n \times \mathbb{N}_p)$: $\mu_i^l < \frac{1}{(K(\mathbf{x}_i, \mathbf{x}_i))_{l,l} + \nu_i^l}$. In practice, we kept the same parameters as in [11]: $\nu_i^l = 10(K(\mathbf{x}_i, \mathbf{x}_i))_{l,l}$ and μ_i^l equal to 0.95 times the bound. Moreover, as it is standard for coordinate descent methods, our implementation uses efficient updates for the computation of both $\sum_{j=1}^n K(\mathbf{x}_i, \mathbf{x}_j)\alpha_j$ and $\bar{\theta}^l$.

6 Numerical experiments

Two sets of experiments are presented, respectively aimed at assessing the ability of our methodology to predict quantiles and at comparing an implementation of Algorithm 1 with an off-the-shelf solver and an augmented Lagrangian scheme. Following the previous sections, a decomposable kernel $K(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}')\mathbf{B}$ is used, where $\mathbf{B} = (\exp(-\gamma(\tau_i - \tau_j)^2))_{1 \leq i, j \leq p}$ and $k(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|_{\ell_2}^2 / 2\sigma^2)$, with σ being the 0.7-quantile of the pairwise distances of the training data $\{\mathbf{x}_i\}_{1 \leq i \leq n}$. Quantile levels of interest are $\tau = (0.1, 0.3, 0.5, 0.7, 0.9)$.

6.1 Quantile regression

Quantile regression is assessed with two criteria: the pinball loss $\frac{1}{n} \sum_{i=1}^n \ell_{\tau}(y_i \mathbb{1} - h(\mathbf{x}_i))$ is the one minimized to build the proposed estimator and the crossing loss $\sum_{j=1}^{p-1} [\frac{1}{n} \sum_{i=1}^n \max(0, h_{j+1}(\mathbf{x}_i) - h_j(\mathbf{x}_i))]$, assuming that $\tau_j > \tau_{j+1}$, quantifies how far h_j goes below h_{j+1} , while h_j is expected to stay always above h_{j+1} . More experiments are in Appendix D.1.

This study focuses on three non-parametric models based on the RKHS theory. Other linear and spline-based models have been dismissed since Takeuchi et al. [26] have already provided a comparison of these ones with kernel methods. First, we considered an independent estimation of quantile regressors (IND.), which boils down to setting $\mathbf{B} = \mathbf{I}$ (this approach could be set up without vector-valued RKHSs but with scalar-valued kernels only). Second, hard non-crossing constraints on the training data have been imposed (IND. (NC)), as proposed in [26]. Third, the proposed joint estimator (JQR) uses the Gaussian matrix \mathbf{B} presented above.

Quantile regression with multi-task feature learning (MTFL), as proposed in [27], is also included. For a fair comparison, each point is mapped with $\psi(\mathbf{x}) = (k(\mathbf{x}, \mathbf{x}_1), \dots, k(\mathbf{x}, \mathbf{x}_n))$ and the estimator $h(\mathbf{x}) = \mathbf{W}^T \psi(\mathbf{x}) + \mathbf{b}$ ($\mathbf{W} \in \mathbb{R}^{n \times p}$) is learned jointly with the PSD matrix $\mathbf{D} \in \mathbb{R}^{n \times n}$ of the

Algorithm 1 Primal-Dual Coordinate Descent.

Initialize $\alpha_i, \theta_i \in \mathbb{R}^p$ ($\forall i \in \mathbb{N}_n$).
repeat
 Choose $(i, l) \in \mathbb{N}_n \times \mathbb{N}_p$ uniformly at random.
 Set $\bar{\theta}^l \leftarrow \text{proj}_{\mathbb{I}}(\theta^l + \text{diag}(\nu^l)\alpha^l)$.
 Set $d_i^l \leftarrow \sum_{j=1}^n (K(\mathbf{x}_i, \mathbf{x}_j)\alpha_j)^l - y_i + 2\bar{\theta}_i^l - \theta_i^l$.
 Set $\bar{\alpha}_i^l \leftarrow \text{proj}_{[C(\tau_l-1), C\tau_l]}(\alpha_i^l - \mu_i^l d_i^l)$.
 Update coordinate (i, l) : $\alpha_i^l \leftarrow \bar{\alpha}_i^l, \theta_i^l \leftarrow \bar{\theta}_i^l$,
 and keep other coordinates unchanged.
until duality gap (1)-(2) is small enough

Table 2: CPU time (s) for training a model.

Size	QP	AUG. LAG.	PDCD
250	8.73 \pm 0.34	261.11 \pm 46.69	18.69 \pm 3.54
500	75.53 \pm 2.98	865.86 \pm 92.26	61.30 \pm 7.05
1000	621.60 \pm 30.37	-	266.50 \pm 41.16
2000	3416.55 \pm 104.41	-	958.93 \pm 107.80

regularizer $\psi(h) = \text{tr}(\mathbf{W}^\top \mathbf{D}^{-1} \mathbf{W})$. This comes down to alternating our approach (with $\mathbf{B} = \mathbf{I}$ and $k(\cdot, \cdot) = \langle \cdot | \mathbf{D} \cdot \rangle_{\ell_2}$) and the update $\mathbf{D} \leftarrow (\mathbf{W} \mathbf{W}^\top)^{1/2} / \text{tr}((\mathbf{W} \mathbf{W}^\top)^{1/2})$.

To present an honorable comparison of these four methods, we did not choose datasets for the benefit of our method but considered the ones used in [26]. These 20 datasets (whose names are indicated in Table 1) come from the UCI repository and three R packages: quantreg, alr3 and MASS. The sample sizes vary from 38 (CobarOre) to 1375 (heights) and the numbers of explanatory variables vary from 1 (5 sets) to 12 (BostonHousing). The datasets were standardized coordinate-wise to have zero mean and unit variance. Results are given in Table 1 thanks to the mean and the standard deviation of the test losses recorded on 20 random splits train-test with ratio 0.7-0.3. The best result of each line is boldfaced and the bullets indicate the significant differences of each competitor from JQR (based on a Wilcoxon signed-rank test with significance level 0.05).

The parameter C is chosen by cross-validation (minimizing the pinball loss) inside a logarithmic grid ($10^{-5}, 10^{-4}, \dots, 10^5$) for all methods and datasets. For our approach (JQR), the parameter γ is chosen in the same grid as C with extra candidates 0 and $+\infty$. Finally, for a balanced comparison, the dual optimization problems corresponding to each approach are solved with CVXOPT [2].

Regarding the pinball loss, joint quantile regression compares favorably to independent and hard non-crossing constraint estimations for 12 vs 8 datasets (5 vs 1 significantly different). These results bear out the assumption concerning the relationship between conditional quantiles and the usefulness of multiple-output methods for quantile regression. Prediction is also enhanced compared to MTFI for 15 vs 5 datasets (11 vs 1 significantly different).

The crossing loss clearly shows that joint regression enables to weaken the crossing problem, in comparison to independent estimation and hard non-crossing constraints (18 vs 1 favorable datasets and 9 vs 0 significantly different). Results are similar compared to MTFI (16 vs 3, 12 vs 1). Note that for IND. (NC), the crossing loss is null on the training data by construction but not necessarily on the test data. In addition, let us remark that model selection (and particularly for γ , which tunes the trade-off between hetero and homoscedastic regressors) has been performed based on the pinball loss only. It seems that, in a way, the pinball loss embraces the crossing loss as a subcriterion.

6.2 Learning algorithms

This section is aimed at comparing three implementations of algorithms for estimating joint quantile regressors (solving Problem 2), following their running (CPU) time. First, the off-the-shelf solver (based on an interior-point method) included in CVXOPT [2] (QP) is applied to Problem (2) turned into a standard form of linearly constrained quadratic program. Second, an augmented Lagrangian scheme (AUG. LAG) is used in order to get rid of the linear constraints and to make it possible to use a coordinate descent approach (detailed procedure in Appendix C). In this scheme, the inner solver is Algorithm 1 when the intercept is dismissed, which boils down to be the algorithm proposed in [23]. The last approach (PDCD) is Algorithm 1.

We use a synthetic dataset (the same as in Figure 1), for which $X \in [0, 1.5]$. The target Y is computed as a sine curve at 1 Hz modulated by a sine envelope at 1/3 Hz and mean 1. Moreover, this pattern is distorted with a random Gaussian noise with mean 0 and a linearly decreasing standard deviation from 1.2 at $X = 0$ to 0.2 at $X = 1.5$. Parameters for the models are: $(C, \gamma) = (10^2, 10^{-2})$.

To compare the implementations of the three algorithms, we first run QP, with a relative tolerance set to 10^{-2} , and store the optimal objective value. Then, the two other methods (AUG. LAG and PDCD) are launched and stopped when they pass the objective value reached by QP (optimal objective values are reported in Appendix D.2). Table 2 gives the mean and standard deviation of the CPU time required by each method for 10 random datasets and several sample sizes. Some statistics are missing because AUG. LAG. ran out of time.

As expected, it appears that for a not too tight tolerance and big datasets, implementation of Algorithm 1 outperforms the two other competitors. Let us remark that QP is also more expensive in memory than the coordinate-based algorithms like ours. Moreover, training time may seem high in comparison to usual SVMs. However, let us first remind that we jointly learn p regressors. Thus, a fair comparison should be done with an SVM applied to an $np \times np$ matrix, instead of $n \times n$. In addition, there is no sample sparsity in quantile regression, which does speed up SVM training.

Last but not least, in order to illustrate the use of our algorithm, we have run it on two 2000-point datasets from economics and medicine: the U.S. 2000 Census data, consisting of annual salary and 9 related features on workers, and the 2014 National Center for Health Statistics’ data, regarding girl birth weight and 16 statistics on parents.¹ Parameters (C, γ) have been set to $(1, 100)$ and $(0.1, 1)$ respectively for the Census and NCHS datasets (determined by cross-validation). Figure 2 depicts 9 estimated conditional quantiles of the salary with respect to the education (17 levels from no schooling completed to doctorate degree) and of the birth weight (in grams) vs mother’s pre-pregnancy weight (in pounds). As expected, the Census data reveal an increasing and heteroscedastic trend while new-born’s weight does not seem correlated to mother’s weight.

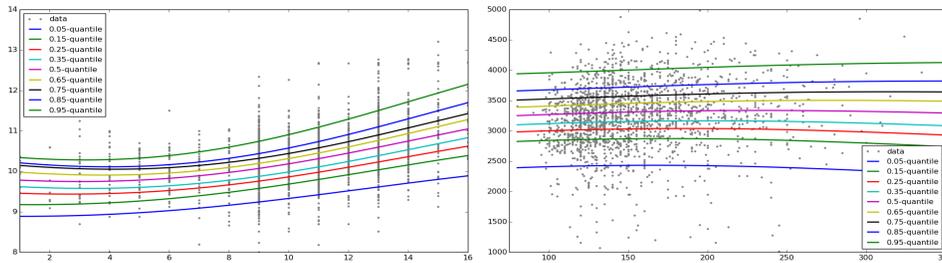


Figure 2: Estimated conditional quantiles for the Census (left, salary vs education) and the NCHS data (right, birth weight vs mother’s pre-pregnancy weight).

7 Conclusion

This paper introduces a novel framework for joint quantile regression, which is based on vector-valued RKHSs. It comes along with theoretical guarantees and an efficient learning algorithm. Moreover, this methodology, which keeps the quantile property, enjoys few curve crossing and enhanced performances compared to independent estimations and hard non-crossing constraints.

To go forward, let us remark that this framework benefits from all the tools now associated with vector-valued RKHSs, such as manifold learning for the semi-supervised setting, multiple kernel learning for measuring feature importance and random Fourier features for very large scale applications. Moreover, extensions of our methodology to multivariate output variables are to be investigated, given that it requires to choose among the various definitions of multivariate quantiles.

Acknowledgments

This work was supported by the industrial chair “Machine Learning for Big Data”.

References

[1] M.A. Alvarez, L. Rosasco, and N.D. Lawrence. Kernels for Vector-Valued Functions: a Review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012. arXiv: 1106.6251.

¹Data are available at www.census.gov/census2000/PUMS5.html and www.nber.org/data/vital-statistics-natality-data.html.

- [2] M.S. Anderson, J. Dahl, and L. Vandenberghe. CVXOPT: A Python package for convex optimization, version 1.1.5., 2012.
- [3] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3): 243–272, 2008.
- [4] P.L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.
- [5] H.D. Bondell, B.J. Reich, and H. Wang. Noncrossing quantile regression curve estimation. *Biometrika*, 97(4):825–838, 2010.
- [6] C. Brouard, F. d’Alché Buc, and M. Szafranski. Semi-supervised Penalized Output Kernel Regression for Link Prediction. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.
- [7] C. Brouard, M. Szafranski, and F. d’Alché Buc. Input Output Kernel Regression: Supervised and Semi-Supervised Structured Output Prediction with Operator-Valued Kernels. *Journal of Machine Learning Research*, 17(176):1–48, 2016.
- [8] V. Chernozhukov, I. Fernández-Val, and A. Galichon. Quantile and Probability Curves Without Crossing. *Econometrica*, 78(3):1093–1125, 2010.
- [9] H. Dette and S. Volgushev. Non-crossing non-parametric estimates of quantile curves. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(3):609–627, 2008.
- [10] F. Dinuzzo, C.S. Ong, P. Gehler, and G. Pillonetto. Learning Output Kernels with Block Coordinate Descent. In *Proceedings of the 28th International Conference of Machine Learning*, 2011.
- [11] O. Fercoq and P. Bianchi. A Coordinate Descent Primal-Dual Algorithm with Large Step Size and Possibly Non Separable Functions. *arXiv:1508.04625 [math]*, 2015.
- [12] M. Hallin and M. Šiman. Elliptical multiple-output quantile regression and convex optimization. *Statistics & Probability Letters*, 109:232–237, 2016.
- [13] X. He. Quantile Curves without Crossing. *The American Statistician*, 51(2):186–192, 1997.
- [14] H. Kadri, E. Duflos, P. Preux, S. Canu, A. Rakotomamonjy, and J. Audiffren. Operator-valued Kernels for Learning from Functional Response Data. *Journal of Machine Learning Research*, 16:1–54, 2015.
- [15] R. Koenker. *Quantile Regression*. Cambridge University Press, Cambridge, New York, 2005.
- [16] R. Koenker and G. Bassett. Regression Quantiles. *Econometrica*, 46(1):33–50, 1978.
- [17] Y. Li, Y. Liu, and J. Zhu. Quantile Regression in Reproducing Kernel Hilbert Spaces. *Journal of the American Statistical Association*, 102(477):255–268, 2007.
- [18] Y. Liu and Y. Wu. Simultaneous multiple non-crossing quantile regression estimation using kernel constraints. *Journal of nonparametric statistics*, 23(2):415–437, 2011.
- [19] A. Maurer. A vector-contraction inequality for Rademacher complexities. In *Proceedings of The 27th International Conference on Algorithmic Learning Theory*, 2016.
- [20] C.A. Micchelli and M.A. Pontil. On Learning Vector-Valued Functions. *Neural Computation*, 17:177–204, 2005.
- [21] H.Q. Minh, L. Bazzani, and V. Murino. A Unifying Framework in Vector-valued Reproducing Kernel Hilbert Spaces for Manifold Regularization and Co-Regularized Multi-view Learning. *Journal of Machine Learning Research*, 17(25):1–72, 2016.
- [22] S.K. Schnabel and P.H.C. Eilers. Simultaneous estimation of quantile curves using quantile sheets. *AStA Advances in Statistical Analysis*, 97(1):77–87, 2012.
- [23] S. Shalev-Shwartz and T. Zhang. Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization. *Journal of Machine Learning Research*, 14:567–599, 2013.
- [24] V. Sindhwani, M.H. Quang, and A.C. Lozano. Scalable Matrix-valued Kernel Learning for High-dimensional Nonlinear Multivariate Regression and Granger Causality. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, 2013.
- [25] I. Takeuchi and T. Furuhashi. Non-crossing quantile regressions by SVM. In *2004 IEEE International Joint Conference on Neural Networks, 2004. Proceedings*, July 2004.
- [26] I. Takeuchi, Q.V. Le, T.D. Sears, and A.J. Smola. Nonparametric Quantile Estimation. *Journal of Machine Learning Research*, 7:1231–1264, 2006.
- [27] I. Takeuchi, T. Hongo, M. Sugiyama, and S. Nakajima. Parametric Task Learning. In *Advances in Neural Information Processing Systems 26*, pages 1358–1366. Curran Associates, Inc., 2013.
- [28] Y. Wu and Y. Liu. Stepwise multiple quantile regression estimation using non-crossing constraints. *Statistics and Its Interface*, 2:299–310, 2009.

Supplementary: Joint quantile regression in vector-valued RKHSs

Maxime Sangnier Olivier Fercoq Florence d'Alché-Buc
LTCI, CNRS, Télécom ParisTech
Université Paris-Saclay
75013, Paris, France
{maxime.sangnier, olivier.fercoq, florence.dalche}
@telecom-paristech.fr

A Detailed proofs

A.1 Generalization

This section describes a proof of the generalization bound given in the corpus of the paper. The result is based on a concentration inequality à la Bartlett and Mendelson [1] for vector-valued functions [4], along with a bound on the Rademacher complexity of operator-valued kernel based hypothesis sets [7]. Before stating these two critical properties, let us remind the definition of the Rademacher complexity, used to quantify the complexity of a class of functions.

Definition A.1 (Rademacher complexity [4]). *Let $(X_i)_{1 \leq i \leq n} \in \mathcal{X}^n$ be an independent and identically distributed (iid) sample of random variables and $(\epsilon_{i,j})_{\substack{1 \leq i \leq n \\ 1 \leq j \leq p}} \in \{-1, 1\}^{n \times p}$ be $n \times p$ independent Rademacher variables (i.e. uniformly distributed on $\{-1, 1\}$). Let now $\mathcal{F} \subset (\mathbb{R}^p)^\mathcal{X}$ be a class of functions from \mathcal{X} to \mathbb{R}^p . The Rademacher complexity (or average) of the class \mathcal{F} is defined as:*

$$\mathcal{R}_n(\mathcal{F}) = \mathbb{E} \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq p}} \epsilon_{i,j} f_j(X_i) \right],$$

where the expectation is computed jointly on $(X_i)_{1 \leq i \leq n}$ and $(\epsilon_{i,j})_{\substack{1 \leq i \leq n \\ 1 \leq j \leq p}}$.

Proposition A.1 (Concentration for Lipschitz hypotheses). *Let $X \in \mathcal{X}$ and $(X_i)_{1 \leq i \leq n} \in \mathcal{X}^n$ be iid random variables, $\mathcal{F} \subset (\mathbb{R}^p)^\mathcal{X}$ a class of functions. Let $\phi: \mathbb{R}^p \rightarrow [a, b]$ ($a, b \in \mathbb{R}$) be a Lipschitz continuous mapping with Lipschitz constant L_ϕ :*

$$\forall (\mathbf{z}, \mathbf{z}') \in \mathbb{R}^p: |\phi(\mathbf{z}) - \phi(\mathbf{z}')| \leq L_\phi \|\mathbf{z} - \mathbf{z}'\|_{\ell_2}.$$

Let $\delta \in (0, 1]$, then with probability at least $1 - \delta$:

$$\sup_{f \in \mathcal{F}} \left(\mathbb{E} [\phi(f(X))] - \frac{1}{n} \sum_{i=1}^n \phi(f(X_i)) \right) \leq 2\sqrt{2}L_\phi \mathcal{R}_n(\mathcal{F}) + (b - a) \sqrt{\frac{\log(1/\delta)}{2n}}.$$

Proof. Thanks to the assumptions above, with probability at least $1 - \delta$, we have [1, 3]:

$$\sup_{f \in \mathcal{F}} \left(\mathbb{E} [\phi(f(X))] - \frac{1}{n} \sum_{i=1}^n \phi(f(X_i)) \right) \leq 2\mathcal{R}_n(\Phi) + (b - a) \sqrt{\frac{\log(1/\delta)}{2n}},$$

where $\Phi = \{\phi \circ f : f \in \mathcal{F}\}$. Then, using [5, Corollary 1], we obtain:

$$\mathcal{R}_n(\Phi) \leq \sqrt{2}L_\phi \mathcal{R}_n(\mathcal{F}).$$

Gathering both equations concludes the proof. □

Proposition A.2 (Bound on the Rademacher average [7, Theorem 3.1]). Assume that there exists $\kappa \in \mathbb{R}_+$ such that: $\sup_{\mathbf{x} \in \mathcal{X}} \text{tr}(K(\mathbf{x}, \mathbf{x})) \leq \kappa$ and let $\mathcal{F} = \{f \in \mathcal{K}_K, \|f\|_{\mathcal{K}} \leq c\}$ for a given $c \in \mathbb{R}_+$. Then:

$$\mathcal{R}_n(\mathcal{F}) \leq c \sqrt{\frac{\kappa}{n}}.$$

Theorem A.3 (Generalization). Let $\tau \in (0, 1)^p$, $((X_i, Y_i))_{1 \leq i \leq n} \in (\mathcal{X} \times \mathcal{Y})^n$ be iid random variables (independent from (X, Y)), $a \in \mathbb{R}_+$ such that $\sup_{y \in \mathcal{Y}} |y| \leq a$ and $\mathbf{b} \in \mathcal{Y}^p$. Let $\mathcal{H} = \{f + \mathbf{b}: f \in \mathcal{K}_K / \|f\|_{\mathcal{K}} \leq c\}$, for a given $c \in \mathbb{R}_+$, be the class of hypotheses. Moreover, assume that there exists $\kappa \in \mathbb{R}_+$ such that: $\sup_{\mathbf{x} \in \mathcal{X}} \text{tr}(K(\mathbf{x}, \mathbf{x})) \leq \kappa$ and, for a hypothesis h , let us denote

$$\hat{R}_n(h) = \frac{1}{n} \sum_{i=1}^n \ell_{\tau}(Y_i \mathbf{1} - h(X_i)),$$

the random variable associated to the empirical risk. Let $\delta \in (0, 1]$, then with probability at least $1 - \delta$:

$$\forall h \in \mathcal{H}: R(h) \leq \hat{R}_n(h) + 2\sqrt{2}c \sqrt{\frac{p\kappa}{n}} + (2pa + c\sqrt{p\kappa}) \sqrt{\frac{\log(1/\delta)}{2n}}.$$

Proof. Let $\mathcal{F} = \{f \in \mathcal{K}_K, \|f\|_{\mathcal{K}} \leq c\}$. The proof begins with the following lemma.

Lemma A.4. Under the assumptions of Theorem A.3: $\forall (f, \mathbf{x}) \in \mathcal{F} \times \mathcal{X}, \|f(\mathbf{x})\|_{\ell_2} \leq c\sqrt{\kappa}$.

Proof of Lemma A.4.

$$\begin{aligned} \forall (f, \mathbf{x}, \mathbf{y}) \in \mathcal{F} \times \mathcal{X} \times \mathbb{R}^p, \\ \langle f(\mathbf{x}) | \mathbf{y} \rangle_{\ell_2} &= \langle K_{\mathbf{x}\mathbf{y}} | f \rangle_{\mathcal{K}} \\ &\leq \|f\|_{\mathcal{K}} \|K_{\mathbf{x}\mathbf{y}}\|_{\mathcal{K}} \\ &\leq c \sqrt{\langle K_{\mathbf{x}\mathbf{y}} | K_{\mathbf{x}\mathbf{y}} \rangle_{\mathcal{K}}} \\ &= c \sqrt{\langle \mathbf{y} | K(\mathbf{x}, \mathbf{x}) \mathbf{y} \rangle_{\ell_2}}. \end{aligned}$$

From the properties of operator-valued kernels, we know that $K(\mathbf{x}, \mathbf{x})$ is symmetric positive semi-definite. Thus, denoting $(\lambda_j)_{1 \leq j \leq p}$ its (non-negative) eigenvalues and $(e_j)_{1 \leq j \leq p}$ the corresponding orthonormal basis, we obtain when $\|\mathbf{y}\|_{\ell_2} \leq 1$:

$$\begin{aligned} \langle \mathbf{y} | K(\mathbf{x}, \mathbf{x}) \mathbf{y} \rangle_{\ell_2} &= \sum_{i,j} \langle \mathbf{y} | e_i \rangle_{\ell_2} \langle \mathbf{y} | e_j \rangle_{\ell_2} \langle e_i | K(\mathbf{x}, \mathbf{x}) e_j \rangle_{\ell_2} \\ &= \sum_j \langle \mathbf{y} | e_j \rangle_{\ell_2}^2 \lambda_j \\ &\leq \sum_j \lambda_j && \text{(since } \|\mathbf{y}\|_{\ell_2} \leq 1) \\ &= \text{tr}(K(\mathbf{x}, \mathbf{x})). \end{aligned}$$

Thus: $\forall (f, \mathbf{x}) \in \mathcal{F} \times \mathcal{X}, \|f(\mathbf{x})\|_{\ell_2} = \sup_{\|\mathbf{y}\|_{\ell_2} \leq 1} \langle f(\mathbf{x}) | \mathbf{y} \rangle_{\ell_2} \leq c\sqrt{\kappa}$. □

In order to apply Proposition A.1, let us observe that the loss function ℓ_τ is \sqrt{p} -Lipschitz:

$$\begin{aligned}
& \forall (\mathbf{r}, \mathbf{r}') \in \mathbb{R}^p, \\
& \ell_\tau(\mathbf{r}) = \ell_\tau(\mathbf{r} - \mathbf{r}' + \mathbf{r}') \\
& = \sum_{j=1}^p \max(\tau_j(r_j - r'_j + r'_j), (\tau_j - 1)(r_j - r'_j + r'_j)) \\
& = \sum_{j=1}^p \begin{cases} \tau_j(r_j - r'_j + r'_j) & \text{if } r_j - r'_j + r'_j \geq 0 \\ (\tau_j - 1)(r_j - r'_j + r'_j) & \text{if } r_j - r'_j + r'_j \leq 0 \end{cases} \\
& = \sum_{j=1}^p \begin{cases} \tau_j(r_j - r'_j) + \tau_j r'_j & \text{if } r_j - r'_j + r'_j \geq 0 \\ (\tau_j - 1)(r_j - r'_j) + (\tau_j - 1)r'_j & \text{if } r_j - r'_j + r'_j \leq 0 \end{cases} \\
& \leq \sum_{j=1}^p \begin{cases} |r_j - r'_j| + \tau_j r'_j & \text{if } r_j - r'_j + r'_j \geq 0 \\ |r_j - r'_j| + (\tau_j - 1)r'_j & \text{if } r_j - r'_j + r'_j \leq 0 \end{cases} \\
& \leq \sum_{j=1}^p (|r_j - r'_j| + \max(\tau_j r'_j, (\tau_j - 1)r'_j)) \\
& = \|\mathbf{r} - \mathbf{r}'\|_{\ell_1} + \ell_\tau(\mathbf{r}').
\end{aligned}$$

Switching \mathbf{r} and \mathbf{r}' we get $|\ell_\tau(\mathbf{r}) - \ell_\tau(\mathbf{r}')| \leq \|\mathbf{r} - \mathbf{r}'\|_{\ell_1}$. Since by Cauchy-Schwarz inequality $\|\mathbf{r} - \mathbf{r}'\|_{\ell_1} \leq \sqrt{p} \|\mathbf{r} - \mathbf{r}'\|_{\ell_2}$, we obtain that ℓ_τ is \sqrt{p} -Lipschitz.

In addition, ℓ_τ is bounded for the residuals of interest:

$$\begin{aligned}
& \forall (f, \mathbf{x}, \mathbf{y}) \in \mathcal{F} \times \mathcal{X} \times \mathcal{Y}^p, \\
& 0 \leq \ell_\tau(\mathbf{y} - f(\mathbf{x}) - \mathbf{b}) \leq \|\mathbf{y} - f(\mathbf{x}) - \mathbf{b}\|_{\ell_1} \\
& \leq \|\mathbf{y} - \mathbf{b}\|_{\ell_1} + \|f(\mathbf{x})\|_{\ell_1} \\
& \leq 2pa + \sqrt{p} \|f(\mathbf{x})\|_{\ell_2} \\
& \leq 2pa + c\sqrt{p\kappa}.
\end{aligned}$$

Let $\mathcal{U} = \{u: (\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathbb{R}^p \mapsto \mathbf{y} - f(\mathbf{x}) - \mathbf{b}, f \in \mathcal{F}\}$. By Proposition A.1 we have with probability at least $1 - \delta$:

$$\sup_{u \in \mathcal{U}} \left(\mathbb{E} [\ell_\tau(u(X, Y))] - \frac{1}{n} \sum_{i=1}^n \ell_\tau(u(X_i, Y_i)) \right) \leq 2\sqrt{2p}\mathcal{R}_n(\mathcal{U}) + (2pa + c\sqrt{p\kappa}) \sqrt{\frac{\log(1/\delta)}{2n}}.$$

Let $(\epsilon_{i,j})_{\substack{1 \leq i \leq n \\ 1 \leq j \leq p}}$ be an *iid* sample of Rademacher random variables. Then:

$$\begin{aligned}
\mathcal{R}_n(\mathcal{U}) &= \mathbb{E} \left[\sup_{u \in \mathcal{U}} \left(\frac{1}{n} \sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq p}} \epsilon_{i,j} u_j(X_i, Y_i) \right) \right] \\
&= \mathbb{E} \left[\sup_{f \in \mathcal{F}} \left(\frac{1}{n} \sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq p}} \epsilon_{i,j} (Y_i - f_j(X_i) - b_j) \right) \right] \\
&= \mathbb{E} \left[\sup_{f \in \mathcal{F}} \left(\frac{1}{n} \sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq p}} \epsilon_{i,j} f_j(X_i) \right) \right] \\
&\quad + \mathbb{E} \left[\frac{1}{n} \sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq p}} \epsilon_{i,j} Y_i \right] + b_j \mathbb{E} \left[\frac{1}{n} \sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq p}} \epsilon_{i,j} \right] \\
&= \mathbb{E} \left[\sup_{f \in \mathcal{F}} \left(\frac{1}{n} \sum_{i=1}^n \epsilon_{i,j} f_j(X_i) \right) \right] \\
&\quad + \frac{1}{n} \sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq p}} \mathbb{E} [\epsilon_{i,j}] \mathbb{E} [Y_i] + b_j \frac{1}{n} \sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq p}} \mathbb{E} [\epsilon_{i,j}] \\
&= \mathbb{E} \left[\sup_{f \in \mathcal{F}} \left(\frac{1}{n} \sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq p}} \epsilon_{i,j} f_j(X_i) \right) \right] \\
&= \mathcal{R}_n(\mathcal{F}) \\
&\leq c \sqrt{\frac{\kappa}{n}} \tag{Proposition A.2}.
\end{aligned}$$

This concludes the proof. \square

A.2 Quantile deviation

Given a vector of probabilities $\boldsymbol{\tau} \in (0, 1)^p$ and a quantile estimator $\hat{h}: \mathcal{X} \rightarrow \mathbb{R}^p$, we are interested in controlling the deviation of $\mathbb{E} \left[\mathbb{P} \left(Y \leq \hat{h}_j(X) \mid X \right) \right]$ from τ_j (for a particular $j \in \mathbb{N}_p$). For this purpose, we would like to derive a uniform bound using the scalar counterpart of Proposition A.1 (which is identical but substituting $\sqrt{2}$ by 1 [3]). Since such a bound is true for all hypothesis h , we do not require $\mathbb{E} [\mathbb{P} (Y \leq h_j(X) \mid X)]$ to be close to τ_j , but to its empirical twin $\frac{1}{n} \sum_{i=1}^n \mathbb{I}_{\mathbb{R}_-} (Y_i - h_j(X_i))$, where $\mathbb{I}_{\mathbb{R}_-}$ is the indicator function of the set \mathbb{R}_- . For a quantile estimator \hat{h} , the quantile property states that τ_j is sufficiently close to $\frac{1}{n} \sum_{i=1}^n \mathbb{I}_{\mathbb{R}_-} (y_i - \hat{h}_j(\mathbf{x}_i))$.

Now, remark that:

$$\mathbb{E} [\mathbb{P} (Y \leq h_j(X) \mid X)] = \mathbb{E} [\mathbb{E} [\mathbb{I}_{\mathbb{R}_-} (Y - h_j(X)) \mid X]] = \mathbb{E} [\mathbb{I}_{\mathbb{R}_-} (Y - h_j(X))].$$

Thus, the two quantities to compare clearly appear as an expected and an empirical costs based on the loss function $\mathbb{I}_{\mathbb{R}_-}$. Unfortunately, that loss function is not Lipschitz continuous. In order to circumvent that pitfall, we introduce an artificial margin $\epsilon > 0$ and two ramp functions Γ_ϵ^- and Γ_ϵ^+ (see definition in Theorem A.5). These surrogate mappings are $\frac{1}{\epsilon}$ -Lipschitz and respectively lower and upper bound $\mathbb{I}_{\mathbb{R}_-}$.

Finally, Theorem A.5 states that $\mathbb{E} [\mathbb{P} (Y \leq h_j(X) \mid X)]$ is uniformly bounded by the *empirical quantile levels* $\frac{1}{n} \sum_{i=1}^n \Gamma_\epsilon^-(Y_i - h_j(X_i))$ and $\frac{1}{n} \sum_{i=1}^n \Gamma_\epsilon^+(Y_i - h_j(X_i))$ to an additive bias in $O(1/\sqrt{n})$.

Theorem A.5 (Quantile deviation). *Let $\tau \in (0, 1)^p$, $((X_i, Y_i))_{1 \leq i \leq n} \in (\mathcal{X} \times \mathcal{Y})^n$ be iid random variables (independent from (X, Y)) and $\mathbf{b} \in \mathbb{R}^p$. Let $\mathcal{H} = \{f + \mathbf{b}: f \in \mathcal{K}_K / \|f\|_{\mathcal{K}} \leq c\}$, for a given $c \in \mathbb{R}_+$, be the class of hypotheses. Moreover, assume that there exists $\kappa \in \mathbb{R}_+$ such that: $\sup_{\mathbf{x} \in \mathcal{X}} \text{tr}(K(\mathbf{x}, \mathbf{x})) \leq \kappa$. Let $\epsilon > 0$ be an artificial margin,*

$$\Gamma_\epsilon^+ : r \in \mathbb{R} \mapsto \text{proj}_{[0,1]} \left(1 - \frac{r}{\epsilon} \right) \quad \text{and} \quad \Gamma_\epsilon^- : r \in \mathbb{R} \mapsto \text{proj}_{[0,1]} \left(-\frac{r}{\epsilon} \right),$$

two ramp functions, $j \in \mathbb{N}_p$ and $\delta \in (0, 1]$. Then with probability at least $1 - \delta$:

$$\forall h \in \mathcal{H}: \frac{1}{n} \sum_{i=1}^n \Gamma_\epsilon^-(Y_i - h_j(X_i)) - \Delta \leq \mathbb{E}[\mathbb{P}(Y \leq h_j(X) | X)] \leq \frac{1}{n} \sum_{i=1}^n \Gamma_\epsilon^+(Y_i - h_j(X_i)) + \Delta,$$

$$\text{where } \Delta = \frac{2c}{\epsilon} \sqrt{\frac{\kappa}{n}} + \sqrt{\frac{\log(2/\delta)}{2n}}.$$

Proof. First, let us remind that: $\mathbb{E}[\mathbb{P}(Y \leq h_j(X) | X)] = \mathbb{E}[\mathbb{I}_{\mathbb{R}_-}(Y - h_j(X))]$, and $\forall r \in \mathbb{R}, \Gamma_\epsilon^-(r) \leq \mathbb{I}_{\mathbb{R}_-}(r) \leq \Gamma_\epsilon^+(r)$. Thus,

$$\mathbb{E}[\Gamma_\epsilon^-(Y - h_j(X))] \leq \mathbb{E}[\mathbb{P}(Y \leq h_j(X) | X)] \leq \mathbb{E}[\Gamma_\epsilon^+(Y - h_j(X))].$$

Then, remarking that Γ_ϵ^+ is $\frac{1}{\epsilon}$ -Lipschitz, we obtain (by the same reasoning as for the proof of Theorem A.3 and using [3, Theorem 1]):

$$\forall h \in \mathcal{H}: \mathbb{E}[\Gamma_\epsilon^+(Y - h_j(X))] \leq \frac{1}{n} \sum_{i=1}^n \Gamma_\epsilon^+(Y_i - h_j(X_i)) + \frac{2c}{\epsilon} \sqrt{\frac{\kappa}{n}} + \sqrt{\frac{\log(1/\delta)}{2n}},$$

with probability at least $1 - \delta$. Respectively, for $-\Gamma_\epsilon^-$, with probability at least $1 - \delta$:

$$\forall h \in \mathcal{H}: -\mathbb{E}[\Gamma_\epsilon^-(Y - h_j(X))] \leq -\frac{1}{n} \sum_{i=1}^n \Gamma_\epsilon^-(Y_i - h_j(X_i)) + \frac{2c}{\epsilon} \sqrt{\frac{\kappa}{n}} + \sqrt{\frac{\log(1/\delta)}{2n}}.$$

Gathering everything with the union bound concludes the proof. \square

B Dual formulation

In this section, we derive a dual problem for learning a joint quantile regressor *with non-crossing constraints*. These last constraints are set thanks to a matrix \mathbf{A} defined below. Considering \mathbf{A} as the null-matrix gives a dual formulation for the learning problem *without non-crossing constraints*.

Let $C \in \mathbb{R}_+$, $\tau \in (0, 1)^p$, such that $\tau_j > \tau_{j+1}$ ($\forall j \in \mathbb{N}_{p-1}$) and the finite difference operator, embodied by the matrix:

$$\mathbf{A} = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 & -1 \end{pmatrix} \in \mathbb{R}^{(p-1) \times p}.$$

The primal problem we are interested in (with the associated dual variables) is:

$$\begin{aligned} & \underset{\substack{f \in \mathcal{H}, \mathbf{b} \in \mathbb{R}^p \\ \boldsymbol{\xi}, \boldsymbol{\xi}^* \in (\mathbb{R}^p)^n}}{\text{minimize}} && \frac{1}{2} \|f\|_{\mathcal{K}}^2 + C \sum_{i=1}^n \langle \tau | \boldsymbol{\xi}_i \rangle_{\ell_2} + C \sum_{i=1}^n \langle \mathbf{1} - \tau | \boldsymbol{\xi}_i^* \rangle_{\ell_2} \\ & \text{s. t.} && \begin{cases} \forall i \in \mathbb{N}_n: \mathbf{y}_i - f(\mathbf{x}_i) - \mathbf{b} = \boldsymbol{\xi}_i - \boldsymbol{\xi}_i^* & : \boldsymbol{\alpha}_i \in \mathbb{R}^p \\ \boldsymbol{\xi}_i \succeq 0 & : \boldsymbol{\beta}_i \in \mathbb{R}_+^p \\ \boldsymbol{\xi}_i^* \succeq 0 & : \boldsymbol{\gamma}_i \in \mathbb{R}_+^p \\ \mathbf{A}(f(\mathbf{x}_i) + \mathbf{b}) \succeq 0 & : \boldsymbol{\delta}_i \in \mathbb{R}_+^{p-1}. \end{cases} \end{aligned}$$

The last constraint enforces the regressors not to cross on the training points (hard non-crossing constraints). Let us write the Lagrangian function:

$$\begin{aligned}
\mathcal{L}(f, \mathbf{b}, \boldsymbol{\xi}_i, \boldsymbol{\xi}_i^*, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \boldsymbol{\gamma}_i, \boldsymbol{\delta}_i) &= \frac{1}{2} \|f\|_{\mathcal{K}}^2 + C \sum_{i=1}^n \langle \boldsymbol{\tau} \mid \boldsymbol{\xi}_i \rangle_{\ell_2} + C \sum_{i=1}^n \langle \mathbf{1} - \boldsymbol{\tau} \mid \boldsymbol{\xi}_i^* \rangle_{\ell_2} \\
&+ \sum_{i=1}^n \langle \boldsymbol{\alpha}_i \mid \mathbf{y}_i \rangle_{\ell_2} - \sum_{i=1}^n \langle \boldsymbol{\alpha}_i \mid f(\mathbf{x}_i) \rangle_{\ell_2} - \sum_{i=1}^n \langle \boldsymbol{\alpha}_i \mid \mathbf{b} \rangle_{\ell_2} \\
&- \sum_{i=1}^n \langle \boldsymbol{\alpha}_i \mid \boldsymbol{\xi}_i \rangle_{\ell_2} + \sum_{i=1}^n \langle \boldsymbol{\alpha}_i \mid \boldsymbol{\xi}_i^* \rangle_{\ell_2} \\
&- \sum_{i=1}^n \langle \boldsymbol{\beta}_i \mid \boldsymbol{\xi}_i \rangle_{\ell_2} - \sum_{i=1}^n \langle \boldsymbol{\gamma}_i \mid \boldsymbol{\xi}_i^* \rangle_{\ell_2} \\
&- \sum_{i=1}^n \langle \boldsymbol{\delta}_i \mid \mathbf{A}(f(\mathbf{x}_i) + \mathbf{b}) \rangle_{\ell_2} \\
&= \frac{1}{2} \|f\|_{\mathcal{K}}^2 - \left\langle \sum_{i=1}^n K_{\mathbf{x}_i} (\boldsymbol{\alpha}_i + \mathbf{A}^\top \boldsymbol{\delta}_i) \mid f \right\rangle_{\mathcal{H}} + \sum_{i=1}^n \langle C\boldsymbol{\tau} - \boldsymbol{\alpha}_i - \boldsymbol{\beta}_i \mid \boldsymbol{\xi}_i \rangle_{\ell_2} \\
&+ \sum_{i=1}^n \langle C(\mathbf{1} - \boldsymbol{\tau}) + \boldsymbol{\alpha}_i - \boldsymbol{\gamma}_i \mid \boldsymbol{\xi}_i^* \rangle_{\ell_2} - \left\langle \sum_{i=1}^n (\boldsymbol{\alpha}_i + \mathbf{A}^\top \boldsymbol{\delta}_i) \mid \mathbf{b} \right\rangle_{\ell_2} \\
&+ \sum_{i=1}^n \langle \boldsymbol{\alpha}_i \mid \mathbf{y}_i \rangle_{\ell_2}.
\end{aligned}$$

First order optimality conditions for the primal variables give:

$$\begin{cases}
\nabla_f \mathcal{L}(f, \mathbf{b}, \boldsymbol{\xi}_i, \boldsymbol{\xi}_i^*, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \boldsymbol{\gamma}_i, \boldsymbol{\delta}_i) = f - \sum_{i=1}^n K_{\mathbf{x}_i} (\boldsymbol{\alpha}_i + \mathbf{A}^\top \boldsymbol{\delta}_i) = 0 \\
\nabla_{\mathbf{b}} \mathcal{L}(f, \mathbf{b}, \boldsymbol{\xi}_i, \boldsymbol{\xi}_i^*, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \boldsymbol{\gamma}_i, \boldsymbol{\delta}_i) = - \sum_{i=1}^n (\boldsymbol{\alpha}_i + \mathbf{A}^\top \boldsymbol{\delta}_i) = 0 \\
\nabla_{\boldsymbol{\xi}_i} \mathcal{L}(f, \mathbf{b}, \boldsymbol{\xi}_i, \boldsymbol{\xi}_i^*, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \boldsymbol{\gamma}_i, \boldsymbol{\delta}_i) = C\boldsymbol{\tau} - \boldsymbol{\alpha}_i - \boldsymbol{\beta}_i = 0 \\
\nabla_{\boldsymbol{\xi}_i^*} \mathcal{L}(f, \mathbf{b}, \boldsymbol{\xi}_i, \boldsymbol{\xi}_i^*, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \boldsymbol{\gamma}_i, \boldsymbol{\delta}_i) = C(\mathbf{1} - \boldsymbol{\tau}) + \boldsymbol{\alpha}_i - \boldsymbol{\gamma}_i = 0.
\end{cases}$$

That is:

$$\begin{cases}
f = \sum_{i=1}^n K_{\mathbf{x}_i} (\boldsymbol{\alpha}_i + \mathbf{A}^\top \boldsymbol{\delta}_i) \\
0 = \sum_{i=1}^n (\boldsymbol{\alpha}_i + \mathbf{A}^\top \boldsymbol{\delta}_i) \\
\boldsymbol{\beta}_i = C\boldsymbol{\tau} - \boldsymbol{\alpha}_i \\
\boldsymbol{\gamma}_i = C(\mathbf{1} - \boldsymbol{\tau}) + \boldsymbol{\alpha}_i.
\end{cases}$$

Recalling that $\boldsymbol{\beta}_i \succcurlyeq 0$ and $\boldsymbol{\gamma}_i \succcurlyeq 0$, we obtain: $C(\boldsymbol{\tau} - \mathbf{1}) \preccurlyeq \boldsymbol{\alpha}_i \preccurlyeq C\boldsymbol{\tau}$. Then, by substitution of the first order equations in the Lagrangian function, the linear expressions in the primal variables vanish and the quadratic part becomes:

$$\begin{aligned}
\frac{1}{2} \|f\|_{\mathcal{K}}^2 - \left\langle \sum_{i=1}^n K_{\mathbf{x}_i} (\boldsymbol{\alpha}_i + \mathbf{A}^\top \boldsymbol{\delta}_i) \mid f \right\rangle_{\mathcal{H}} &= \frac{1}{2} \sum_{i,j=1}^n \left\langle K_{\mathbf{x}_i} (\boldsymbol{\alpha}_i + \mathbf{A}^\top \boldsymbol{\delta}_i) \mid K_{\mathbf{x}_j} (\boldsymbol{\alpha}_j + \mathbf{A}^\top \boldsymbol{\delta}_j) \right\rangle_{\mathcal{H}} \\
&- \sum_{i,j=1}^n \left\langle K_{\mathbf{x}_i} (\boldsymbol{\alpha}_i + \mathbf{A}^\top \boldsymbol{\delta}_i) \mid K_{\mathbf{x}_j} (\boldsymbol{\alpha}_j + \mathbf{A}^\top \boldsymbol{\delta}_j) \right\rangle_{\mathcal{H}} \\
&= -\frac{1}{2} \sum_{i,j=1}^n \left\langle (\boldsymbol{\alpha}_i + \mathbf{A}^\top \boldsymbol{\delta}_i) \mid K(\mathbf{x}_i, \mathbf{x}_j) (\boldsymbol{\alpha}_j + \mathbf{A}^\top \boldsymbol{\delta}_j) \right\rangle_{\ell_2}.
\end{aligned}$$

Gathering every thing, the dual problem writes:

$$\begin{aligned} & \underset{\substack{\boldsymbol{\alpha}_i \in \mathbb{R}^p, \boldsymbol{\delta}_i \in \mathbb{R}^{p-1} \\ \forall i \in \mathbb{N}_n}}{\text{maximize}} & -\frac{1}{2} \sum_{i,j=1}^n \left\langle (\boldsymbol{\alpha}_i + \mathbf{A}^\top \boldsymbol{\delta}_i) \mid K(\mathbf{x}_i, \mathbf{x}_j) (\boldsymbol{\alpha}_j + \mathbf{A}^\top \boldsymbol{\delta}_j) \right\rangle_{\ell_2} + \sum_{i=1}^n \langle \boldsymbol{\alpha}_i \mid \mathbf{y}_i \rangle_{\ell_2} \\ \text{s. t.} & \begin{cases} \forall i \in \mathbb{N}_n : C(\boldsymbol{\tau} - \mathbf{1}) \preceq \boldsymbol{\alpha}_i \preceq C\boldsymbol{\tau} \\ \boldsymbol{\delta}_i \succeq 0 \\ \sum_{i=1}^n (\boldsymbol{\alpha}_i + \mathbf{A}^\top \boldsymbol{\delta}_i) = \mathbf{0}_{\mathbb{R}^p}. \end{cases} \end{aligned}$$

In order to simplify the previous problem, let $\mathbf{u}_i = \boldsymbol{\alpha}_i + \mathbf{A}^\top \boldsymbol{\delta}_i$ and remark that $\mathbf{A}\mathbf{y}_i = y_i(\mathbf{A}\mathbf{1}) = 0$. The new dual problem then becomes:

$$\begin{aligned} & \underset{\substack{\mathbf{u}_i \in \mathbb{R}^p, \boldsymbol{\delta}_i \in \mathbb{R}^{p-1} \\ \forall i \in \mathbb{N}_n}}{\text{maximize}} & -\frac{1}{2} \sum_{i,j=1}^n \langle \mathbf{u}_i \mid K(\mathbf{x}_i, \mathbf{x}_j) \mathbf{u}_j \rangle_{\ell_2} + \sum_{i=1}^n \langle \mathbf{u}_i \mid \mathbf{y}_i \rangle_{\ell_2} \\ \text{s. t.} & \begin{cases} \forall i \in \mathbb{N}_n : C(\boldsymbol{\tau} - \mathbf{1}) \preceq \mathbf{u}_i - \mathbf{A}^\top \boldsymbol{\delta}_i \preceq C\boldsymbol{\tau} \\ \boldsymbol{\delta}_i \succeq 0 \\ \sum_{i=1}^n \mathbf{u}_i = \mathbf{0}_{\mathbb{R}^p}. \end{cases} \end{aligned}$$

Primal variables are recovered thanks to first order conditions. First, $f = \sum_{i=1}^n K_{\mathbf{x}_i} \mathbf{u}_i$. Second, the intercept \mathbf{b} can be obtained either by detecting couples $(i, \ell) \in \mathbb{N}_n \times \mathbb{N}_p$ such that $C(\boldsymbol{\tau} - \mathbf{1}) < (u_i)_\ell - (\mathbf{A}^\top \boldsymbol{\delta}_i)_\ell < C\boldsymbol{\tau}_\ell$ (in this case $b_\ell = y_i - f_\ell(\mathbf{x}_i)$), or by remarking that \mathbf{b} is a dual vector for the linear constraint $\sum_{i=1}^n \mathbf{u}_i = \mathbf{0}_{\mathbb{R}^p}$ (if one uses a primal-dual algorithm to solve the previous optimization problem).

When non-crossing constraints are dismissed ($\mathbf{A} = \mathbf{0}_{\mathbb{R}^{(p-1) \times p}}$), the regressor $h = f + \mathbf{b}$ satisfies the quantile property. Thus, knowing f , the intercepts b_ℓ can be recovered as τ_ℓ -quantiles of $(y_i - f_\ell(\mathbf{x}_i))_{1 \leq i \leq n}$.

C Algorithmic details

This section details an augmented Lagrangian scheme for estimating quantile regressors. We start with the dual formulation of the learning problem (without non-crossing constraints):

$$\begin{aligned} & \underset{\boldsymbol{\alpha} \in (\mathbb{R}^p)^n}{\text{minimize}} & \frac{1}{2} \sum_{i,j=1}^n \langle \boldsymbol{\alpha}_i \mid K(\mathbf{x}_i, \mathbf{x}_j) \boldsymbol{\alpha}_j \rangle_{\ell_2} - \sum_{i=1}^n y_i \langle \boldsymbol{\alpha}_i \mid \mathbf{1} \rangle_{\ell_2} \\ \text{s. t.} & \begin{cases} \forall i \in \mathbb{N}_n : C(\boldsymbol{\tau} - \mathbf{1}) \preceq \boldsymbol{\alpha}_i \preceq C\boldsymbol{\tau} \\ \sum_{i=1}^n \boldsymbol{\alpha}_i = \mathbf{0}_{\mathbb{R}^p}. \end{cases} \end{aligned}$$

The method consists in solving the saddle point problem with an additional squared penalty [2]:

$$\begin{aligned} & \underset{\mathbf{b} \in \mathbb{R}^p}{\text{maximize}} \underset{\boldsymbol{\alpha} \in (\mathbb{R}^p)^n}{\text{minimize}} & \frac{1}{2} \sum_{i,j=1}^n \langle \boldsymbol{\alpha}_i \mid K(\mathbf{x}_i, \mathbf{x}_j) \boldsymbol{\alpha}_j \rangle_{\ell_2} - \sum_{i=1}^n y_i \langle \boldsymbol{\alpha}_i \mid \mathbf{1} \rangle_{\ell_2} \\ & & + \left\langle \mathbf{b} \mid \sum_{i=1}^n \boldsymbol{\alpha}_i \right\rangle_{\ell_2} + \frac{\mu}{2} \left\| \sum_{i=1}^n \boldsymbol{\alpha}_i \right\|_{\ell_2}^2 \\ \text{s. t.} & \forall i \in \mathbb{N}_n : C(\boldsymbol{\tau} - \mathbf{1}) \preceq \boldsymbol{\alpha}_i \preceq C\boldsymbol{\tau}, \end{aligned}$$

where μ is a positive scalar. The next step is to split the optimization program into an outer problem (depending only on the variable \mathbf{b}) and an inner one (depending on $\boldsymbol{\alpha}$). This latter problem is:

$$\begin{aligned} & \underset{\boldsymbol{\alpha} \in (\mathbb{R}^p)^n}{\text{minimize}} & \frac{1}{2} \sum_{i,j=1}^n \langle \boldsymbol{\alpha}_i \mid (K(\mathbf{x}_i, \mathbf{x}_j) + \mu \mathbf{I}) \boldsymbol{\alpha}_j \rangle_{\ell_2} + \sum_{i=1}^n \langle \boldsymbol{\alpha}_i \mid \mathbf{b} - y_i \mathbf{1} \rangle_{\ell_2} \\ \text{s. t.} & \forall i \in \mathbb{N}_n : C(\boldsymbol{\tau} - \mathbf{1}) \preceq \boldsymbol{\alpha}_i \preceq C\boldsymbol{\tau}, \end{aligned} \tag{1}$$

Algorithm 1 Augmented Lagrangian algorithm

Initialize $\mu \leftarrow 10$, $\mathbf{b} \leftarrow 0_{\mathbb{R}^p}$.**repeat**

Solve Optimization Problem (1).

Make a gradient step on \mathbf{b} with step size μ .**until** $\|\sum_{i=1}^n \alpha_i\|_{\ell_2}^2$ is small enough

Table 1: Empirical quantile loss $\times 100$ (the closer to 0, the better).

Data set	IND.	IND. (NC)	MTFL	JQR
caution	4.50 \pm 39.08	3.33 \pm 37.84	7.00 \pm 32.57	ooo 7.17 \pm 36.40
ftcollinssnow	1.43 \pm 38.49	1.79 \pm 38.12	1.25 \pm 38.50	ooo 0.54 \pm 35.77
highway	10.83 \pm 70.12	10.83 \pm 71.20	7.50 \pm 62.96	ooo 15.00 \pm 67.11
heights	-1.15 \pm 9.88	-1.14 \pm 9.88	-0.76 \pm 9.39	ooo -1.26 \pm 9.31
sniffer	-6.58 \pm 26.45	-6.58 \pm 27.59	-3.95 \pm 27.57	ooo -4.08 \pm 29.23
snowgeese	-0.00 \pm 44.03	1.07 \pm 43.41	1.43 \pm 50.03	ooo -6.43 \pm 44.94
ufc	0.58 \pm 12.13	0.89 \pm 11.97	-0.31 \pm 13.59	●●o -1.79 \pm 13.42
birthwt	2.02 \pm 29.80	2.02 \pm 29.80	2.11 \pm 34.55	ooo 0.88 \pm 33.86
crabs	-2.42 \pm 20.52	-1.25 \pm 22.08	-2.17 \pm 22.21	ooo -0.50 \pm 21.75
GAGurine	1.95 \pm 17.43	1.74 \pm 17.39	0.89 \pm 16.71	ooo 1.84 \pm 16.89
geyser	1.22 \pm 18.84	1.17 \pm 18.66	0.22 \pm 19.20	ooo 1.61 \pm 19.04
gilgais	0.95 \pm 18.35	0.95 \pm 18.19	-0.64 \pm 16.12	ooo 0.18 \pm 20.22
topo	-19.38 \pm 70.18	-19.38 \pm 71.34	-20.00 \pm 64.18	ooo -20.31 \pm 65.73
BostonHousing	7.40 \pm 18.26	7.30 \pm 17.94	5.72 \pm 16.72	ooo 5.30 \pm 15.74
CobarOre	26.67 \pm 72.49	25.42 \pm 72.58	-22.08 \pm 77.16	ooo 20.00 \pm 72.62
engel	-3.66 \pm 18.71	-3.73 \pm 18.72	-3.73 \pm 19.73	ooo -2.39 \pm 17.18
mecycle	1.37 \pm 29.20	1.75 \pm 30.73	4.75 \pm 28.42	ooo 6.25 \pm 30.84
BigMac2003	-4.76 \pm 51.10	-0.00 \pm 45.24	1.67 \pm 52.19	ooo 0.24 \pm 45.76
UN3	4.44 \pm 19.11	4.21 \pm 19.23	2.62 \pm 17.06	ooo 4.76 \pm 20.68
cpus	2.38 \pm 20.30	3.57 \pm 19.63	1.51 \pm 15.02	ooo 1.67 \pm 31.78

where \mathbf{I} is the identity matrix. This inner optimization problem is a quadratic program with a box constraint. This is quite easily solvable. Thus, following [2], we can learn quantile estimators thanks to the simple alternate scheme described in Algorithm 1. In practice, the inner solver used in Algorithm 1 in order to get an approximate solution for Problem (1) is the primal dual-dual coordinate descent proposed in [6] with 10^4 as the maximum number of iterations.

D Numerical results

D.1 Quantile regression

Another criterion for assessing quantile regression methods is the quantile loss $\sum_{j=1}^p \left[\left[\frac{1}{n} \sum_{i=1}^n \mathbf{I}_{\mathbb{R}_-}(y_i - h_j(\mathbf{x}_i)) \right] - \tau_j \right]$, where $\mathbf{I}_{\mathbb{R}_-}$ is the indicator function of the set \mathbb{R}_- . This loss measures the deviation of the estimators h_j to the prescribed quantile levels τ_j .

However, the quantile loss is quite an equivocal criterion, since it measures *how much* the *unconditional* quantile property is satisfied. This unconditional indicator is indeed the only way to get a piece of information concerning the *conditional* quantile property. For instance, Takeuchi et al. [8] empirically showed (with the same datasets) that the constant function based on the unconditional quantile estimator performs best under this criterion, even though it is expected to be a poor conditional quantile regressor. The numerical results in Table 1 follow this remark and the results previously obtained [8]. No significant ranking comes out.

D.2 Training algorithms

In order to compare the implementations of the three algorithms for solving the dual optimization problem of joint quantile regression, the following procedure has been set up: we first run QP, with a relative tolerance set to 10^{-2} , and store the optimal objective value. Then, the two other methods (AUG. LAG and PDCD) are launched and stopped when they pass the objective value reached by QP.

During the descent of PDCD, we used an efficient accumulated objective value, which is not exact since the iterate α is not feasible to the linear constraint. Table 2 describes the average objective values (divided by the sample size) reached by each algorithm after projection of the best candidate

Table 2: Average objective value (divided by the sample size) reached in the second numerical experiment presented in the corpus of the paper.

Size	QP	AUG. LAG.	PDCD
250	-109.69 \pm 5.41	-109.73 \pm 5.41	-109.72 \pm 5.40
500	-109.84 \pm 2.09	-109.85 \pm 2.08	-109.88 \pm 2.11
1000	-104.13 \pm 1.49	–	-104.17 \pm 1.49
2000	-106.35 \pm 2.36	–	-106.39 \pm 2.38

onto the set of constraints. We can check that our approach (PDCD) always reaches a smaller objective value than the target QP. This validates our procedure.

References

- [1] P.L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.
- [2] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999. ISBN 9781886529007.
- [3] S.M. Kakade, K. Sridharan, and A. Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in Neural Information Processing Systems*. 2009.
- [4] A. Maurer. Bounds for Linear Multi-Task Learning. *Journal of Machine Learning Research*, 7:117–139, 2006.
- [5] A. Maurer. A vector-contraction inequality for Rademacher complexities. In *Proceedings of The 27th International Conference on Algorithmic Learning Theory*, 2016.
- [6] S. Shalev-Shwartz and T. Zhang. Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization. *Journal of Machine Learning Research*, 14:567–599, 2013.
- [7] V. Sindhvani, M.H. Quang, and A.C. Lozano. Scalable Matrix-valued Kernel Learning for High-dimensional Nonlinear Multivariate Regression and Granger Causality. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, 2013.
- [8] I. Takeuchi, Q.V. Le, T.D. Sears, and A.J. Smola. Nonparametric Quantile Estimation. *Journal of Machine Learning Research*, 7:1231–1264, 2006.