Non-crossing Quantile Regression Curves by Support Vector Machine and Its Efficient Implementation

Ichiro Takeuchi (Dept. of Information Engineering, Mie University, Japan)

Abstract: In nonparametric regression quantile estimations, we sometimes face an embarrassing phenomenon called *Quantile Crossing*. Quantile crossing signifies the crossing or overlaps of two or more regression quantile curves which are estimated from the same samples with different quantile orders. To draw *non-crossing* regression quantile curves, we need to impose the *non-crossing* constraint. Since we are trying to estimate flexible nonlinear functions, say f_1 and f_2, the non-crossing constraint: $f_1(x) < f_2(x) \; \forall x$, also has to be flexible and nonlinear. Optimization with such a complex constraint is, in general, very difficult. One of the possible approaches is to transform such a complex constraint into a more simple tractable one like He's *Restricted Regression Quantile*, in which the non-crossing constraint was transformed into a positivity constraint.

In this paper, we address the non-crossing problem with an approach used in support vector machines (SVM). SVM has been intensively studied in the fields of machine learning and optimization. One of the key concepts of SVM is implicitly defined *kernel induced feature space (KIFS)*. SVM is defined as a linear model in KIFS but it can be a flexible nonlinear function in the original covariate-response space. Since non-crossing constraint for two or more linear models is also linear constraints, we can easily introduce the non-crossing constraint for SVMs in KIFS. Using a trick called *kernel trick*, the optimization problem defined in the KIFS is transformed into a constrained maximization of a convex piecewise quadratic function in the original covariate-response space. The above optimization problem can be solved by publicly available quadratic programming package although it is very computationaly costly when the dimension of the covariates or the number of samples is large. To alleviate this burden, we also propose, in this paper, an efficient implementation of the above optimization problem. The implementation fully exploits the nature of the problem and much faster than using standard quadratic programing package.