Distribution-Free Prediction Intervals for Kernel Regression

László Keresztes, *ELTE TTK*
Supervisor: Balázs Csanád Csáji, *SZTAKI*

I. Introduction

One of the most important aspects of *machine learning* methods is their *generalization* capability. A model that was trained on a finite sample makes predictions on new, unseen examples. We want to give *guarantees* on the goodness of these predictions. For a regression problem, one could construct an interval that contains the output variable with a given probability for a new input variable. One of the solutions to this problem is the framework of *conformal prediction*.

The illustration of the strength of conformal prediction could be shown even on $I \to \mathbb{R}$ regression. The conformal prediction requires a point estimaton method, which will be the *support vector regression* (SVR) method for the illustration. The kernel methods (including SVR) have emerged because they could learn in large (and even in infinite dimensional) feature spaces, without constructing the feature vectors.

II. Kernel methods

Let $\mathbb{X}$ be the input domain, let $\mathbb{Y}$ be the output domain (both are assumed to be measurable spaces), and let D be a distribution on $\mathbb{X} \times \mathbb{Y}$. Let $(x_1, y_1), \ldots, (x_m, y_m)$ be a finite i.i.d sample of input-output measurements having distribution D. Let $\ell : \mathbb{Y} \times \mathbb{Y} \to \mathbb{R}_{\geq 0}$ be a (measurable) *loss* function. A fundamental problem in *statistical learning* is the following: construct a function $\hat{f} : \mathbb{X} \to \mathbb{Y}$ which minimizes the *risk*, that is $R(\hat{f}) = \mathbb{E}_{(X,Y)\sim D}\left[\ell(\hat{f}(X), Y)\right]$ should be minimized.

We usually have no information about D. We only have the i.i.d sample with some prior knowledge of the problem (if we have). If $\mathbb{Y}$ is a continuous domain, we call the estimation problem a *regression* problem. For now, restrict our $\mathbb{Y}$ to $\mathbb{R}$.

Kernel methods could be seen through a similarity measure between the points in $\mathbb{X}$. With the assumption that if x_1 and x_2 are similar, then y_1 and y_2 will be also similar, we could use a $k : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_{\geq 0}$ similarity (or dissimilarity) function which helps us constructing $\hat{f}$. This is one of the intuitions behind kernel methods. Some of the most important definition and theorems are from the book "Learning with Kernels" [2].

From the optimization perspective positive definite kernels are more desired because they lead to convex optimization problems. The positive definiteness is determined by the (data-dependent) Gram matrices.

Definition 1. *(Gram matrix)*

Given a function $k : \mathbb{X}^2 \to \mathbb{R}$ and patterns $x_1, \ldots, x_m \in \mathbb{X}$, the $m \times m$ matrix K with elements $K_{i,j} = k(x_i, x_j)$ is called the Gram matrix (or kernel matrix) of k with respect to the data points $x_1, \ldots, x_m \in \mathbb{X}$.

Definition 2. *(Positive definite kernel)*

Let $\mathbb{X}$ be a nonempty set. A symmetric $k : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ function which $\forall m\ \forall x_1, \ldots x_m \in \mathbb{X}$ points gives a positive semi-definite Gram matrix is called a positive definite kernel (or kernel). If for all m and distinct $\{x_i\}$ the Gram matrix is positive definite, the kernel is called strictly positive definite.

For any real-valued kernel: $k(x, x) \geq 0\ \forall x \in \mathbb{X}$ and $k(x_i, x_j) = k(x_j, x_i)\ \forall x_i, x_j \in \mathbb{X}$.

The feature map is a $\Phi : \mathbb{X} \to \mathbb{H}$ mapping, where $\mathbb{H}$ is a subset of some vector space. We call $\mathbb{H}$ a feature space, if it is a vector space, that contains $\Phi(x)\ \forall x \in \mathbb{X}$.

Any (positive definite) kernel corresponds to a dot product in some feature space (pre-Hilbert space). The opposite is also true, any dot product in a feature space corresponds to a kernel.

The most simple example is the $\Phi : x \mapsto k(\cdot, x)$ mapping. This mapping corresponds to the so-called Reproducing Kernel Hilbert Spaces, which states whether a dot product in a Hilbert space corresponds to a kernel.

Definition 3. *Reproducing Kernel Hilbert Spaces*

Let $\mathbb{X}$ be a nonempty set and $\mathbb{H}$ a Hilbert space of functions $f : \mathbb{X} \to \mathbb{R}$ endowed with the dot product $\langle \cdot, \cdot \rangle$. Then $\mathbb{H}$ is called a Reproducing Kernel Hilbert Space if $\exists k : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ with the following properties:

1) *k has the reproducing property*
   $\langle f, k(\cdot, x) \rangle = f(x)\ \forall f \in \mathbb{H}\ \forall x \in \mathbb{X}$
2) *k spans $\mathbb{H}$*
   $\mathbb{H} = \overline{span\{k(x, \cdot) : x \in \mathbb{X}\}}$

Back to the estimation problem, when we construct the $\hat{f}$ function, we require that $\ell(\hat{f}(X), Y)$ be small on average. As we only have the training sample, this leads to searching for a function that minimizes the empirical risk. As the No-Free-Lunch-Theorem states [4], this is not enough. One option is to restrict the possible space of functions explicitly, which leads to the classical statistical learning setting with VC-dimension and PAC learning.

The other option is to add a regularization term to the empirical loss and minimize the sum of the two terms. The regularization term tries to bound the "complexity" of $\hat{f}$. A common regularization term is $\Omega(\hat{f}) = \frac{1}{2}\|\hat{f}\|^2_{\mathbb{H}}$, see SVR.

We only require that Ω be a convex function of f. Because the empirical risk is also required to be convex, there is only one global minimum of the regularized risk $R_{reg}(\hat{f}) = \frac{1}{m}\sum_{i=1}^{m} \ell(\hat{f}(x_i), y_i) + \lambda\Omega(\hat{f})$, where $\lambda > 0$ is the regularization parameter.

The *Representer Theorem* [2] states that given an RKHS, $\mathbb{H}$, with the kernel k, for minimizing the regularized risk, one

should only consider the linear span of functions $\{k(\cdot, x_i)\}$.

Statement 1. *(Representer Theorem) Let $\mathbb{H}$ be a Reproducing Kernel Hilbert Space associated to the kernel k. Denote by $\Omega : [0, \infty] \to \mathbb{R}$ a strictly monotonic increasing function, by $\mathbb{X}$ a set, and by $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ an arbitrary loss function. Then each minimizer $\hat{f} \in \mathbb{H}$ of the regularized risk $\frac{1}{m}\sum_{i=1}^{m} \ell(\hat{f}(x_i), y_i) + \Omega(\|\hat{f}\|_{\mathbb{H}})$ admits a representation of the form $\hat{f}(x) = \sum_{i=1}^{m} \alpha_i k(x, x_i)$*

The Representer Theorem gives a guide to how and where to search the minimizer of the regularized risk. The theorem explains, e.g., the form of the decision functions in SVRs.

III. Split conformal prediction

The general idea behind the conformal prediction is due to Vovk [3]. There, the construction of the intervals depends on a point estimator and could be computationally intensive.

The *split conformal prediction* is an alternative way to construct these intervals [1]. The idea is that we split our training set into two halves. We fit our point estimator (or regression algorithm) on the first half, and compute the absolute error for every instance in the second half. Then, we compute the corresponding percentile, determined by the miscoverage level. Finally from this percentile, one could construct an interval, that contains the true target variable with high probability.

Algorithm 1: Split Conformal Prediction

Input : Data $(X_i, Y_i), i = 1, \ldots, n$, miscoverage level $\alpha \in (0, 1)$, regression algorithm $\mathcal{A}$
Output: Prediction band, over $x \in \mathbb{R}^d$
Randomly split $\{1, \ldots, n\}$ into two equal-sized subsets $\mathcal{I}_1, \mathcal{I}_2$
$\hat{\mu} = \mathcal{A}(\{(X_i, Y_i) : i \in \mathcal{I}_1\})$
$R_i = |Y_i - \hat{\mu}(X_i)|, i \in \mathcal{I}_2$
$d =$ the kth smallest value in $\{R_i : i \in \mathcal{I}_2\}$, where $k = \lceil (n/2 + 1)(1 - \alpha) \rceil$
Return $C_{split}(x) = [\hat{\mu}(x) - d, \hat{\mu}(x) + d]$

The following theorems provide stochastic guarantees for the constructed prediction regions [1].

Statement 2. *If $(X_i, Y_i), i = 1, \ldots, n$ are i.i.d., then for a new i.i.d. draw (X_{n+1}, Y_{n+1})*

$$\mathbb{P}(Y_{n+1} \in C_{split}(X_{n+1})) \geq 1 - \alpha,$$

for the split conformal prediction band C_{split} constructed in Algorithm 1. Moreover, if we assume additionally that the residuals $R_i, i \in \mathcal{I}_2$ have a continuous joint distribution, then

$$\mathbb{P}(Y_{n+1} \in C_{split}(X_{n+1})) \leq 1 - \alpha + \frac{2}{n+2}$$

Statement 3. *Under the conditions of Statement 2., there is an absolute constant $c > 0$ such that, for any $\epsilon > 0$,*

$$\mathbb{P}(|\frac{2}{n}\sum_{i \in \mathcal{I}_2} \mathbb{I}\{Y_i \in C_{split}(X_i)\} - (1 - \alpha)| \geq \epsilon) \leq 2\exp(-cn^2(\epsilon - 4/n)_+^2)$$

IV. Numerical experiments

In the test section, I applied the split conformal prediction approach on the regression problem $f : [0, 1] \to \mathbb{R}$, $f(x) = x\sin(cx)$ with $c > 0$. I selected the support vector regression method with RBF kernel as the point estimator that is used by the split conformal prediction. I tested how the number of training examples and the α miscoverage level affect the split conformal prediction regions.

A. Support Vector Regression with RBF kernel

In the SVR setting, we select the regularization as $\Omega(f) = \frac{\lambda}{2}\|f\|^2$, if $f \in \mathbb{H}$ where $\mathbb{H}$ is an RKHS, then we could write the regularization term as the function of α_i and x_i.

The loss $\ell(y, \hat{y}) = max(|y - \hat{y}| - \epsilon, 0)$ is the ϵ-insensitive loss. Note that any positive definite kernel could work with SVR, because of the representer theorem, but here the RBF kernel was applied. The Gaussian RBF kernel has the form $k(x, x') = \exp(-\|x - x'\|^2/2\sigma^2)$, with any $\sigma > 0$. The RBF kernel satisfies $k(x, x) = 1\ \forall x \in \mathbb{X}$ and $k(x, x') > 0\ \forall x, x' \in X$. It can be proven that the Gram matrix with respect to different $x_1, \ldots, x_m$ points of an RBF kernel always has full rank, which means that the search space is going to be infinite dimensional as we add more training examples.

B. Setting and parameters

Let $\mathbb{I} = [0, 1]$. Let $f : \mathbb{I} \to \mathbb{R}$, $f(x) = x\sin(cx)$. Sample: $(X_i, Y_i), i = 1, \ldots, m$ i.i.d, where $X_i \sim U([0, 1])$ and $Y_i = f(X_i) + N_i$, $N_i \sim Laplace(0, b)$ i.i.d. variables.

The point estimator is Support Vector Regression algorithm, with $\epsilon > 0$ in the loss function and $C > 0$ regularization parameter. The kernel is RBF with $\sigma > 0$.

The prediction intervals have the miscoverage level α.

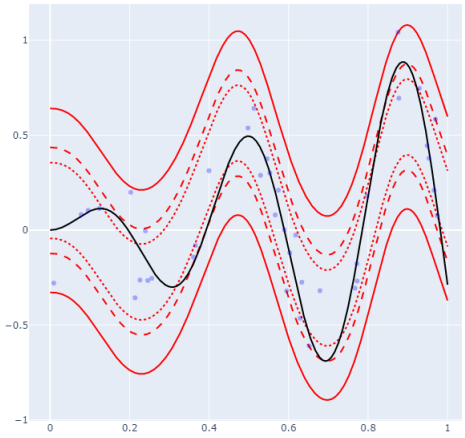
Parameter values: $c = 16.0$, $b = 0.1$, $\epsilon = 0.01$, $C = 1.0$, $\sigma = 0.1$.

I was interested in how the α and m parameters affect the result, therefore I only changed these, while leaving the others on the default value.

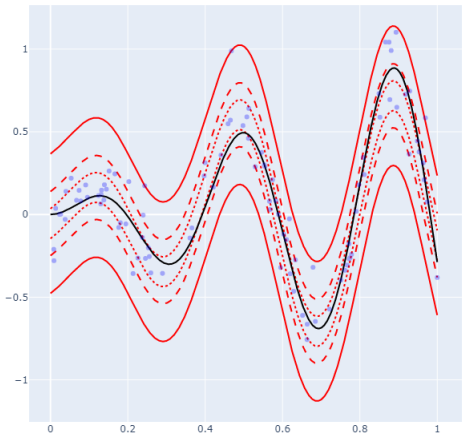
C. Plots

On the plots, the black line is f, the blue points are the (X_i, Y_i) sample points, the dotted red region has $\alpha = 0.5$, the dashed red region has $\alpha = 0.2$, and the solid red region has $\alpha = 0.05$ miscoverage level.

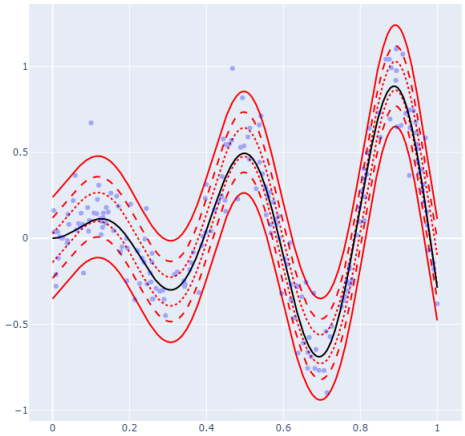
m=40 samples



m=100 samples



m=200 samples



From the resulting plots we can conclude, that for a fixed m, as $\alpha \to 1$, the conformal regions are centered around the point estimate function (cf. Algorithm 1).

The other thing, we can conclude is that if we fix α, then as $m \to \infty$, the split conformal prediction region tends to the real α quantile region (cf. Section 3 of [1]).

REFERENCES

[1] Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

[2] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[3] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *J. Mach. Learn. Res.*, 9:371–421, June 2008.

[4] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, USA, 2014.