Fast non-parametric regression using different approximation methods

Aniruddha R Rao

April 24, 2018

Introduction

Kernel ridge regression : Consider the supervised problem of learning a function given a training set of n examples (x_i, y_i), i =1,2.....,n , where $x_i \in$ X, X = R^d and $y_i \in$ R. Kernel methods are nonparametric approaches defined by a kernel K : X * X $\rightarrow$ R, that is a symmetric and positive definite (PD) function. A particular instance is kernel ridge regression given by,

$$f_\lambda(x) = \Sigma_{i=1}^n \alpha_i k(x_i, x)$$

The convex problem is,
$\hat{f} = argmin_{f \in \mathcal{H}}[1/N \Sigma_{i=1}^n (f(x_i) - y_i)^2 + \lambda ||f||_{\mathcal{H}}^2]$

Equivalently, $\hat{\alpha} = argmin_{\alpha \in R^n}[(y - k\alpha)'(y - k\alpha) - \lambda\alpha' k\alpha]$
Solution, $\hat{\alpha} = (k + \lambda I)^{-1} y$

Kernel Ridge Regression

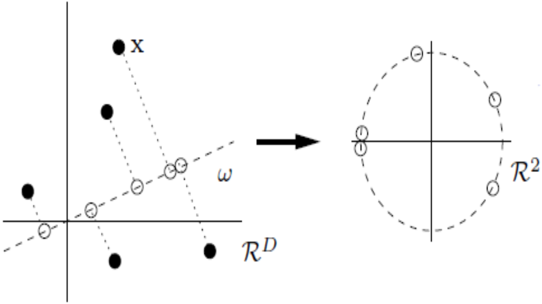
What is the Problem?

- Kernel matrix (Gram matrix) is fully dense and scales poorly with the size of the training dataset.
- k(x, y)= $<\phi(x), \phi(y)>$. Memory O(n^2) and computation time O(n^3).

Solution is to approximate the kernel matrix or approximates the kernel function directly.

Random Fourier Features

Random features consists of random Fourier bases z(x)=cos(w'x) where w$\in R^d$. Each component of the feature map z(x) projects x onto a random direction w drawn from the Fourier transform p(w) of $k(x, y)$ and wraps this line onto the unit circle in R^2. After transforming two points x and y in this way, their inner product is an unbiased estimator of k(x,y).



RFF Algorithm

- Select a positive definite shift-invariant kernel k(x, y) = $k(x-y)$.
- Compute the Fourier transform p of the kernel k:
  $p(w) = 1/(2\pi) \int e^{jw'\delta} K(\delta) d \triangle$.
- Draw D iid samples w_1,.....,$w_D \in R^d$ from p.
- Construct a randomized feature map z(x) : $R^d \rightarrow R^D$ so that
  $z(x)'z(y) \approx k(x-y)$.
  z(x)=$\sqrt{1/D}[cos(w_1'x), ..., cos(w_D'x), sin(w_1'x), ..., sin(w_D'x)]$

Note : z(x) takes O(nD^2 + D^3) in time, O(nD) in space, and and
k(x,y) =($e^{-s*||x-y||^2/2}$) follows p(w) =N_d(μ=0,Σ=s*I)

Sketching's Method

Sketching's method approximates of KRR based on m-dimensional randomized sketches (projections) of the kernel matrix.

Algorithm:

- Define $\alpha_{n*1} = S_{n*m} * W_{m*1}$ where S is a matrix defined by random sketches where m<<n.
- $\hat{W} = argmin_{W \in R^m}[(y - kSW)^{'}(y - kSW) - \lambda W^{'} S^{'} kSW]$
  $\hat{W} = [S^{'}(k + n\lambda I)S]^{-1} S^{'} y$
- $f_\lambda(x) = \Sigma_{i=1}^{n} (SW)_i k(x_i, x)$

Note : The computation time gets reduced to O(n^2 log(m)+m^3) and storage space is O(nm+m^2)

Standard Nystrom

Algorithm:

- Decide m (<<n). Randomly sample m columns from k(x,y). Get W_{m*m}.
- $$k(x,y) = \begin{bmatrix} W_{m*m} & k' \\ k_{(n-m)*m} & f(W,k) \end{bmatrix} \quad C_{n*m} = \begin{bmatrix} W \\ k \end{bmatrix}$$
- $k(x,y) \approx CW^{-}C^{T}$ & (k+λ*I)$^{-1}$=(I-C[λ*I+$W^{-}C^{T}C]^{-}W^{-}C^{T}$)/$\lambda$
- Substitute above values in KRR method.

Note : The computation time gets reduced to O(m^3+nm^2) and storage space is O(m^2+nm)

Other methods

1. **Model Averaging**
   - Randomly partitions a dataset of size n into m subsets of equal size
   - Compute an independent kernel ridge regression model for each subset
   - Average the local solutions into a global predictor.

Note : The computation time gets reduced to $O(n^3/m^2)$ and storage space is $O(n^2/m^2)$

2. **Cholesky Decomposition**

Simulation

One Dimensional example:

y =sin(x)+cos(x) where x $\in$ (-6,6) and y $\in$ (-1.42,1.42)

Gaussian kernel : k(x,y)=exp(-s*$||x-y||^2$)

The bandwidth (s) and penalty (λ) are constant across the methods and the other tuning parameters are functions for n. (ϵ=0.03)

Multi Dimensional example:

y$\sim$ $N_{d=5}(u=0,\Sigma=I)$ where I is identity matrix.

Gaussian kernel : k(x,y)=exp(-s*$||x-y||^2$)

All the parameters had to be tuned for every case of n in each of the approximate method.(ϵ=1*10^{-8})

Simulation result : 1-D example

Figure 1: Comparison plot of different methods for n=5000

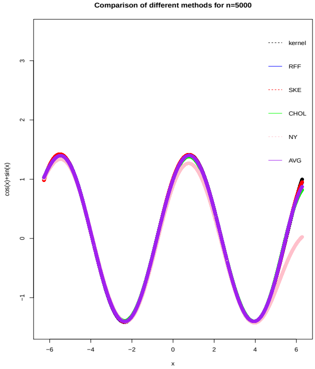
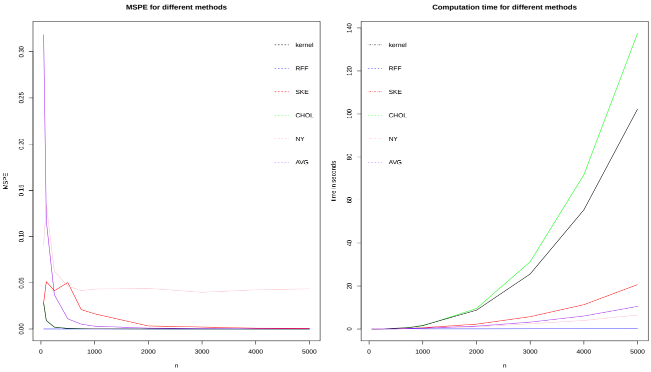


Table 1: Comparison for n=5000

Method	MSPE
KRR	3.580697e-04
RFF	5.722305e-05
NYSTROM	4.166946e-02
SKETCHING	2.104542e-02
CHOLESKY	3.580697e-04
MODEL AVG	5.276929e-03

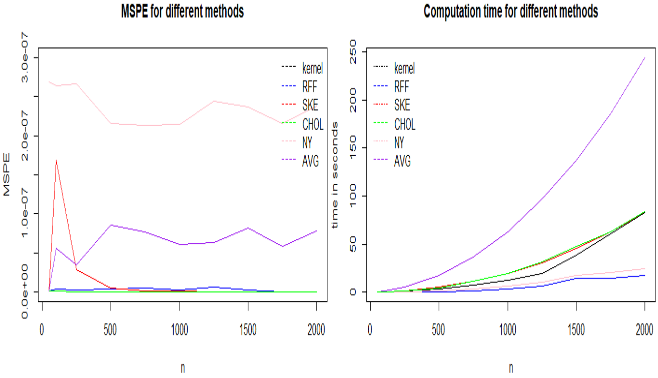
Simulation result : 1-D example

Figure 2: Plots for MSPE and Computation time



Simulation result : Multi-D example

Figure 3: Plots for MSPE and Computation time



Real Data Example

Parkinsons Telemonitoring Data Set from UCI Data Repository.
The dataset is composed of a range of biomedical voice measurements.
The goal is to predict UPDRS score from the different voice measures.

Table 2: Comparison wrt MSPE

Method	MSPE
KRR	409.8344
RFF	105.8395
NYSTROM	416.6946
SKETCHING	868.1353
CHOLESKY	409.8344
MODEL AVG	783.3628

n=5875, d=19
In all it took around 8 hours to run all the methods. I have not tuned it yet but the results look not so bad.

Problems and extension ideas

Problems:

- In the 1-D example, Nystrom method results in huge distance between the Matrices. Also could not apply direct method.
- Tuning all the methods was time consuming. Some methods had 3 tunning parameters.

Extension ideas:

- Applying different methods within Model Averaging.
- Better selection of subset for Nystrom method.
- Exploring different types of random sketches in sketching's method.