Introduction to Machine Learning - CS725

Instructor: Prof. Ganesh Ramakrishnan

Lecture 14 - RKHS, Non-parametric Regression

Sequential Minimial Optimization Algorithm for Solving SVR

Solving the SVR Dual Optimization Problem

- It can be shown that the objective:
  $max_{\alpha_i,\alpha_i^*} - \frac{1}{2}\sum_i \sum_j (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(x_i, x_j)$
  $-\epsilon \sum_i (\alpha_i + \alpha_i^*) + \sum_i y_i(\alpha_i - \alpha_i^*)$
- can be written as:
  $max_{\beta_i} - \frac{1}{2}\sum_i \sum_j \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) - \epsilon \sum_i |\beta_i| + \sum_i y_i \beta_i$
  s.t.

[1] https://en.wikipedia.org/wiki/Quadratic_programming#Solvers_and_scripting_.28programming.29_languages

Solving the SVR Dual Optimization Problem

- It can be shown that the objective:
  $max_{\alpha_i,\alpha_i^*} - \frac{1}{2}\sum_i \sum_j (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i, x_j)$
  $-\epsilon \sum_i (\alpha_i + \alpha_i^*) + \sum_i y_i(\alpha_i - \alpha_i^*)$
- can be written as:
  $max_{\beta_i} - \frac{1}{2}\sum_i \sum_j \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) - \epsilon \sum_i |\beta_i| + \sum_i y_i \beta_i$
  s.t.
  - $\sum_i \beta_i = 0$
  - $\beta_i \in [-C, C], \forall i$
- Even for this form, standard QP (LCQP) solvers[1] can be used
- Question: How about (iteratively) solving for two β_i's at a time?
  - This is the idea of the Sequential Minimal Optimization (SMO) algorithm

[1] https://en.wikipedia.org/wiki/Quadratic_programming#Solvers_and_scripting_.28programming.29_languages

Sequential Minimal Optimization (SMO) for SVR

- Consider:
  $max_{\beta_i} - \frac{1}{2} \sum_i \sum_j \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) - \epsilon \sum_i |\beta_i| + \sum_i y_i \beta_i$
  s.t.
  - $\sum_i \beta_i = 0$
  - $\beta_i \in [-C, C], \forall i$
- The SMO subroutine can be defined as:

Sequential Minimal Optimization (SMO) for SVR

- Consider:
  $max_{\beta_i} - \frac{1}{2} \sum_i \sum_j \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) - \epsilon \sum_i |\beta_i| + \sum_i y_i \beta_i$
  s.t.
  - $\sum_i \beta_i = 0$
  - $\beta_i \in [-C, C], \forall i$
- The SMO subroutine can be defined as:
  1. Initialise $\beta_1, \ldots, \beta_n$ to some value $\in [-C, C]$
  2. Pick β_i, β_j to estimate closed form expression for next iterate (i.e. β_i^{new}, β_j^{new})
  3. Check if the KKT conditions are satisfied
     - If not, choose β_i and β_j that worst violate the KKT conditions and reiterate

Lasso and Dual Form

- $$\mathbf{w}^* = \underset{\mathbf{w}}{\mathbf{argmin}} \|\phi\mathbf{w} - \mathbf{y}\|^2 \ \text{s.t.} \ \|\mathbf{w}\|_1 \leq \eta, \tag{1}$$
  where
  $$\|\mathbf{w}\|_1 = \left(\sum_{i=1}^{n} |w_i|\right) \tag{2}$$
- Since $\|\mathbf{w}\|_1$ is not differentiable, one can express (2) as a set of constraints
  $$\sum_{i=1}^{n} \xi_i \leq \eta, \ w_i \leq \xi_i, \ -w_i \leq \xi_i$$
- The resulting problem is a linearly constrained Quadratic optimization problem (LCQP):
  $$\mathbf{w}^* = \underset{\mathbf{w}}{\mathbf{argmin}} \|\phi\mathbf{w} - \mathbf{y}\|^2 \ \text{s.t.} \ \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \xi_{\mathbf{i}} \leq \eta, \ \mathbf{w_i} \leq \xi_{\mathbf{i}}, \ -\mathbf{w_i} \leq \xi_{\mathbf{i}} \tag{3}$$

Non Parametric Regression

Basis function expansion and the Kernel trick: Additional Discussion 1

Consider regression function $f(\mathbf{x}) = \sum_{j=1}^{p} w_j \phi_j(\mathbf{x})$ with weight vector $\mathbf{w}$ estimated as

$$\mathbf{w}_{Pen} = \underset{\mathbf{w}}{\text{argmin}} \; \mathcal{L}(\phi, \mathbf{w}, \mathbf{y}) + \lambda \Omega(\mathbf{w})$$

It can be shown that for $p \in [0, \infty)$, under certain conditions on K, the following can be equivalent representations

-
$$f(\mathbf{x}) = \sum_{j=1}^{p} w_j \phi_j(\mathbf{x})$$

- And[2]
$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}_i)$$

[2] Section 5.8.1 of Tibshi.

The Representer Theorem & Reproducing Kernel Hilbert Space (RKHS)

1. The solution $f^* \in \mathcal{H}$ (Hilbert space) to the following problem

$$f^* = \arg\min_{f \in \mathcal{H}} \sum_{i=1}^{m} \mathbf{E}\left(f\left(\mathbf{x}^{(i)}\right), y^{(i)}\right) + \Omega(\|f\|_K)$$

can be always written as $f^*(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}^{(i)})$, provided $\Omega(\|f\|_K)$ is a ....

2. More specifically, if $f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b$ and $K(\mathbf{x}', \mathbf{x}) = \phi^T(\mathbf{x})\phi(\mathbf{x}')$ then the solution $\mathbf{w}^* \in \Re^n$ to the following problem

$$(\mathbf{w}^*, b^*) = \arg\min_{\mathbf{w}, b} \sum_{i=1}^{m} \mathbf{E}\left(f\left(\mathbf{x}^{(i)}\right), y^{(i)}\right) + \Omega(\|\mathbf{w}\|_2)$$

can be always written as $\phi^T(\mathbf{x})\mathbf{w}^* + b = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}^{(i)})$, provided $\Omega(\|\mathbf{w}\|_2)$ is a monotonically increasing function of $\|\mathbf{w}\|_2$. $\Re^n$ is the Hilbert space and $K(., \mathbf{x}) : \mathcal{X} \to \Re$ is the **Reproducing (RKHS) Kernel**

The Reproducing Kernel Hilbert Space (RKHS)

Consider the set of functions $\mathcal{K} = \{K(.,\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\}$ and let $\mathcal{H}$ be the set of all functions that are **finite** linear combinations of functions in $\mathcal{K}$. That is, any function $h \in \mathcal{H}$ can be written as $\mathbf{h}(.) = \sum_{t=1}^{T} \alpha_t K(.,\mathbf{x}_t)$ for some T and $\mathbf{x}_t \in \mathcal{X}, \alpha_t \in \Re$. One can easily verify that $\mathcal{H}$ is a vector space[3] with an inner product.

[3] Try it yourself. Prove that $\mathcal{H}$ is closed under vector addition and (real) scalar multiplication.

Inner Product over RKHS $\mathcal{H}$

For any $g(.) = \sum_{t=1}^{S} \beta_s K(., \mathbf{x}'_s) \in \mathcal{H}$ and $h(.) = \sum_{t=1}^{T} \alpha_t K(., \mathbf{x}_t) \in \mathcal{H}$, define the inner product[4]

[4] Again, you can verify that $\langle f, g \rangle$ is indeed an inner product following properties such as symmetry, linearity in the first argument and positive-definiteness: `https://en.wikipedia.org/wiki/Inner_product_space`

Inner Product over RKHS $\mathcal{H}$

For any $g(.) = \sum_{t=1}^{S} \beta_s K(., \mathbf{x}'_s) \in \mathcal{H}$ and $h(.) = \sum_{t=1}^{T} \alpha_t K(., \mathbf{x}_t) \in \mathcal{H}$, define the inner product[4]

$$\langle h, g \rangle = \sum_{s=1}^{S} \beta_s \sum_{t=1}^{T} \alpha_t K(\mathbf{x}'_s, \mathbf{x}_t) \tag{4}$$

Further simplifying (4),

$$\langle h, g \rangle = \sum_{s=1}^{S} \beta_s \sum_{t=1}^{T} \alpha_t K(\mathbf{x}'_s, \mathbf{x}_t) = \sum_{s=1}^{S} \beta_s f(\mathbf{x}_s) \tag{5}$$

One immediately observes that in the special case that $g() = K(., \mathbf{x})$,

[4] Again, you can verify that $\langle f, g \rangle$ is indeed an inner product following properties such as symmetry, linearity in the first argument and positive-definiteness: `https://en.wikipedia.org/wiki/Inner_product_space`

Inner Product over RKHS $\mathcal{H}$

For any $g(.) = \sum_{t=1}^{S} \beta_s K(., \mathbf{x}'_s) \in \mathcal{H}$ and $h(.) = \sum_{t=1}^{T} \alpha_t K(., \mathbf{x}_t) \in \mathcal{H}$, define the inner product[4]

$$\langle h, g \rangle = \sum_{s=1}^{S} \beta_s \sum_{t=1}^{T} \alpha_t K(\mathbf{x}'_s, \mathbf{x}_t) \tag{4}$$

Further simplifying (4),

$$\langle h, g \rangle = \sum_{s=1}^{S} \beta_s \sum_{t=1}^{T} \alpha_t K(\mathbf{x}'_s, \mathbf{x}_t) = \sum_{s=1}^{S} \beta_s f(\mathbf{x}_s) \tag{5}$$

One immediately observes that in the special case that $g() = K(., \mathbf{x})$,

$$\langle h, K(., \mathbf{x}) \rangle = h(\mathbf{x}) \tag{6}$$

[4] Again, you can verify that $\langle f, g \rangle$ is indeed an inner product following properties such as symmetry, linearity in the first argument and positive-definiteness: `https://en.wikipedia.org/wiki/Inner_product_space`

Orthogonal Decomposition

Since $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(m)}\} \subseteq \mathcal{X}$ and $\mathcal{K} = \{K(., \mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\}$ with $\mathcal{H}$ being the set of all finite linear combinations of function in $\mathcal{K}$, we also have that

$$lin\_span\left\{K(., \mathbf{x}^{(1)}), K(.\mathbf{x}^{(2)}), \ldots, K(., \mathbf{x}^{(m)})\right\} \subseteq \mathcal{H}$$

Thus, we can use orthogonal projection to

Orthogonal Decomposition

Since $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(m)}\} \subseteq \mathcal{X}$ and $\mathcal{K} = \{K(., \mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\}$ with $\mathcal{H}$ being the set of all finite linear combinations of function in $\mathcal{K}$, we also have that

$$lin\_span\left\{K(., \mathbf{x}^{(1)}), K(.\mathbf{x}^{(2)}), \ldots, K(., \mathbf{x}^{(m)})\right\} \subseteq \mathcal{H}$$

Thus, we can use orthogonal projection to decompose any $h \in \mathcal{H}$ into a sum of two functions, one lying in $lin\_span\left\{K(., \mathbf{x}^{(1)}), K(.\mathbf{x}^{(2)}), \ldots, K(., \mathbf{x}^{(m)})\right\}$, and the other lying in the orthogonal complement:

$$h = h^{\parallel} + h^{\perp} = \sum_{i=1}^{m} \alpha_i K(., \mathbf{x}^{(i)}) + h^{\perp} \tag{7}$$

where $\langle K(., \mathbf{x}^{(i)}), h^{\perp} \rangle = 0$, for each $i = [1..m]$.

For a specific training point $\mathbf{x}^{(j)}$, substituting from (7) into (6) for any $h \in \mathcal{H}$, using the fact that $\langle K(., \mathbf{x}^{(i)}), h^{\perp} \rangle = 0$

$$h(\mathbf{x}^{(j)}) = \langle \sum_{i=1}^{m} \alpha_i K(., \mathbf{x}^{(i)}) + h^{\perp}, K(., \mathbf{x}^{(j)}) \rangle = \sum_{i=1}^{m} \alpha_i \langle K(., \mathbf{x}^{(i)}), K(., \mathbf{x}^{(j)}) \rangle = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \tag{8}$$

which we observe is independent of $h^{\perp}$.

Basis function expansion & Kernel: Additional Discussion 2

Consider regression function $f(\mathbf{x}) = \sum_{j=1}^{p} w_j \phi_j(\mathbf{x})$ with weight vector $\mathbf{w}$ estimated as

$$\mathbf{w}_{Pen} = \underset{\mathbf{w}}{\operatorname{argmin}}\ \mathcal{L}(\phi, \mathbf{w}, \mathbf{y}) + \lambda\Omega(\mathbf{w})$$

It can be shown that for $p \in [0, \infty)$, under certain conditions on K, the following can be equivalent representations

- $$f(\mathbf{x}) = \sum_{j=1}^{p} w_j \phi_j(\mathbf{x})$$
- And[5]
$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}_i)$$

[5] Section 5.8.1 of Tibshi.

Basis function expansion & Kernel: Additional Discussion 2

- We could also begin with (Eg: NadarayaWatson kernel regression)

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}_i) = \frac{\sum_{i=1}^{m} y_i k_n(||\mathbf{x} - \mathbf{x}_i)||}{\sum_{i=1}^{m} k_n(||\mathbf{x} - \mathbf{x}_i||)}$$

A non-parametric kernel k_n is a non-negative real-valued integrable function satisfying the following two requirements: $\int_{-\infty}^{+\infty} k_n(u)du = 1$ and $k_n(-u) = k_n(u)$ for all values of u

[6] Section 2.8.2 of Tibshi

Basis function expansion & Kernel: Additional Discussion 2

- We could also begin with (Eg: NadarayaWatson kernel regression)

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}_i) = \frac{\sum_{i=1}^{m} y_i k_n(||\mathbf{x} - \mathbf{x}_i)||}{\sum_{i=1}^{m} k_n(||\mathbf{x} - \mathbf{x}_i||)}$$

  A non-parametric kernel k_n is a non-negative real-valued integrable function satisfying the following two requirements: $\int_{-\infty}^{+\infty} k_n(u)du = 1$ and $k_n(-u) = k_n(u)$ for all values of u

- *E.g.:* $k_n(x_i - x) = I(||x_i - x|| \leq ||x_{(k)} - x||)$ where $x_{(k)}$ is the training observation ranked k^{th} in distance from x and $I(S)$ is the indicator of the set S
- This is precisely the Nearest Neighbor Regression model
- Kernel regression and density models are other examples of such *local regression* methods[6]

[6] Section 2.8.2 of Tibshi

Basis function expansion & Kernel: Additional Discussion 2

- We could also begin with (Eg: NadarayaWatson kernel regression)

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}_i) = \frac{\sum_{i=1}^{m} y_i k_n(||\mathbf{x} - \mathbf{x}_i)||}{\sum_{i=1}^{m} k_n(||\mathbf{x} - \mathbf{x}_i||)}$$

  A non-parametric kernel k_n is a non-negative real-valued integrable function satisfying the following two requirements: $\int_{-\infty}^{+\infty} k_n(u)du = 1$ and $k_n(-u) = k_n(u)$ for all values of u
- *E.g.:* $k_n(x_i - x) = I(||x_i - x|| \le ||x_{(k)} - x||)$ where $x_{(k)}$ is the training observation ranked k^{th} in distance from x and $I(S)$ is the indicator of the set S
- This is precisely the Nearest Neighbor Regression model
- Kernel regression and density models are other examples of such *local regression* methods[6]
- The broader class - **Non-Parametric Regression:** $y = g(\mathbf{x}) + \epsilon$ where functional form of $g(\mathbf{x})$ is not fixed

[6] Section 2.8.2 of Tibshi

Non-parametric Kernel weighted (Local Linear) Regression: Tut 5, Prob 3

Given $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_i, y_i), \ldots, (\mathbf{x}_n, y_n)\}$, predict $f(\mathbf{x}') = (\mathbf{w}'^\top \phi(\mathbf{x}') + b)$ for each test (or query point) $\mathbf{x}'$ as:

$$(\mathbf{w}', b') = \underset{\mathbf{w}, b}{\text{argmin}} \sum_{i=1}^{n} K(\mathbf{x}', \mathbf{x}_i) \left(y_i - (\mathbf{w}^\top \phi(x_i) + b)\right)^2$$

1. If there is a closed form expression for $(\mathbf{w}', b')$ and therefore for $f(x')$ in terms of the known quantities, derive it.
2. How does this model compare with linear regression and $k-$nearest neighbor regression? What are the relative advantages and disadvantages of this model?
3. In the one dimensional case (that is when $\phi(x) \in \Re$), graphically try and interpret what this regression model would look like, say when $K(.,.)$ is the linear kernel[7].

[7] Hint: What would the regression function look like at each training data point?

Answer to Question 1

The weighing factor $r_i^{x'}$ of each training data point $(\mathbf{x}_i, y_i)$ is now also a function of the query or test data point $(\mathbf{x}', ?)$, so that we write it as $r_i^{x'} = K(\mathbf{x}', \mathbf{x}_i)$ for $i = 1, \ldots, m$. Let $r_{m+1}^{x'} = 1$ and let R be an $(m+1) \times (m+1)$ diagonal matrix of $r_1^{x'}, r_2^{x'}, \ldots, r_{m+1}^{x'}$.

$$R = \begin{bmatrix} r_1^{x'} & 0 & \ldots & 0 & \\ 0 & r_2^{x'} & \ldots & 0 & \\ \ldots & \ldots & \ldots & \ldots & 1 \\ 0 & 0 & 0 & \ldots & r_{m+1}^{x'} \end{bmatrix}$$

Further, let

$$\Phi = \begin{bmatrix} \phi_1(x_1) & \ldots & \phi_p(x_1) & 1 \\ \ldots & \ldots & \ldots & 1 \\ \phi_1(x_m) & \ldots & \phi_p(x_m) & 1 \end{bmatrix}$$

and

Answer to Question 1 (contd.)

$$\widehat{\mathbf{w}} = \begin{bmatrix} w_1 \\ \dots \\ w_p \\ b \end{bmatrix}$$

and

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \dots \\ y_m \end{bmatrix}$$

The sum-square error function then becomes

$$\frac{1}{2}\sum_{i=1}^{m} r_i(y_i - (\widehat{\mathbf{w}}^T\phi(x_i) + b))^2 = \frac{1}{2}||\sqrt{R}\mathbf{y} - \sqrt{R}\Phi\widehat{\mathbf{w}}||_2^2$$

where $\sqrt{R}$ is a diagonal matrix such that each diagonal element of $\sqrt{R}$ is the square root of the corresponding element of R.

Answer to Question 1 (contd.)

The sum-square error function:

$$\frac{1}{2}\sum_{i=1}^{m} r_i(y_i - (\widehat{\mathbf{w}}^T\phi(x_i) + b))^2 = \frac{1}{2}||\sqrt{R}\mathbf{y} - \sqrt{R}\Phi\widehat{\mathbf{w}}||_2^2$$

This convex function has a global minimum at $\widehat{\mathbf{w}}_*^{x'}$ such that

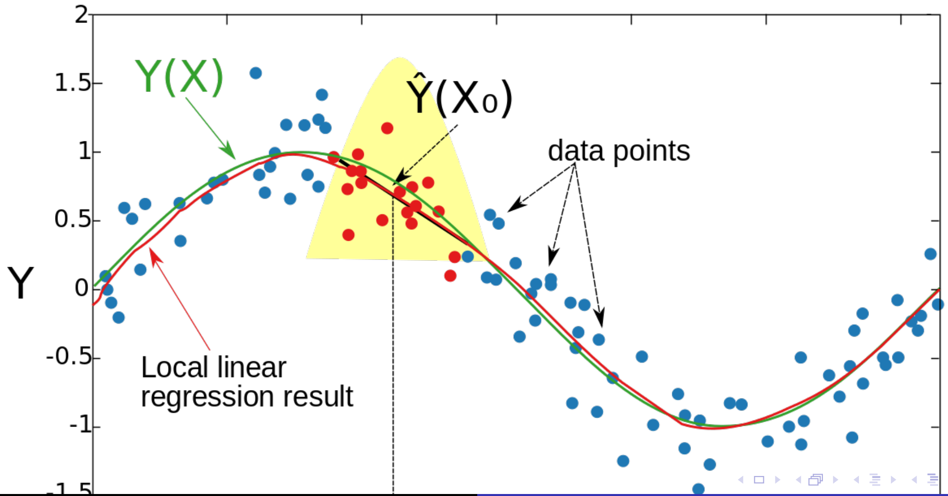
$$\widehat{\mathbf{w}}_*^{x'} = (\Phi^T R\Phi)^{-1}\Phi^T R\mathbf{y}$$

This is referred to as local linear regression (Section 6.1.1 of Tibshi).

Answer to Question 2

1. Local linear regression gives more importance (than linear regression) to points in $\mathcal{D}$ that are closer/similar to $\mathbf{x}'$ and less importance to points that are less similar.
2. Important if the regression curve is supposed to take different shapes in different parts of the space.
3. Local linear regression comes close to k-nearest neighbor. But unlike k-nearest neighbor, local linear regression gives you a smooth solution

Answer to Question 3



Gaussian Process Regressionn