Introduction to Machine Learning - CS725
Instructor: Prof. Ganesh Ramakrishnan
Lecture 13 - Mercer and Positive Definite Kernels, SMO Algorithm

The Kernelized version of SVR

- The kernelized dual problem:

$$max_{\alpha_i,\alpha_i^*} - \frac{1}{2}\sum_i \sum_j (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(\mathbf{x}_i, \mathbf{x}_j)$$

$$-\epsilon \sum_i (\alpha_i + \alpha_i^*) + \sum_i y_i(\alpha_i - \alpha_i^*)$$

  s.t.
  - $\sum_i (\alpha_i - \alpha_i^*) = 0$
  - $\alpha_i, \alpha_i^* \in [0, C]$
- The kernelized decision function:
  $f(\mathbf{x}) = \sum_i (\alpha_i - \alpha_i^*)K(\mathbf{x}_i, \mathbf{x}) + b$
- Using any point x_j with $\alpha_j \in (0, C)$:
  $b = y_j - \sum_i (\alpha_i - \alpha_i^*)K(\mathbf{x}_i, \mathbf{x}_j)$
- Computing $K(\mathbf{x}_1, \mathbf{x}_2)$ often does not even require computing $\phi(\mathbf{x}_1)$ or $\phi(\mathbf{x}_2)$ explicitly

An Example Kernel

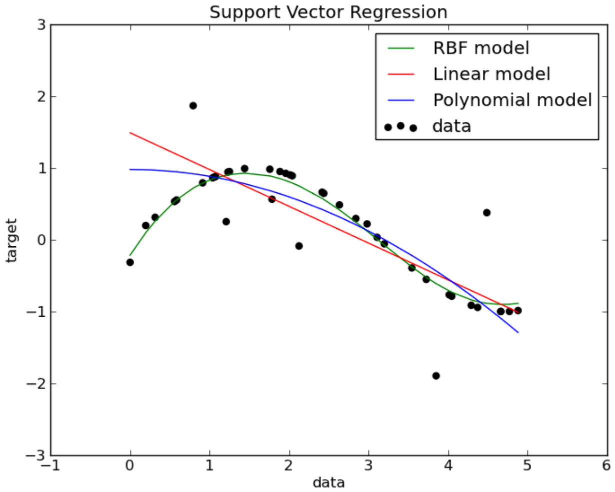
- Let $K(\mathbf{x}_1, \mathbf{x}_2) = (1 + \mathbf{x}_1^\top \mathbf{x}_2)^2$
- Which value of $\phi(\mathbf{x})$ will yield $\phi^\top(\mathbf{x}_1)\phi(\mathbf{x}_2) = K(\mathbf{x}_1, \mathbf{x}_2) = (1 + \mathbf{x}_1^\top \mathbf{x}_2)^2$
- Is such a ϕ guaranteed to exist?
- Is there a unique ϕ for given K?

An Example Kernel

- We can prove that such a ϕ exists
- For example, for a 2-dimensional $\mathbf{x}_i$:

$$\phi(\mathbf{x}_i) = \begin{bmatrix} 1 \\ x_{i1}\sqrt{2} \\ x_{i2}\sqrt{2} \\ x_{i1}x_{i2}\sqrt{2} \\ x_{i1}^2 \\ x_{i2}^2 \end{bmatrix}$$

- $\phi(\mathbf{x}_i)$ exists in a 6-dimensional space
- But, to compute $K(\mathbf{x}_1, \mathbf{x}_2)$, all we need is $x_1^\top x_2$ without having to enumerate $\phi(\mathbf{x}_i)$


Support Vector Regression
RBF model
Linear model
Polynomial model
data
target
data
3
2
1
0
−1
−2
−3
−1
0
1
2
3
4
5
6

More on the Kernel Trick

- **Kernels** operate in a *high-dimensional*, *implicit* feature space without necessarily computing the coordinates of the data in that space, but rather by simply computing the Kernel function
- This approach is called the "*kernel trick*" and will subsequently talk about *valid kernels*
- This operation is often computationally cheaper than the explicit computation of the coordinates
- Claim: If $\mathcal{K}_{ij} = K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$ are entries of an $n \times n$ **Gram Matrix** $\mathcal{K}$ then
  - $\mathcal{K}$ must be positive semi-definite
  - Proof: $\mathbf{b}^T \mathcal{K} \mathbf{b} = \sum_{i,j} b_i \mathcal{K}_{ij} b_j = \sum_{i,j} b_i b_j \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$

$$= \langle \sum_i b_i \phi(\mathbf{x}_i), \sum_j b_j \phi(\mathbf{x}_j) \rangle = || \sum_i b_i \phi(\mathbf{x}_i) ||_2^2 \geq 0$$

Existence of basis expansion ϕ for symmetric K?

- *Positive-definite kernel:* For any dataset $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ and for any m, the Gram matrix $\mathcal{K}$ must be positive definite

$$\mathcal{K} = \begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & \ldots & K(\mathbf{x}_1, \mathbf{x}_n) \\ \ldots & K(\mathbf{x}_i, \mathbf{x}_j) & \ldots \\ K(\mathbf{x}_m, \mathbf{x}_1) & \ldots & K(\mathbf{x}_m, \mathbf{x}_m) \end{bmatrix}$$

so that $\mathcal{K} = U\Sigma U^T = (U\Sigma^{\frac{1}{2}})(U\Sigma^{\frac{1}{2}})^T = RR^T$ where rows of U are linearly independent and Σ is a positive diagonal matrix

[1] Eigen-decomposition wrt linear operators. See `https://en.wikipedia.org/wiki/Mercer%27s_theorem`

[2] That is, if every Cauchy sequence is convergent.

Existence of basis expansion ϕ for symmetric K?

- *Positive-definite kernel:* For any dataset $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ and for any m, the Gram matrix $\mathcal{K}$ must be positive definite

$$\mathcal{K} = \begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & \ldots & K(\mathbf{x}_1, \mathbf{x}_n) \\ \ldots & K(\mathbf{x}_i, \mathbf{x}_j) & \ldots \\ K(\mathbf{x}_m, \mathbf{x}_1) & \ldots & K(\mathbf{x}_m, \mathbf{x}_m) \end{bmatrix}$$

  so that $\mathcal{K} = U\Sigma U^T = (U\Sigma^{\frac{1}{2}})(U\Sigma^{\frac{1}{2}})^T = RR^T$ where rows of U are linearly independent and Σ is a positive diagonal matrix

- *Mercer kernel:* Extending to eigenfunction decomposition[1]:

$$K(\mathbf{x}_1, \mathbf{x}_2) = \sum_{j=1}^{\infty} \alpha_j \phi_j(\mathbf{x}_1)\phi_j(\mathbf{x}_2) \text{ where } \alpha_j \geq 0 \text{ and } \sum_{j=1}^{\infty} \alpha_j^2 < \infty$$

- *Mercer kernel* and *Positive-definite kernel* turn out to be equivalent if the input space $\{x\}$ is *compact*[2]

[1] Eigen-decomposition wrt linear operators. See https://en.wikipedia.org/wiki/Mercer%27s_theorem

[2] That is, if every Cauchy sequence is convergent.

Mercer's theorem

- **Mercer kernel:** $K(\mathbf{x}_1, \mathbf{x}_2)$ is a Mercer kernel if $\int\int K(\mathbf{x}_1, \mathbf{x}_2) g(\mathbf{x}_1) g(\mathbf{x}_2)\, d\mathbf{x}_1 d\mathbf{x}_2 \geq 0$ for all square integrable functions $g(\mathbf{x})$ ($g(\mathbf{x})$ is square integrable *iff* $\int (g(\mathbf{x}))^2\, dx$ is finite)
- **Mercer's theorem:**
  An implication of the theorem:
  for any *Mercer kernel* $K(\mathbf{x}_1, \mathbf{x}_2)$, $\exists\, \phi(\mathbf{x}) : \mathbb{R}^n \mapsto H$,
  s.t. $K(\mathbf{x}_1, \mathbf{x}_2) = \phi^\top(\mathbf{x}_1)\phi(\mathbf{x}_2)$
  - where H is a *Hilbert space*[3], the infinite dimensional version of the Eucledian space.
  - Eucledian space: $(\Re^n, < .,. >)$ where $< .,. >$ is the standard dot product in $\Re^n$
  - Advanced: Formally, *Hibert Space* is an inner product space with associated norms, where every Cauchy sequence is convergent

[3] Do you know Hilbert? No? Then what are you doing in his space? :)

Prove that $(\mathbf{x}_1^\top \mathbf{x}_2)^d$ is a Mercer kernel ($d \in \mathbb{Z}^+$, $d \geq 1$)

- We want to prove that
  $\int_{\mathbf{x}_1} \int_{\mathbf{x}_2} (\mathbf{x}_1^\top \mathbf{x}_2)^d g(\mathbf{x}_1) g(\mathbf{x}_2)\, d\mathbf{x}_1 d\mathbf{x}_2 \geq 0$,
  for all square integrable functions $g(\mathbf{x})$
- Here, $\mathbf{x}_1$ and $\mathbf{x}_2$ are vectors s.t $\mathbf{x}_1, \mathbf{x}_2 \in \Re^t$
- Thus, $\int_{\mathbf{x}_1} \int_{\mathbf{x}_2} (\mathbf{x}_1^\top \mathbf{x}_2)^d g(\mathbf{x}_1) g(\mathbf{x}_2)\, d\mathbf{x}_1 d\mathbf{x}_2$

$$= \int_{x_{11}} .. \int_{x_{1t}} \int_{x_{21}} .. \int_{x_{2t}} \left[\sum_{n_1..n_t} \frac{d!}{n_1!..n_t!} \prod_{j=1}^{t} (x_{1j} x_{2j})^{n_j} \right] g(x_1) g(x_2)\, dx_{11}..dx_{1t} dx_{21}..dx_{2t}$$

$$\text{s.t.} \sum_{i=1}^{t} n_i = d$$

(taking a leap)

Prove that $(\mathbf{x}_1^\top \mathbf{x}_2)^d$ is a Mercer kernel ($d \in \mathbb{Z}^+$, $d \geq 1$)

$$= \sum_{n_1 \dots n_t} \frac{d!}{n_1! \dots n_t!} \int_{\mathbf{x}_1} \int_{\mathbf{x}_2} \prod_{j=1}^{t} (x_{1j} x_{2j})^{n_j} \, g(x_1) g(x_2) \, dx_1 \, dx_2$$

$$= \sum_{n_1 \dots n_t} \frac{d!}{n_1! \dots n_t!} \int_{\mathbf{x}_1} \int_{\mathbf{x}_2} (x_{11}^{n_1} x_{12}^{n_2} \dots x_{1t}^{n_t}) g(x_1) \, (x_{21}^{n_1} x_{22}^{n_2} \dots x_{2t}^{n_t}) g(x_2) \, dx_1 \, dx_2$$

Prove that $(\mathbf{x}_1^\top \mathbf{x}_2)^d$ is a Mercer kernel ($d \in \mathbb{Z}^+$, $d \geq 1$)

$$= \sum_{n_1 \ldots n_t} \frac{d!}{n_1! \ldots n_t!} \int_{\mathbf{x}_1} \int_{\mathbf{x}_2} \prod_{j=1}^{t} (x_{1j} x_{2j})^{n_j} \, g(x_1) g(x_2) \, dx_1 dx_2$$

$$= \sum_{n_1 \ldots n_t} \frac{d!}{n_1! \ldots n_t!} \int_{\mathbf{x}_1} \int_{\mathbf{x}_2} (x_{11}^{n_1} x_{12}^{n_2} \ldots x_{1t}^{n_t}) g(x_1) \, (x_{21}^{n_1} x_{22}^{n_2} \ldots x_{2t}^{n_t}) g(x_2) \, dx_1 dx_2$$

$$= \sum_{n_1 \ldots n_t} \frac{d!}{n_1! \ldots n_t!} \left(\int_{\mathbf{x}_1} (x_{11}^{n_1} \ldots x_{1t}^{n_t}) g(x_1) \, dx_1 \right) \left(\int_{\mathbf{x}_2} (x_{21}^{n_1} \ldots x_{2t}^{n_t}) g(x_2) \, dx_2 \right)$$

(integral of decomposable product as product of integrals)

$$\text{s.t.} \sum_{i}^{t} n_i = d$$

Prove that $(\mathbf{x}_1^\top \mathbf{x}_2)^d$ is a Mercer kernel ($d \in \mathbb{Z}^+$, $d \geq 1$)

- Realize that both the integrals are basically the same, with different variable names
- Thus, the equation becomes:

$$\sum_{n_1 \ldots n_t} \frac{d!}{n_1! \ldots n_t!} \left(\int_{\mathbf{x}_1} (x_{11}^{n_1} \ldots x_{1t}^{n_t}) g(x_1) \, dx_1 \right)^2 \geq 0$$

(the square is non-negative for reals)

- Thus, we have shown that $(\mathbf{x}_1^\top \mathbf{x}_2)^d$ is a Mercer kernel.

What about $\sum_{d=1}^{r} \alpha_d (\mathbf{x}_1^\top \mathbf{x}_2)^d$ s.t. $\alpha_d \geq 0$?

- $K(\mathbf{x}_1, \mathbf{x}_2) = \sum_{d=1}^{r} \alpha_d (\mathbf{x}_1^\top \mathbf{x}_2)^d$
- Is $\int_{\mathbf{x}_1} \int_{\mathbf{x}_2} \left(\sum_{d=1}^{r} \alpha_d (x_1^\top x_2)^d \right) g(\mathbf{x}_1) g(\mathbf{x}_2) \, d\mathbf{x}_1 d\mathbf{x}_2 \geq 0$?
- We have

$$\int_{\mathbf{x}_1} \int_{\mathbf{x}_2} \left(\sum_{d=1}^{r} \alpha_d (\mathbf{x}_1^\top \mathbf{x}_2)^d \right) g(x_1) g(x_2) \, dx_1 dx_2 =$$

What about $\sum_{d=1}^{r} \alpha_d (\mathbf{x}_1^\top \mathbf{x}_2)^d$ s.t. $\alpha_d \geq 0$?

- $K(\mathbf{x}_1, \mathbf{x}_2) = \sum_{d=1}^{r} \alpha_d (\mathbf{x}_1^\top \mathbf{x}_2)^d$
- Is $\int_{\mathbf{x}_1} \int_{\mathbf{x}_2} \left(\sum_{d=1}^{r} \alpha_d (x_1^\top x_2)^d \right) g(\mathbf{x}_1) g(\mathbf{x}_2) \, d\mathbf{x}_1 d\mathbf{x}_2 \geq 0$?
- We have

$$\int_{\mathbf{x}_1} \int_{\mathbf{x}_2} \left(\sum_{d=1}^{r} \alpha_d (\mathbf{x}_1^\top \mathbf{x}_2)^d \right) g(x_1) g(x_2) \, dx_1 dx_2 =$$

$$\sum_{d=1}^{r} \alpha_d \int_{\mathbf{x}_1} \int_{\mathbf{x}_2} (\mathbf{x}_1^\top \mathbf{x}_2)^d g(\mathbf{x}_1) g(\mathbf{x}_2) \, d\mathbf{x}_1 d\mathbf{x}_2$$

What about $\sum_{d=1}^{r} \alpha_d (\mathbf{x}_1^\top \mathbf{x}_2)^d$ s.t. $\alpha_d \geq 0$?

- We have already proved that $\int_{\mathbf{x}_1} \int_{\mathbf{x}_2} (\mathbf{x}_1^\top \mathbf{x}_2)^d g(\mathbf{x}_1) g(\mathbf{x}_2) \, d\mathbf{x}_1 d\mathbf{x}_2 \geq 0$
- Also, $\alpha_d \geq 0$, $\forall d$
- Thus,
$$\sum_{d=1}^{r} \alpha_d \int_{\mathbf{x}_1} \int_{\mathbf{x}_2} (\mathbf{x}1^\top \mathbf{x}_2)^d g(\mathbf{x}_1) g(\mathbf{x}_2) \, d\mathbf{x}_1 d\mathbf{x}_2 \geq 0$$
- By which, $K(\mathbf{x}_1, \mathbf{x}_2) = \sum_{d=1}^{r} \alpha_d (\mathbf{x}_1^\top \mathbf{x}_2)^d$ is a Mercer kernel.
- Examples of Mercer Kernels: Linear Kernel, Polynomial Kernel, Radial Basis Function Kernel

Closure properties of Kernels

Let $K_1(\mathbf{x}_1, \mathbf{x}_2)$ and $K_2(\mathbf{x}_1, \mathbf{x}_2)$ be positive definite (valid) kernels. Then the following are also kernels.

- $\alpha_1 K_1(\mathbf{x}_1, \mathbf{x}_2) + \alpha_2 K_2(\mathbf{x}_1, \mathbf{x}_2)$ for $\alpha_1, \alpha_2 \geq 0$.
  Proof:

Closure properties of Kernels

Let $K_1(\mathbf{x}_1, \mathbf{x}_2)$ and $K_2(\mathbf{x}_1, \mathbf{x}_2)$ be positive definite (valid) kernels. Then the following are also kernels.

- $\alpha_1 K_1(\mathbf{x}_1, \mathbf{x}_2) + \alpha_2 K_2(\mathbf{x}_1, \mathbf{x}_2)$ for $\alpha_1, \alpha_2 \geq 0$.
  Proof:
- $K_1(\mathbf{x}_1, \mathbf{x}_2) K_2(\mathbf{x}_1, \mathbf{x}_2)$
  Proof:

Kernels in SVR

- Recall:
  $max_{\alpha_i,\alpha_i^*} -\frac{1}{2}\sum_i\sum_j(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)K(\mathbf{x}_i,\mathbf{x}_j)-\epsilon\sum_i(\alpha_i+\alpha_i^*)+\sum_i y_i(\alpha_i-\alpha_i^*)$
  and the decision function:
  $f(x)=\sum_i(\alpha_i-\alpha_i^*)K(\mathbf{x}_i,\mathbf{x})+b$
  are all in terms of the kernel $K(\mathbf{x}_i,\mathbf{x}_j)$ only
- *One can now employ any mercer kernel in SVR or Ridge Regression to implicitly perform linear regression in higher dimensional spaces*

Solving the SVR Dual Optimization Problem

- The SVR dual objective is:
  $max_{\alpha_i,\alpha_i^*} - \frac{1}{2}\sum_i\sum_j(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i, x_j)$
  $-\epsilon\sum_i(\alpha_i + \alpha_i^*) + \sum_i y_i(\alpha_i - \alpha_i^*)$ such that $\sum_i(\alpha_i - \alpha_i^*) = 0$, $\alpha_i, \alpha_i^* \in [0, C]$
- This is a linearly constrained quadratic program (LCQP), just like the

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

Solving the SVR Dual Optimization Problem

- The SVR dual objective is:
  $max_{\alpha_i,\alpha_i^*} - \frac{1}{2}\sum_i \sum_j (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i, x_j)$
  $-\epsilon \sum_i (\alpha_i + \alpha_i^*) + \sum_i y_i(\alpha_i - \alpha_i^*)$ such that $\sum_i (\alpha_i - \alpha_i^*) = 0$, $\alpha_i, \alpha_i^* \in [0, C]$
- This is a linearly constrained quadratic program (LCQP), just like the constrained version of Lasso
- There exists no closed form solution to this formulation
- Standard QP (LCQP) solvers[4] can be used
- Question: Are there more specific and efficient algorithms for solving SVR in this form?

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

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.

[5] 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[5] 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

[5] 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