Labs
Machine Learning Course
Fall 2022

EPFL
School of Computer and Communication Sciences
Nicolas Flammarion & Martin Jaggi
www.epfl.ch/labs/mlo/machine-learning-cs-433

Problem Set 7, Nov 3, 2022
(Theory Questions Part)

2. Support Vector Machines using Coordinate Descent

1. The dual objective function that we have to optimise is the following :

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{maximize}} \quad & f(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^\top \mathbf{1} - \tfrac{1}{2\lambda}\boldsymbol{\alpha}^\top \boldsymbol{Q}\boldsymbol{\alpha} \\ \text{subject to} \quad & \boldsymbol{\alpha} \in [0,1]^N \end{aligned}$$

where $\boldsymbol{Q} := \text{diag}(\boldsymbol{y})\boldsymbol{X}\boldsymbol{X}^\top \text{diag}(\boldsymbol{y})$. For computing coordinate update for one coordinate n, consider the following one variable sub-problem:

$$\begin{aligned} \underset{\gamma \in \mathbb{R}}{\text{maximize}} \quad & f(\boldsymbol{\alpha} + \gamma \boldsymbol{e}_n) \\ \text{subject to} \quad & 0 \leq \alpha_n + \gamma \leq 1 \end{aligned}$$

where $\boldsymbol{e}_n = [0, \cdots, 1, \cdots, 0]^\top$ (all zero vector except at the n^{th} position). We can explicitly develop $f(\boldsymbol{\alpha} + \gamma \boldsymbol{e}_n)$ as a polynomial in γ, indeed:

$$\begin{aligned} f(\boldsymbol{\alpha} + \gamma \boldsymbol{e}_n) &= (\boldsymbol{\alpha} + \gamma \boldsymbol{e}_n)^\top \mathbf{1} - \tfrac{1}{2\lambda}(\boldsymbol{\alpha} + \gamma \boldsymbol{e}_n)^\top \boldsymbol{Q}(\boldsymbol{\alpha} + \gamma \boldsymbol{e}_n) \\ &= \boldsymbol{\alpha}^\top \mathbf{1} - \tfrac{1}{2\lambda}\boldsymbol{\alpha}^\top \boldsymbol{Q}\boldsymbol{\alpha} + \gamma - \tfrac{1}{2\lambda}(\gamma^2 \boldsymbol{e}_n^\top \boldsymbol{Q}\boldsymbol{e}_n + \gamma \boldsymbol{\alpha}^\top \boldsymbol{Q}\boldsymbol{e}_n + \gamma \boldsymbol{e}_n^\top \boldsymbol{Q}\boldsymbol{\alpha}) \\ &= f(\boldsymbol{\alpha}) - \tfrac{\gamma^2}{2\lambda}\boldsymbol{Q}_{nn} + \gamma(1 - \tfrac{1}{\lambda}\boldsymbol{\alpha}^\top \boldsymbol{Q}\boldsymbol{e}_n) \end{aligned}$$

Hence we recognize a concave second degree polynomial, which has a unique maximum over $\mathbb{R}$. Differentiating with respect to γ and setting to 0 to obtain its maximum over $\mathbb{R}$, we get :

$$\begin{aligned} -\tfrac{\gamma^\star}{\lambda}\boldsymbol{Q}_{nn} + (1 - \tfrac{1}{\lambda}\boldsymbol{\alpha}^\top \boldsymbol{Q}\boldsymbol{e}_n) &= 0 \\ \gamma^\star &= \frac{\lambda}{\boldsymbol{Q}_{nn}}(1 - \tfrac{1}{\lambda}\boldsymbol{\alpha}^\top \boldsymbol{Q}\boldsymbol{e}_n) \end{aligned}$$

Note that $\boldsymbol{Q}_{nn} = \boldsymbol{x}_n^\top \boldsymbol{x}_n y_n^2 = \boldsymbol{x}_n^\top \boldsymbol{x}_n$ and $\boldsymbol{\alpha}^\top \boldsymbol{Q}\boldsymbol{e}_n = \sum_{i=1}^N \alpha_i \boldsymbol{Q}_{i,n} = \sum_{i=1}^N \alpha_i y_i \boldsymbol{x}_i^\top \boldsymbol{x}_n y_n$. Using $\boldsymbol{w}(\boldsymbol{\alpha}) = \frac{1}{\lambda}\sum_{i=1}^N \alpha_i y_i \boldsymbol{x}_i$, we get $\boldsymbol{\alpha}^\top \boldsymbol{Q}\boldsymbol{e}_n = \lambda y_n \boldsymbol{w}^\top \boldsymbol{x}_n$ and thus

$$\gamma^\star = \frac{\lambda}{\boldsymbol{x}_n^\top \boldsymbol{x}_n}(1 - y_n \boldsymbol{w}^\top \boldsymbol{x}_n)$$

We conclude

$$\begin{aligned} \alpha_n^{\text{new}} &= \alpha_n + \gamma^\star \\ &= \alpha_n + \frac{\lambda}{\boldsymbol{x}_n^\top \boldsymbol{x}_n}(1 - y_n \boldsymbol{w}^\top \boldsymbol{x}_n) \end{aligned}$$

In the previous equation it *seems* as if α_n^{new} depends on α_n, which should not be the case, you can check that it is indeed not the case. Since we have a constraint $\boldsymbol{\alpha} \in [0,1]^N$ and we know that function f is quadratic with respect to α_n, the optimal α_n is the projection of α_n^{new} onto the set $[0,1]^N$:

$$\alpha_n^{\text{new}} := \min\Big\{ \max\Big\{ \alpha_n + \frac{\lambda}{\boldsymbol{x}_n^\top \boldsymbol{x}_n}(1 - y_n \boldsymbol{w}^\top \boldsymbol{x}_n), 0 \Big\}, 1 \Big\}$$

2. In lecture we seen the relation between primal and dual variables that is

$$\boldsymbol{w}(\boldsymbol{\alpha}) = \frac{1}{\lambda}\sum_{n=1}^{N}\alpha_n y_n \boldsymbol{x}_n$$

Assume that for current dual variable $\boldsymbol{\alpha}$ the corresponding $\boldsymbol{w}(\boldsymbol{\alpha})$ is known. After a coordinate update over a coordinate n_1 the dual variable changes as $\boldsymbol{\alpha}^{\text{new}} = \boldsymbol{\alpha} + \gamma^* \boldsymbol{e}_{n_1}$, and then the primal variable changes as

$$\boldsymbol{w}^{\text{new}}(\boldsymbol{\alpha}^{\text{new}}) = \frac{1}{\lambda}\sum_{n=1}^{N}\alpha_n^{\text{new}} y_n \boldsymbol{x}_n = \frac{1}{\lambda}\sum_{n=1}^{N}\alpha_n y_n \boldsymbol{x}_n + \frac{1}{\lambda}\gamma^* y_{n_1}\boldsymbol{x}_{n_1} = \boldsymbol{w} + \frac{1}{\lambda}\gamma^* y_{n_1}\boldsymbol{x}_{n_1}$$

4. We see that for the dataset used in this exercise, optimizing with coordinate descent is faster than using SGD. We can also see that the duality gap goes to zero.

   In general practical performance of SGD for SVM with well tuned stepsize is identical to dual Coordinate Descent, however, the advantage of CD is that no stepsize tuning is needed.

3. Kernels

1. • First we will prove that the sum or two valid kernels k_1 and k_2 $k = k_1 + k_2$ is a valid kernel. We need to construct a feature vector $\phi(\boldsymbol{x})$ such that $k(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})^\top \phi(\boldsymbol{x}')$, then by definition k would be a valid kernel.

   Because kernels k_1 and k_2 are valid kernels

   $$k_1(\boldsymbol{x}, \boldsymbol{x}') = \phi_1(\boldsymbol{x})^\top \phi_1(\boldsymbol{x}'), \qquad k_2(\boldsymbol{x}, \boldsymbol{x}') = \phi_2(\boldsymbol{x})^\top \phi_2(\boldsymbol{x}'),$$

   for some feature vectors $\phi_1(\boldsymbol{x})$ and $\phi_2(\boldsymbol{x})$.

   Lets take $\phi(\boldsymbol{x}) = \begin{pmatrix}\phi_1(\boldsymbol{x}) \\ \phi_2(\boldsymbol{x})\end{pmatrix}$, then

   $$\begin{aligned}\phi(\boldsymbol{x})^\top \phi(\boldsymbol{x}') &= \left(\phi_1(\boldsymbol{x})^\top, \phi_2(\boldsymbol{x})^\top\right)\begin{pmatrix}\phi_1(\boldsymbol{x}') \\ \phi_2(\boldsymbol{x}')\end{pmatrix} = \phi_1(\boldsymbol{x})^\top \phi_1(\boldsymbol{x}') + \phi_2(\boldsymbol{x})^\top \phi_2(\boldsymbol{x}') \\ &= k_1(\boldsymbol{x}, \boldsymbol{x}') + k_2(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}, \boldsymbol{x}')\end{aligned}$$

   Therefore $k = k_1 + k_2$ is a valid kernel.

   • Second, we will prove that the product $k = k_1 \cdot k_2$ of two valid kernels is a valid kernel.

   Let's denote n_1 and n_2 dimensions of a feature vectors $\phi_1(\boldsymbol{x})$ and $\phi_2(\boldsymbol{x})$ (i.e. $\phi_1(\boldsymbol{x}) \in \mathbb{R}^{n_1}$, $\phi_2(\boldsymbol{x}) \in \mathbb{R}^{n_1}$).

   $$k_1(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=0}^{n_1-1}\phi_{1,i}(\boldsymbol{x})\phi_{1,i}(\boldsymbol{x}'), \qquad k_2(\boldsymbol{x}, \boldsymbol{x}') = \sum_{j=0}^{n_2-1}\phi_{2,j}(\boldsymbol{x})\phi_{2,j}(\boldsymbol{x}'),$$

   Then the kernel $k = k_1 \cdot k_2$ is

   $$k(\boldsymbol{x}, \boldsymbol{x}') = \left(\sum_{i=0}^{n_1-1}\phi_{1,i}(\boldsymbol{x})\phi_{1,i}(\boldsymbol{x}')\right)\left(\sum_{j=0}^{n_2-1}\phi_{2,j}(\boldsymbol{x})\phi_{2,j}(\boldsymbol{x}')\right) = \sum_{i=0}^{n_1-1}\sum_{j=0}^{n_2-1}\left(\phi_{1,i}(\boldsymbol{x})\phi_{2,j}(\boldsymbol{x})\right)\left(\phi_{1,i}(\boldsymbol{x}')\phi_{2,j}(\boldsymbol{x}')\right)$$

   Lets introduce a feature vector $\phi(\boldsymbol{x}) \in \mathbb{R}^{n_1 n_2}$, such that $\phi_{in_2+j}(\boldsymbol{x}) = \phi_{1,i}(\boldsymbol{x})\phi_{2,j}(\boldsymbol{x})$ for $i \in [0, \ldots, n_1 - 1], j \in [0, \ldots, n_2 - 1]$. Note that for such i and j the index of the feature vector ϕ ic correct: $in_2 + j \in [0, \ldots, n_1 n_2 - 1]$. Then,

   $$\begin{aligned}\phi(\boldsymbol{x})^\top \phi(\boldsymbol{x}') &= \sum_{l=0}^{n_1 n_2-1}\phi_l(\boldsymbol{x})\phi_l(\boldsymbol{x}') = \sum_{i=0}^{n_1-1}\sum_{j=0}^{n_2-1}\phi_{in_2+j}(\boldsymbol{x})\phi_{in_2+j}(\boldsymbol{x}') \\ &= \sum_{i=0}^{n_1-1}\sum_{j=0}^{n_2-1}\left(\phi_{1,i}(\boldsymbol{x})\phi_{2,j}(\boldsymbol{x})\right)\left(\phi_{1,i}(\boldsymbol{x}')\phi_{2,j}(\boldsymbol{x}')\right) = k_1(\boldsymbol{x}, \boldsymbol{x}') \cdot k_2(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}, \boldsymbol{x}').\end{aligned}$$

   Therefore $k(\boldsymbol{x}, \boldsymbol{x}') = k_1(\boldsymbol{x}, \boldsymbol{x}') \cdot k_2(\boldsymbol{x}, \boldsymbol{x}')$ is a valid kernel.

- Third we need to show that if k_1 is a valid kernel, then $k = c \times k_1$ with $c \geq 0$ is also a valid kernel. Since k_1 is valid, we can write for all x and x': $k_1(x, x') = \phi_1(x)^T \phi_1(x')$. Now let $\phi(x) = \sqrt{c}\phi_1(x)$. Notice that $k(x, x') = c \cdot k_1(x, x') = (\sqrt{c}\phi_1(x))^T(\sqrt{c}\phi_1(x')) = \phi(x)^T\phi(x')$. Hence $c \times k_1$ is also a valid kernel.
- Since f is only composed of the three previous operations (sum, product, multiplication by positive scalar), we can conclude that $(x, x') \mapsto f(k_1(x, x'))$ is a valid kernel.

2. It suffices to apply the hint to the sequence of kernels $k_n(x, x') = \sum_{i=0}^{n} \frac{1}{i!} k_1(x, x')^i$. According to our previous result these are valid kernels. Since $k_n(x, x') \underset{n \to +\infty}{\longrightarrow} \exp(k_1(x, x')$ we can apply the hint and conclude that k is a valid kernel.

Bonus. For the curious who are familiar with matrices and the trace operator, here is an elegant and more natural way of showing that the product of two valid kernels is a valid kernel. Notice that for $x, x' \in \mathcal{X}^2$:

$$\begin{aligned}
k(x, x') &= k_1(x, x') \cdot k_2(x, x') \\
&= \phi_1(x)^T \phi_1(x') \phi_2(x)^T \phi_2(x') \\
&= \phi_1(x)^T \phi_1(x') \phi_2(x')^T \phi_2(x) \\
&= \text{trace}(\phi_1(x)^T \phi_1(x') \phi_2(x')^T \phi_2(x)) \\
&= \text{trace}(\phi_1(x') \phi_2(x')^T \phi_2(x) \phi_1(x)^T) \\
&= \text{trace}((\phi_2(x') \phi_1(x')^T)^T \phi_2(x) \phi_1(x)^T) \\
&= \langle \phi_2(x) \phi_1(x)^T,\ \phi_2(x') \phi_1(x')^T \rangle_F
\end{aligned}$$

Second equality is the definition of the valid kernels k_1 and k_2, third is due to $x^T y = y^T x$, fourth is noticing that for $z \in \mathbb{R}$ trace$(z) = z$, fifth is that trace$(AB) =$ trace(BA), seventh is due to $xy^T = (yx^T)^T$, and last is the definition of the Frobenius inner product for matrices. Hence by letting $\phi(x) = \phi_2(x)\phi_1(x)^T \in \mathbb{R}^{n_2 \times n_1}$ we obtain $k(x, x') = \langle \phi(x), \phi(x') \rangle_F$.