Kernels

Optimization for Machine Learning — Exercise #3

Monday 15th May, 2023

Part I: Theory

I.1. Kernel Regression

Exercise I.1 (Kernel derivations)**.** Given a dataset $\mathcal{D} = \{x^{(j)}, t^{(j)}\}_{j\in[N]}$, where, for $j \in [N]$, $(x^{(j)}, t^{(j)}) \in \mathbb{R}^d \times \mathbb{R}$. Let $X \in \mathbb{R}^{d\times N}$ be the stacked samples as columns, and $t \in \mathbb{R}^N$ be the stacked targets. Denote by $\mathrm{span}(X)$ the set of vectors spanned by the columns of X: $\mathrm{span}(X) = \{y \mid \exists(\alpha_j)_{j\in[N]} \in \mathbb{R}^N,\, y = \sum_{j\in[N]} \alpha_j x^{(j)}\}$.

1. Show that any weight for a linear prediction $y(x) = \langle x, w\rangle$ is such that $w \in \mathrm{span}(X)$.
2. What if we had features, i.e. $\phi\colon \mathbb{R}^d \to \mathbb{R}^M$, such that $w \in \mathbb{R}^M$ and $y(x) = \langle \phi(x), w\rangle$?

Answer I.1.

1. Given the linear subspace $\mathrm{span}(X)$ and its orthogonal complement $\mathrm{span}(X)^\perp$, the space $\mathbb{R}^d$ can be orthogonally decomposed as $\mathbb{R}^d = \mathrm{span}(X) \oplus \mathrm{span}(X)^\perp$, meaning that each $w \in \mathbb{R}^d$ can be uniquely written as $w = w_\parallel + w_\perp$ with $w_\parallel \in \mathrm{span}(X)$ and $w_\perp \in \mathrm{span}(X)^\perp$. Recall that $\mathrm{span}(X)^\perp = \{z \in \mathbb{R}^d \mid \forall j \in [N], \langle z, x^{(j)}\rangle = 0\}$.

   With this decomposition, the prediction on the dataset becomes

   $$\forall j \in [N], \quad y(x^{(j)}) = \langle x^{(j)}, w\rangle = \langle x^{(j)}, w_\parallel + w_\perp\rangle = \langle x^{(j)}, w_\parallel\rangle + \langle x^{(j)}, w_\perp\rangle = \langle x^{(j)}, w_\parallel\rangle,$$

   where $\langle x^{(j)}, w_\perp\rangle = 0$ since $w_\perp \in \mathrm{span}(X)^\perp$.

   Since, by definition, the training only evaluates the prediction on the training dataset, we have $w = w_\parallel \in \mathrm{span}(X)$.

2. In the case where features are used, the same argument can be written with $\Phi \in \mathbb{R}^{M\times N}$ instead of X, and we get $w \in \mathrm{span}(\Phi)$.

Exercise I.2 (Primal representation – Curse of dimensionality)**.** This exercise portrays an example of a model that scales badly (exponentially) with the dimension of the samples. This phenomenon is called the curse of dimensionality.

Imagine regressing a function $f\colon \mathcal{C} \to \mathbb{R}$ on the unit cube in dimension d: $\mathcal{C} = [0,1]^d$.

One way to do it is by using M basis functions $\{\phi_i\}_{i\in[M]}$, and try to express the unknown f as a sum of those basis functions, i.e. find $w \in \mathbb{R}^M$ such that $f = \sum_{i\in[M]} w_i\phi_i$.

In Exercise Sheet #1 I.1, we have seen the polynomial example $\phi_i(x) = x^i$ for $d = 1$. An issue of the polynomial is the behaviour when x grows or shrinks at infinity.

Another typical family of functions, which does not suffer from this behaviour, is called the Radial Basis Functions (RBF). They are of this form

$$\begin{array}{rccl} \phi_i & : & \mathbb{R}^d & \longrightarrow & \mathbb{R} \\ & & x & \longmapsto & \exp\left(-\frac{\|x-\mu^{(i)}\|^2}{2\sigma^2}\right) \end{array},$$

with $i \in [M]$, $\mu^{(i)}$ different points at which the exponential are centred, and σ is the variance of the Gaussian functions.

Notice how the functions ϕ_i have to be "dense" in the cube in order to approximate the target correctly.

1. What is the issue with this formulation?
2. Using the result from Ex. I.1, what could be a solution to this problem?

Answer I.2.

1. This formulation requires M basis functions in order to estimate the target function, one for each mean $\mu^{(i)}$. For a given resolution, this number grows exponentially with the dimension, as the grid made of the different locations $\mu^{(i)}$ have to fill the volume $\mathcal{C} = [0,1]^d$. For instance, for a resolution r, we need $M = r$ locations in 1D, $M = r^2$ locations in 2D, $M = r^d$ in dimension d. This is problematic in high dimensions.
2. From Ex I.1 (2.), we known that the parameters w can be written as a linear combination of the training features: $w \in \operatorname{span}(\Phi) \implies \exists(\alpha_j)_{j\in[N]} : w = \sum_{j\in[N]} \alpha_j\phi(x^{(j)})$.

   Now, only the N coefficients $\alpha_{i i\in[N]}$ have to be estimated, and their number is fixed with respect to the dimension of the data points.

Exercise I.3 (Dual representation)**.** With the notations from Ex. I.1, in the case we use a feature map $\phi\colon \mathbb{R}^d \to \mathbb{R}^M$, so that features $\Phi = \{\phi_i(x^{(j)})\}_{(i,j)\in[M]\times[N]} \in \mathbb{R}^{M\times N}$, we know that $w \in \operatorname{span}\Phi \subseteq \mathbb{R}^M$.

Let $\alpha \in \mathbb{R}^N$ be such that $w = \sum_{j\in[N]} \alpha_j\phi(x^{(j)}) =: \sum_{j\in[N]} \alpha_j\phi^{(j)}$.

For two points $(x, y) \in \mathbb{R}^d \times \mathbb{R}^d$, define the kernel $\mathcal{K}$ as

$$\mathcal{K}(x, y) := \langle\phi(x), \phi(y)\rangle = \phi(x)^\top\phi(y).$$

1. What becomes with the prediction $y(x) = \langle\phi(x), w\rangle$?
2. Define the Gram matrix K as $K_{k\ell} := \mathcal{K}(x^{(k)}, x^{(\ell)})$. What are the prediction on the complete training set Φ?

3. Using the squared-norm error $E(\alpha) = \frac{1}{2}\|y - K\alpha\|_2^2$, and assuming that K is invertible, show that
$$\alpha^* \in \operatorname*{argmin}_\alpha E(\alpha) \iff \alpha^* = (K^\top K)^{-1} K^\top y.$$
Compare with the primal problem $w^* \in \operatorname{argmin}_w \frac{1}{2}\|y - \Phi^\top w\|_2^2 \iff w^* = (\Phi\Phi^\top)^{-1}\Phi y$.

4. Compute $\mathcal{K}$ when, for all $i \in [M]$,
   a) $x \in \mathbb{R}$, $\phi_i(x) = (x)^{i-1}$
   b) $x \in \mathbb{R}^d$, $\phi_i(x) = \exp\left(-\frac{\|x-\mu^{(i)}\|^2}{2\sigma^2}\right)$
   c) $x \in \mathbb{R}$, $\phi_i(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right)\frac{\left(\frac{x}{\sigma}\right)^i}{\sqrt{i!}}$, $i \to \infty$.

Answer I.3. 1. The prediction becomes $y(x) = \sum_{j\in[N]} \alpha_j \langle \phi^{(j)}, \phi(x)\rangle = \sum_{j\in[N]} \alpha_j K(x^{(j)}, x)$.

2. On the whole dataset, the prediction becomes
$$\begin{aligned} y = \Phi^\top w &= \Phi^\top \sum_{j\in[N]} \alpha_j \phi^{(j)} \\ &= \sum_{j\in[N]} \alpha_j \Phi^\top \phi^{(j)} \\ &= \sum_{j\in[N]} \alpha_j \left(\langle \phi^{(k)}, \phi^{(j)}\rangle\right)_{k\in[N]} \\ &= \sum_{j\in[N]} \alpha_j K_{:,j} \\ &= K\alpha \end{aligned}$$

3. First order optimality condition for a convex loss $\alpha^* \in \operatorname{argmin}_\alpha E(\alpha) \iff \nabla E(\alpha^*) = 0 \iff K^\top(y - K\alpha^*) = 0 \iff \alpha^* = (K^\top K)^{-1} K^\top y$, assuming K and hence $K^\top K = K^2$ is invertible.

4. a) With $\phi_i = (x)^{i-1}$, one computes for all $(x, y) \in \mathbb{R} \times \mathbb{R}$,
$$\begin{aligned} \mathcal{K}(x,y) = \langle \phi(x), \phi(y)\rangle = \sum_{i\in[M]} \phi_i(x)\phi_i(y) = \sum_{i\in[M]} (x)^{i-1}(y)^{i-1} = \sum_{i=0}^{M-1} (xy)^i \\ = \begin{cases} \frac{1-(xy)^M}{1-xy} & \text{if } xy \neq 1, \\ M & \text{if } xy = 1. \end{cases}, \end{aligned}$$
since this is the sum of the terms of a geometric sequence with ratio xy. There is no hyperparameter in this case (except for M), and one can directly compute the kernel without summing over all dimensions of the features.

b) With $\phi_i(x) = \exp\left(-\frac{\|x-\mu^{(i)}\|^2}{2\sigma^2}\right)$, one computes for all $(x, y) \in \mathbb{R}^d \times \mathbb{R}^d$,

$$\begin{aligned}
\mathcal{K}(x,y) = \langle \phi(x), \phi(y) \rangle &= \sum_{i\in[M]} \phi_i(x)\phi_i(y) \\
&= \sum_{i\in[M]} \exp\left(-\frac{\|x-\mu^{(i)}\|^2}{2\sigma^2}\right) \exp\left(-\frac{\|y-\mu^{(i)}\|^2}{2\sigma^2}\right) \\
&= \sum_{i\in[M]} \exp\left(-\frac{\|x\|^2 + \|y\|^2 - 2\langle \mu^{(i)}, x+y\rangle + 2\|\mu^{(i)}\|^2}{2\sigma^2}\right) \\
&= \exp\left(-\frac{\|x\|^2+\|y\|^2}{2\sigma^2}\right) \sum_{i\in[M]} \exp\left(\frac{\langle \mu^{(i)}, x+y\rangle - \|\mu^{(i)}\|^2}{\sigma^2}\right).
\end{aligned}$$

Here, we have M different locations $\{\mu^{(i)}\}_{i\in[M]}$ and the variance σ^2 as hyperparameters.

c) With $\phi_i(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right) \frac{\left(\frac{x}{\sigma}\right)^i}{\sqrt{i!}}$, it makes sense to let $i \in \mathbb{N}$. In this case, $\phi(x) \in \ell^2$ (space of sequences such that the series $\sum_{i=0}^{\infty} |\phi_i|^2 < \infty$ converges), associated with the inner product $\langle u, v\rangle = \sum_{i\in\mathbb{N}} u_i v_i$ for $(u, v) \in \ell^2 \times \ell^2$.

We can therefore compute, for all $(x, y) \in \mathbb{R} \times \mathbb{R}$,

$$\begin{aligned}
\mathcal{K}(x,y) = \langle \phi(x), \phi(y)\rangle = \sum_{i=0}^{\infty} \phi_i(x)\phi_i(y) &= \sum_{i=0}^{\infty} \left[\exp\left(-\frac{x^2}{2\sigma^2}\right) \frac{\left(\frac{x}{\sigma}\right)^i}{\sqrt{i!}} \exp\left(-\frac{y^2}{2\sigma^2}\right) \frac{\left(\frac{y}{\sigma}\right)^i}{\sqrt{i!}}\right] \\
&= \exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right) \sum_{i=0}^{\infty} \frac{\left(\frac{xy}{\sigma^2}\right)^i}{i!} \\
&= \exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right) \exp\left(\frac{xy}{\sigma^2}\right) \\
&= \exp\left(-\frac{(x^2+y^2)}{2\sigma^2} + \frac{2xy}{2\sigma^2}\right) = \exp\left(-\frac{(x-y)^2}{2\sigma^2}\right).
\end{aligned}$$

This means that even if the feature space is infinitely dimensional, the kernel $\mathcal{K}$ is easily computed. This is the Gaussian kernel and plays an important role in kernel learning.

Note that this extends to the case where $(x, y) \in \mathbb{R}^d$ as $\mathcal{K}(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$. The variance σ^2 is the only hyperparameter (one could also have a covariance matrix Σ and get $\mathcal{K}(x, y) = \exp\left(-\frac{1}{2}(x-y)^\top \Sigma^{-1}(x-y)\right)$ in more generality).

I.2. Kernel SVMs

Exercise I.4. Support Vector Machines can leverage kernels in order to deal with non-linearly separable data. Let $\mathcal{D} = \{(x^{(i)}, t^{(i)})\}_{i\in[n]}$ a dataset with, for $i \in [n]$, $(x^{(i)}, t^{(i)}) \in \mathbb{R}^d \times \{-1, 1\}$. Let $\phi\colon \mathbb{R}^d \to \mathbb{R}^m$ be a feature map, and $(w, b) \in \mathbb{R}^m \times \mathbb{R}$ be the weights and biases parametrizing the hyperplane in the feature space. The model is therefore $y(x) = \langle w, \phi(x)\rangle + b$.

1. Show that the distance of a point to the decision boundary is $|y(x)|/\|w\|$.
2. Explain why we can choose to make the model equal to ± 1 for the closest points to the decision boundary and conclude that the margin is $\frac{2}{\|w\|}$.
3. Write the constrained optimization problem of minimizing $\frac{1}{2}\|w\|^2$ while classifying the points correctly.
4. Write the Lagrangian $L(w, b, \alpha)$ of the problem, where $\alpha \in \mathbb{R}^n$ are the Lagrange multipliers.
5. Show that the associated dual problem is

$$\text{maximize} \quad g(\alpha) = \sum_{j\in[n]} \alpha_j - \frac{1}{2}\sum_{i\in[n]}\sum_{j\in[n]} t_i t_j \alpha_i \alpha_j \langle \phi(x^{(i)}), \phi(x^{(j)})\rangle \tag{1a}$$

$$\text{subject to} \quad \sum_{j\in[n]} \alpha_j t_j = 0, \tag{1b}$$

$$\forall j \in [n], \alpha_j \geqslant 0. \tag{1c}$$

6. Is the solution found by solving the dual problem also solving the primal problem? State the condition that allows the conclude.
7. How could one go and solve the dual problem? (*Hint:* what is the nice property about g?) Why would a simple gradient ascent *not* work?

Answer I.4. 1. To show that, we write any point $z \in \mathbb{R}^m$ as the sum $z = z_\perp + \lambda\frac{w}{\|w\|}$, where $z_\perp$ is the orthogonal projection of z onto the hyperplane of equation $\langle w, z'\rangle + b = 0$, and $\lambda \in \mathbb{R}$ is the (signed) distance of z to the hyperplane.

Therefore, $\langle w, z_\perp\rangle + b = 0 \implies \langle w, z_\perp\rangle = -b$.

Evaluating $\langle w, z\rangle$ gives

$$\langle w, z\rangle = \langle w, z_\perp\rangle + \langle w, \lambda\frac{w}{\|w\|}\rangle = -b + \lambda\frac{\|w\|^2}{\|w\|} = -b + \lambda\|w\|$$

$$\implies \qquad \lambda = \frac{\langle w, z\rangle + b}{\|w\|}.$$

This is also true for $z = \phi(x)$, and therefore, the absolute distance for a given point is

$$|\lambda| = \frac{|\langle w, \phi(x)\rangle + b|}{\|w\|} = \frac{|y(x)|}{\|w\|}.$$

2. We notice that $\frac{|y(x)|}{\|w\|}$ is left invariant by any scaling $w \leftarrow \kappa w, b \leftarrow \kappa b$. Therefore, we can choose the scaling that makes the points closest to the hyperplane such that $|y(x)| = 1$ and decide that $y(x) = \pm 1$ for the closest points on each side of the hyperplane.

   The size of the margin is therefore $2 \cdot \frac{1}{\|w\|} = \frac{2}{\|w\|}$.

3. The margin is $\frac{2}{\|w\|}$, which we want to maximize. This is equivalent to minimizing $\frac{1}{2}\|w\|^2$.

   For each data point x_i, the constraint can be written $t_i y(x_i) \geqslant 1$, since $t_i y(x_i) = 1$ for the closest points to the boundary. Letting $y_i := y(x_i) = \langle w, \phi(x_i)\rangle + b$ for all $i \in [n]$, the overall constrained problem is

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 \tag{2a}$$

$$\text{subject to} \quad \forall i \in [n],\ t_i y_i - 1 \geqslant 0. \tag{2b}$$

4. The Lagrangian is derived by considering the constraints as penalties. If a constraint $t_i y_i - 1 \geqslant 0$ is not satisfied (i.e. $t_i y_i - 1 < 0$), we want to count it as a **positive penalty** for our *minimization* problem (i.e. add a term that increases our objective $\frac{1}{2}\|w\|^2$). This is why the constraints can also be written $1 - t_i y_i \leqslant 0$, so that when they are not satisfied, a positive term $1 - t_i y_i > 0$ appears.

   The Lagrangian multiplies (one per constraint) are always positive by convention. We denote them $\alpha \in \mathbb{R}^n$. The overall penalty term is therefore $\sum_{i\in[n]} \alpha_i(1 - t_i y_i)$ or $-\sum_{i\in[n]} \alpha_i(t_i y_i - 1)$, and the Lagrangian is

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 + \sum_{i\in[n]} \alpha_i(1 - t_i y_i) = \frac{1}{2}\|w\|^2 - \sum_{i\in[n]} \alpha_i(t_i y_i - 1). \tag{3}$$

5. For a given $\alpha \in (\mathbb{R}^+)^n$, the optimization problem is therefore

$$\min_{w,b} L(w, b, \alpha)$$

   This gives us a *lower bound* on the original problem. Denote $g(\alpha) := \min_{w,b} L(w, b, \alpha)$. This is the **Lagrangian dual function**.

   It can be analytically computed when the objective function and constraints are convex by solving $\nabla_{w,b} L(w^*, b^*, \alpha) = 0$.

$$\nabla_w L(w,b,\alpha) = w - \sum_{i\in[n]} \alpha_i t_i \nabla_w y_i = w - \sum_{i\in[n]} \alpha_i t_i \phi_i,$$
$$\nabla_b L(w,b,\alpha) = -\sum_{i\in[n]} \alpha_i t_i,$$
$$\nabla_w L(w^*,b,\alpha) = 0 \implies \quad w^* = \sum_{i\in[n]} \alpha_i t_i \phi_i,$$
$$\nabla_b L(w,b^*,\alpha) = 0 \implies \quad \sum_{i\in[n]} \alpha_i t_i = 0$$

Now, $g(\alpha) = L(w^*, b^*, \alpha)$ can be computed. Injecting the expression $w^* = \sum_{i\in[n]} \alpha_i t_i \phi_i$ into (3), and using the fact that $b \sum_{i\in[n]} \alpha_i t_i = 0$, one gets

$$\begin{aligned}
g(\alpha) &= \frac{1}{2}\langle w^*, w^*\rangle - \sum_{i\in[n]} \alpha_i (t_i(\langle w^*, \phi_i\rangle + b^*) - 1) \\
&= \frac{1}{2} \sum_{i\in[n]} \sum_{j\in[n]} \alpha_i \alpha_j t_i t_j \langle \phi_i, \phi_j\rangle - \sum_{i\in[n]} \alpha_i t_i \left\langle \sum_{j\in[n]} \alpha_j t_j \phi_j, \phi_i \right\rangle + \sum_{i\in[n]} \alpha_i \\
&= \sum_{i\in[n]} \alpha_i + \frac{1}{2} \sum_{i\in[n]} \sum_{j\in[n]} \alpha_i \alpha_j t_i t_j \langle \phi_i, \phi_j\rangle - \sum_{i\in[n]} \sum_{j\in[n]} \alpha_i t_i \alpha_j t_j \langle \phi_j, \phi_i\rangle \\
&= \sum_{i\in[n]} \alpha_i - \frac{1}{2} \sum_{i\in[n]} \sum_{j\in[n]} \alpha_i \alpha_j t_i t_j \langle \phi_i, \phi_j\rangle.
\end{aligned}$$

This Lagrangian dual function is a lower-bound on the original problem. We aim to find the maximal lower-bound possible (which will be equal to the original solution under some conditions). We still have the constraints $\alpha \in (\mathbb{R}^+)^n$ from the Lagrangian multipliers, and the additional constraint $\sum_{\alpha_i t_i} = 0$ from the condition $\nabla_b L(w, b^*, \alpha) = 0$. Therefore, the dual problem is (1).

6. With the weak Slater's condition, it is enough for the duality gap to be zero to have a convex primal objective and affine constraints. This is the case for our problem, and therefore the duality gap is 0 (i.e. the maximum lower-bound found by solving the dual problem (1)) will be equal to the original solution.

7. Since g is convex in α (as the point-wise minimum of affine functions in α), one idea is to find a maximizer of g by gradient *ascent*. **But** the constraints have to be taken care of, making gradient ascent not viable.

Part II: Programming

Exercise II.1 (SMO algorithm)**.** The goal of this exercise is to implement and solve the SVM problem from Ex. I.4. We will do it by solving the dual problem (1).

In order to solve (1), one possible solution is to look at g as depending on only two variables, α_k and α_ℓ, and see how one can improve the objective g while satisfying the constraints. This method is called Sequential Minimal Optimization.

Recall the constraints from (1):

$$\sum_{j\in[n]} \alpha_j t_j = 0, \tag{1a}$$

$$\forall j \in [n], \alpha_j \geqslant 0 \tag{1b}$$

We only update two coordinates, k and ℓ. From the condition (1a), one sees that given two indices k and ℓ, and denoting by α_k^{old} and α_ℓ^{old} the coefficients before the update, one has

$$\alpha_k t_k + \alpha_\ell t_\ell = \alpha_k^{\text{old}} t_k + \alpha_\ell^{\text{old}} t_\ell$$

From this, we can write

$$\begin{aligned}\alpha_k &= \alpha_k^{\text{old}} + \alpha_\ell^{\text{old}} t_\ell t_k - \alpha_\ell t_\ell t_k \\ &= \gamma - s\alpha_\ell,\end{aligned}$$

with $s := t_k t_\ell$ and $\gamma := \alpha_k^{\text{old}} + s\alpha_\ell^{\text{old}}$.

The second constraint (1b) allows to write

$$\begin{aligned}\alpha_k \geqslant 0 &\implies \gamma - s\alpha_\ell \geqslant 0 \\ &\implies \begin{cases} \alpha_\ell \leqslant \alpha_k^{\text{old}} + \alpha_\ell^{\text{old}} & \text{if } s = 1, \\ \alpha_\ell \geqslant \max(0, \alpha_\ell^{\text{old}} - \alpha_k^{\text{old}}) & \text{if } s = -1. \end{cases}\end{aligned}$$

Therefore, let

$$L = \begin{cases} 0 & \text{if } s = 1, \\ \max(0, \alpha_\ell^{\text{old}} - \alpha_k^{\text{old}}) & \text{if } s = -1. \end{cases}, \text{and } H = \begin{cases} \alpha_k^{\text{old}} + \alpha_\ell^{\text{old}} & \text{if } s = 1, \\ \infty & \text{if } s = -1. \end{cases} \tag{2}$$

be the lower and upper bounds for α_ℓ

Consider the function g in (1) as a function of α_k and α_ℓ only. With $\phi_i := \phi(x^{(i)})$, we have

$$g(\alpha_k, \alpha_\ell) = \alpha_k + \alpha_\ell - \frac{1}{2}\alpha_k^2\langle\phi_k, \phi_k\rangle - \frac{1}{2}\alpha_\ell^2\langle\phi_\ell, \phi_\ell\rangle - \alpha_k\alpha_\ell t_k t_\ell\langle\phi_k, \phi_\ell\rangle - \alpha_k t_k v_k - \alpha_\ell t_\ell v_\ell,$$

with $v_i := \sum_{j\in[n]\setminus\{k,\ell\}} \alpha_j t_j \langle\phi_i, \phi_j\rangle$.

Plugging-in $\alpha_k = \gamma - s\alpha_\ell$ gives

$$g(\alpha_\ell) = (\gamma - s\alpha_\ell) + \alpha_\ell - \frac{1}{2}(\gamma - s\alpha_\ell)^2\langle\phi_k, \phi_k\rangle - \frac{1}{2}\alpha_\ell^2\langle\phi_\ell, \phi_\ell\rangle - (\gamma - s\alpha_\ell)\alpha_\ell t_k t_\ell\langle\phi_k, \phi_\ell\rangle - (\gamma - s\alpha_\ell)t_k v_k - \alpha_\ell t_\ell v_\ell,$$

and solving $\frac{\partial}{\partial\alpha_\ell} g(\alpha_\ell^*) = 0$ gives

$$\alpha_\ell^* = \alpha_\ell^{\text{old}} + \frac{t_l(E_k - E_\ell)}{\langle\phi_k, \phi_k\rangle + \langle\phi_\ell, \phi_\ell\rangle - 2\langle\phi_k, \phi_l\rangle},$$

where $E_i := f(x^{(i)}) - t_i$.

Therefore, α_ℓ^* is the value for α_ℓ that maximizes $g(\alpha_k, \alpha_\ell)$, subject to the constraint (1a). It could be that α_ℓ^* does not satisfy the bounds (2); that's why we clip it and obtain

$$\alpha_\ell^{\text{new}} = \begin{cases} L & \text{if } \alpha_\ell^* < L, \\ \alpha_\ell^* & \text{if } \alpha_\ell^* \in [L, H], \\ H & \text{if } \alpha_\ell^* > H. \end{cases} \tag{3}$$

The pseudo-code of the SMO algorithm can be summarized as

initialization $\alpha = \mathbf{0}$
while $g(\alpha)$ not converged **do**
    choose a training sample x_k that violates the KKT conditions
    choose a second training sample x_ℓ so that $|E_k - E_\ell|$ is maximized (heuristic)
    $\alpha_\ell \leftarrow \alpha_\ell^{\text{new}}$ from (3)
    $\alpha_k \leftarrow \alpha_k^{\text{old}} + t_k t_\ell(\alpha_\ell^{\text{old}} - \alpha_\ell^{\text{new}})$
output α

1. Implement the SMO algorithm.

2. Test it on some data.