Introduction to Kernel methods

Matthieu R. Bloch

1 The kernel trick

Consider the maximum margin hyperplane with non linear transform $\Phi : \mathbb{R}^d \to \mathbb{R}^p$:

$$\underset{\mathbf{w},b}{\text{argmin}} \frac{1}{2}\|\mathbf{w}\|_2^2 \text{ s.t. } \forall i \quad y_i(\mathbf{w}^\intercal \Phi(\mathbf{x}_i) + b) \geqslant 1 \tag{1}$$

We will show later that the optimal $\mathbf{w}$ is a linear combination of the data points $\mathbf{w} = \sum_{i=1}^N \alpha_i \Phi(\mathbf{x}_i)$, so that

$$\|\mathbf{w}\|_2^2 = \mathbf{w}^\intercal \mathbf{w} = \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j \Phi(\mathbf{x}_i)^\intercal \Phi(\mathbf{x}_j) = \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\rangle \tag{2}$$

and

$$\mathbf{w}^\intercal \Phi(\mathbf{x}_j) = \sum_{i=1}^N \alpha_i \Phi(\mathbf{x}_i)^\intercal \Phi(\mathbf{x}_j) = \sum_{i=1}^N \alpha_i \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\rangle. \tag{3}$$

Note that the only quantities that really matter in this optimization problem are the inner products $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\rangle$. Irrespective of the codomain of Φ, there are only N^2 inner products. Perhaps surprisingly, the dimension of $\Phi(\mathbf{x})$ is hidden in the inner products and does not explicitly appear, all other operations are in the original feature space $\mathbb{R}^d$. The nonlinear features may not even be computed explicitly in $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\rangle$.

The "kernel trick" consists in exploiting these observations to replace the inner products of transformed feature vectors $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\rangle$ by a *kernel* $k(\mathbf{x}_i, \mathbf{x}_j)$, without ever having to specify Φ. The main challenge in kernelizing is to understand what is needed to define a valid kernel. Based on our previous discussion, the kernel should define inner products $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\rangle$, defined in a Hilbert space $\mathcal{H}$ such that $\Phi : \mathbb{R}^d \to \mathcal{H}$.

Definition 1.1 (Inner product kernel). *An inner product kernel is a mapping $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ for which there exists a Hilbert space $\mathcal{H}$ and a mapping $\Phi : \mathbb{R}^d \to \mathcal{H}$ such that*

$$\forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^d \quad k(\mathbf{u}, \mathbf{v}) = \langle \Phi(\mathbf{u}), \Phi(\mathbf{v})\rangle_{\mathcal{H}}$$

Example 1.2 (Quadratic kernel). *Quadratic kernel $k(\mathbf{u}, \mathbf{v}) = (\mathbf{u}^\intercal \mathbf{v})^2$*

What makes kernel useful is an alternative and much more tangible characterization, which we now establish.

Definition 1.3 (Positive semidefinite kernel). *A function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a positive semidefinite kernel if*

1. *k is symmetric, i.e., $k(\mathbf{u}, \mathbf{v}) = k(\mathbf{v}, \mathbf{u})$*

2. *for all* $\{\mathbf{x}_i\}_{i=1}^N$*, the Gram matrix* $\mathbf{K}$ *is positive semidefinite, i.e.,*

$$\mathbf{x}^\intercal \mathbf{K}\mathbf{x} \geqslant 0 \text{ with } \mathbf{K} = [K_{i,j}] \text{ and } K_{i,j} \triangleq k(\mathbf{x}_i, \mathbf{x}_j)$$

Positive semidefinite kernels are quite common (you might have encountered them in other contexts), and turn out to be all that we need to characterize inner product kernels.

Theorem 1.4. *A function* $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ *is an inner product kernel if and only if* k *is a positive semidefinite kernel.*

Proof. Comning soon ∎

Useful examples of kernels include:

- Homogeneous polynomial kernel: $k(\mathbf{u}, \mathbf{v}) = (\mathbf{u}^\intercal \mathbf{v})^m$ with $m \in \mathbb{N}^*$
- Inhomogenous polynomial kernel: $k(\mathbf{u}, \mathbf{v}) = (\mathbf{u}^\intercal \mathbf{v} + c)^m$ with $c > 0$, $m \in \mathbb{N}^*$
- Radial basis function (RBF) kernel: $k(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{\|\mathbf{u}-\mathbf{v}\|^2}{2\sigma^2}\right)$ with $\sigma^2 > 0$

The main lingering question is how to effectively kernelize the maximum margin optimization problem. This requires us to review some notions of optimization, and in particular the notion of *duality*.

2 Introduction to Lagrangian duality

We will consider the following canonical form of constrained optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) \text{ such that } \begin{cases} g_i(\mathbf{x}) \leqslant 0 & \forall i \in [\![1, m]\!] \\ h_j(\mathbf{x}) = 0 & \forall j \in [\![1, p]\!] \end{cases} . \tag{4}$$

Unlike unconstrained optimization problems that you may be familiar with, it is not enough to set the derivative of f equal to zero (assuming it exists) to find an optimizer, even if the function is convex. This happens because the constraints that we add restrict the domain over which we are looking for a minimizer. We will repeatedly use the following terminology.

- f is called the objective function;
- $g_i(\mathbf{x})$ is called an inequality constraint;
- $h_j(\mathbf{x})$ is called an equality constraint;
- if $\mathbf{x}$ satisfies all the constraints, we say that it is *feasible*.

Remark 2.1. *There is no loss of generality in considering problems as in* (4)*. In fact, any inequality and equality can be put in the form given in* (4)*, and maximization problems can be turned into minimization problems by considering* $-f$ *in place of* f*.*

Definition 2.2. *A constrained optimization problem is *convex* if* f *is convex, the* g_i*'s are convex, and the* h_j*'s are affine*

Rather than dealing with the constraints separately from the objective function, it is convenient to group everything together; specifically, we turn the constrained optimization into an unconstrained one using the *Lagrangian*

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \triangleq f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) + \sum_{j=1}^{p} \mu_j h_j(\mathbf{x}) \text{ with } \boldsymbol{\lambda} \geqslant \mathbf{0}, \tag{5}$$

where we have defined the *dual variables* (also called *Lagrange multipliers*) $\boldsymbol{\lambda} \triangleq [\lambda_1, \cdots, \lambda_m]^\intercal$ and $\boldsymbol{\mu} \triangleq [\mu_1, \cdots, \mu_p]^\intercal$.

2.1 Primal and dual problems

Based on the Lagrangian in (5), one can define two problems of interest.

Definition 2.3. *The Lagrange dual function is*

$$\mathcal{L}_D(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}). \tag{6}$$

The dual optimization problem is

$$\max_{\boldsymbol{\lambda} \geqslant \mathbf{0}, \boldsymbol{\mu}} \mathcal{L}_D(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \max_{\boldsymbol{\lambda} \geqslant \mathbf{0}, \boldsymbol{\mu}} \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}). \tag{7}$$

Definition 2.4. *The primal function is*

$$\mathcal{L}_P(\mathbf{x}) \triangleq \max_{\boldsymbol{\lambda} \geqslant \mathbf{0}, \boldsymbol{\mu}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \tag{8}$$

The primal optimization problem is

$$\min_{\mathbf{x}} \mathcal{L}_P(\mathbf{x}) = \min_{\mathbf{x}} \max_{\boldsymbol{\lambda} \geqslant \mathbf{0}, \boldsymbol{\mu}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \tag{9}$$

We will see shortly how these two problems relate to our original problem in (4), but we can already make a few observations.

Proposition 2.5. *The dual function $\mathcal{L}_D(\boldsymbol{\lambda}, \boldsymbol{\mu})$ is concave in $\boldsymbol{\lambda}, \boldsymbol{\mu}$.*

Proof. For a fixed $\mathbf{x}$, it follows from the definition in (5) that $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ is affine in $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, and therefore concave. The dual function is then the pointwise minimum of concave functions, and therefore concave as well (check it!). ∎

Proposition 2.5 should be intriguing because we have *not* said anything about f. In particular, f need not be convex or well-behaved in anyway. Consequently, the dual optimization problem is a concave maximization that is always well-behaved.

Proposition 2.6. *Denote by $\mathcal{F}$ the set of feasible values of $\mathbf{x}$, i.e.,*

$$\mathcal{F} \triangleq \{\mathbf{x} \in \mathbb{R}^d : \forall i \in [\![1, m]\!]\, g_i(\mathbf{x}) \leqslant 0 \text{ and } \forall j \in [\![1, p]\!]\, h_j(\mathbf{x}) = 0\}. \tag{10}$$

The minimum of the primal problem is the minimum of the original problem, i.e.,

$$\min_{\mathbf{x}} \mathcal{L}_P(\mathbf{x}) = \min_{\mathbf{x} \in \mathcal{F}} f(\mathbf{x}). \tag{11}$$

Proof. We first note that if a point $\mathbf{x}$ is *not* feasible then there exists $i^* \in [\![1, m]\!]$ such that $g_{i^*}(\mathbf{x}) > 0$ or $j^* \in [\![1, p]\!]$ such that $h_{j^*} \neq 0$. In the former case, note that for $\lambda_{i^*} \geqslant 0$, the quantity $\lambda_{i^*} g_{i^*}(\mathbf{x}) \geqslant 0$ can be made arbitrarily large. In the latter case, we can always find μ_{j^*} such that $\operatorname{sgn}(\mu_{j^*}) = \operatorname{sgn}(h_{j^*}(\mathbf{x}))$ and $\mu_{j^*} h_{j^*}(\mathbf{x}) \geqslant 0$ can be made arbitrarily large. In either case, we obtain

$$\mathcal{L}_P(\mathbf{x}) = +\infty \quad \text{when } \mathbf{x} \text{ is not feasible.} \tag{12}$$

Next, note that

$$\min_{x \in \mathcal{F}} f(x) \overset{(a)}{=} \min_{x \in \mathcal{F}} \mathcal{L}(\mathbf{x}, \mathbf{0}, \mathbf{0}) \overset{(b)}{\leqslant} \min_{x \in \mathcal{F}} \max_{\boldsymbol{\lambda} \geqslant \mathbf{0}, \boldsymbol{\mu}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \overset{(c)}{=} \min_{x \in \mathcal{F}} \mathcal{L}_P(\mathbf{x}) \overset{(d)}{=} \min_{x} \mathcal{L}_P(\mathbf{x}). \tag{13}$$

Note that (a) follows by definition of the Lagrangian; (b) follows because we are maximizing over $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$; (c) follows by definition of the primal function; (d) follows by (12).

In addition, if we let $\mathbf{x}^* \triangleq \operatorname{argmin}_{\mathbf{x} \in \mathcal{F}} f(\mathbf{x})$, note that $\mathbf{x}^*$ is feasible so that $g_i(\mathbf{x}^*) \leqslant 0$ and $h_j(\mathbf{x}^*) = 0$. Consequently, for $\boldsymbol{\lambda} \geqslant \mathbf{0}$, we have

$$\mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(x^*) + \sum_{i=1}^{m} \lambda_i \underbrace{g_i(\mathbf{x}^*)}_{\leqslant 0} + \sum_{i=1}^{p} \mu_i \underbrace{h_j(\mathbf{x}^*)}_{=0} \leqslant f(\mathbf{x}^*), \tag{14}$$

with equality if $\boldsymbol{\lambda} = \mathbf{0}$, and for any $\boldsymbol{\mu}$, which we can choose equal to $\mathbf{0}$. Therefore,

$$\min_{x} \mathcal{L}_P(\mathbf{x}) \leqslant \mathcal{L}_P(\mathbf{x}^*) = \mathcal{L}(\mathbf{x}^*, \mathbf{0}, \mathbf{0}) = f(\mathbf{x}^*), \tag{15}$$

so that the inequality in (b) of (13) is an equality. ∎

Proposition 2.6 shows why the primal function is called *primal*. It is essentially equivalent to the original problem but is formulated as an unconstrained optimization problem.

2.2 Weak duality

One of the key results that justifies the introduction of the primal and dual problems is the so called *weak duality*.

Theorem 2.7 (Weak duality)**.**

$$d^* \triangleq \max_{\boldsymbol{\lambda} \geqslant \mathbf{0}, \boldsymbol{\mu}} \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \leqslant p^* \triangleq \min_{\mathbf{x}} \max_{\boldsymbol{\lambda} \geqslant \mathbf{0}, \boldsymbol{\mu}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}). \tag{16}$$

Proof. Let $\mathbf{x} \in \mathcal{F}$ be feasible, and let $\boldsymbol{\lambda} \geqslant \mathbf{0}$, $\boldsymbol{\mu}$ be fixed. By definition, $g_i(\mathbf{x}) \leqslant 0$ and $h_j(\mathbf{x}) = 0$ so that

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \sum_{i=1}^{m} \underbrace{\lambda_i}_{\geqslant 0} \underbrace{g_i(\mathbf{x})}_{\leqslant 0} + \sum_{j=1}^{P} \mu_j \underbrace{h_j(\mathbf{x})}_{=0} \leqslant f(\mathbf{x}). \tag{17}$$

Consequently,

$$\mathcal{L}_D(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \overset{(a)}{\leqslant} \min_{\mathbf{x} \in \mathcal{F}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \overset{(b)}{\leqslant} \min_{\mathbf{x} \in \mathcal{F}} f(\mathbf{x}) \overset{(c)}{=} \min_{\mathbf{x}} \mathcal{L}_P(\mathbf{x}), \tag{18}$$

where (a) follows because restricting the set of points can only increase the minimum; (b) follows because of (17); and (c) follows by from the equality of the primal solution with the original problem solution. Since $\boldsymbol{\lambda} \geqslant \mathbf{0}$ and $\boldsymbol{\mu}$ are arbitrary, we obtain

$$\max_{\boldsymbol{\lambda} \geqslant \mathbf{0}, \boldsymbol{\mu}} \mathcal{L}_D(\boldsymbol{\lambda}, \boldsymbol{\mu}) \leqslant \min_{\mathbf{x}} \mathcal{L}_P(\mathbf{x}). \tag{19}$$

■

Weak duality tells us that we can solve the dual problem , which is always a concave maximization problem and obtain a lower bound for the primal. This result is useful by itself in many situations. There are situations where knowing a lower bound on the minimum (e.g., a minimum revenue) is perhaps all we care about. Perhaps more importantly, the dual problem allows us to check whether a proposed primal solution is valid or not. In general, without additional assumptions, there is no reason to have $d^* = p^*$ and the gap $p^* - d^* \geqslant 0$ is called the *duality gap*.

The situation in which $p^* - d^* = 0$ is called *strong duality* and makes the dual formulation particularly interesting. There are too many interesting applications of strong duality to list, and I will just mention a few.

- *Certificates.* The optimizers of the dual problem $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ can serve as a certificate to check if a proposed minimizer $\mathbf{x}^*$ is indeed correct. In fact, if strong duality holds, we can merely check that $\mathcal{L}_D(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \mathcal{L}_P(\mathbf{x}^*)$ and be guaranteed that we have the optimal solution. Note that this does *not* require us to know how the solutions were obtained. Another application of certificates is to decide when to stop an iterative algorithm when the duality gap is zero. The smaller the duality gap at a given iteration, the closer we are to the optimal solution.
- *Primal-dual methods.* There exist many algorithms that iteratively solve primal and dual problems to converge to the optimal solution. When running such an iterative algorithm, we can compute the duality gap at each iteration for the specific values of $\mathbf{x}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ and measure how much progress the algorithm is making.

2.3 Karush-Kuhn Tucker conditions

The Karush-Kuhn Tucker (KKT) conditions can be though of the extension of the well known *stationary condition* ($\frac{df}{dx}(x) = 0$) used in non-constrained optimization to identify the extremum of a function.

Definition 2.8 (KKT conditions). *Consider a function $f : \mathbb{R}^d \to \mathbb{R}$, assumed to be differentiable in its domain. There are four KKT conditions.*

1. *Stationarity*

$$0 = \nabla f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}) + \sum_{j=1}^{p} \mu_j \nabla h_j(\mathbf{x}) \tag{20}$$

2. *Primal optimality*

$$\forall i \in [\![1, m]\!] \quad g_i(\mathbf{x}) \leqslant 0 \qquad \forall j \in [\![1, p]\!] \quad h_j(\mathbf{x}) = 0 \tag{21}$$

3. *Dual optimality*

$$\forall i \in [\![1, m]\!] \quad \lambda_i \geqslant 0 \tag{22}$$

4. Complementary slackness

$$\forall i \in [\![1, m]\!] \quad \lambda_i g_i(\mathbf{x}) = 0 \tag{23}$$

Proposition 2.9. *If* $\mathbf{x}^*$ *and* $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ *are primal and dual solutions with zero duality gap, then* $\mathbf{x}^*$ *and* $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ *satisfy the KKT conditions.*

Proof. Coming soon. ∎

Proposition 2.10. *If the original problem is convex and* $\tilde{\mathbf{x}}$ *and* $(\tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}})$ *satisfy the KKT conditions, then* $\tilde{\mathbf{x}}$ *is primal optimal,* $(\tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}})$ *is dual optimal, and the duality gap is zero.*

Proof. Coming soon. ∎

Consequently, if a constrained optimization problem is differentiable and convex, the KKT conditions are necessary and sufficient for primal/dual optimality (with zero duality gap); in addition, we can use the KKT conditions to find a solution to our optimization problem. Conveniently, the optimal soft-margin hyperplane problem falls in this category.

3 Kernelization of optimal soft-margin hyperplane classifier

The optimal soft-margin hyperplane is the solution of the following

$$\underset{\mathbf{w},b,\boldsymbol{\xi}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{C}{N}\sum_{i=1}^{N}\xi_i \quad \text{s.t.} \quad \forall i \in [\![1, N]\!] \quad y_i(\mathbf{w}^\intercal \mathbf{x}_i + b) \geqslant 1 - \xi_i \text{ and } \xi_i \geqslant 0 \tag{24}$$

This optimization problem is differentiable and convex, so that the KKT conditions are necessary and sufficient and the duality gap is zero. We will kernelize the (equivalent) dual problem.

The Lagrangian is

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \triangleq \frac{1}{2}\mathbf{w}^\intercal\mathbf{w} + \frac{C}{N}\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\lambda_i(1 - \xi_i - y_i(\mathbf{w}^\intercal\mathbf{x}_i + b)) - \sum_{i=1}^{N}\mu_i\xi_i \tag{25}$$

with $\boldsymbol{\lambda} \geqslant \mathbf{0}, \boldsymbol{\mu} \geqslant \mathbf{0}$. The Lagrange dual function is

$$\mathcal{L}_D(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \min_{\mathbf{w},b,\boldsymbol{\xi}} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \tag{26}$$

and the dual problem is

$$\max_{\boldsymbol{\lambda}\geqslant\mathbf{0},\boldsymbol{\mu}\geqslant\mathbf{0}} \mathcal{L}_D(\boldsymbol{\lambda}, \boldsymbol{\mu}). \tag{27}$$

This dual problem is not particularly more convenient to work with. The key insight is that we can use the KKT conditions to simplify $\mathcal{L}_D(\boldsymbol{\lambda}, \boldsymbol{\mu})$.

Lemma 3.1 (Simplification of dual function)**.** *The dual function is*

$$\mathcal{L}_D(\boldsymbol{\lambda}, \boldsymbol{\mu}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j y_i y_j \mathbf{x}_i^\intercal\mathbf{x}_j + \sum_{i=1}^{N}\lambda_i \tag{28}$$

Proof. Follows from KKT conditions. ∎

Lemma 3.2. *The dual optimization problem function is*

$$\max_{\boldsymbol{\lambda},\boldsymbol{\mu}} -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j y_i y_j \mathbf{x}_i^\intercal \mathbf{x}_j + \sum_{i=1}^{N}\lambda_i \; s.t. \begin{cases} \forall i \in [\![1,N]\!] & \sum_{i=1}^{N}\lambda_i y_i = 0 \\ \forall i \in [\![1,N]\!] & 0 \leqslant \lambda_i \leqslant \frac{C}{N} \end{cases} \tag{29}$$

Proof. Coming soon. ∎

We can very efficiently solve for $\boldsymbol{\lambda}^*$ using numerical algorithms. It remains to show how one can relate the solution $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ of the dual problem to the solution $(\mathbf{w}^*, b^*)$ of the primal problem.

Lemma 3.3.

$$\mathbf{w}^* = \sum_{i=1}^{N}\lambda_i^* y_i \mathbf{x}_i \quad \textit{and} \quad b^* = y_i - \mathbf{w}^{*\intercal}\mathbf{x}_i$$

for some $i \in [\![1,N]\!]$ *such that* $0 < \lambda_i^* < \frac{C}{N}$

Proof. Coming soon. ∎

Note that the only data points that matter are those for which $\lambda_i^* \neq 0$. By completementary slackness they are the ones for which $y_i(\mathbf{w}^{*\intercal}\mathbf{x}_i + b) = 1 - \xi_i^*$. These points are called *support vectors* and are located on or inside the margin. In practice, the number of support vectors is often $\ll N$, so that the optimal classifier can be described with a small number of parameters.

Now that we have described the optimal soft margin classifier in terms of the optimization problem in Lemma 3.2, all we have to do to kernelize is replace the inner products $\mathbf{x}_j^\intercal \mathbf{x}_i$ by kernel values $k(\mathbf{x}_j, \mathbf{x}_i)$. Given an inner product kernel $k(\cdot,\cdot)$, the *support vector machine* classifier is

$$h^{\text{SVM}}(\mathbf{x}) \triangleq \text{sgn}\left(\sum_{i\in[\![1,N]\!]}\lambda_i^* y_i k(\mathbf{x}_i, \mathbf{x}) + b^*\right) \tag{30}$$

where $\boldsymbol{\lambda}^*$ is the solution of

$$\max_{\boldsymbol{\lambda},\boldsymbol{\mu}} -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^{N}\lambda_i \text{ s .t. } \begin{cases} \forall i \in [\![1,N]\!] & \sum_{i=1}^{N}\lambda_i y_i = 0 \\ \forall i \in [\![1,N]\!] & 0 \leqslant \lambda_i \leqslant \frac{C}{N} \end{cases} \tag{31}$$

and

$$b^* = y_i - \sum_{j=1}^{N}\lambda_j^* y_j k(\mathbf{x}_i, \mathbf{x}_j) \tag{32}$$

for some $i \in [\![1,N]\!]$ such that $0 < \lambda_i^* < \frac{C}{N}$.

We will see other examples of kernelization later in the course.