The Karush-Kuhn-Tucker (KKT) conditions

For several lectures we have been alluding to the **Karush-Kuhn-Tucker (KKT) conditions**. We are finally in a position to provide an intuitive account of what these are and where they come from. Simply put, the KKT conditions are a set of sufficient (and at most times necessary) conditions for an $\boldsymbol{x}^\star$ to be the solution of a given convex optimization problem. The conditions involve the existence of Lagrange multipliers satisfying certain natural properties, and they play a fundamental role in both the theory and practice of convex optimization.

We will start here by considering a general convex program with **inequality** constraints only. Specifically, we will consider the convex program given by

$$\begin{aligned} &\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \;\; f(\boldsymbol{x}) \\ &\text{subject to} \;\; g_m(\boldsymbol{x}) \leq 0, \quad m = 1, \ldots, M. \end{aligned} \tag{1}$$

Note that our initial restriction to inequality constraints only is merely to make the exposition easier – after we have this established, we will show how to include equality constraints (which must always be affine in convex programming).

Our analysis of the KKT conditions will exploit concepts from Lagrange duality. Recall that the Lagrangian for (1) is given by

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_{m=1}^{M} \lambda_m g_m(\boldsymbol{x}),$$

the Lagrange dual function is given by

$$d(\boldsymbol{\lambda}) = \inf_{\boldsymbol{x}\in\mathbb{R}^N} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}),$$

and the associated dual problem is given by

$$\underset{\boldsymbol{\lambda} \in \mathbb{R}^M}{\text{maximize}} \; d(\boldsymbol{\lambda}) \quad \text{subject to} \quad \boldsymbol{\lambda} \geq \mathbf{0}. \tag{2}$$

Here and throughout this lecture, we assume that the functions $f(\boldsymbol{x}), g_1(\boldsymbol{x}), \ldots, g_M(\boldsymbol{x})$ are convex and differentiable. The KKT conditions for the optimization program in (1) are as follows.

KKT (inequality only)
The KKT conditions for $\boldsymbol{x} \in \mathbb{R}^N$ and $\boldsymbol{\lambda} \in \mathbb{R}^M$ are

$$\begin{aligned}
g_m(\boldsymbol{x}) &\leq 0, \quad m = 1, \ldots, M, && \text{(K1)} \\
\lambda_m &\geq 0, \quad m = 1, \ldots, M, && \text{(K2)} \\
\lambda_m g_m(\boldsymbol{x}) &= 0, \quad m = 1, \ldots, M, && \text{(K3)} \\
\nabla f(\boldsymbol{x}) + \sum_{m=1}^{M} \lambda_m \nabla g_m(\boldsymbol{x}) &= \mathbf{0}. && \text{(K4)}
\end{aligned}$$

Below we will argue that under certain conditions, a necessary and sufficient condition for $\boldsymbol{x}^\star$ to be a solution to (1) is for there to exist some $\boldsymbol{\lambda}^\star$ such that $\boldsymbol{x}^\star$ and $\boldsymbol{\lambda}^\star$ satisfy the KKT conditions. Moreover, under the same conditions, a necessary and sufficient condition for $\boldsymbol{\lambda}^\star$ to be a solution to (2) is for there to exist some $\boldsymbol{x}^\star$ such that $\boldsymbol{x}^\star$ and $\boldsymbol{\lambda}^\star$ satisfy the KKT conditions.

Sufficiency

We start by establishing that these are sufficient.

> Suppose that $f(\boldsymbol{x}), g_1(\boldsymbol{x}), \ldots, g_M(\boldsymbol{x})$ are convex and differentiable. If the KKT conditions hold for some $\boldsymbol{x}^\star \in \mathbb{R}^N$ and some $\boldsymbol{\lambda}^\star \in \mathbb{R}^M$, then $\boldsymbol{x}^\star$ is a solution to (1) and $\boldsymbol{\lambda}^\star$ is a solution to (2).

Suppose that $\boldsymbol{x}'$ and $\boldsymbol{\lambda}'$ satisfy the KKT conditions. We will not use the notation $\boldsymbol{x}^\star$ and $\boldsymbol{\lambda}^\star$ just yet to avoid any confusion, since we have not yet proven that they are solutions. We begin by noting that K4 is equivalent to the condition that

$$\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}', \boldsymbol{\lambda}') = \mathbf{0}.$$

Since f and $g_1, \ldots, g_M$ are convex, this implies that $\boldsymbol{x}'$ is a minimizer of $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}')$, i.e.,

$$\mathcal{L}(\boldsymbol{x}', \boldsymbol{\lambda}') \leq \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}')$$

for all $\boldsymbol{x} \in \mathbb{R}^N$. This implies that

$$d(\boldsymbol{\lambda}') = \mathcal{L}(\boldsymbol{x}', \boldsymbol{\lambda}'),$$

and thus

$$\begin{aligned} d(\boldsymbol{\lambda}') &= f(\boldsymbol{x}') + \sum_{m=1}^{M} \lambda_m' g_m(\boldsymbol{x}') \\ &= f(\boldsymbol{x}'), \end{aligned}$$

where the last equality follows from (K3).

Recall that for *any* feasible $\boldsymbol{x}$ and $\boldsymbol{\lambda}$ we always have $d(\boldsymbol{\lambda}) \leq f(\boldsymbol{x})$. Since (K1) and (K2) ensure that $\boldsymbol{x}'$ and $\boldsymbol{\lambda}'$ are both feasible, the

above result implies that $\boldsymbol{x}'$ is a solution to (1) and $\boldsymbol{\lambda}'$ is a solution to (2): for any feasible $\boldsymbol{x}$ we have $f(\boldsymbol{x}) \geq d(\boldsymbol{\lambda}') = f(\boldsymbol{x}')$ and for any $\lambda \geq \mathbf{0}$ we have $d(\boldsymbol{\lambda}) \leq f(\boldsymbol{x}') = d(\boldsymbol{\lambda}')$.

Note that along the way we have also shown that the existence of $\boldsymbol{x}, \boldsymbol{\lambda}$ satisfying the KKT conditions also implies strong duality.

Necessity

We have just shown that for any convex problem of the form (1), if we can find a $\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star$ satisfying the KKT conditions, then $\boldsymbol{x}^\star$ must be a solution to (1). However, this on its own does not ensure that for any solution $\boldsymbol{x}^\star$ we must necessarily be able to find a $\boldsymbol{\lambda}^\star$ such that $\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star$ obey the KKT conditions. Indeed, we do not necessarily even know if a $\boldsymbol{x}, \boldsymbol{\lambda}$ satisfying the KKT conditions exists.

Here we show that if we make the additional assumption that *strong duality* holds (for example, if our constraints satisfy Slater's condition), then the KKT conditions are also necessary.

> Suppose that $f(\boldsymbol{x}), g_1(\boldsymbol{x}), \ldots, g_M(\boldsymbol{x})$ are convex and differentiable. Let $\boldsymbol{x}^\star$ be a solution to (1) and $\boldsymbol{\lambda}^\star$ be a solution to (2). If strong duality holds (e.g., Slater's condition holds) then $\boldsymbol{x}^\star$ and $\boldsymbol{\lambda}^\star$ obey the KKT conditions.

We trivially have (K1) and (K2) simply because $\boldsymbol{x}^\star$ and $\boldsymbol{\lambda}^\star$ must be feasible to be solutions to (1) and (2) respectively. We very nearly proved the remaining conditions last time without explicitly saying so. In particular, we previously argued that if we have strong duality

then

$$\begin{aligned} f(\boldsymbol{x}^\star) &= d(\boldsymbol{\lambda}^\star) \\ &= \inf_{\boldsymbol{x}\in\mathbb{R}^N} \left(f(\boldsymbol{x}) + \sum_{m=1}^{M} \lambda_m^\star g_m(\boldsymbol{x}) \right) \\ &\leq f(\boldsymbol{x}^\star) + \sum_{m=1}^{M} \lambda_m^\star g_m(\boldsymbol{x}^\star) \\ &\leq f(\boldsymbol{x}^\star). \end{aligned} \tag{3}$$

where the last inequality follows from the facts that we must have $\lambda_m^\star \geq 0$ and $g_m(\boldsymbol{x}^\star) \leq 0$. Looking at this entire chain of inequalities, where the first and last term are both $f(\boldsymbol{x}^\star)$, means that

$$f(\boldsymbol{x}^\star) = \min_{\boldsymbol{x}\in\mathbb{R}^N} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}^\star) = \mathcal{L}(\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star).$$

Since $f(\boldsymbol{x}), g_1(\boldsymbol{x}), \ldots g_M(\boldsymbol{x})$ are convex and differentiable, so is $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}^\star)$ for any $\boldsymbol{\lambda}^\star \geq \mathbf{0}$. Thus, if $\boldsymbol{x}^\star$ is a minimizer of $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}^\star)$ then we must have $\nabla \mathcal{L}(\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star) = \mathbf{0}$, which is precisely (K4).

Moreover, the chain of (in)equalities concluding in (3) also implies that

$$\sum_{m=1}^{M} \lambda_m^\star g_m(\boldsymbol{x}^\star) = 0.$$

Since each $\lambda_m^\star g_m(\boldsymbol{x}^\star)$ is non-positive, the only way the sum can equal zero is if every term is zero, which yields (K3).

KKT conditions with equality constraints

Now suppose that we have an optimization problem involving equality constraints:

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \;\; & f(\boldsymbol{x}) \\ \text{subject to} \;\; & g_m(\boldsymbol{x}) \leq 0, \quad m = 1, \ldots, M_I \\ & h_m(\boldsymbol{x}) = 0, \quad m = 1, \ldots, M_E. \end{aligned} \tag{4}$$

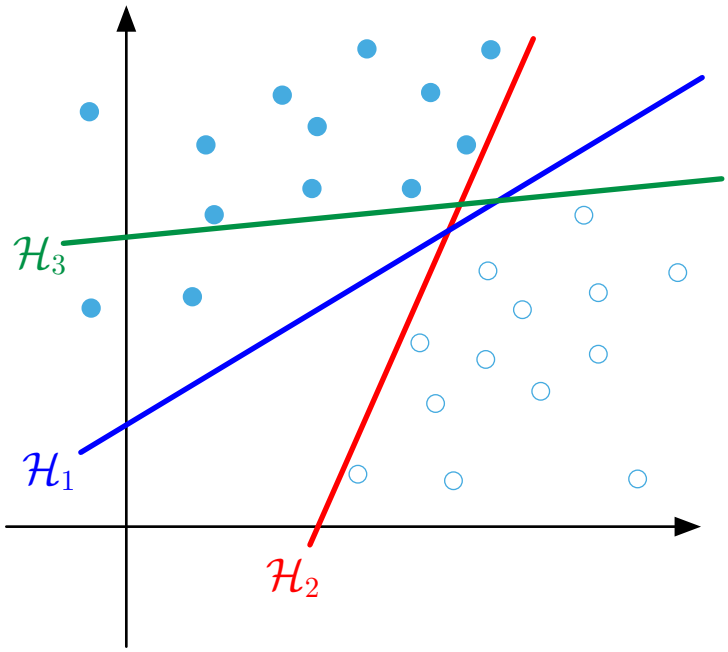
Using essentially the same arguments as before, we can show that the following KKT conditions on $\boldsymbol{x}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\nu}$ are sufficient for $\boldsymbol{x}$ and $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ to be solutions of (4) and its dual, respectively. Moreover, if strong duality holds they are also necessary.

KKT (with equality constraints)
The KKT conditions for $\boldsymbol{x} \in \mathbb{R}^N$, $\boldsymbol{\lambda} \in \mathbb{R}^{M_I}$, and $\boldsymbol{\nu} \in \mathbb{R}^{M_E}$ are

$$g_m(\boldsymbol{x}) \leq 0, \quad m = 1, \ldots, M_I, \tag{K1}$$
$$h_m(\boldsymbol{x}) = 0, \quad m = 1, \ldots, M_E,$$
$$\lambda_m \geq 0, \quad m = 1, \ldots, M_I, \tag{K2}$$
$$\lambda_m g_m(\boldsymbol{x}) = 0, \quad m = 1, \ldots, M_I, \tag{K3}$$
$$\nabla f(\boldsymbol{x}) + \sum_{m=1}^{M_I} \lambda_m \nabla g_m(\boldsymbol{x}) + \sum_{m=1}^{M_E} \nu_m \nabla h_m(\boldsymbol{x}) = \mathbf{0}. \tag{K4}$$

Example: Support vector machines

Consider the following fundamental binary classification problem. We are given points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M \in \mathbb{R}^N$ with class labels $y_1, \ldots, y_M$, where $y_m \in \{-1, +1\}$. We would like to find a hyperplane (i.e., affine functional) which *separates* the classes:



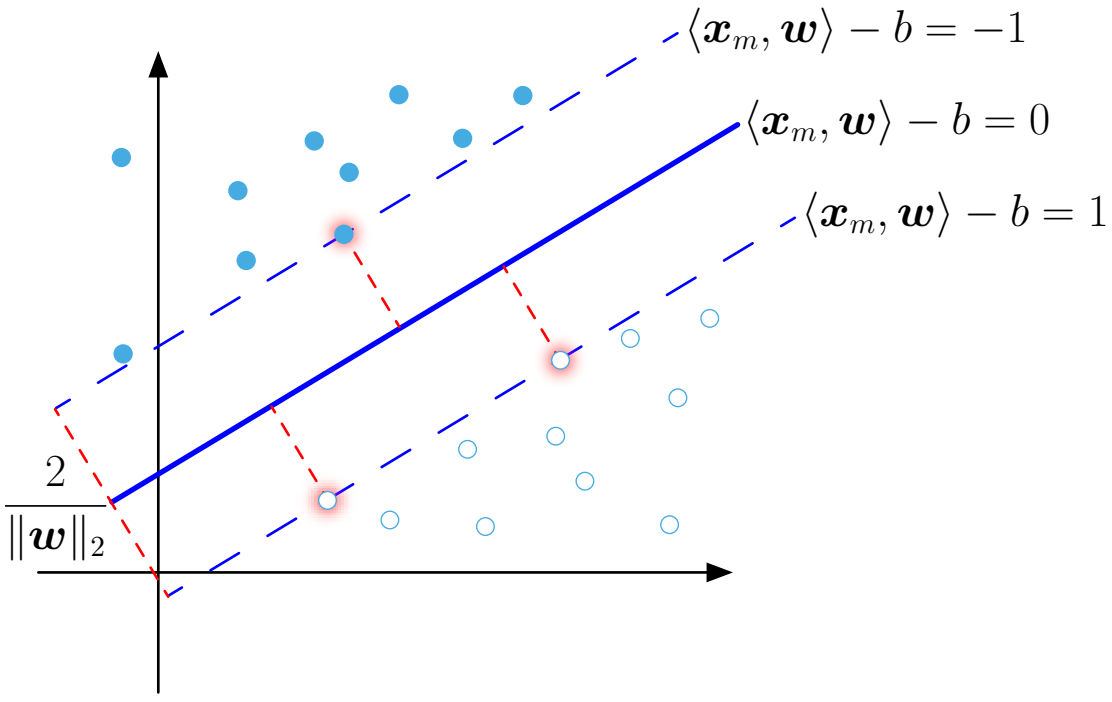
$\mathcal{H}_1$ and $\mathcal{H}_2$ above both separate the classes in $\mathbb{R}^2$, but $\mathcal{H}_3$ does not. While separating the classes is obviously desirable, we still need a good method to choose from among the many hyperplanes that do separate the classes – and some will perform better than others. Support vector machines (SVMs) take the one with *maximum margin*, i.e., we choose the hyperplane that maximizes the distance to the closest point in either class.

To restate this, we want to find a $\boldsymbol{w} \in \mathbb{R}^N$ and $b \in \mathbb{R}$ such that

$$\begin{aligned} \langle \boldsymbol{x}_m, \boldsymbol{w} \rangle - b &\geq 1, \quad \text{when } y_m = 1, \\ \langle \boldsymbol{x}_m, \boldsymbol{w} \rangle - b &\leq -1, \quad \text{when } y_m = -1. \end{aligned}$$

Of course, it is possible that no separating hyperplane exists; in this case, there will be no feasible points in the program above. It is straightforward, though, to modify this discussion to allow "mislabeled" points.

In the formulation above, the distance between the two (parallel) hyperplanes is $2/\|\boldsymbol{w}\|_2$:



Thus maximizing this distance is the same as minimizing $\|\boldsymbol{w}\|_2$.

This leads to the program

$$\begin{aligned} &\underset{\boldsymbol{w}\in\mathbb{R}^N,\ b\in\mathbb{R}}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ &\text{subject to} \ \ y_m(b - \langle \boldsymbol{x}_m, \boldsymbol{w}\rangle) + 1 \ \leq 0, \quad m = 1, \ldots, M. \end{aligned}$$

This is a linearly constrained quadratic program, and is clearly convex. The Lagrangian is

$$\begin{aligned} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda}) &= \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \sum_{m=1}^{M} \lambda_m \left[y_m(b - \langle \boldsymbol{x}_m, \boldsymbol{w}\rangle) + 1\right] \\ &= \frac{1}{2}\|\boldsymbol{w}\|_2^2 + b\,\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w} + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{1}, \end{aligned}$$

where $\boldsymbol{X}$ is the $N \times M$ matrix

$$\boldsymbol{X} = \begin{bmatrix} y_1\boldsymbol{x}_1 & y_2\boldsymbol{x}_2 & \cdots & y_M\boldsymbol{x}_M \end{bmatrix}.$$

The dual function is

$$d(\boldsymbol{\lambda}) = \inf_{\boldsymbol{w},b} \left(\frac{1}{2}\|\boldsymbol{w}\|_2^2 + b\,\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w} + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{1} \right).$$

Since b is unconstrained above, we see that the presence of $b\,\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{y}$ means that the dual will be $-\infty$ unless $\langle \boldsymbol{\lambda}, \boldsymbol{y}\rangle = 0$. Minimizing over $\boldsymbol{w}$, we need the gradient equal to zero,

$$\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda}) = \mathbf{0}, \quad \Rightarrow \quad \boldsymbol{w} - \boldsymbol{X}\boldsymbol{\lambda} = \mathbf{0}.$$

This means that we must have $\boldsymbol{w} = \boldsymbol{X}\boldsymbol{\lambda}$, which itself is a very handy fact as it gives us a direct passage from the dual solution to the primal

solution. With these substitutions, the dual function is

$$d(\boldsymbol{\lambda}) = \begin{cases} \frac{1}{2}\|\boldsymbol{X}\boldsymbol{\lambda}\|_2^2 - \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{\lambda} + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{1}, & \langle \boldsymbol{\lambda}, \boldsymbol{y} \rangle = 0, \\ -\infty, & \text{otherwise.} \end{cases}$$

Thus, the dual SVM program is then

$$\begin{aligned} \underset{\boldsymbol{\lambda}}{\text{maximize}} \quad & -\frac{1}{2}\|\boldsymbol{X}\boldsymbol{\lambda}\|_2^2 + \sum_{m=1}^{M} \lambda_m \\ \text{subject to} \quad & \langle \boldsymbol{\lambda}, \boldsymbol{y} \rangle = 0, \quad \boldsymbol{\lambda} \geq \mathbf{0}. \end{aligned}$$

Given the solution $\boldsymbol{\lambda}^\star$ to the dual, we can take $\boldsymbol{w}^\star = \boldsymbol{X}\boldsymbol{\lambda}^\star$, and the classifier is

$$\begin{aligned} f(\boldsymbol{x}) &= \langle \boldsymbol{x}, \boldsymbol{w}^\star \rangle - b^\star \\ &= \langle \boldsymbol{x}, \boldsymbol{X}\boldsymbol{\lambda}^\star \rangle - b^\star \\ &= \sum_{m=1}^{M} \lambda_m^\star y_m \langle \boldsymbol{x}, \boldsymbol{x}_m \rangle - b^\star. \end{aligned}$$

Notice that the data $\boldsymbol{x}_m$ appear only through inner products with $\boldsymbol{x}$.

A key realization about the SVM is that the for the dual program, the objective function depends on the data $\boldsymbol{x}_m$ only through inner products, as

$$\|\boldsymbol{X}\boldsymbol{\lambda}\|_2^2 = \sum_{\ell=1}^{M}\sum_{m=1}^{M} y_\ell y_m \langle \boldsymbol{x}_\ell, \boldsymbol{x}_m \rangle.$$

This means that we can replace $\langle \boldsymbol{x}_\ell, \boldsymbol{x}_m \rangle$ with any "positive kernel function" $K(\boldsymbol{x}_\ell, \boldsymbol{x}_m) : \mathbb{R}^N \otimes \mathbb{R}^N \to \mathbb{R}$ – a positive kernel just means that the $M \times M$ matrix $K(\boldsymbol{x}_\ell, \boldsymbol{x}_m)$ is in S_+^M for all choices of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M$.

For example, you might take

$$K(\boldsymbol{x}_\ell, \boldsymbol{x}_m) = (1 + \langle \boldsymbol{x}_\ell, \boldsymbol{x}_m \rangle)^2 = 1 + 2\langle \boldsymbol{x}_\ell, \boldsymbol{x}_m \rangle + \langle \boldsymbol{x}_\ell, \boldsymbol{x}_m \rangle^2.$$

This means we have replaced the inner product of two vectors with the inner product between two vectors which have been mapped into a higher-dimensional space:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_N \\ x_1^2 \\ x_2^2 \\ \vdots \\ x_N^2 \\ \sqrt{2}x_1x_2 \\ \vdots \\ \sqrt{2}x_{N-1}x_N \end{bmatrix}.$$

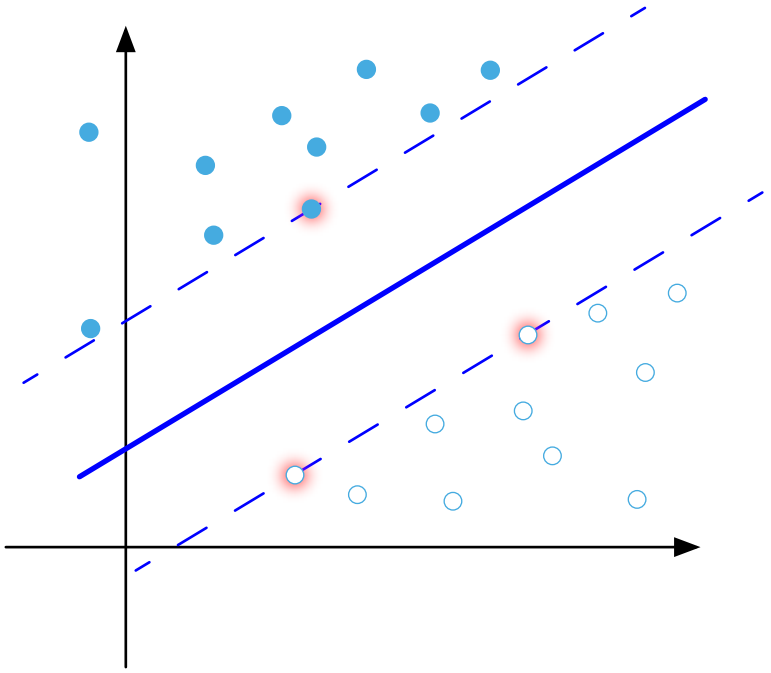
A set of linear constraints on the coordinates on the right, then, corresponds to a second order curve constraint (parabola, ellipse, hyperbola) on the coordinates on the left.

Many kernels are possible. The advantage is that to train and use the classifier, you never have to explicitly move to the higher-dimensional space – you just need to be able to compute $K(\boldsymbol{x}_\ell, \boldsymbol{x}_m)$ for any pair of inputs in $\mathbb{R}^N$. A popular choice of kernel is

$$K(\boldsymbol{x}_\ell, \boldsymbol{x}_m) = \exp\left(-\gamma \|\boldsymbol{x}_\ell - \boldsymbol{x}_m\|_2^2\right).$$

This is a perfectly valid positive kernel, and it is straightforward to compute it for any pair of inputs. But it corresponds to mapping the $\boldsymbol{x}_m$ into an infinite dimensional space, then finding a separating hyperplane.

Here is an example of a linear classifier in a higher-dimensional space:



that results in a nonlinear classifier in a lower-dimensional space: