CONVEX OPTIMIZATION, DUALITY, AND THEIR APPLICATION TO SUPPORT VECTOR MACHINES

DANIEL HENDRYCKS

Abstract. This paper develops the fundamentals of convex optimization and applies them to Support Vector Machines, a machine learning model. To do this, first we cover basic definitions and consequences in convexity. We then utilize the Lagrange dual function to aid with general optimization. Last, we apply these results to train a Support Vector Machine efficiently.

Contents

1. Introduction

Function optimization occurs in contexts from traveling (minimize time spent driving) to ethics (maximize expected pleasure). Finding a function's extrema can be difficult in general, but if the function is convex, the task is tractable. Other important tasks like object classification can be made convex, and therefore tractable, with the help of objects called support vector machines. These machines are covered in depth in machine learning courses, but they are made more useful with concepts from convex optimization. As such, this document aims to cover the rudiments of convexity, basics of optimization, and consequences of duality. These methods culminate into a way for support vector machines to "learn" to classify objects efficiently.

2. Convex Sets

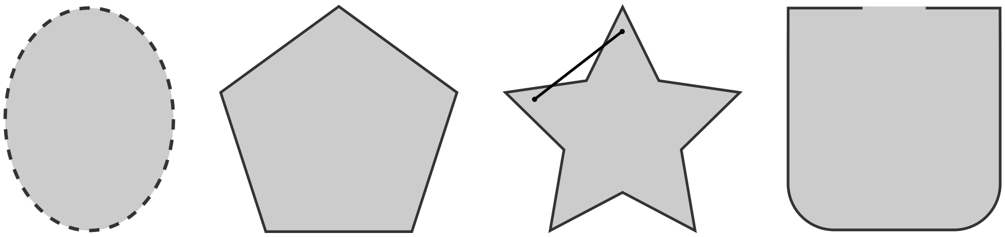
In order to learn convex optimization, we must first learn some basic vocabulary. We begin by defining convexity and internalize the definition through various examples.

2.1. **Definition.** A set $C \subseteq \mathbb{R}^n$ is convex if the the line segment between any two points in C lies in C. That is, for any $x_1, x_2 \in C$ and any $0 \leq t \leq 1$, we have

$$tx_1 + (1-t)x_2 \in C.$$

2.2. **Examples.**

(1) **Planar shapes.**



To start with the examples, we notice that some simple shapes are convex. For example, the left two sets are convex. However, the star is not—the segment is not within the star. Last, the right box is also not convex. This is evident by taking a segment which extends from the top left corner to the top right, and see that the set lacks points necessary for convexity.

(2) **The nonnegative region.** Let's move to more abstract sets. Clearly $\mathbb{R}^n$ is convex. Within $\mathbb{R}^n$ is the subset $\mathbb{R}^n_+ = \{x = (x_1, \ldots, x_n) | x_i \geq 0,\ 1 \leq i \leq n\}$. This is convex because should we take any $x, y \in \mathbb{R}^n_+$, $0 \leq t \leq 1$, then

$$(tx + (1-t)y)_i \geq 0,$$

where x_i is the ith component of vector x.

(3) **Hyperplanes and halfspaces.** Also within $\mathbb{R}^n$ are the hyperplanes and halfspaces, and they are convex. In order show this, we will symbolically represent hyperplanes and show they satisfy the definition of convexity. To start, recall that hyperplanes are the translates of $(n-1)$-dimensional subspaces. Let us represent $(n-1)$-dimensional subspaces. A subspace S and its orthogonal complement $S^\perp = \{y | x^T y = 0 \text{ for all } x \in S\}$ of course satisfy $\dim S + \dim S^\perp = n$. Any $(n-1)$-dimensional subspace S has the form $\{x | x^T b = 0\}$ for some nonzero b in $S^\perp$. We translate the $(n-1)$-dimensional subspace by arbitrary vector a to get a hyperplane

$$\{x_1 + a | x_1 \perp b\} = \{x_2 | (x_2 - a)^T b = 0\} = \{x_2 | x_2^T b = \alpha\},$$

where $\alpha = a^T b$. Finally, hyperplanes are a convex set because for $x_1, x_2 \in \{x | x^T b = \alpha\}$, we have $(tx + (1-t)y)^T b = t\alpha + (1-t)\alpha = \alpha$. Halfspaces $\{x | x^T b \leq \alpha\}$ are convex by the same argument.

(4) **Norm balls.** There are many convex objects with more curvature than the previous example. Let $\| \cdot \|$ be a norm on $\mathbb{R}^n$. The norm ball centered at x_c with radius r is the set $B = \{x | \|x - x_c\| \leq r\}$, and it is convex. To see how, suppose $x, y \in B$. Again,

$$\|tx + (1-t)y - x_c\| \leq t\|x - x_c\| + (1-t)\|y - x_c\| \leq r.$$

(5) **Positive semidefinite matrices.** As seen above, several familiar objects turn out to be convex. Let us now look beyond $\mathbb{R}^n$. At the same time, we will establish notation and look at a trivial example. The set $\mathbb{S}^n$ is the set of symmetric $n \times n$ matrices, and the set of symmetric postive semidefinite matrices is $\mathbb{S}^n_+ := \{X \in \mathbb{S}^n | z^T X z \geq 0,\ z \in \mathbb{R}^n\}$. Easily $z^T(tX + (1-t)X)z = tz^T X z + (1-t)z^T X z \geq 0$. Notice that the convexity of $\mathbb{S}^n_+$ follows from the convexity of $\mathbb{R}^n_+$.

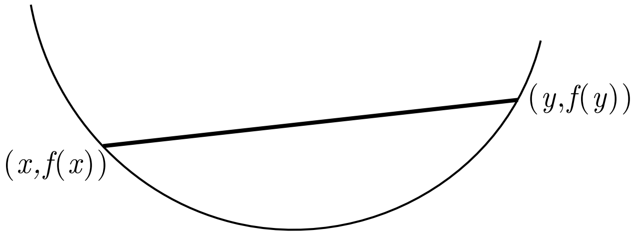
These various sets are building blocks for more complicated convex sets. We must use this knowledge of convex sets to confirm whether a function is convex.

3. Convex Functions

3.1. **Definition.** A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if **dom** f, the domain of f, is a convex set and if for all x, $y \in$ **dom** f, and $0 \leq t \leq 1$, we have

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y).$$

We can use the previous section's content to verify that the domain is convex. For the inequality in the definition, it is helpful to have a geometric interpretation.



The geometric interpretation of this definition is that the chord from x to y lies above (or on) the graph of f.

Should the chord lie below (or on) the graph, f is called *concave*. Equivalently, f is concave if $-f$ is convex. Fundamental definitions aside, it is now time for a fundamental property.

3.2. **First-order convexity condition.** Suppose that f is differentiable. Then f is convex if and only if **dom** f is convex and for all $x, y \in$ **dom** f,

$$f(y) \geq f(x) + \nabla_x f(x)^T (y - x),$$

where $\nabla_x f(x) = \left(\dfrac{\partial f(x)}{\partial x_1}, \dfrac{\partial f(x)}{\partial x_2}, \ldots, \dfrac{\partial f(x)}{\partial x_n}\right)$.

Lemma. A set $C \subseteq \mathbb{R}^n$ is convex if and only if its intersection with an arbitrary line is convex.

Proof. Easily, the intersection of convex sets is convex. As a line is convex, so too is the intersection of a convex set and a line.

Conversely, suppose C intersected with a line is convex. Take $x, y \in C$. Note that, for $0 \leq t \leq 1$, $tx + (1-t)y \in C$ because the line through x and y when intersected with C is convex. ■

Lemma. The function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if its restriction to any line is convex. That is, for any $x, v \in \mathbb{R}^n$ and $s \in \mathbb{R}$, the function $g(s) := f(x + sv)$ is convex.

Proof. To start, let f be convex. The domain of g is the convex set **dom** f intersected with a line, so **dom** g is convex, as is necessary for g to be convex. Second, by the convexity of f,

$$\begin{aligned} g(ts_1 + (1-t)s_2) &= f(x + [ts_1 + (1-t)s_2]v) \\ &= f(x + ts_1 v + (1-t)s_2 v) \\ &= f(t(s_1 v + x) + (1-t)(s_2 v + x)) \\ &\leq tf(s_1 v + x) + (1-t)f(s_2 v + x) \\ &= tg(s_1) + (1-t)g(s_2). \end{aligned}$$

Showing the converse is similar. ■

Proof of the first-order convexity condition. We will start the proof by considering when $n = 1$. After considering this case, we consider arbitrary n, restrict the function $f : \mathbb{R}^n \to \mathbb{R}$ to a line, and thereby rely on the $n = 1$ case.

Require that $f : \mathbb{R} \to \mathbb{R}$ be convex, x and $y \in$ **dom** f, and $0 < t \leq 1$. As a consequence, $f(x + t(y - x)) \leq (1-t)f(x) + tf(y)$. Manipulating this, we see

$$f(y) \geq f(x) + \frac{f(x + t(y - x)) - f(x)}{t},$$

and the limit as $t \to 0$ gives our desired inequality.

Conversely, let $x, y \in$ **dom** f. More, choose $x \neq y$, $0 \leq t \leq 1$, and $z = tx + (1-t)y$. By assumption,

$$f(x) \geq f(z) + f'(z)(x - z), \qquad f(y) \geq f(z) + f'(z)(y - z).$$

Weigh the first inequality by t and the second by $1 - t$, then add. This yields an equation establishing convexity; viz.,

$$tf(x) + (1-t)f(y) \geq f(x).$$

More generally, $f : \mathbb{R}^n \to \mathbb{R}$. In this setting, we restrict f to the line passing through $x, y \in \mathbb{R}^n$. Call the restriction $g(t) = f(ty + (1-t)x)$. Then $g'(t) = \nabla f(ty + (1-t)x)^T(y-x)$. Also, assume that f is convex; then g is convex by the lemma. Therefore, $g(1) \geq g(0) + g'(0)$, so

$$f(y) \geq f(x) + \nabla f(x)^T (y - x).$$

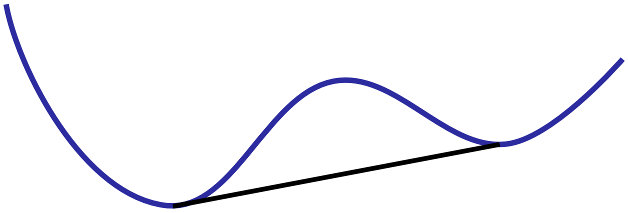
Conversely, suppose this inequality is true for any x, y. If $ty + (1-t)x \in \textbf{dom } f$ and $\psi y + (1-\psi)x \in \textbf{dom } f$, then

$$f(ty + (1-t)x) \geq f(\psi y + (1-\psi)x) + \nabla f(\psi y + (1-\psi)x)^T (y-x)(t-\psi),$$

and $g(t) \geq g(\psi) + g'(\psi)(t - \psi)$. Therefore, g is convex. ■

Corollary. If $\nabla f(x) = 0$, then $f(x)$ is the minimum of f.

Therefore, local minima are automatically global minima. Thus there is no need to worry about there existing better minima whenever a local minima is found. However, this is not to suggest convex functions tend to have many local minima. A non-affine convex function has at most one global minimum. If there were two local minima with higher points between, then the chord would lie below the graph—this violates convexity.



3.3. **Second-order convexity condition.** Since local minima of a convex function are automatically global minima, other necessary and sufficient conditions for convexity are highly valuable. The second-order convexity condition provides us with another way to detect convexity in a function.

It says that a twice differentiable function f is convex if and only if the Hessian is positive semidefinite and the domain is convex.

Proof. Assume $n = 1$, like in the previous proof. Let $x, y \in \textbf{dom } f$ be such that $x < y$. From the first-order condition,

$$f(y) \geq f(x) + f'(x)(y - x)$$
$$-f(x) \leq -f(y) - f'(x)(x - y).$$

Then

$$f'(x)(y-x) \leq f(y) - f(x) \leq -f'(y)(x-y) = f'(y)(y-x),$$

so

$$\frac{f'(y) - f'(x)}{y - x} \geq 0$$

gives the result when we take the limit as $y \to x$.

Conversely, suppose z is in the convex set $\textbf{dom } f$ and $f''(z) \geq 0$. With x and y as before, we have

$$\begin{aligned} 0 &\leq \int_x^y f''(z)(y-z)\,\mathrm{d}z \\ &= -f'(x)(y-x) + f(y) - f(x). \end{aligned}$$

As before, generalizing this to $n > 1$ involves a tedious yet straightforward restriction of the Hessian to a line. ■

3.4. **Examples.** The usefulness of these conditions hold for several familiar functions which are convex. Further, by being the negative of convex functions, concave functions can exploit the conditions discussed above (after we multiply the function by -1). Notice the many familiar functions which can exploit the above properties!

Convex examples on $\mathbb{R}$.

- Affine functions are of the form $ax + b$, where $a, b \in \mathbb{R}$. On $\mathbb{R}$, affine functions are convex.

- Exponential functions e^{ax}, $a \in \mathbb{R}$ are convex on its domain $\mathbb{R}$. Verifying the exponential function's convexity is simple by taking its second derivative.
- Powers x^a for $a \geq 1$ or $a \leq 0$ are convex on $\mathbb{R}_{>0}$. However, x^{-1} is not convex on $\mathbb{R} \setminus \{0\}$ due to its singularity. Bear in mind that for a function to be convex, its domain must be convex too.
- The quadratic form $f(x) = x^T Ax/2 + b^T x + c$ where $A \in \mathbb{S}^n$ is a convex function if and only if A is positive semidefinite. This follows because $\nabla^2 f(x) = A$.

Concave examples on $\mathbb{R}$.

- Affine functions are concave on $\mathbb{R}$. Affine functions are *both* convex and concave.
- The logarithm $\log x$ is concave on $\mathbb{R}_{>0}$.
- Powers x^a for $0 \leq a \leq 1$ are concave on $\mathbb{R}_{>0}$.

There are several more advanced functions which are convex. The three following convex functions are not of particular importance for our coverage of support vector machines, but these are worth knowing for deeper study of convexity.

Composition of scalar functions. Let $g : \mathbb{R}^n \to \mathbb{R}$, $h : \mathbb{R} \to \mathbb{R}$, and $f(x) = h(g(x))$. Then f is convex if g is convex and h is convex and nondecreasing. Also, f is convex if g is concave and h is convex and nonincreasing. This is evident when $n = 1$ and g and h are twice differentiable. Under the conditions described,

$$f''(x) = h''(g(x))[g'(x)]^2 + h'(g(x))g''(x) \geq 0.$$

The proof is more involved when $n > 1$.

Epigraph. Recall that the graph of a function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as $\{(x, f(x)) | x \in \textbf{dom } f\}$. The *epigraph* of a function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as

$$\textbf{epi } f = \{(x, s) | x \in \textbf{dom } f,\, f(x) \leq s\}.$$

Proposition. Function f is convex if and only if **epi** f is convex.

Proof. To see this, assume that f is convex, $0 \leq t \leq 1$, and $(x_1, s_1), (x_2, s_2) \in \textbf{epi } f$. Then

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2) \leq ts_1 + (1-t)s_2.$$

Just as we wanted, $t(x_1, s_1) + (1-t)(x_2, s_2) \in \textbf{epi } f$.

Conversely, let **epi** f be convex. This means that

$$f(tx_1 + (1-t)x_2) \leq ts_1 + (1-t)s_2.$$

Say $s_1 = f(x_1), s_2 = f(x_2)$. Therefore, f is convex. ■

A function's epigraph ties together the definition of convex functions and convex sets. This central relation is used in the next example.

Pointwise supremum of convex functions. Assume that each function f_i for $i \in \mathcal{I}$ is convex. Using these convex functions, the function $f(x) = \sup\{f_i(x) | i \in \mathcal{I}\}$ is convex. To see this, first recall that the intersection of an arbitrary collection of convex sets is convex. Now see that

$$\textbf{epi } f = \bigcap_{i \in \mathcal{I}} \textbf{epi } f_i,$$

so the pointwise supremum is convex. We use this result later when studying duality!

With the fundamentals and a few extras from convexity covered, we now turn to optimization and duality.

4. Duality

The spirit of duality is to bound or solve an optimization problem with a different optimization problem. Before bounding optimization problems, we need to formalize what a convex optimization problem is.

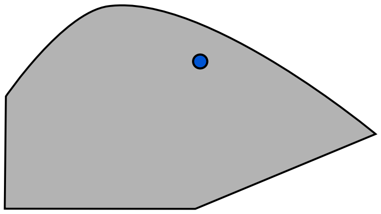
4.1. Optimization.

Optimizing differentiable, convex functions is presumably easy; merely set the derivative to zero to find the minima. However, when the function is constrained, a point where the derivative is zero may not exist in the domain. Worse, if the function is not differentiable, then the first- and second-order conditions do not apply. Despite these wrinkles, the problem still may be convex. We can represent these various situations with the standard form for a convex optimization problem as follows:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad 1 \leq i \leq m, \\ & h_i(x) = 0, \quad 1 \leq i \leq p, \end{array} \tag{4.1}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$, $f_0, f_1, \ldots, f_m$ are convex, and $h_i : \mathbb{R}^n \to \mathbb{R}$ are affine. Denote the optimal value $p^\star$.

Let us obtain intuition about convex optimization problems more generally. To do this, consider where $p^\star$ can be achieved within a convex set. Suppose that we are told $p^\star$ is achieved on the blue point in the following convex set and *only* at this point. That cannot occur. Here is why.



Proposition. A convex function on a closed interval attains it maximum at an endpoint.
Proof. Without loss of generality, consider convex $f : [0,1] \to \mathbb{R}$ and take any $x \in [0,1]$. Notice

$$\begin{aligned} f(x) &= f(x \cdot 1 + (1-x) \cdot 0) \\ &\leq x f(1) + (1-x) f(0) \\ &\leq [x + (1-x)] \max\{f(0), f(1)\} \\ &= \max\{f(0), f(1)\}. \end{aligned}$$

The maximum must occur on one of the endpoints. ■

A line through the blue point intersected with the convex set is a segment between points from two boundary points of the convex set. Because of this, the maximum must occur at an endpoint of the segment. The implication is that $p^\star$ cannot be attained exclusively at the blue point. However, if $p^\star$ is attained at the blue point, then it must occur everywhere on the segment.

4.2. **The Lagrangian and the dual function.** With some intuition about the convex optimization problems in general, we consider ways to manage the problem's constraints, as these add more difficulty to the problem. To meet this challenge, we define the *Lagrangian* $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ as

$$\mathcal{L}(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x).$$

Vectors λ such that $\lambda_i \geq 0$ $(1 \leq i \leq m)$ and ν are called *Lagrange multipliers* or *dual variables.* For simplicity, we write $\lambda \succeq 0$ to indicate that each component of λ is nonnegative. As we will see shortly, this function lets us bound the original optimization problem, hence the Lagrangian is useful. Before discussing that bound, we motivate these multipliers by considering how they "penalize" the Lagrangian over infeasible vectors.
Proposition. The Lagrangian satisfies

$$\sup_{\lambda \succeq 0, \nu} \mathcal{L}(x, \lambda, \nu) = \begin{cases} f_0(x) & \text{for feasible } x \\ \infty & \text{otherwise.} \end{cases}$$

Proof. To prove the first equation, assume that x is feasible. The convex optimization problem setup stipulated that $f_i(x) \leq 0$ for $1 \leq i \leq m$ and $h_i(x) = 0$ for $1 \leq i \leq p$. Since $\lambda \succeq 0$,

$$\sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \leq 0.$$

Thus for feasible x, $\sup\limits_{\lambda \succeq 0, \nu} \mathcal{L}(x, \lambda, \nu) = f_0(x)$.
In the other situation, x is infeasible. Necessarily, there exists and i such that $f_i(0) > 0$ or $h_i \neq 0$. Then let $\lambda_i \to \infty$ or $\nu_i \to \text{sign}(h_i)\infty$. ■
The proposition lets us represent problem (4.1) more compactly as $\inf\limits_{x} \sup\limits_{\lambda \succeq 0, \nu} \mathcal{L}(x, \lambda, \nu)$. Another usage of the Lagrangian is the *dual function* $g : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$

$$g(\lambda, \nu) = \inf_{x \text{ feasible}} \mathcal{L}(x, \lambda, \nu) = \inf_{x \text{ feasible}} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right).$$

Notice that $g(\lambda, v) \leq p^\star$ because $\sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \leq 0$. Already the dual is useful because when $\lambda \succeq 0$, the dual function gives a nontrivial lower bound on $p^\star$. That is, the dual allows us to bound the original optimization problem.

4.3. **The dual problem.** We have identified that for $\lambda \succeq 0$, $g(\lambda, \nu)$ gives a lower bound on $p^\star$. The *dual problem* is to find the best lower bound. Thus we have

$$\begin{aligned} \text{maximize} \quad & g(\lambda, \nu) \\ \text{subject to} \quad & \lambda \succeq 0. \end{aligned}$$

Notice that the dual is concave since it is an infimum of affine functions in (λ, ν); recall that the pointwise supremum of convex functions is convex. Because of this, the dual problem can be solved with convex optimization, and that is efficient! We can efficiently gain valuable information about the original problem with the dual problem regardless of the original problem's convexity. This alone justifies studying duality. After solving the dual problem, we obtain $d^\star$, the dual's best lower bound of $p^\star$. Value $d^\star$ from the dual problem allows us to obtain valuable information efficiently about the original optimization problem (4.1). Knowing that $d^\star \leq p^\star$, we may wonder under what conditions $d^\star = p^\star$. When these conditions are satisfied, the dual would not only bound the original problem efficiently, it would solve the original problem. A condition where $d^\star = p^\star$ is called Slater's constraint qualification.
Theorem (Strong Duality). Consider problem (4.1). Assume there exists feasible x such that the constraint

$$f_i(x) < 0, \quad 1 \leq i \leq m$$

holds. This is called Slater's constraint qualification; there must be a feasible x where the inequalities are strict. Then the Lagrange dual function satisfies

$$d^\star = \sup_{\lambda \succeq 0, \nu} \inf_{x} \mathcal{L}(x, \lambda, \nu) = \inf_{x} \sup_{\lambda \succeq 0, \nu} \mathcal{L}(x, \lambda, \nu) = p^\star,$$

and this is called *strong duality.*
The theorem's proof is technical and is available in standard reference textbooks (e.g., Boyd and Vandenberghe, pg. 235).

Strong duality is useful not only because it implies that $d^\star = p^\star$ but also because it gives information about λ. To see this, first say that $x^\star$ is optimal for problem (4.1) and $(\lambda^\star, \nu^\star)$ is dual optimal. Then when strong duality holds,

$$\begin{aligned} p^\star = d^\star &= \inf_{x} \mathcal{L}(x, \lambda^\star, \nu^\star) \\ &\leq \mathcal{L}(x^\star, \lambda^\star, \nu^\star) \\ &= f_0(x^\star) + \sum_{i=1}^{m} \lambda_i^\star f_i(x^\star) + \sum_{i=1}^{p} \nu_i^\star h_i(x^\star) \\ &\leq f(x^\star) = p^\star. \end{aligned}$$

Evidently, $\sum_{i=1}^{m} \lambda_i^\star f_i(x^\star) + \sum_{i=1}^{p} \nu_i^\star h_i(x^\star) = 0$, so $\lambda_i^\star f_i(x^\star) = 0$, $1 \leq i \leq m$. This new information about λ is the last piece of a theorem which wraps together many concepts in duality.

Theorem (Karush-Kuhn-Tucker for convex functions). Consider the following conditions in the context of problem (4.1).

$$\begin{aligned} f_i(x^\star) &\leq 0, \quad 1 \leq i \leq m, \\ h_i(x^\star) &= 0, \quad 1 \leq i \leq p, \\ \lambda^\star &\succeq 0, \\ \lambda_i^\star f_i(x^\star) &= 0, \quad 1 \leq i \leq m, \\ \nabla_x \left(f_0(x^\star) + \sum_{i=1}^{m} \lambda_i^\star f_i(x^\star) + \sum_{i=1}^{p} \nu_i^\star h_i(x^\star) \right) &= 0, \end{aligned}$$

These conditions hold if and only if $x^\star$ and $(\lambda^\star, \nu^\star)$ are optimal for the original convex optimization problem and the dual problem, and $d^\star = p^\star$.[1]

Essentially, the KKT theorem provides an excellent checklist for whether $p^\star = d^\star$, and that lets us know whether we can solve the original problem with the dual problem.

4.4. **Examples.** Through examples, we demonstrate the KKT conditions in action on two elementary problems found in machine learning.

(1) **Least-squares solution.** Consider the affine space determined by $Xw = y$ where $X \in \mathbb{R}^{m \times n}$, $w \in \mathbb{R}^n$, and $y \in \mathbb{R}^m$. We may wish to find the least "complex" but satisfactory w to this equation. A way to capture complexity is with the ℓ_2-norm of w defined as $\|w\|_2 := \sqrt{x^T x}$. More precisely, our least-norm convex optimization problem is

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|w\|_2^2 \\ \text{subject to} \quad & Xw = y \end{aligned}$$

after we square the norm and divide it by two for simplicity. We can solve this with the Lagrangian. With the dual variable $v \in \mathbb{R}^m$, the Lagrangian is $\mathcal{L}(w, \nu) = w^T w/2 + \nu^T (Xw - y)$. After computing $\nabla_w \mathcal{L}(w, \nu) = w + X^T \nu$, we see that the optimality conditions are

$$Xw^\star - y = 0, \qquad w^\star + X^T \nu^\star = 0.$$

Consequently, $y = Xw^\star = -XX^T \nu^\star$ and $w^\star = -X^T \nu^\star$, so we have

$$w^\star = X^T (XX^T)^{-1} y.$$

(2) **Regularization.** Many times we may seek to reduce the least squared error while keeping the "complexity" of w small. A way to do this is *regularization.* Formally, for some fixed, positive real number a, regularization means adding $a\|w\|_2^2$ to the objective $\frac{1}{2}\|Xw - y\|_2^2$, and this gives us the unconstrained convex optimization problem

$$\text{minimize} \quad \frac{1}{2}\|Xw - y\|_2^2 + a\|w\|_2^2.$$

We wish to show that this is equivalent to the constrained convex optimization problem

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|Xw - y\|_2^2 \\ \text{subject to} \quad & \|w\|_2^2 \leq b \end{aligned}$$

for some constant b. Our constrained problem has the Lagrangian $\mathcal{L}(w, \lambda) = \frac{1}{2}\|Xw - y\|_2^2 + \lambda \left(\|w\|_2^2 - b \right)$, and thus we have the optimality conditions

$$\|w^\star\|_2^2 - b \leq 0, \quad \lambda^\star \geq 0, \quad \lambda^\star \left(\|w^\star\|_2^2 - b \right) = 0, \quad \nabla_w \mathcal{L}(w^\star, \lambda^\star) = 0.$$

[1] Karush discovered this in an unpublished Master's thesis at the University of Chicago. A simplified proof by Brezhneva, Tretyakov, and Wright is available here: http://link.springer.com/article/10.1007%2Fs11590-008-0096-3.

Meanwhile, the unconstrained problem must have that $\nabla_w \left(\frac{1}{2}\|Xw-y\|_2^2 + a\|w\|_2^2\right) = X^TXw - X^Ty + 2aw = 0$ for optimality. Call the solution of the unconstrained problem $w^\star(a)$. Now notice that

$$\begin{aligned}\nabla_w \mathcal{L}(w^\star, \lambda^\star) &= X^TXw^\star - X^Ty + 2\lambda^\star w^\star \\ &= 0 \\ &= X^TXw^\star(a) - X^Ty + 2aw^\star(a).\end{aligned}$$

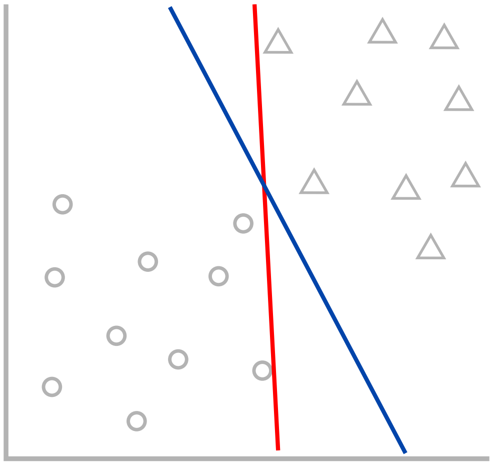
In light of this, if we let $\lambda^\star = a$, $w^\star = w^\star(a)$, and $b = \|w^\star(a)\|_2^2$, we satisfy the KKT conditions for the constrained problem. This shows that both problems have the same solution.

5. Support Vector Machines

Having just had a taste for problems which occur in machine learning, we turn to support vector machines and see the role duality plays in them.

5.1. Computing separation lines through support vectors.

Imagine that we want to categorize vectors into one of two classes, either circles or triangles. Given examples of circles and triangles, we could make a line of separation between the classes. Here are two of the infinitely many ways to separate the classes.



Both lines separate the classes, so both are sufficient for classification. However, suppose a new datum is received but its class is unknown. Say that this datum is located at the bottom left corner. Using either the red or blue line, we can predict the datum belongs to the circle class, as it is on the side of the lines with the other circles. Should, however, we receive a datum near the bottom center, the red line might have us classify it as a circle or triangle (depending on what side of the line the datum is), while the blue line would readily have us predict a circle. Clearly the blue line provides a more reasonable class prediction. What, more generally, makes the blue line better?

We want to capture what makes the blue line superior to the red line so we can better predict unclassified data. A highly relevant factor making the blue line better is that the minimum of the distances between the lines and the points (these distances are *margins*) is greater with the blue line than with the red. That is, the blue line has a larger smallest margin than the red line. This suggests we should want to find the separation line with the largest minimum margin. Is satisfying this heuristic feasible?

Fortunately, this objective can be made into a convex optimization problem. In trying to develop the convex problem, we first formalize our notions. Let us pretend a machine will "learn" or discover the best line according to the margin heuristic. In order for the machine to make predictions about unclassified data, it needs examples of points in the circle class and points in the triangle class to generate a line of separation. Data that the machine learns from is called training data; it is the set $\left\{\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right) \ldots, \left(x^{(m)}, y^{(m)}\right)\right\}$ where $x^{(i)} \in \mathbb{R}^n$ is the coordinates of a vector and $y^{(i)} \in \{-1, 1\}$ gives the label for point $x^{(i)}$, $1 \le i \le m$. For our running example, let -1 represent the circle class, 1 the triangle class. Moreover, let the line of separation be $w^Tx + b = 0$, where $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$. With this notation, we can try to construct the appropriate optimization problem.

In our first attempt to make the problem convex, we first notice that we want all classified points to be beyond or on some margin γ. Equivalently, we demand that $y^{(i)}(w^T x^{(i)} + b) \geq \gamma$ for $1 \leq i \leq m$. It stands to reason that we could try to solve the problem

$$\begin{aligned} \text{maximize} \quad & \gamma \\ \text{subject to} \quad & y^{(i)}(w^T x^{(i)} + b) \geq \gamma, \quad 1 \leq i \leq m. \end{aligned}$$

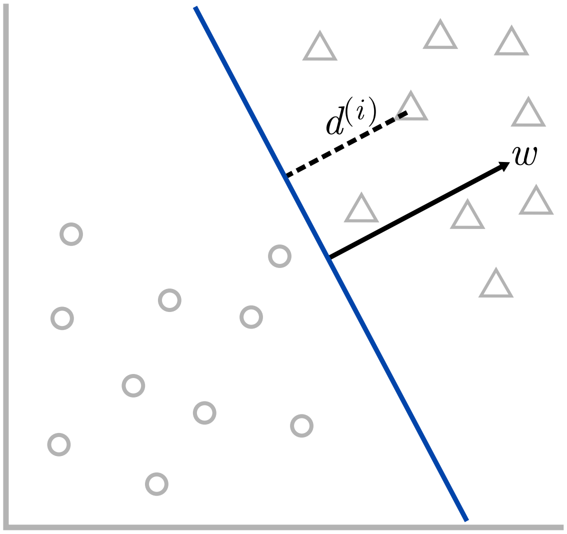
However, a sufficient γ implies that $c\gamma$, $c > 1$, is also sufficient—merely make w larger by scaling. Therefore, a sufficient γ implies the problem has no maximum, as γ's are unbounded.

We tackle the problem of γ becoming unbounded by penalizing γ with the magnitude of w. Then we have

$$\begin{aligned} \text{maximize} \quad & \gamma/\|w\|_2 \\ \text{subject to} \quad & y^{(i)}(w^T x^{(i)} + b) \geq \gamma, \quad 1 \leq i \leq m. \end{aligned}$$

This formulation is close to the upcoming, final formulation, but first the replacement of γ with $\gamma/\|w\|_2$ requires discussion.

Before modifying this new formulation further, we provide theoretical justification for our replacement of γ with $\gamma/\|w\|_2$ because this step may seem ad-hoc. To see the justification for that step, first let us denote the distance from $x^{(i)}$ to the separating line with $d^{(i)}$. The vector from the line to the point $x^{(i)}$ is thus $x^{(i)} - d^{(i)} \cdot w/\|w\|_2$ if $y^{(i)} = 1$.



Therefore, $w^T \left(x^{(i)} - d^{(i)} \cdot w/\|w\|_2\right) + b = 0$. Rearrangement implies that $d^{(i)} = (w/\|w\|_2)^T x^{(i)} + b/\|w\|_2$ or

$$d^{(i)} = y^{(i)} \left[\left(\frac{w}{\|w\|_2} \right)^T x^{(i)} + \frac{b}{\|w\|_2} \right]$$

when we take into account the sign of $y^{(i)}$. Consequently, maximizing $\gamma/\|w\|_2$ is equivalent to maximizing the geometric distance $d = \min_i d^{(i)}$.

Having justified our problem's reformulation, we can tweak it so that we have the equivalent problem

$$\begin{aligned} \text{minimize} \quad & \|w\|_2/\gamma \\ \text{subject to} \quad & y^{(i)}(w^T x^{(i)} + b) \geq \gamma, \quad 1 \leq i \leq m. \end{aligned}$$

Modifying it further, we can square $\|w\|_2/\gamma$, multiply it by half, and modify the constraints. The convex problem then becomes

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|w/\gamma\|_2^2 \\ \text{subject to} \quad & y^{(i)}((w/\gamma)^T x^{(i)} + b) \geq 1, \quad 1 \leq i \leq m. \end{aligned}$$

Observe now that the problem has us utilize the ratio w/γ and not the separate components w and γ. Also, recognize that scaling w by $1/\gamma$ does not change which value of w solves the problem. We therefore let $\gamma = 1$ and obtain the final form of the convex optimization problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|w\|_2^2 \\ \text{subject to} & y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad 1 \leq i \leq m. \end{array}$$

Let us solve it. Starting by using the Lagrangian, we have

$$\mathcal{L}(w, b, \lambda) = \frac{1}{2}\|w\|_2^2 - \sum_{i=1}^{m} \lambda_i \left[y^{(i)}(w^T x^{(i)} + b) - 1 \right].$$

As usual, in deriving the dual, we differentiate. Note

$$\nabla_w \mathcal{L}(w, b, \lambda) = w - \sum_{i=1}^{m} \lambda_i y^{(i)} x^{(i)} = 0,$$

and

$$\nabla_b \mathcal{L}(w, b, \lambda) = \sum_{i=1}^{m} \lambda_i y^{(i)} = 0.$$

Substituting and simplifying reveals

$$\mathcal{L}(w, b, \lambda) = \sum_{i=1}^{m} \lambda_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i \lambda_j y^{(i)} y^{(j)} (x^{(i)})^T x^{(j)}.$$

Therefore, the dual problem is

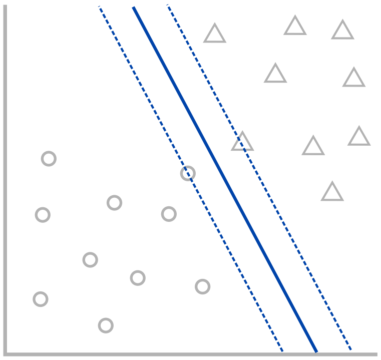
$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{m} \lambda_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i \lambda_j y^{(i)} y^{(j)} (x^{(i)})^T x^{(j)} \\ \text{subject to} & \lambda \succeq 0 \\ & \sum_{i=1}^{m} \lambda_i y^{(i)} = 0. \end{array}$$

For $p^\star$ to equal $d^\star$, the KKT conditions must hold, and indeed they do.

The dual gave us another way to find $p^\star$, but why go through that work? We saw that duality lets us bound or solve the original problem, but the original problem is convex. Why not put the original convex problem in a convex optimization solver?

To motivate why we derived the dual problem, we must preliminarily remind the reader of the constraints

$$f_i(w) = -y^{(i)}(w^T x^{(i)} + b) + 1 \leq 0.$$

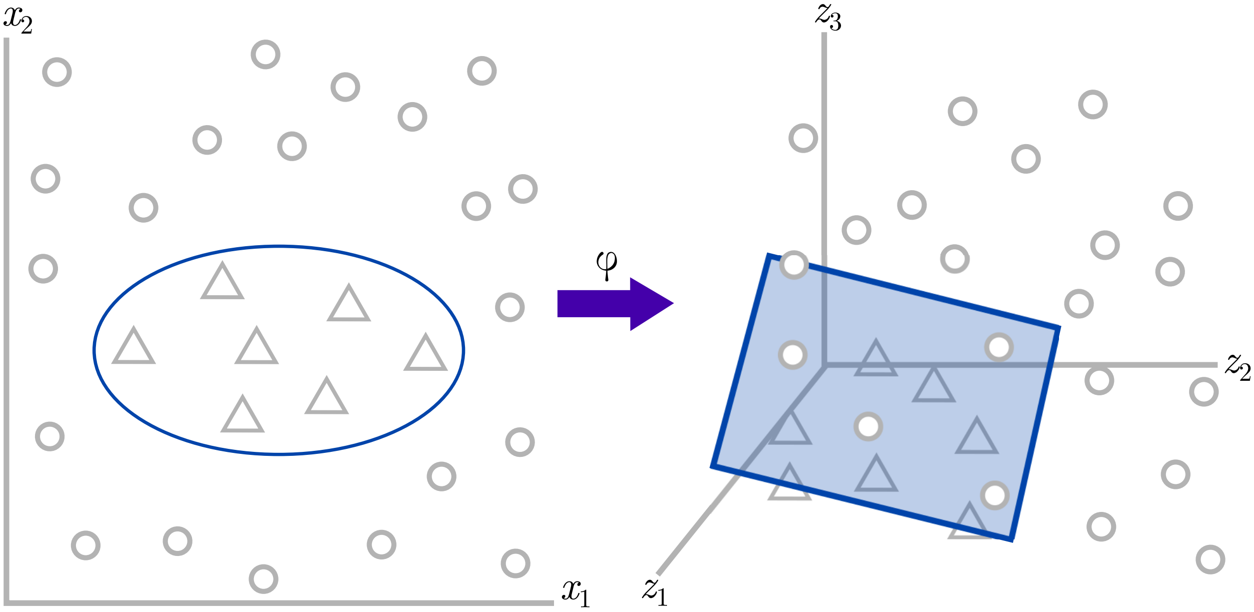


Now, the two vectors touching the dotted line make $f_i(w) = 0$, as they are the vectors with distance $d = 1/\|w\|_2 = 1$ from the separation line. All other vectors have $f_i(w) < 0$. By the KKT conditions, we have that if $\lambda_i > 0$, then $f_i(w) = 0$, and if $f_i(w) > 0$, then $\lambda_i = 0$. This is exceedingly useful because then

$$\begin{aligned} w^T x + b &= \left(\sum_{i=1}^{m} \lambda_i y^{(i)} x^{(i)} \right)^T x + b \\ &= \sum_{i:\lambda_i > 0} \lambda_i y^{(i)} (x^{(i)})^T x + b. \end{aligned}$$

Computing the separation line is now highly computationally efficient because we need only to compute the inner product of the two vectors—called support vectors—seeing that all other $\lambda_i's = 0$.[2] In other problems, the number of support vectors could be greater, but they are likely to be a small proportion of the vectors from the training set. By needing only to consider a few of the training set's vectors by exploiting duality, we can have our machine compute the best separation line rapidly. By relying only on these support vectors, a *support vector machine* is the machine that learns a separation line.

5.2. **The kernel trick.** So far, support vector machines are useful only when training data is linearly separable. Fortunately, mapping inseparable data to new coordinates can let SVM's learn to separate the data. For example, in the figure below, the data on the left becomes linearly separable in the right coordinate system.



Here, we say that

$$\varphi(x_1, x_2) = \left(x_1^2, \sqrt{2} x_1 x_2, x_2^2 \right).$$

With this in mind, we see that the elliptical decision boundary becomes linear in the new space because

$$\frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} = c \implies \frac{z_1}{a^2} + \frac{z_3}{b^2} = c.$$

Therefore, we want the machine to find a separation hyperplane of the form $w^T \varphi(x) + b = 0$ instead of $w^T x + b = 0$. Now let us consider mappings which can handle more complicated data, as the separation hyperplane expressed by $\varphi(x) = \left(x_1^2, \sqrt{2} x_1 x_2, x_2^2\right)$ may be insufficient for there to exist a separating hyperplane between classes. We may need a more complicated mapping like $\varphi : \mathbb{R}^n \to \mathbb{R}^{n^2}$ where

$$\varphi(x) = (x_1 x_1, x_1 x_2, \ldots, x_1 x_n, \ldots, x_n x_1, x_n x_2, \ldots, x_n x_n).$$

[2] Some readers may wonder about the value of b. It is $b^\star = -\left(\max_{i:y^{(i)}=-1} (w^\star)^T x^{(i)} + \min_{i:y^{(i)}=1} (w^\star)^T x^{(i)} \right) /2.$

By the previous section, the hyperplane is

$$\sum_{i:\lambda_i>0} \lambda_i y^{(i)} \varphi(x^{(i)})^T \varphi(x) + b = \sum_{i:\lambda_i>0} \lambda_i y^{(i)} \left[\sum_{k=1}^{n} \sum_{l=1}^{n} x_k^{(i)} x_l^{(i)} x_k x_l \right] + b.$$

This computation is messy and expensive. As an attempt to reduce its complexity, we note that the term $\varphi\left(x^{(i)}\right)^T \varphi(x)$ can be more compactly expressed as $\left(\left(x^{(i)}\right)^T x\right)^2 =: \kappa(x^{(i)}, x)$. This form is computationally cheaper, as $\left(\left(x^{(i)}\right)^T x\right)^2$ does not require computing the n^2 terms of the mapping—it only requires $O(n)$ computations. To gain this computational saving, we can substitute $\varphi\left(x^{(i)}\right)^T \varphi(x)$ with $\kappa(x^{(i)}, x)$ and obtain the hyperplane

$$\sum_{i:\lambda_i>0} \lambda_i y^{(i)} \kappa(x^{(i)}, x) + b.$$

Our above substitution is called the *kernel trick*. Notice that we did not explicitly map to $\mathbb{R}^{n^2}$. In contrast, we computed a simple function between vectors in $\mathbb{R}^n$. Meanwhile, we captured the effect of mapping to $\mathbb{R}^{n^2}$ by computing a function equivalent to $\varphi\left(x^{(i)}\right)^T \varphi(x)$. Thus we get the benefit of linear separability from a high-dimensional space without the computationally expensive $O\left(n^2\right)$ mapping by φ. More generally, the kernel trick lets us avoid the cost of computing the inner product between high-dimensional or infinite-dimensional vectors by instead computing a simpler function called a kernel. Other popular kernels include

$$\exp\left(-\frac{\|x^{(i)} - x\|_2^2}{2\sigma^2}\right) \qquad \text{and} \qquad \left((x^{(i)})^T z + c\right)^d.$$

Kernels provide a way to obtain linear separability in a high-dimensional space without paying the cost of mapping to a high-dimensional space. This makes SVM's far more powerful than those from the previous section all while remaining efficient.

After building the fundamentals of convexity and duality, we saw that support vector machines can achieve a reasonable separation line by considering only the vectors with corresponding constants $\lambda_i \neq 0$. Often, this is a small fraction of the data, so exploiting duality is useful computationally. Thereafter, recognizing that many datasets are not linearly separable in typical coordinates, we aimed to map the data to a different space where it is linearly separable. In many situations, that sought space is very high-dimensional or infinite-dimensional. Yet this concern is manageable because we can successfully capture the inner product between vectors in the mapped space with kernels, functions much easier to compute. Ultimately, kernels give SVM's much power for classification, and duality lets us find the separator efficiently.

6. Acknowledgments

It is a pleasure to thank my mentor Sergei Sagatov for his advice, help, and instruction. I would also like to thank Professor Laszlo Babai for his fantastic lectures. Finally, I would like to thank Peter May for organizing this REU. I am grateful to have taken part in it.

References

[1] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*, Cambridge University Press. http://stanford.edu/ boyd/cvxbook/

[2] Alex Smola and S.V.N. Vishwanathan. *Introduction to Machine Learning*, Cambridge University Press. http://alex.smola.org/drafts/thebook.pdf

[3] R. Tyrrell Rockafellar. *Convex Analysis*, Princeton University Press.

[4] Andrew Ng. *Support Vector Machines.* http://cs229.stanford.edu/notes/cs229-notes3.pdf

[5] Tommi Jaakkola. *The Support Vector Machine.* http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-867-machine-learning-fall-2006/lecture-notes/lec3.pdf