SVM

CDS, NYU

Feb 14, 2023

Today's lecture:

- Support Vector Machines: one of the most widely used classification model
- We will focus on linear SVM today (non-linear SVM next week!)
- Plan:
    - Derive the SVM learning objective (in two ways)
    - Solve the optimization problem
    - Get insight from its dual problem
- (Requires some background knowledge on convex optimization)

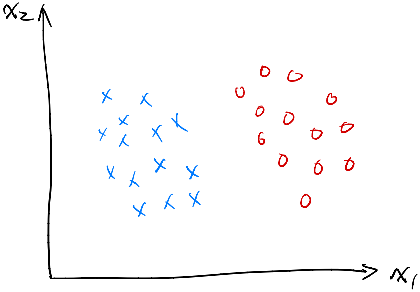
Part I: Derive the SVM Objective

- Start with the inductive bias: what makes a good linear decision boundary?
- Start with the loss function and regularization

Maximum Margin Classifier

Linearly Separable Data

Consider a linearly separable dataset $\mathcal{D}$:



Find a separating hyperplane such that

- $w^T x_i > 0$ for all x_i where $y_i = +1$
- $w^T x_i < 0$ for all x_i where $y_i = -1$

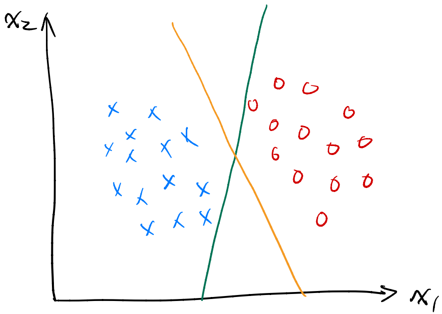
The Perceptron Algorithm

- Initialize $w \leftarrow 0$
- While not converged (exists misclassified examples)
  - For $(x_i, y_i) \in \mathcal{D}$
    - If $y_i w^T x_i < 0$ (wrong prediction)
    - Update $w \leftarrow w + y_i x_i$
- Intuition: move towards misclassified positive examples and away from negative examples
- Guarantees to find a zero-error classifier (if one exists) in finite steps
- What is the loss function if we consider this as a SGD algorithm?

Maximum-Margin Separating Hyperplane

For separable data, there are infinitely many zero-error classifiers.

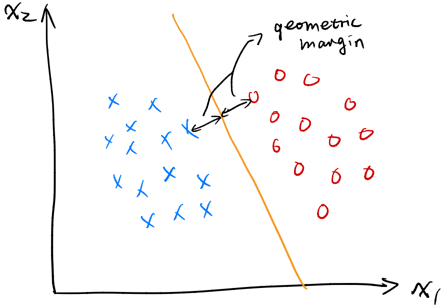
Which one do we pick?



(Perceptron does not return a unique solution.)

Maximum-Margin Separating Hyperplane

We prefer the classifier that is farthest from both classes of points



- Geometric margin: smallest distance between the hyperplane and the points
- Maximum margin: *largest* distance to the closest points

Geometric Margin

We want to maximize the distance between the separating hyperplane and the closest points.

Let's formalize the problem.

Definition (separating hyperplane)

We say (x_i, y_i) for $i = 1, \ldots, n$ are **linearly separable** if there is a $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ such that $y_i(w^T x_i + b) > 0$ for all i. The set $\{v \in \mathbb{R}^d \mid w^T v + b = 0\}$ is called a **separating hyperplane**.

Definition (geometric margin)

Let H be a hyperplane that separates the data (x_i, y_i) for $i = 1, \ldots, n$. The **geometric margin** of this hyperplane is

$$\min_i d(x_i, H),$$

the distance from the hyperplane to the closest data point.

Distance between a Point and a Hyperplane

- Projection of $v \in \mathbb{R}^d$ onto $w \in \mathbb{R}^d$: $\frac{v \cdot w}{\|w\|_2}$

- Distance between x_i and H:

$$d(x_i, H) = \left| \frac{w^T x_i + b}{\|w\|_2} \right| = \frac{y_i(w^T x_i + b)}{\|w\|_2}$$

Maximize the Margin

We want to maximize the geometric margin:

$$\text{maximize } \min_i d(x_i, H).$$

Given separating hyperplane $H = \left\{v \mid w^T v + b = 0\right\}$, we have

$$\text{maximize } \min_i \frac{y_i(w^T x_i + b)}{\|w\|_2}.$$

Let's remove the inner minimization problem by

$$\begin{array}{ll} \text{maximize} & M \\ \text{subject to} & \frac{y_i(w^T x_i + b)}{\|w\|_2} \geqslant M \quad \text{for all } i \end{array}$$

Note that the solution is not unique (why?).

Maximize the Margin

Let's fix the norm $\|w\|_2$ to $1/M$ to obtain:

$$\begin{array}{ll} \text{maximize} & \frac{1}{\|w\|_2} \\ \text{subject to} & y_i(w^T x_i + b) \geqslant 1 \quad \text{for all } i \end{array}$$

It's equivalent to solving the minimization problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|w\|_2^2 \\ \text{subject to} & y_i(w^T x_i + b) \geqslant 1 \quad \text{for all } i \end{array}$$

Note that $y_i(w^T x_i + b)$ is the (functional) margin.

In words, it finds the minimum norm solution which has a margin of at least 1 on all examples.

Soft Margin SVM

What if the data is *not* linearly separable?

For any w, there will be points with a negative margin.

Introduce **slack variables** to penalize small margin:

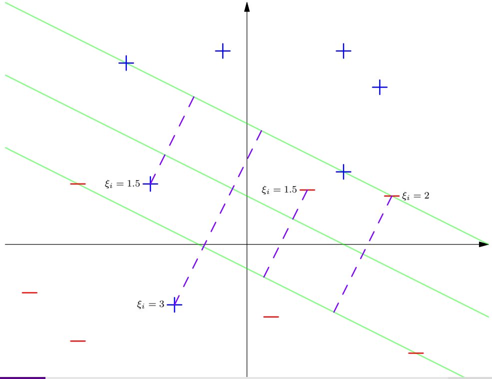
$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|w\|_2^2 + \frac{C}{n}\sum_{i=1}^n \xi_i \\ \text{subject to} & y_i(w^Tx_i + b) \geqslant 1 - \xi_i \quad \text{for all } i \\ & \xi_i \geqslant 0 \quad \text{for all } i \end{array}$$

- If $\xi_i = 0\ \forall i$, it's reduced to hard SVM.
- What does $\xi_i > 0$ mean?
- What does C control?

Slack Variables

$d(x_i, H) = \frac{y_i(w^T x_i + b)}{\|w\|_2} \geqslant \frac{1-\xi_i}{\|w\|_2}$, thus ξ_i measures the violation by multiples of the geometric margin:

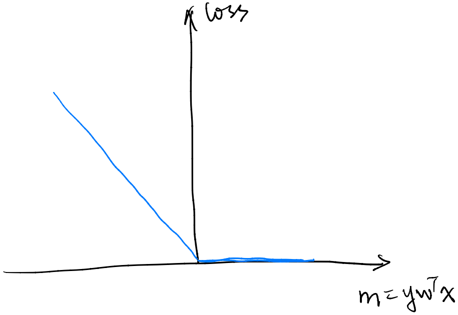
- $\xi_i = 1$: x_i lies on the hyperplane
- $\xi_i = 3$: x_i is past 2 margin width beyond the decision hyperplane



Minimize the Hinge Loss

Perceptron Loss

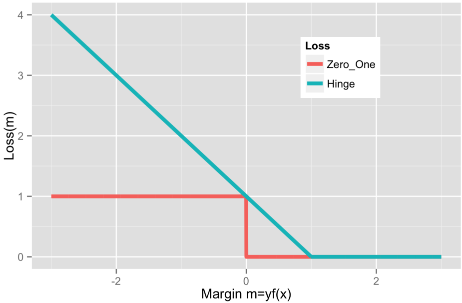
$$\ell(x, y, w) = \max(0, -yw^T x)$$



If we do ERM with this loss function, what happens?

Hinge Loss

- SVM/Hinge loss: $\ell_{\text{Hinge}} = \max\{1-m, 0\} = (1-m)_+$
- Margin $m = yf(x)$; "Positive part" $(x)_+ = x\mathbb{1}(x \geqslant 0)$.



Hinge is a **convex**, **upper bound** on $0-1$ loss. Not differentiable at $m=1$.
We have a **"margin error"** when $m < 1$.

Support Vector Machine

Using ERM:

- Hypothesis space $\mathcal{F} = \left\{ f(x) = w^T x + b \mid w \in \mathrm{R}^d, b \in \mathrm{R} \right\}$.
- ℓ_2 regularization (Tikhonov style)
- Hinge loss $\ell(m) = \max\{1 - m, 0\} = (1 - m)_+$
- The SVM prediction function is the solution to

$$\min_{w \in \mathrm{R}^d, b \in \mathrm{R}} \frac{1}{2}||w||^2 + \frac{c}{n} \sum_{i=1}^{n} \max\left(0, 1 - y_i \left[w^T x_i + b \right] \right).$$

- Not differentiable because of the max

SVM as a Constrained Optimization Problem

- The SVM optimization problem is equivalent to

$$\begin{array}{ll}\text{minimize} & \frac{1}{2}\|w\|^2 + \frac{c}{n}\sum_{i=1}^{n}\xi_i \\ \text{subject to} & \xi_i \geqslant \max\left(0, 1 - y_i\left[w^T x_i + b\right]\right) \text{ for } i = 1, \ldots, n.\end{array}$$

- Which is equivalent to

$$\begin{array}{ll}\text{minimize} & \frac{1}{2}\|w\|^2 + \frac{c}{n}\sum_{i=1}^{n}\xi_i \\ \text{subject to} & \xi_i \geqslant \left(1 - y_i\left[w^T x_i + b\right]\right) \text{ for } i = 1, \ldots, n \\ & \xi_i \geqslant 0 \text{ for } i = 1, \ldots, n\end{array}$$

Summary

Two ways to derive the SVM optimization problem:

- Maximize the (geometric) margin
- Minimize the hinge loss with ℓ_2 regularization

Both leads to the minimum norm solution satisfying certain margin constraints.

- **Hard-margin SVM**: all points must be correctly classified with the margin constraints
- **Soft-margin SVM**: allow for margin constraint violation with some penalty

Part II: Subgradient Descent for SVM

Now that we have the objective, can we do SGD on it?

Subgradient: generalize gradient for non-differentiable convex functions

SVM Optimization Problem (no intercept)

- SVM objective function:

$$J(w)=\frac{1}{n}\sum_{i=1}^{n}\max\left(0,1-y_iw^Tx_i\right)+\lambda\|w\|^2.$$

- Not differentiable... but let's think about gradient descent anyway.
- Hinge loss: $\ell(m)=\max(0,1-m)$

$$\begin{aligned}\nabla_wJ(w) &= \nabla_w\left(\frac{1}{n}\sum_{i=1}^{n}\ell\left(y_iw^Tx_i\right)+\lambda\|w\|^2\right)\\ &= \frac{1}{n}\sum_{i=1}^{n}\nabla_w\ell\left(y_iw^Tx_i\right)+2\lambda w\end{aligned}$$

"Gradient" of SVM Objective

- Derivative of hinge loss $\ell(m) = \max(0, 1-m)$:

$$\ell'(m) = \begin{cases} 0 & m > 1 \\ -1 & m < 1 \\ \text{undefined} & m = 1 \end{cases}$$

- By chain rule, we have

$$\begin{aligned} \nabla_w \ell\left(y_i w^T x_i\right) &= \ell'\left(y_i w^T x_i\right) y_i x_i \\ &= \begin{cases} 0 & y_i w^T x_i > 1 \\ -y_i x_i & y_i w^T x_i < 1 \\ \text{undefined} & y_i w^T x_i = 1 \end{cases} \end{aligned}$$

“Gradient” of SVM Objective

$$\nabla_w \ell\left(y_i w^T x_i\right) = \begin{cases} 0 & y_i w^T x_i > 1 \\ -y_i x_i & y_i w^T x_i < 1 \\ \text{undefined} & y_i w^T x_i = 1 \end{cases}$$

So

$$\begin{aligned} \nabla_w J(w) &= \nabla_w \left(\frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i w^T x_i\right) + \lambda \|w\|^2 \right) \\ &= \frac{1}{n} \sum_{i=1}^{n} \nabla_w \ell\left(y_i w^T x_i\right) + 2\lambda w \\ &= \begin{cases} \frac{1}{n} \sum_{i: y_i w^T x_i < 1} \left(-y_i x_i\right) + 2\lambda w & \text{all } y_i w^T x_i \neq 1 \\ \text{undefined} & \text{otherwise} \end{cases} \end{aligned}$$

Gradient Descent on SVM Objective?

- The gradient of the SVM objective is

$$\nabla_w J(w) = \frac{1}{n} \sum_{i: y_i w^T x_i < 1} (-y_i x_i) + 2\lambda w$$

  when $y_i w^T x_i \neq 1$ for all i, and otherwise is undefined.

Potential arguments for why we shouldn't care about the points of nondifferentiability:

- If we start with a random w, will we ever hit exactly $y_i w^T x_i = 1$?
- If we did, could we perturb the step size by ε to miss such a point?
- Does it even make sense to check $y_i w^T x_i = 1$ with floating point numbers?

However, would gradient descent work if the objective is not differentiable?

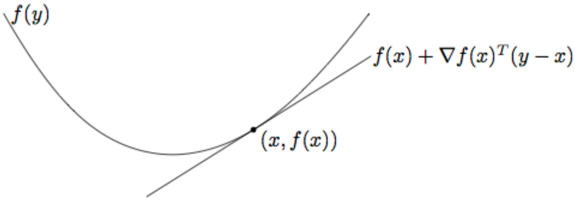
Subgradient

First-Order Condition for Convex, Differentiable Function

- Suppose $f : \mathbf{R}^d \to \mathbf{R}$ is convex and differentiable Then for any $x, y \in \mathbf{R}^d$

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

- The linear approximation to f at x is a global underestimator of f:



- This implies that if $\nabla f(x) = 0$ then x is a global minimizer of f.

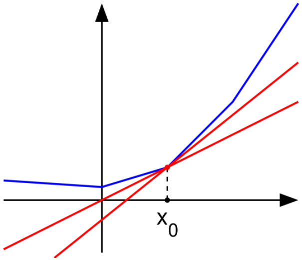
Figure from Boyd & Vandenberghe Fig. 3.2; Proof in Section 3.1.3

Subgradients

Definition

A vector $g \in \mathbb{R}^d$ is a **subgradient** of a *convex* function $f : \mathbb{R}^d \to \mathbb{R}$ at x if for all z,

$$f(z) \geqslant f(x) + g^T(z - x).$$



Blue is a graph of $f(x)$.
Each red line $x \mapsto f(x_0) + g^T(x - x_0)$ is a global lower bound on $f(x)$.

Properties

Definitions

- The set of all subgradients at x is called the **subdifferential**: $\partial f(x)$
- f is **subdifferentiable** at x if $\exists$ at least one subgradient at x.

For convex functions:

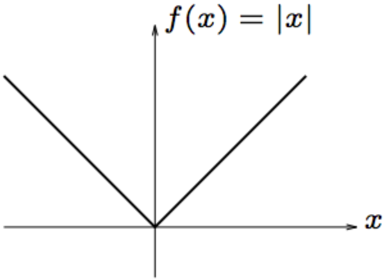
- f is differentiable at x iff $\partial f(x) = \{\nabla f(x)\}$.
- Subdifferential is always non-empty ($\partial f(x) = \emptyset \implies f$ is not convex)
- x is the global optimum iff $0 \in \partial f(x)$.

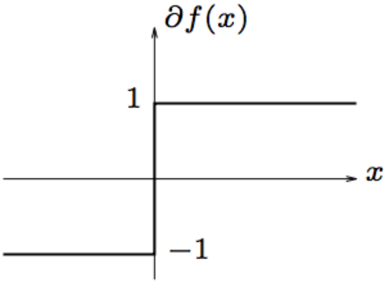
For non-convex functions:

- The subdifferential may be an empty set (no global underestimator).

Subdifferential of Absolute Value

- Consider $f(x) = |x|$



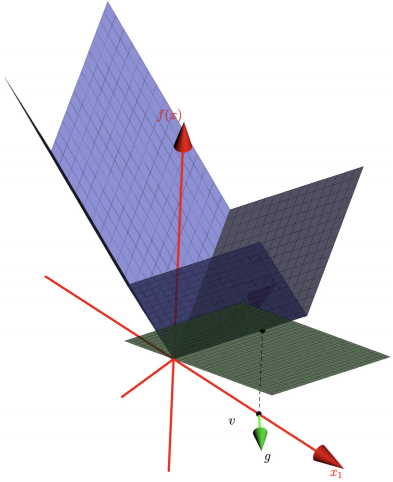


- Plot on right shows $\{(x, g) \mid x \in \mathbf{R},\ g \in \partial f(x)\}$

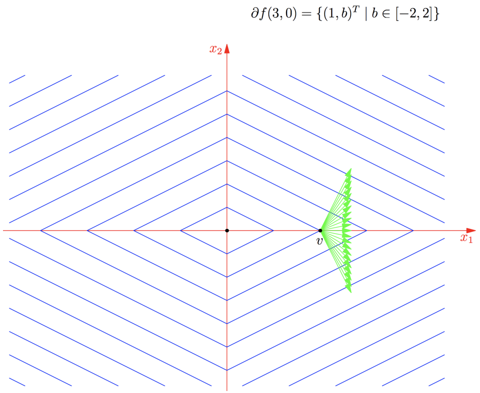
Boyd EE364b: Subgradients Slides

Subgradients of $f(x_1, x_2) = |x_1| + 2|x_2|$

- Let's find the subdifferential of $f(x_1, x_2) = |x_1| + 2|x_2|$ at $(3, 0)$.
- First coordinate of subgradient must be 1, from $|x_1|$ part (at $x_1 = 3$).
- Second coordinate of subgradient can be anything in $[-2, 2]$.
- So graph of $h(x_1, x_2) = f(3, 0) + g^T(x_1 - 3, x_2 - 0)$ is a global underestimate of $f(x_1, x_2)$, for any $g = (g_1, g_2)$, where $g_1 = 1$ and $g_2 \in [-2, 2]$.



Subdifferential on Contour Plot



Contour plot of $f(x_1, x_2) = |x_1| + 2|x_2|$, with set of subgradients at $(3,0)$. .

Plot courtesy of Brett Bernstein.

Basic Rules for Calculating Subdifferential

- Non-negative scaling: $\partial \alpha f(x) = \alpha \partial f(x)$ for $(\alpha > 0)$
- Summation: $\partial(f_1(x) + f_2(x)) = d_1 + d_2$ for any $d_1 \in \partial f_1$ and $d_2 \in \partial f_2$
- Composing with affine functions: $\partial f(Ax+b) = A^T \partial f(z)$ where $z = Ax+b$
- max: convex combinations of argmax gradients

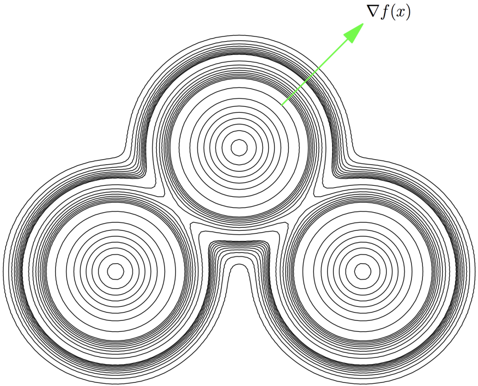
$$\partial \max(f_1(x), f_2(x)) = \begin{cases} \nabla f_1(x) & \text{if } f_1(x) > f_2(x), \\ \nabla f_2(x) & \text{if } f_1(x) < f_2(x), \\ \nabla \theta f_1(x) + (1-\theta) f_2(x) & \text{if } f_1(x) = f_2(x), \end{cases}$$

where $\theta \in [0,1]$.

Subgradient Descent

Gradient orthogonal to level sets

We know that gradient points to the fastest ascent direction. What about subgradients?



Plot courtesy of Brett Bernstein.

Contour Lines and Subgradients

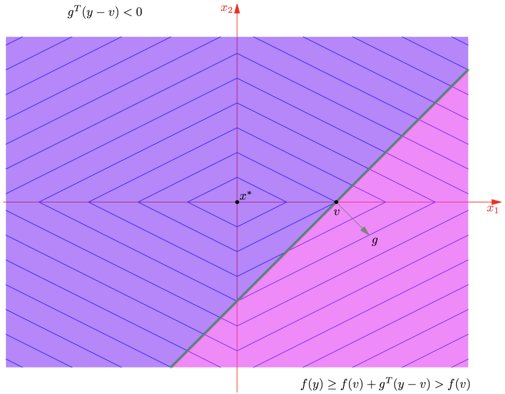
A hyperplane H **supports** a set S if H intersects S and all of S lies one one side of H.

Claim: If $f : \mathbf{R}^d \to \mathbf{R}$ has subgradient g at x_0, then the hyperplane H orthogonal to g at x_0 must **support** the level set $S = \left\{x \in \mathbf{R}^d \mid f(x) = f(x_0)\right\}$.

Proof:

- For any y, we have $f(y) \geqslant f(x_0) + g^T(y - x_0)$. (def of subgradient)
- If y is strictly on side of H that g points in,
  - then $g^T(y - x_0) > 0$.
  - So $f(y) > f(x_0)$.
  - So y is not in the level set S.
- $\therefore$ All elements of S must be on H or on the $-g$ side of H.

Subgradient of $f(x_1, x_2) = |x_1| + 2|x_2|$



- Points on g side of H have larger f-values than $f(x_0)$. (from proof)
- But points on $-g$ side may **not** have smaller f-values.
- So $-g$ may **not** be a descent direction. (shown in figure)

Plot courtesy of Brett Bernstein.

Subgradient Descent

- Move along the negative subgradient:

$$x^{t+1} = x^t - \eta g \quad \text{where } g \in \partial f(x^t) \text{ and } \eta > 0$$

- This can increase the objective but gets us closer to the minimizer if f is convex and η is small enough:

$$\|x^{t+1} - x^*\| < \|x^t - x^*\|$$

- Subgradients don't necessarily converge to zero as we get closer to x^*, so we need decreasing step sizes, e.g. $O(1/t)$ or $O(1/\sqrt{t})$.

- Subgradient methods are slower than gradient descent, e.g. $O(1/\epsilon^2)$ vs $O(1/\epsilon)$ for convex functions.

Based on https://www.cs.ubc.ca/~schmidtm/Courses/5XX-S20/S4.pdf

Subgradient descent for SVM (HW3)

SVM objective function:

$$J(w) = \frac{1}{n} \sum_{i=1}^{n} \max\left(0, 1 - y_i w^T x_i\right) + \lambda \|w\|^2.$$

Pegasos: stochastic subgradient descent with step size $\eta_t = 1/(t\lambda)$

Input: $\lambda > 0$. Choose $w_1 = 0, t = 0$
While termination condition not met
  For $j = 1, \ldots, n$ (assumes data is randomly permuted)
    $t = t + 1$
    $\eta_t = 1/\left(t\lambda\right);$
    If $y_j w_t^T x_j < 1$
      $w_{t+1} = (1 - \eta_t \lambda) w_t + \eta_t y_j x_j$
    Else
      $w_{t+1} = (1 - \eta_t \lambda) w_t$

Summary

- Subgradient: generalize gradient for non-differentiable convex functions
- Subgradient "descent":
  - General method for non-smooth functions
  - Simple to implement
  - Slow to converge

Part III: The Dual Problem

In addition to subgradient descent, we can directly solve the optimization problem using a QP solver.

Let's study its dual problem to gain addition insights (which will be useful for next week!)

SVM as a Quadratic Program

- The SVM optimization problem is equivalent to

$$
\begin{array}{ll}
\text{minimize} & \frac{1}{2}||w||^2 + \frac{c}{n}\sum_{i=1}^{n}\xi_i \\
\text{subject to} & -\xi_i \leqslant 0 \quad \text{for } i = 1, \dots, n \\
& \left(1 - y_i\left[w^T x_i + b\right]\right) - \xi_i \leqslant 0 \quad \text{for } i = 1, \dots, n
\end{array}
$$

- Differentiable objective function
- $n + d + 1$ unknowns and $2n$ affine constraints.
- A **quadratic program** that can be solved by any off-the-shelf QP solver.
- Let's learn more by examining the dual.

Why Do We Care About the Dual?

The Lagrangian

The general [inequality-constrained] optimization problem is:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leqslant 0, \quad i = 1, \ldots, m \end{array}$$

Definition

The **Lagrangian** for this optimization problem is

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

- λ_i's are called **Lagrange multipliers** (also called the **dual variables**).
- Weighted sum of the objective and constraint functions
- Hard constraints $\rightarrow$ soft constraints

Lagrange Dual Function

Definition

The **Lagrange dual function** is

$$g(\lambda) = \inf_x L(x, \lambda) = \inf_x \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) \right)$$

- $g(\lambda)$ is concave
- **Lower bound property**: if $\lambda \succeq 0$, $g(\lambda) \leqslant p^*$ where p^* is the optimal value of the optimization problem.
- $g(\lambda)$ can be $-\infty$ (uninformative lower bound)

The Primal and the Dual

- For any **primal form** optimization problem,

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leqslant 0, \;\; i = 1, \ldots, m, \end{array}$$

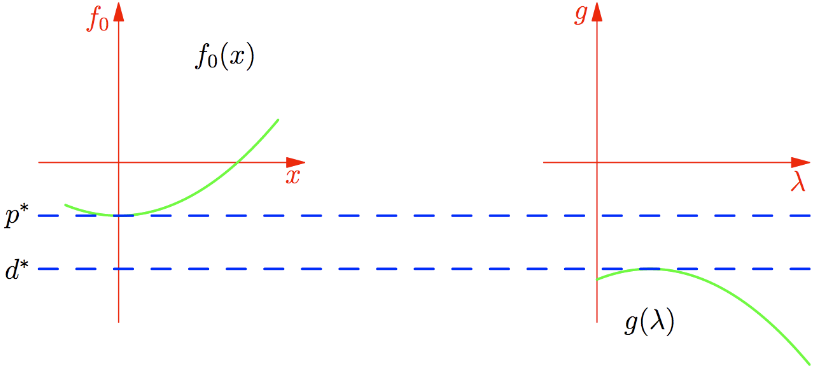
  there is a recipe for constructing a corresponding **Lagrangian dual problem:**

$$\begin{array}{ll} \text{maximize} & g(\lambda) \\ \text{subject to} & \lambda_i \geqslant 0, \;\; i = 1, \ldots, m, \end{array}$$

- The dual problem is always a convex optimization problem.
- The dual variables often have interesting and relevant interpretations.
- The dual variables provide certificates for optimality.

Weak Duality

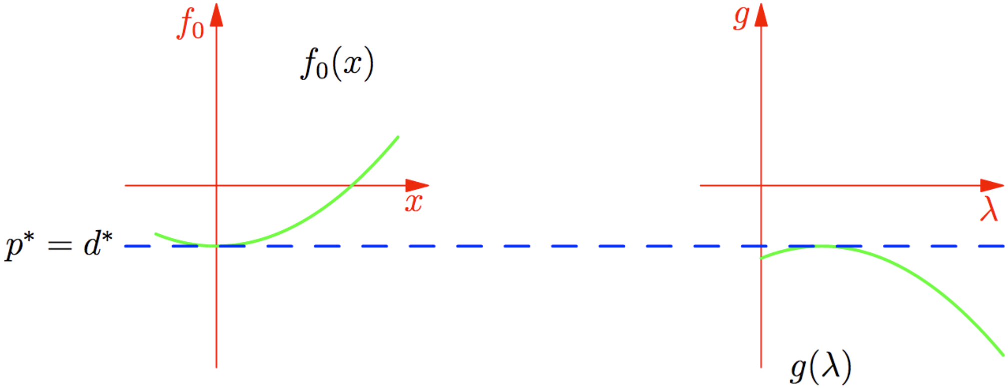
We always have **weak duality:** $p^* \geqslant d^*$.



Plot courtesy of Brett Bernstein.

Strong Duality

For some problems, we have **strong duality**: $p^* = d^*$.



For convex problems, strong duality is fairly typical.

Plot courtesy of Brett Bernstein.

Complementary Slackness

- Assume strong duality. Let x^* be primal optimal and λ^* be dual optimal. Then:

$$
\begin{aligned}
f_0(x^*) &= g(\lambda^*) = \inf_x L(x, \lambda^*) \quad \text{(strong duality and definition)} \\
&\leqslant L(x^*, \lambda^*) \\
&= f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* f_i(x^*) \\
&\leqslant f_0(x^*).
\end{aligned}
$$

Each term in sum $\sum_{i=1} \lambda_i^* f_i(x^*)$ must actually be 0. That is

$$
\lambda_i > 0 \implies f_i(x^*) = 0 \quad \text{and} \quad f_i(x^*) < 0 \implies \lambda_i = 0 \quad \forall i
$$

This condition is known as **complementary slackness**.

The SVM Dual Problem

SVM Lagrange Multipliers

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|w\|^2 + \frac{c}{n}\sum_{i=1}^{n}\xi_i \\ \text{subject to} & -\xi_i \leqslant 0 \quad \text{for } i=1,\ldots,n \\ & \left(1-y_i\left[w^Tx_i+b\right]\right)-\xi_i \leqslant 0 \quad \text{for } i=1,\ldots,n \end{array}$$

Lagrange Multiplier	Constraint
λ_i	$-\xi_i \leqslant 0$
α_i	$\left(1-y_i\left[w^Tx_i+b\right]\right)-\xi_i \leqslant 0$

$$L(w,b,\xi,\alpha,\lambda) = \frac{1}{2}\|w\|^2 + \frac{c}{n}\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i\left(1-y_i\left[w^Tx_i+b\right]-\xi_i\right) + \sum_{i=1}^{n}\lambda_i\left(-\xi_i\right)$$

Dual optimum value: $d^* = \sup_{\alpha,\lambda\succeq 0}\inf_{w,b,\xi} L(w,b,\xi,\alpha,\lambda)$

Strong Duality by Slater's Constraint Qualification

The SVM optimization problem:

$$
\begin{array}{ll}
\text{minimize} & \frac{1}{2}\|w\|^2 + \frac{c}{n}\sum_{i=1}^{n}\xi_i \\
\text{subject to} & -\xi_i \leqslant 0 \text{ for } i = 1,\ldots,n \\
& \left(1 - y_i\left[w^T x_i + b\right]\right) - \xi_i \leqslant 0 \text{ for } i = 1,\ldots,n
\end{array}
$$

Slater's constraint qualification:

- Convex problem + affine constraints $\Longrightarrow$ strong duality iff problem is feasible
- Do we have a feasible point?
- For SVM, we have strong duality.

SVM Dual Function: First Order Conditions

Lagrange dual function is the inf over primal variables of L:

$$g(\alpha,\lambda) = \inf_{w,b,\xi} L(w,b,\xi,\alpha,\lambda)$$

$$= \inf_{w,b,\xi}\left[\frac{1}{2}w^Tw + \sum_{i=1}^n \xi_i\left(\frac{c}{n}-\alpha_i-\lambda_i\right) + \sum_{i=1}^n \alpha_i\left(1-y_i\left[w^Tx_i+b\right]\right)\right]$$

$$\partial_w L = 0 \iff w - \sum_{i=1}^n \alpha_i y_i x_i = 0 \iff \boxed{w = \sum_{i=1}^n \alpha_i y_i x_i}$$

$$\partial_b L = 0 \iff -\sum_{i=1}^n \alpha_i y_i = 0 \iff \boxed{\sum_{i=1}^n \alpha_i y_i = 0}$$

$$\partial_{\xi_i} L = 0 \iff \frac{c}{n} - \alpha_i - \lambda_i = 0 \iff \boxed{\alpha_i + \lambda_i = \frac{c}{n}}$$

SVM Dual Function

- Substituting these conditions back into L, the second term disappears.
- First and third terms become

$$\begin{aligned}
\frac{1}{2}w^T w &= \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \\
\sum_{i=1}^{n} \alpha_i (1 - y_i \left[w^T x_i + b\right]) &= \sum_{i=1}^{n} \alpha_i - \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_j^T x_i - b \underbrace{\sum_{i=1}^{n} \alpha_i y_i}_{=0}.
\end{aligned}$$

- Putting it together, the dual function is

$$g(\alpha, \lambda) = \begin{cases} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_j^T x_i & \begin{matrix} \sum_{i=1}^{n} \alpha_i y_i = 0 \\ \alpha_i + \lambda_i = \frac{c}{n}, \text{ all } i \end{matrix} \\ -\infty & \text{otherwise.} \end{cases}$$

SVM Dual Problem

- The **dual function** is

$$g(\alpha,\lambda)=\begin{cases}\sum_{i=1}^n \alpha_i-\frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_j y_i y_j x_j^T x_i & \substack{\sum_{i=1}^n \alpha_i y_i=0\\ \alpha_i+\lambda_i=\frac{c}{n},}\text{ all } i\\ -\infty & \text{otherwise.}\end{cases}$$

- The **dual problem** is $\sup_{\alpha,\lambda\succeq 0} g(\alpha,\lambda)$:

$$\begin{aligned}\sup_{\alpha,\lambda} \quad & \sum_{i=1}^n \alpha_i-\frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_j y_i y_j x_j^T x_i\\ \text{s.t.} \quad & \sum_{i=1}^n \alpha_i y_i=0\\ & \alpha_i+\lambda_i=\frac{c}{n} \quad \alpha_i,\lambda_i\geqslant 0,\ i=1,\ldots,n\end{aligned}$$

Insights from the Dual Problem

KKT Conditions

For convex problems, if Slater's condition is satisfied, then **KKT conditions** provide necessary and sufficient conditions for the optimal solution.

- Primal feasibility: $f_i(x) \leqslant 0 \quad \forall i$
- Dual feasibility: $\lambda \succeq 0$
- Complementary slackness: $\lambda_i f_i(x) = 0$
- First-order condition:

$$\frac{\partial}{\partial x} L(x, \lambda) = 0$$

The SVM Dual Solution

- We found the SVM dual problem can be written as:

$$
\begin{aligned}
\sup_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_j^T x_i \\
\text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\
& \alpha_i \in \left[0, \frac{c}{n}\right] \ \ i = 1, \ldots, n.
\end{aligned}
$$

- Given solution α^* to dual, primal solution is $w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$.
- The solution is in the space spanned by the inputs.
- Note $\alpha_i^* \in [0, \frac{c}{n}]$. So c controls max weight on each example. (**Robustness**!)
    - What's the relation between c and regularization?

Complementary Slackness Conditions

- Recall our primal constraints and Lagrange multipliers:

Lagrange Multiplier	Constraint
λ_i	$-\xi_i \leqslant 0$
α_i	$(1 - y_i f(x_i)) - \xi_i \leqslant 0$

- Recall first order condition $\nabla_{\xi_i} L = 0$ gave us $\lambda_i^* = \frac{c}{n} - \alpha_i^*$.
- By strong duality, we must have **complementary slackness**:

$$\alpha_i^* \left(1 - y_i f^*(x_i) - \xi_i^*\right) = 0$$

$$\lambda_i^* \xi_i^* = \left(\frac{c}{n} - \alpha_i^*\right) \xi_i^* = 0$$

Consequences of Complementary Slackness

By strong duality, we must have **complementary slackness**.

$$\alpha_i^* \left(1 - y_i f^*(x_i) - \xi_i^*\right) = 0$$
$$\left(\frac{c}{n} - \alpha_i^*\right) \xi_i^* = 0$$

Recall "**slack variable**" $\xi_i^* = \max(0, 1 - y_i f^*(x_i))$ is the hinge loss on (x_i, y_i).

- If $y_i f^*(x_i) > 1$ then the margin loss is $\xi_i^* = 0$, and we get $\alpha_i^* = 0$.
- If $y_i f^*(x_i) < 1$ then the margin loss is $\xi_i^* > 0$, so $\alpha_i^* = \frac{c}{n}$.
- If $\alpha_i^* = 0$, then $\xi_i^* = 0$, which implies no loss, so $y_i f^*(x) \geqslant 1$.
- If $\alpha_i^* \in \left(0, \frac{c}{n}\right)$, then $\xi_i^* = 0$, which implies $1 - y_i f^*(x_i) = 0$.

Complementary Slackness Results: Summary

If α^* is a solution to the dual problem, then primal solution is

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i \quad \text{where} \alpha_i^* \in [0, \frac{c}{n}].$$

Relation between margin and example weights (α_i's):

$$\begin{aligned}
\alpha_i^* = 0 \quad &\implies \quad y_i f^*(x_i) \geqslant 1 \\
\alpha_i^* \in \left(0, \frac{c}{n}\right) \quad &\implies \quad y_i f^*(x_i) = 1 \\
\alpha_i^* = \frac{c}{n} \quad &\implies \quad y_i f^*(x_i) \leqslant 1
\end{aligned}$$

$$\begin{aligned}
y_i f^*(x_i) < 1 \quad &\implies \quad \alpha_i^* = \frac{c}{n} \\
y_i f^*(x_i) = 1 \quad &\implies \quad \alpha_i^* \in \left[0, \frac{c}{n}\right] \\
y_i f^*(x_i) > 1 \quad &\implies \quad \alpha_i^* = 0
\end{aligned}$$

Support Vectors

- If α^* is a solution to the dual problem, then primal solution is

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$$

  with $\alpha_i^* \in [0, \frac{c}{n}]$.

- The x_i's corresponding to $\alpha_i^* > 0$ are called **support vectors.**
- Few margin errors or "on the margin" examples $\implies$ sparsity in input examples.

Teaser for Kernelization

Dual Problem: Dependence on x through inner products

- SVM Dual Problem:

$$
\begin{aligned}
\sup_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_j^T x_i \\
\text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\
& \alpha_i \in \left[0, \frac{c}{n}\right] \;\; i = 1, \ldots, n.
\end{aligned}
$$

- Note that all dependence on inputs x_i and x_j is through their inner product: $\langle x_j, x_i \rangle = x_j^T x_i$.
- We can replace $x_j^T x_i$ by other products...
- This is a "kernelized" objective function.