# Support Vector Machine

Mengye Ren

(Slides credit to David Rosenberg, He He, et al.)

NYU

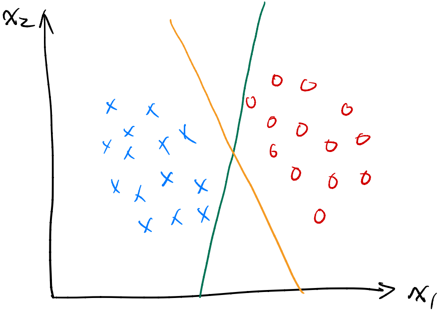
September 24, 2024

# Slides

# Maximum-Margin Separating Hyperplane

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

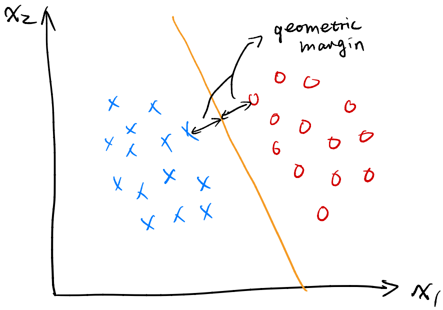
Which one do we pick?




(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**.

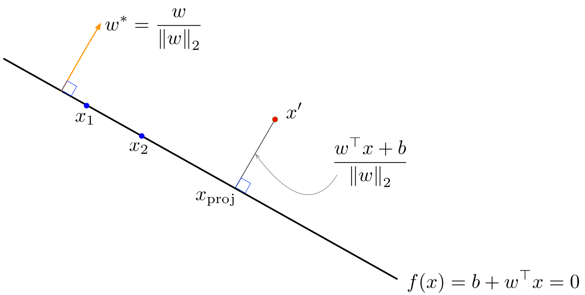
**Definition (geometric margin)**

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




- Any point on the plane $p$, and normal vector $w/\|w\|_2$
- Projection of x onto the normal: $\frac{(x'-p)^T w}{\|w\|_2}$
- $(x'-p)^T w = x'^T w - p^T w = x'^T w + b$ (since $p^T w + b = 0$)
- Signed distance between $x'$ and Hyperplane $H$: $\frac{w^T x' + b}{\|w\|_2}$
- Taking into account of the label $y$: $d(x', H) = \frac{y(w^T x' + b)}{\|w\|_2}$

# Maximize the Margin

We want to maximize the geometric margin:

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

Given separating hyperplane $H = \{v \mid w^T v + b = 0\}$, 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. The optimization finds the minimum norm solution which has a margin of at least 1 on all examples.

# Not linearly separable

What if the data is *not* linearly separable?

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

# Soft Margin SVM

Introduce **slack variables** $\xi$'s to penalize small margin:

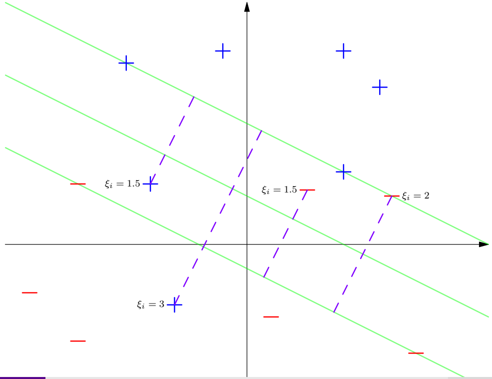
$$
\begin{array}{ll}
\text{minimize} & \frac{1}{2}\|w\|_2^2 + \frac{c}{n}\sum_{i=1}^n \xi_i \\
\text{subject to} & y_i(w^T x_i + b) \geqslant 1 - \xi_i \quad \text{for all } i \\
& \xi_i \geqslant 0 \quad \text{for all } i
\end{array}
$$

- 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 $\xi_i$ measures the violation by multiples of the geometric margin:

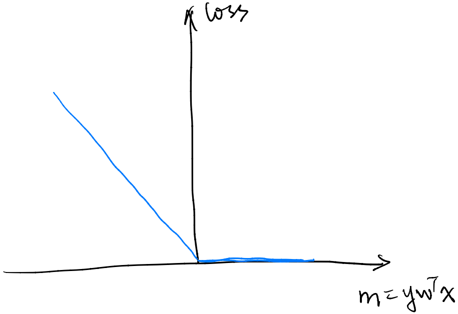
- $\xi_i = 1$: $x_i$ lies on the hyperplane
- $\xi_i = 3$: $x_i$ is past 2 margin width beyond the decision hyperplane



# Minimize the Hinge Loss

# Perceptron Loss

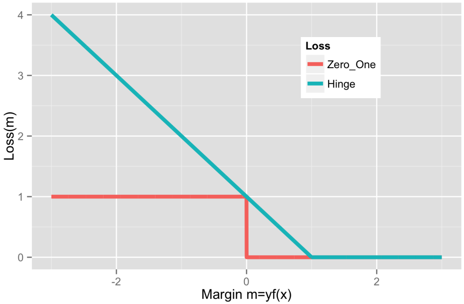
$$\ell(x, y, w) = \max(0, -yw^T x)$$




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 $m < 1$.

# SVM as an Optimization Problem

- The SVM optimization problem is equivalent to

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

which is equivalent to

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

# SVM as an Optimization Problem

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

Move the constraint into the objective:

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

- The first term is the L2 regularizer.
- The second term is the Hinge loss.

# Support Vector Machine

Using ERM:

- Hypothesis space $\mathcal{F} = \left\{ f(x) = w^T x + b \mid w \in \mathrm{R}^d, b \in \mathrm{R} \right\}$.
- $\ell_2$ regularization (Tikhonov style)
- Hinge loss $\ell(m) = \max\{1 - m, 0\} = (1 - m)_+$
- The SVM prediction function is the solution to

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

# Summary

Two ways to derive the SVM optimization problem:

- Maximize the margin
- Minimize the hinge loss with $\ell_2$ 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

# SVM Optimization Problem

- SVM objective function (letting $\lambda = \frac{1}{2c}$):

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

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

- Hinge loss: $\ell(m) = \max(0, 1 - m)$

$$\begin{aligned} \nabla_w J(w) &= \nabla_w \left( \frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i w^T x_i\right) + \lambda||w||^2 \right) \\ &= \frac{1}{n} \sum_{i=1}^{n} \nabla_w \ell\left(y_i w^T x_i\right) + 2\lambda w \end{aligned}$$

# "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 $\varepsilon$ to miss such a point?
- Does it even make sense to check $y_i w^T x_i = 1$ with floating point numbers?

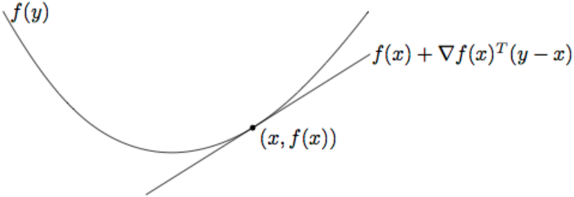
# Subgradient

# First-Order Condition for Convex, Differentiable Function

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

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

- 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 $f$.

---

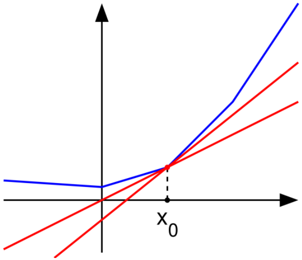
Figure from Boyd & Vandenberghe Fig. 3.2; Proof in Section 3.1.3

# 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:

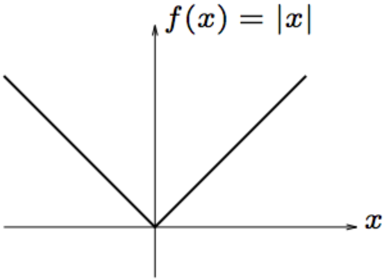
- $f$ is differentiable at $x$ iff $\partial f(x) = \{\nabla f(x)\}$.
- Subdifferential is always non-empty ($\partial f(x) = \emptyset \implies f$ is not convex)
- $x$ is the global optimum iff $0 \in \partial f(x)$.

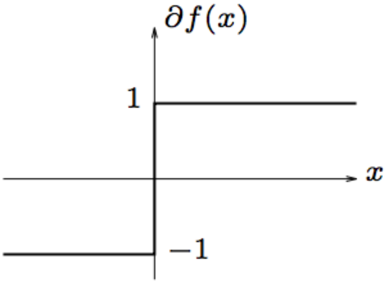
For non-convex functions:

- 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

# 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 $\eta$ 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.

- Subgradient methods are slower than gradient descent.

# Subgradient descent for SVM

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)
    $t = t + 1$
    $\eta_t = 1/(t\lambda)$;
    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
      $w_{t+1} = (1 - \eta_t \lambda) w_t$

What subgradient is the Pegasos algorithm taking?

# Summary

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

# The Dual Problem

- In addition to subgradient descent, we can directly solve the optimization problem using a Quadratic Programming (QP) solver.
- For convex optimization problem, we can also look into its **dual problem**.

# SVM as a Quadratic Program

- The SVM optimization problem is equivalent to

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

- 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 get more insights by examining 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

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

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

# 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,

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

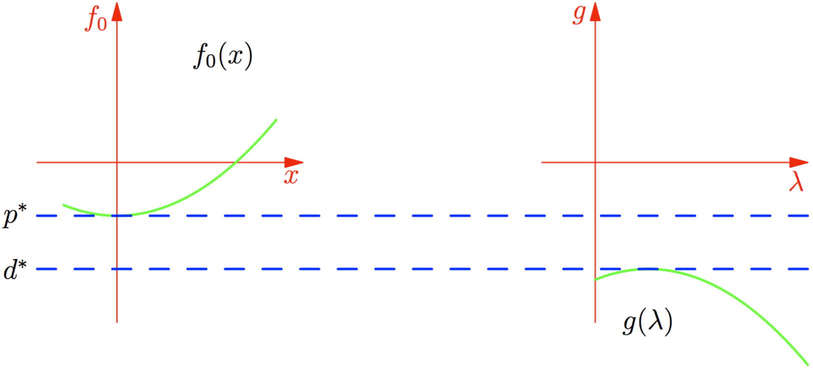
  there is a recipe for constructing a corresponding **Lagrangian dual problem:**

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

- The dual problem is always a convex optimization problem.

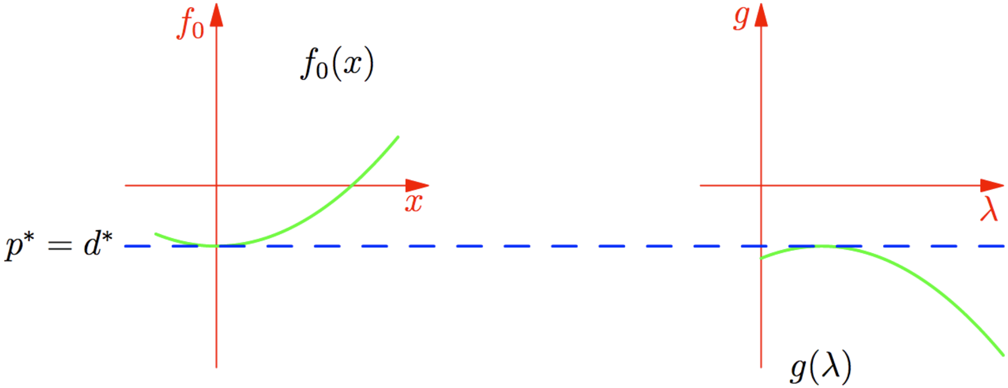
# Weak Duality

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




Plot courtesy of Brett Bernstein.

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 $\lambda^*$ 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

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

| Lagrange Multiplier | Constraint |
|---|---|
| $\lambda_i$ | -$\xi_i \leqslant 0$ |
| $\alpha_i$ | $\left(1 - y_i\left[w^T x_i + b\right]\right) - \xi_i \leqslant 0$ |

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

# Strong Duality by Slater's Constraint Qualification

The SVM optimization problem:

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

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 $L$:

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

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

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

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

# SVM Dual Function

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

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

- Putting it together, the dual function is

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

# SVM Dual Problem

- The **dual function** is

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

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

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

# Insights from the Dual Problem

# 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 $\alpha^*$ 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 |
|---|---|
| $\lambda_i$ | $-\xi_i \leqslant 0$ |
| $\alpha_i$ | $(1-y_i f(x_i))-\xi_i \leqslant 0$ |

- Recall first order condition $\nabla_{\xi_i} L = 0$ gave us $\lambda_i^* = \frac{c}{n} - \alpha_i^*$.

# 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 $1-y_i f^*(x_i)=0$.

# Support Vectors

- If $\alpha^*$ 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 $\Longrightarrow$ sparsity in input examples.

# Dual Problem: Dependence on $x$ through inner products

- SVM Dual Problem:

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

- 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.

# Feature Maps

# The Input Space $\mathcal{X}$

- Our general learning theory setup: no assumptions about $\mathcal{X}$
- But $\mathcal{X} = \mathbf{R}^d$ for the specific methods we've developed:
  - Ridge regression
  - Lasso regression
  - Support Vector Machines
- Our hypothesis space for these was all affine functions on $\mathbf{R}^d$:

$$\mathcal{F} = \left\{ x \mapsto w^T x + b \mid w \in \mathbf{R}^d, b \in \mathbf{R} \right\}.$$

- What if we want to do prediction on inputs not natively in $\mathbf{R}^d$?

# The Input Space $\mathcal{X}$

- Often want to use inputs not natively in $\mathbb{R}^d$:
  - Text documents
  - Image files
  - Sound recordings
  - DNA sequences
- They may be represented in numbers, but...
- The $i$th entry of each sequence should have the same "meaning"
- All the sequences should have the same length

**Definition**

Mapping an input from $\mathcal{X}$ to a vector in $\mathrm{R}^d$ is called **feature extraction** or **featurization**.

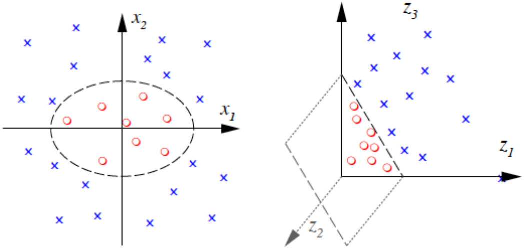
# Linear Models with Explicit Feature Map

- Input space: $\mathcal{X}$ (no assumptions)
- Introduce **feature map** $\phi : \mathcal{X} \to \mathbb{R}^d$
- The feature map maps into the **feature space** $\mathbb{R}^d$.
- Hypothesis space of affine functions on feature space:

$$\mathcal{F} = \left\{x \mapsto w^T \phi(x) + b \mid w \in \mathbb{R}^d, b \in \mathbb{R}\right\}.$$

# Geometric Example: Two class problem, nonlinear boundary

$$\Phi : R^2 \to R^3$$
$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{(}2)x_1x_2, x_2^2)$$




- With identity feature map $\phi(x) = (x_1, x_2)$ and linear models, can't separate regions
- With appropriate featurization $\phi(x) = \left(x_1, x_2, x_1^2 + x_2^2\right)$, becomes linearly separable .
- Video: http://youtu.be/3liCbRZPrZA

# Expressivity of Hypothesis Space

- For linear models, to grow the hypothesis spaces, we must add features.
- Sometimes we say a larger hypothesis is more expressive.
    - (can fit more relationships between input and action)
- Many ways to create new features.

# Handling Nonlinearity with Linear Methods

# Example Task: Predicting Health

- General Philosophy: Extract every feature that might be relevant
- Features for medical diagnosis
  - height
  - weight
  - body temperature
  - blood pressure
  - etc...

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

# Feature Issues for Linear Predictors

- For linear predictors, it's important **how** features are added
  - The relation between a feature and the label may not be linear
  - There may be complex dependence among features
- Three types of nonlinearities can cause problems:
  - Non-monotonicity
  - Saturation
  - Interactions between features

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

# Non-monotonicity: The Issue

- Feature Map: $\phi(x) = [1, \text{temperature}(x)]$
- Action: Predict health score $y \in \mathbf{R}$ (positive is good)
- Hypothesis Space $\mathcal{F}$={affine functions of temperature}
- Issue:
  - Health is not an affine function of temperature.
  - Affine function can either say
    - Very high is bad and very low is good, or
    - Very low is bad and very high is good,
    - But here, both extremes are bad.

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

# Non-monotonicity: Solution 1

- Transform the input:
$$\phi(x) = \left[1, \{\text{temperature(x)-37}\}^2\right],$$
where 37 is "normal" temperature in Celsius.

- Ok, but requires manually-specified domain knowledge
  - Do we really need that?
  - What does $w^T\phi(x)$ look like?

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

# Non-monotonicity: Solution 2

- Think less, put in more:

$$\phi(x) = \left[1, \text{temperature}(x), \{\text{temperature}(x)\}^2\right].$$

- More expressive than Solution 1.

### General Rule

Features should be simple building blocks that can be pieced together.

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

# Saturation: The Issue

- Setting: Find products relevant to user's query
- Input: Product $x$
- Output: Score the relevance of $x$ to user's query
- Feature Map:
$$\phi(x) = [1, N(x)],$$
where $N(x) =$ number of people who bought $x$.
- We expect a monotonic relationship between $N(x)$ and relevance, but also expect diminishing return.

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

# Saturation: Solve with nonlinear transform

- Smooth nonlinear transformation:

$$\phi(x) = [1, \log\{1 + N(x)\}]$$

  - $\log(\cdot)$ good for values with large dynamic ranges

- Discretization (a discontinuous transformation):

$$\phi(x) = (\mathbb{1}[0 \leqslant N(x) < 10], \mathbb{1}[10 \leqslant N(x) < 100], \ldots)$$

  - Small buckets allow quite flexible relationship

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

# Interactions: The Issue

- Input: Patient information $x$
- Action: Health score $y \in R$ (higher is better)
- Feature Map

$$\phi(x) = [\text{height}(x), \text{weight}(x)]$$

- Issue: It's the weight *relative* to the height that's important.
- Impossible to get with these features and a linear classifier.
- Need some **interaction** between height and weight.

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

# Interactions: Approach 1

- Google "ideal weight from height"
- J. D. Robinson's "ideal weight" formula:

$$\text{weight(kg)} = 52 + 1.9\,[\text{height(in)} - 60]$$

- Make score square deviation between height($h$) and ideal weight($w$)

$$f(x) = (52 + 1.9\,[h(x) - 60] - w(x))^2$$

- WolframAlpha for complicated Mathematics:

$$f(x) = 3.61h(x)^2 - 3.8h(x)w(x) - 235.6h(x) + w(x)^2 + 124w(x) + 3844$$

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

# Interactions: Approach 2

- Just include all second order features:

$$\phi(x) = \left[1, h(x), w(x), h(x)^2, w(x)^2, \underbrace{h(x)w(x)}_{\text{cross term}}\right]$$

- More flexible, no Google, no WolframAlpha.

**General Principle**

Simpler building blocks replace a single "smart" feature.

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.