ITCS 6156/8156 Fall 2023
Machine Learning

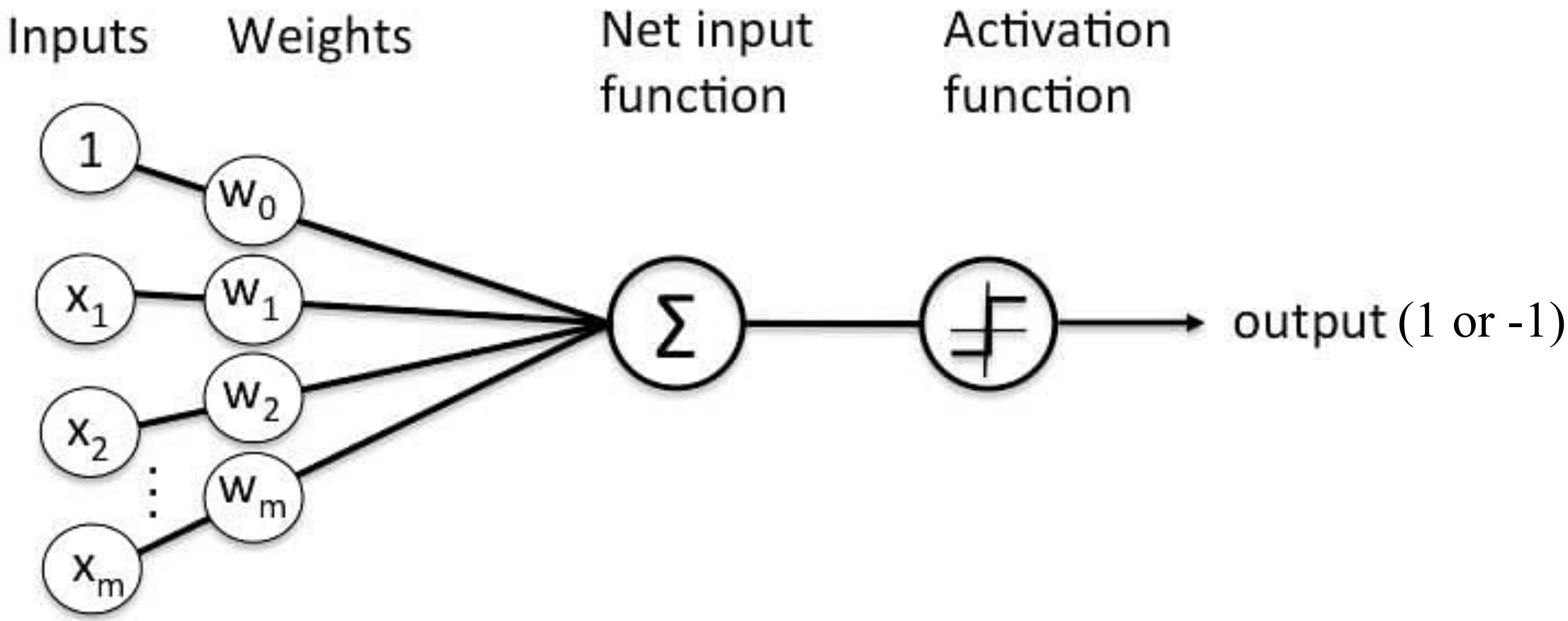
Support Vector Machine

Instructor: Hongfei Xue
Email: hongfei.xue@charlotte.edu
Class Meeting: Mon & Wed, 4:00 PM – 5:15 PM, CHHS 376

UNIVERSITY OF NORTH CAROLINA CHARLOTTE

Some content in the slides is based on Dr. Varun's lecture

Perceptron



- $$h_{\mathbf{w}}(X) = \mathbf{w}^T X = [w_0, w_1, \dots, w_d]^T [1, x, \dots, x_d]$$
  $$= w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d$$
- If $h_{\mathbf{w}}(X) > 0$, output will be 1; otherwise, output will be -1
- Activation function is $sign(z)$:

$$sign(z) = \begin{cases} 1, & if\ z > 0 \\ -1, & otherwise \end{cases}$$

Training

- Training algorithm:

  initialize parameters $\mathbf{w} = 0$

  for $n = 1 \dots \mathrm{N}$

  $\quad h_n = \mathbf{w}^{\mathrm{T}} \mathbf{x}_n$

  $\quad$ **if** $h_{\mathrm{n}} \geq 0$ and $t_n = -1$

  $\qquad \mathbf{w} = \mathbf{w} - \mathbf{x}_n$

  $\quad$ **if** $h_{\mathrm{n}} \leq 0$ and $t_n = +1$

  $\qquad \mathbf{w} = \mathbf{w} + \mathbf{x}_n$

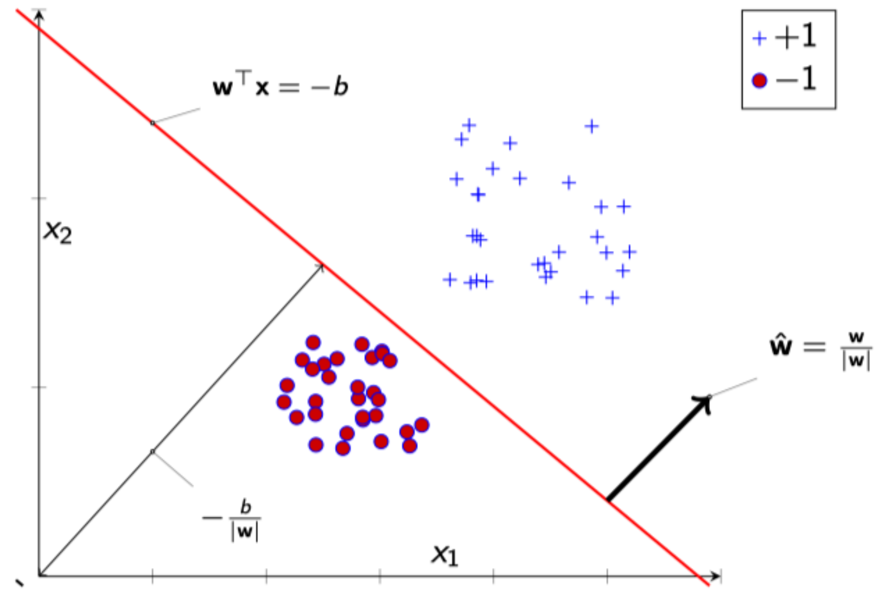
Repeat:
- until converge
- for a number of epochs

- Theorem:
  - If the training dataset is linearly separable, the perceptron learning algorithm is **guaranteed** to find a solution in a finite number of steps.

Maximum Margin Classifiers

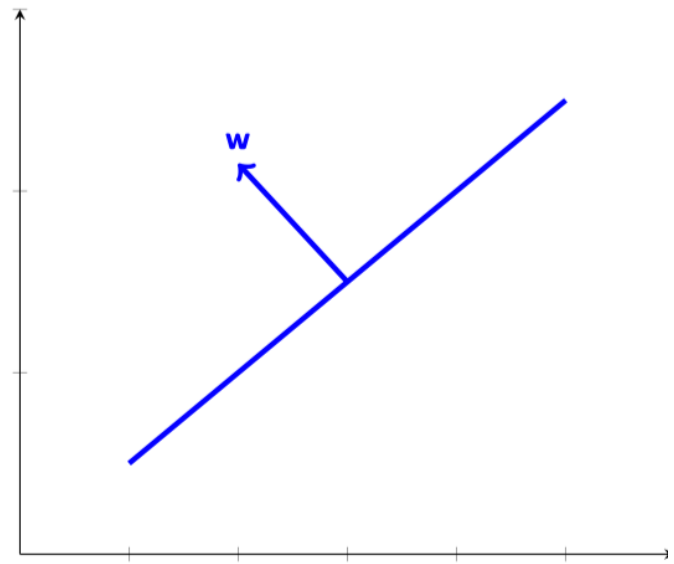
$$y = \mathbf{w}^\top \mathbf{x} + b$$

- Remember the Perceptron!
- If data is linearly separable
  - Perceptron training guarantees learning the decision boundary
- There can be other boundaries
  - Depends on initial value for $\mathbf{w}$
- **But what is the best boundary?**



Linear Hyperplane

- Separates a D-dimensional space into two half-spaces
- Defined by $\mathbf{w} \in \Re^D$
  - *Orthogonal* to the hyperplane
  - This $\mathbf{w}$ goes through the origin
  - How do you check if a point lies "above" or "below" $\mathbf{w}$?
  - What happens for points **on** $\mathbf{w}$?
- Add a bias b
- How to check if point lies above or below $\mathbf{w}$?
  - If $\mathbf{w}^\top \mathbf{x} + b > 0$ then $\mathbf{x}$ is *above*
  - Else, *below*



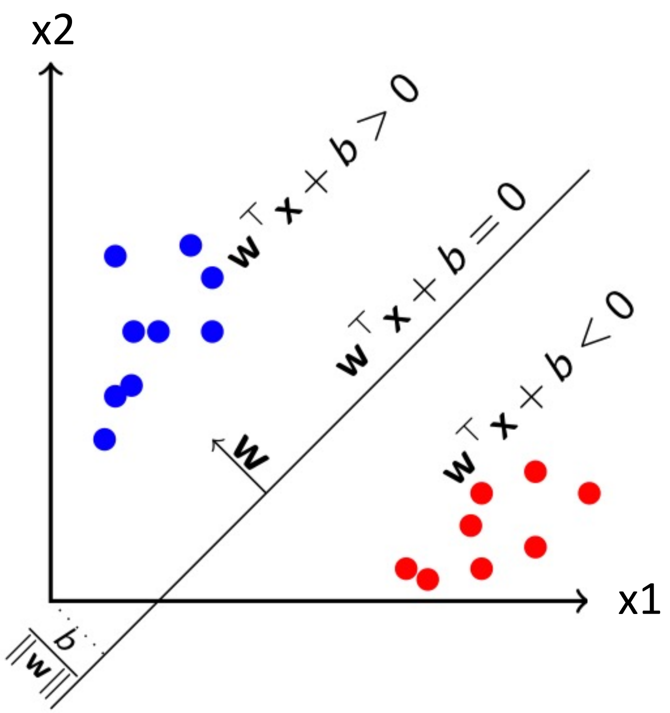
Line as a Decision Surface

- Decision boundary represented by the hyperplane $\mathbf{w}$
- For binary classification, $\mathbf{w}$ points **towards** the positive class

Decision Rule

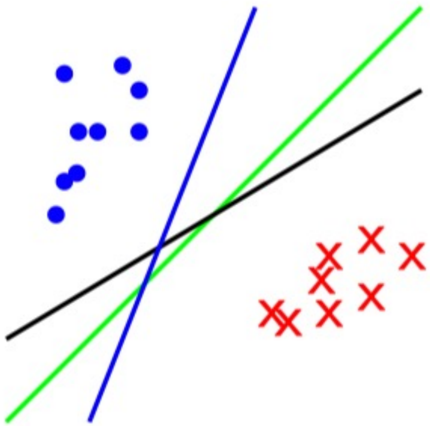
$$y = sign(\mathbf{w}^\top \mathbf{x} + b)$$

- $\mathbf{w}^\top \mathbf{x} + b > 0 \Rightarrow y = +1$
- $\mathbf{w}^\top \mathbf{x} + b < 0 \Rightarrow y = -1$



Best Hyperplane Separator

- **Perceptron** can find a hyperplane that separates the data
  - . . . if the data is linearly separable
- But there can be many choices!
- Find the one with best separability (largest margin)
- Gives better generalization performance



Concept of Margin

- **Margin** is the distance between an example and the decision line
- Denoted by γ
- For a positive point:

$$\gamma = \frac{\mathbf{w}^\top \mathbf{x} + b}{||\mathbf{w}||}$$

- For a negative point:

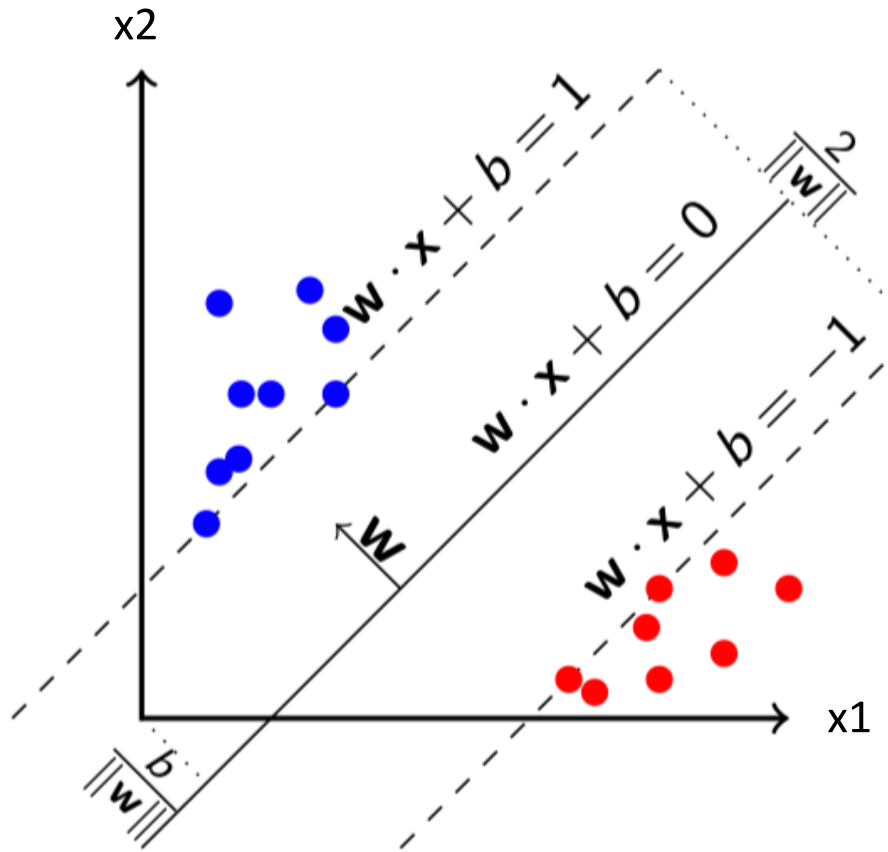
$$\gamma = -\frac{\mathbf{w}^\top \mathbf{x} + b}{||\mathbf{w}||}$$

Functional Interpretation

- Margin **positive** if prediction is **correct**; **negative** if prediction is **incorrect**

Maximum Margin Principle

- Figure after normalization:



From the figure one can note that the size of the margin is $\frac{2}{\|\mathbf{w}\|}$. We can show this as follows. Since the data is separable, we can get two parallel lines represented by $\mathbf{w}^\top\mathbf{x}+b=+1$ and $\mathbf{w}^\top\mathbf{x}+b=-1$. Using result from (1) and (2), the distance between the two lines is given by $2\gamma=\frac{2}{\|\mathbf{w}\|}$.

Support Vector Machines

- A hyperplane based classifier defined by $\mathbf{w}$ and b
- Like perceptron
- Find hyperplane with *maximum separation margin* on the training data
- Assume that data is linearly separable (will relax this later)
  - Zero training error (loss)

SVM Prediction Rule

$$y = sign(\mathbf{w}^\top \mathbf{x} + b)$$

SVM Learning

- **Input**: Training data $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$
- **Objective**: Learn $\mathbf{w}$ and b that maximizes the margin

SVM Learning

- SVM learning task as an optimization problem
- Find $\mathbf{w}$ and b that gives zero training error
- Maximizes the margin $(= \frac{2}{\|w\|})$
- Same as minimizing $\|\mathbf{w}\|$

Optimization Formulation

$$\begin{aligned} \underset{\mathbf{w},b}{\text{minimize}} \quad & \frac{\|\mathbf{w}\|^2}{2} \\ \text{subject to} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \; i = 1, \ldots, n. \end{aligned}$$

- **Optimization** with N linear inequality constraint

A Different Interpretation of Margin

- What impact does the margin have on $\mathbf{w}$?
- Large margin $\Rightarrow$ Small $\|\mathbf{w}\|$
- Small $\|\mathbf{w}\|$ $\Rightarrow$ regularized/simple solutions
- Simple solutions $\Rightarrow$ Better generalizability (*Occam's Razor*)

Optimization Problem

Optimization Formulation

$$\begin{aligned} \underset{\mathbf{w},b}{\text{minimize}} \quad & \frac{\|\mathbf{w}\|^2}{2} \\ \text{subject to} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \; i = 1, \ldots, n. \end{aligned}$$

- There is an quadratic objective function to minimize with N inequality constraints
- "Off-the-shelf" packages - quadprog (MATLAB), CVXOPT
- Is that the best way?

An Optimization Problem

- An optimization problem without constraint:

$$\boxed{\underset{x,y}{\text{minimize}} \quad f(x,y) = \quad x^2 + 2y^2 - 2}$$

- An optimization problem with constraint:

$$\boxed{\begin{array}{lll} \underset{x,y}{\text{minimize}} & f(x,y) = & x^2 + 2y^2 - 2 \\ \text{subject to} & h(x,y) = & x + y - 1 = 0. \end{array}}$$

An Optimization Problem

- Tool for solving constrained optimization problems of differentiable functions

$$\begin{aligned} \underset{x,y}{\text{minimize}} \quad & f(x,y) = \quad x^2 + 2y^2 - 2 \\ \text{subject to} \quad & h(x,y) : \quad x + y - 1 = 0. \end{aligned}$$

- A Lagrangian multiplier (β) lets you combine the two equations into one

$$\underset{x,y,\beta}{\text{minimize}} \quad L(x,y,\beta) = \quad f(x,y) + \beta h(x,y)$$

An Optimization Problem

Solution 1. *Writing the objective as Lagrangian.*

$$L(x, y, \beta) = x^2 + 2y^2 - 2 + \beta(x + y - 1)$$

Setting the gradient to 0 with respect to x, y and β will give us the optimal values.

$$\frac{\partial L}{\partial x} = 2x + \beta = 0$$

$$\frac{\partial L}{\partial y} = 4y + \beta = 0$$

$$\frac{\partial L}{\partial \beta} = x + y - 1 = 0$$

Multiple Constraints

$$\begin{aligned}
\underset{x,y,z}{\text{minimize}} \quad & f(x,y,z) = & x^2 + 4y^2 + 2z^2 + 6y + z \\
\text{subject to} \quad & h_1(x,y,z): & x + z^2 - 1 = 0 \\
& h_2(x,y,z): & x^2 + y^2 - 1 = 0.
\end{aligned}$$

$$L(x,y,z,\boldsymbol{\beta}) = f(x,y,z) + \sum_i \beta_i h_i(x,y,z)$$

Handling Inequality Constraints

$$
\boxed{\begin{array}{lcr} \underset{x,y}{\text{minimize}} & f(x,y) = & x^3 + y^2 \\ \text{subject to} & g(x): & x^2 - 1 \leq 0. \end{array}}
$$

- Inequality constraints are **transferred** as constraints on the Lagrangian, α

The Lagrangian in the above example becomes:

$$
\begin{aligned}
L(x, y, \alpha) &= f(x, y) + \alpha g(x, y) \\
&= x^3 + y^2 + \alpha(x^2 - 1)
\end{aligned}
$$

Handling Inequality Constraints

Solving for the gradient of the Lagrangian gives us:

$$\frac{\partial}{\partial x} L(x, y, \alpha) = 3x^2 + 2\alpha x = 0$$
$$\frac{\partial}{\partial y} L(x, y, \alpha) = 2y = 0$$
$$\frac{\partial}{\partial \alpha_1} L(x, y, \alpha) = x^2 - 1 = 0$$

Furthermore we require that:

$$\alpha \geq 0$$

From above equations we get $y = 0$, $x = \pm 1$ and $\alpha = \pm\frac{3}{2}$. But since $\alpha \geq 0$, hence $\alpha = \frac{3}{2}$. This gives $x = 1$, $y = 0$, and $f = 1$.

Generalized Lagrangian

Handling Both Types of Constraints

$$
\begin{array}{|lll|}
\hline
\underset{\mathbf{w}}{\text{minimize}} & f(\mathbf{w}) & \\
\text{subject to} & g_i(\mathbf{w}) \leq 0 & i = 1, \ldots, k \\
\text{and} & h_i(\mathbf{w}) = 0 & i = 1, \ldots, l. \\
\hline
\end{array}
$$

Generalized Lagrangian

$$L(\mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\boldsymbol{w}) + \sum_{i=1}^{k} \alpha_i g_i(\mathbf{w}) + \sum_{i=1}^{l} \beta_i h_i(\mathbf{w})$$

subject to, $\alpha_i \geq 0, \forall i$

Primal and Dual Formulations

Primal Optimization

- Let θ_P be defined as:

$$\theta_P(\mathbf{w}) = \underset{\boldsymbol{\alpha},\boldsymbol{\beta}:\alpha_i \geq 0}{max} L(\mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

- One can prove that the optimal value for the original constrained problem is same as:

$$p^* = \underset{\mathbf{w}}{min} \theta_P(\mathbf{w}) = \underset{\mathbf{w}}{min} \underset{\boldsymbol{\alpha},\boldsymbol{\beta}:\alpha_i \geq 0}{max} L(\mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

Consider

$$\begin{aligned}
\theta_P(\mathbf{w}) &= \underset{\boldsymbol{\alpha},\boldsymbol{\beta}:\alpha_i \geq 0}{max} L(\mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \\
&= \underset{\boldsymbol{\alpha},\boldsymbol{\beta}:\alpha_i \geq 0}{max} f(\mathbf{w}) + \sum_{i=1}^{k} \alpha_i g_i(\mathbf{w}) + \sum_{i-1}^{l} \beta_i h_i(\mathbf{w})
\end{aligned}$$

It is easy to show that if any constraints are not satisfied, i.e., if either $g_i(\mathbf{w}) > 0$ or $h_i(\mathbf{w}) \neq 0$, then $\theta_P(\mathbf{w}) = \infty$. Which means that:

$$\theta_P(\mathbf{w}) = \begin{cases} f(\mathbf{w}) & \text{if primal constraints are satisfied} \\ \infty & \text{otherwise,} \end{cases}$$

A Toy Example

Optimization Formulation

$$\underset{\mathbf{w},b}{\text{minimize}} \quad \frac{\|\mathbf{w}\|^2}{2}$$

$$\text{subject to} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \; i = 1, \ldots, n.$$

A Toy Example

- $\mathbf{x} \in \Re^2$
- Two training points:

$$\mathbf{x}_1, y_1 = (1, 1), -1$$

$$\mathbf{x}_2, y_2 = (2, 2), +1$$

- Find the best hyperplane $\mathbf{w} = (w_1, w_2)$

A Toy Example

Optimization problem for the toy example

$$\begin{array}{lrl} \underset{\mathbf{w}}{\text{minimize}} & f(\mathbf{w}) = & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} & g_1(\mathbf{w}, b) = & y_1(\mathbf{w}^\top \mathbf{x}_1 + b) - 1 \geq 0 \\ & g_2(\mathbf{w}, b) = & y_2(\mathbf{w}^\top \mathbf{x}_2 + b) - 1 \geq 0. \end{array}$$

- Substituting actual values for $\mathbf{x}_1, y_1$ and $\mathbf{x}_2, y_2$.

$$\begin{array}{lrl} \underset{\mathbf{w}}{\text{minimize}} & f(\mathbf{w}) = & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} & g_1(\mathbf{w}, b) = & -(\mathbf{w}^\top \mathbf{x}_1 + b) - 1 \geq 0 \\ & g_2(\mathbf{w}, b) = & (\mathbf{w}^\top \mathbf{x}_2 + b) - 1 \geq 0. \end{array}$$

The above problem can be also written as:

$$\begin{array}{lrl} \underset{w_1, w_2, b}{\text{minimize}} & f(w_1, w_2) = & \frac{1}{2}(w_1^2 + w_2^2) \\ \text{subject to} & g_1(w_1, w_2, b) = & -(w_1 + w_2 + b) - 1 \geq 0 \\ & g_2(w_1, w_2, b) = & (2w_1 + 2w_2 + b) - 1 \geq 0. \end{array}$$

A Toy Example

To solve the toy optimization problem, we rewrite it in the Lagrangian form:

$$L(w_1, w_2, b, \alpha) \quad = \quad \frac{1}{2}(w_1^2 + w_2^2) + \alpha_1(w_1 + w_2 + b + 1) - \alpha_2(2w_1 + 2w_2 + b - 1)$$

Setting $\nabla L = 0$, we get:

$$\begin{aligned}
\frac{\partial}{\partial w_1} L(w_1, w_2, b, \alpha) &= w_1 + \alpha_1 - 2\alpha_2 = 0 \\
\frac{\partial}{\partial w_2} L(w_1, w_2, b, \alpha) &= w_2 + \alpha_1 - 2\alpha_2 = 0 \\
\frac{\partial}{\partial b} L(w_1, w_2, b, \alpha) &= \alpha_1 - \alpha_2 = 0 \\
\frac{\partial}{\partial \alpha_1} L(w_1, w_2, b, \alpha) &= w_1 + w_2 + b + 1 = 0 \\
\frac{\partial}{\partial \alpha_2} L(w_1, w_2, b, \alpha) &= 2w_1 + 2w_2 + b - 1 = 0
\end{aligned}$$

Solving the above equations, we get, $w_1 = w_2 = 1$ and $b = -3$.

Primal and Dual Formulations

Dual Optimization

- Consider θ_D, defined as:

$$\theta_D(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \underset{\mathbf{w}}{min} L(\mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

- The **dual** optimization problem can be posed as:

$$d^* = \underset{\boldsymbol{\alpha},\boldsymbol{\beta}:\alpha_i \geq 0}{max} \theta_D(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \underset{\boldsymbol{\alpha},\boldsymbol{\beta}:\alpha_i \geq 0}{max} \underset{\mathbf{w}}{min} L(\mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

$d^* == p^*$**?**

- Note that $d^* \leq p^*$
- "Max min" of a function is always less than or equal to "Min max"
- When will they be equal?
  - $f(\mathbf{w})$ is convex
  - Constraints are affine

Why Dual Formulations?

- For the primal optimization problem, directly differentiating the Lagrangian is one approach. However, this can lead to a **non-convex optimization problem**.
- The dual form is a **convex quadratic programming problem**. There are many mature optimization algorithms available for this kind of problem that can find the **global optimum**.
- The dual form provides insights into the structure of the data, especially regarding the properties of the **support vectors**.

Relation between Primal and Dual

- In general $d^* \le p^*$, for SVM optimization the equality holds
- Certain conditions should be true
- Known as the **Kahrun-Kuhn-Tucker** conditions
- For $d^* = p^* = L(\mathbf{w}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$:

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) &= 0 \\
\frac{\partial}{\partial \beta_i} L(\mathbf{w}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) &= 0, \quad i = 1, \ldots, l \\
\alpha_i^* g_i(\mathbf{w}^*) &= 0, \quad i = 1, \ldots, k \\
g_i(\mathbf{w}^*) &\le 0, \quad i = 1, \ldots, k \\
\alpha_i^* &\ge 0, \quad i = 1, \ldots, k
\end{aligned}$$

Review

Optimization Formulation

$$\begin{aligned}&\underset{\mathbf{w},b}{\text{minimize}} \quad \frac{\|\mathbf{w}\|^2}{2}\\&\text{subject to} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \; i = 1, \ldots, n.\end{aligned}$$

- Introducing Lagrange Multipliers,α_i, $i = 1, \ldots, n$

Rewriting as a (primal) Lagrangian

$$\begin{aligned}&\underset{\mathbf{w},b,\alpha}{\text{minimize}} \quad L_P(\mathbf{w}, b, \alpha) = \frac{\|\mathbf{w}\|^2}{2} + \sum_{i=1}^{n} \alpha_i \{1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\}\\&\text{subject to} \quad \alpha_i \geq 0 \; i = 1, \ldots, n.\end{aligned}$$

Solving the Lagrangian

- Set gradient of L_P to 0

$$\frac{\partial L_P}{\partial \mathbf{w}} = 0 \Rightarrow \boxed{\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i}$$

$$\frac{\partial L_P}{\partial b} = 0 \Rightarrow \boxed{\sum_{i=1}^{n} \alpha_i y_i = 0}$$

- Substituting in L_P to get the dual L_D

Review

Dual Lagrangian Formulation

$$\underset{b,\alpha}{\text{maximize}} \quad L_D(\mathbf{w}, b, \alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{m,n=1}^{N} \alpha_m \alpha_i y_m y_i (\mathbf{x}_m^\top \mathbf{x}_i)$$

$$\text{subject to} \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \alpha_i \geq 0 \; i = 1, \ldots, n.$$

- Dual Lagrangian is a *quadratic programming problem* in α_i's
  - Use "off-the-shelf" solvers

Investigating Kahrun Kuhn Tucker Conditions

- For the primal and dual formulations
- We can optimize the dual formulation (as shown earlier)
- Solution should satisfy the **Karush-Kuhn-Tucker** (KKT) Conditions

KKT Conditions

$$
\begin{aligned}
\frac{\partial}{\partial \mathbf{w}} L_P(\mathbf{w}, b, \alpha) &= \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0 && (1)\\
\frac{\partial}{\partial b} L_P(\mathbf{w}, b, \alpha) &= -\sum_{i=1}^{n} \alpha_i y_i = 0 && (2)\\
1 - y_i\{\mathbf{w}^\top \mathbf{x}_i + b\} &\leq 0 && (3)\\
\alpha_i &\geq 0 && (4)\\
\alpha_i(1 - y_i\{\mathbf{w}^\top \mathbf{x}_i + b\}) &= 0 && (5)
\end{aligned}
$$

- Use KKT condition #5
- For $\alpha_i > 0$

$$(y_i\{\mathbf{w}^\top \mathbf{x}_i + b\} - 1) = 0$$

- Which means that:

$$b = -\frac{\max\limits_{n:y_i=-1} \mathbf{w}^\top \mathbf{x}_i + \min\limits_{n:y_i=1} \mathbf{w}^\top \mathbf{x}_i}{2}$$

Support Vectors

Most α_i's are 0

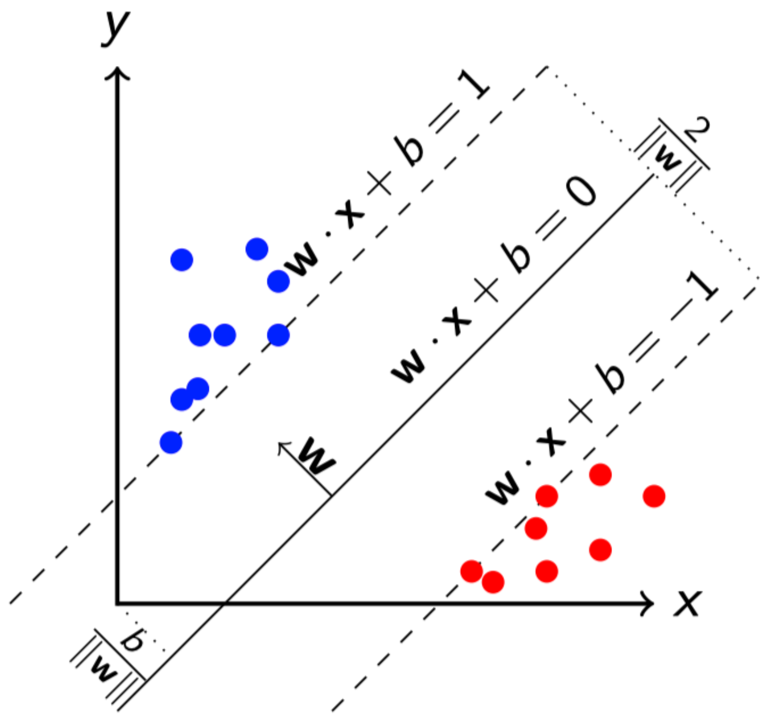
- KKT condition #5:

$$\alpha_i(1 - y_i\{\mathbf{w}^\top \mathbf{x}_i + b\}) = 0$$

- If $\mathbf{x}_i$ **not** on margin

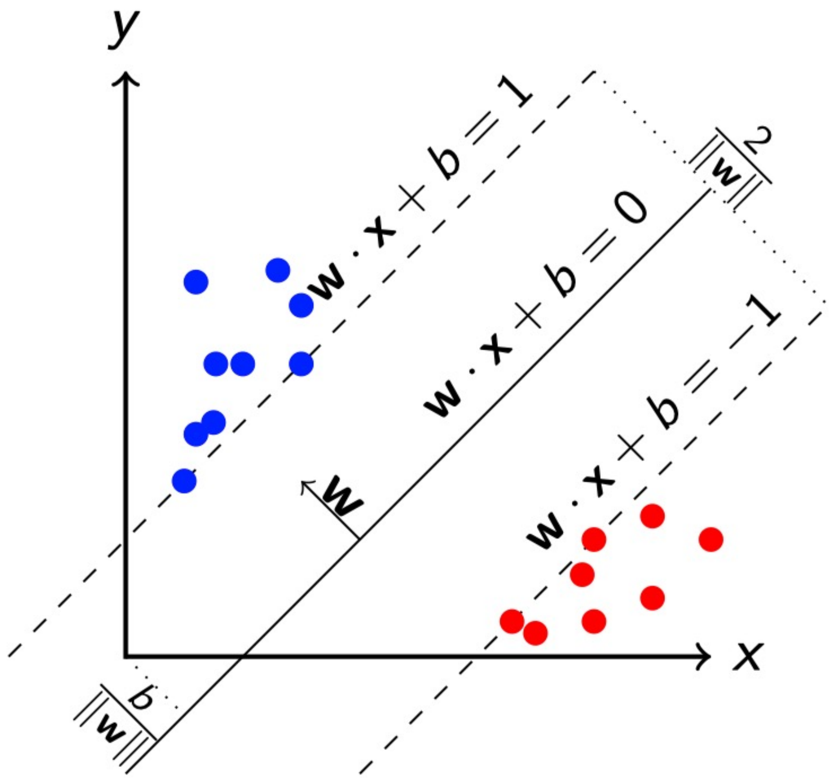
$$y_i\{\mathbf{w}^\top \mathbf{x}_i + b\} > 1$$
$$\Rightarrow \qquad \alpha_i = 0$$

- $\alpha_i \neq 0$ only for $\mathbf{x}_i$ on margin
- These are the **support vectors**
- Only need these for prediction



SVM for Linearly Separable Data

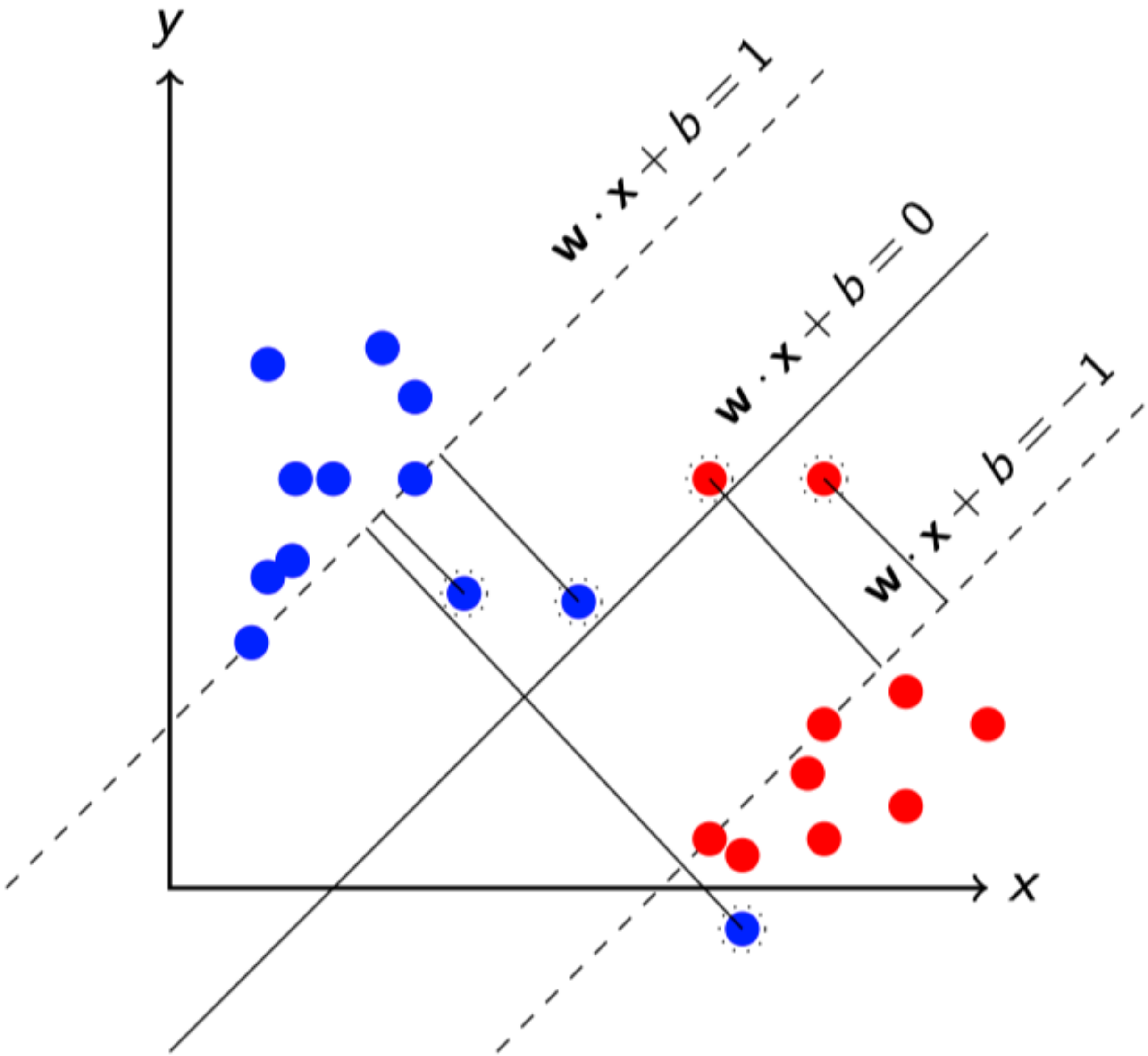
- For linearly separable data, SVM learns a weight vector **w**
- Maximizes the margin
- SVM training is a **constrained optimization problem**
  - Each training example should lie outside the margin
  - N constraints



What if data is not linearly separable?

- Cannot go for zero training error
- Still learn a maximum margin hyperplane
  1. Allow some examples to be misclassified
  2. Allow some examples to fall **inside** the margin
- How do you set up the optimization for SVM training

Cutting Some Slack



Slack Variables

- **Separable Case**: To ensure zero training loss, constraint was

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \quad \forall i = 1 \ldots n$$

- **Non-separable Case**: Relax the constraint

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i = 1 \ldots n$$

- ξ_i is called **slack variable** ($\xi_i \geq 0$)
- For misclassification, $\xi_i > 1$

Relaxing the Constraint

- It is OK to have some misclassified training examples
  - Some ξ_i's will be non-zero
- Minimize the number of such examples
  - Minimize $\boxed{\sum_{i=1}^{n} \xi_i}$
- Optimization Problem for Non-Separable Case

$$\boxed{\begin{array}{ll} \underset{\mathbf{w},b}{\text{minimize}} & f(\mathbf{w}, b) = \|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \\ \text{subject to} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0 \; i = 1, \ldots, n. \end{array}}$$

- C controls the impact of margin and the margin error.

Parameter C

- C determining the trade-off between maximizing the margin and minimizing the classification error on the training data.

  - High C:
    - more emphasis on correctly classifying the training examples, even at the risk of overfitting.
    - less tolerant of noise and outliers.
    - result in a smaller margin.

  - Low C:
    - prioritize maximizing the margin, which might lead to some misclassifications on the training set (or even potentially underfit the data.
    - more tolerant of noise and outliers.
    - result in a wider margin.

Estimating Weights

- Similar optimization procedure as for the separable case (QP for the dual)
- Weights have the same expression

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

- Support vectors are slightly different
  1. Points on the margin ($\xi_i = 0$)
  2. Inside the margin but on the correct side ($0 < \xi_i < 1$)
  3. On the wrong side of the hyperplane ($\xi_i \geq 1$)

Questions?

UNIVERSITY OF NORTH CAROLINA
CHARLOTTE