CSE417T – Lecture 20

- Please **mute** yourself and **turn off videos** to save bandwidth.

- If you have questions during the lecture
  - Use chatrooms to post your questions
    - I’ll review chatrooms in batches
  - You can also un-mute yourself and ask the questions directly
- The slides are posted on the course website

- RECORD THE LECTURE!
  - Please remind me if I forget to do so.

Logistics: Homework

- Homework 4 will be due April 13 (Monday)
  - Please start it early
    - It was on average the most time consuming assignment for students in the past
  - Keep track of your own late days
    - Gradescope doesn't allow separate deadlines
    - Your submissions won't be graded if you exceed the late-day limit
    - Up to 3 late days can be used if you still have late days left

- Homework 5 is posted on the course website.
  - Due on April 19 (Sunday), **11:30AM**
  - At most two late days can be used in this homework
  - We have covered all topics except for Problem 4 (today) and 5 (this Thursday)

Logistics: Exam 2

- Exam 2 will be held online on April 23 using Canvas.
- Exam duration: **80 minutes**
  - 5 more minutes than Exam 1 as the buffer for online exam
- Start time
  - By default, please start the exam around the lecture time (11:30am CST).
    - Small deviations are fine
    - The clock starts ticking when you start the exam
  - I can only guarantee to be online to deal with issues during the lecture time.
  - If you cannot make it
    - please let me know by next Friday, Apri 17
      - I'll make the exam available for a longer period of time (likely 6-8 hours). But only people who get approved can take the exam at a different time
    - Unless there is a strong reason, everyone should take the exam on April 23.

Logistics: Exam 2

- Topic
  - The focus will be on the 2^{nd} half of the semester (Everything from Decision trees to the end of the semester).
  - Note that that knowledge is cumulative, and concepts in Exam 1 might also be included.
- Format
  - Similar to Exam 1.
  - A mix of long questions and multiple choice questions.
  - Likely with more multiple-choice questions.
  - I will try to minimize the need to write math in the long questions.
- Open book
  - You can reference any materials in hard copies. Searching information online is not allowed. Talking with other people is not allowed.
- Randomized questions
  - The questions will be randomly drawn from a “question bank”, so everyone might be getting different set of questions. I’ll take that into account in final grades.

Logistics: Exam 2

- Internet connections
  - If you disconnect, you should be able to come back and keep doing it. The clock will keep ticking during your disconnection (reason for the 5-min buffer).
  - If you encounter serious technical issues and cannot connect back within 5~10 minutes. Please inform me as soon as you can.
- Dry run
  - I plan to have a dry run (with dummy exam questions) before the exam. You are strongly encouraged (but not required) to do the dry run to make sure you are familiar with the flow of the exam.
- Proctoring
  - I plan to use “Lockdown browser”: it’s a standalone browser for the exam. It will prevent you from doing things other than answering the exam during exam time.
    - It needs to be installed and only work on Mac or Windows.
    - Let me know if it would be an issue for you with this requirement.

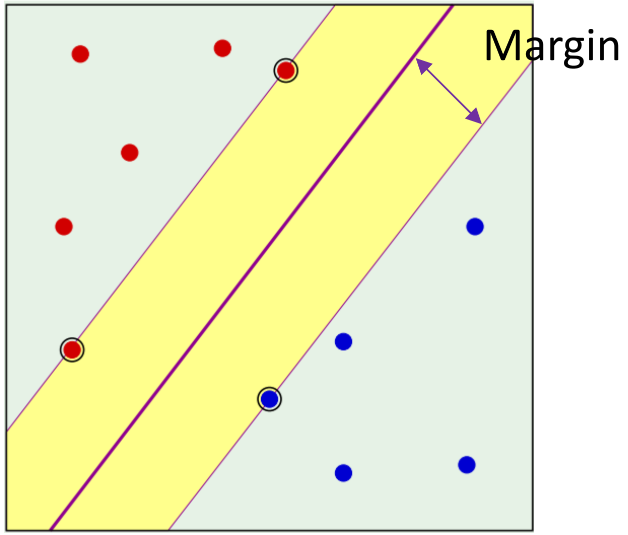
Recap

Support Vector Machine

- Goal: Find the max-margin linear separator that separates the data

- Hard-Margin SVM (Assume data is linearly separable)

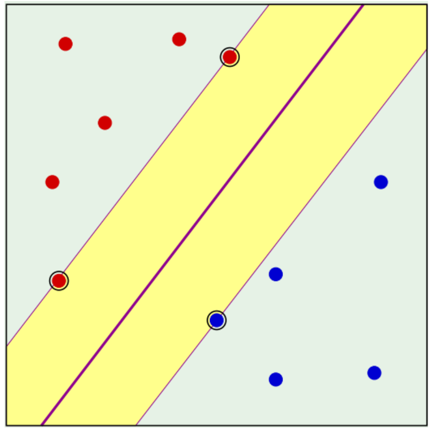
$$\text{minimize}_{\vec{w},b} \quad \frac{1}{2}\vec{w}^T\vec{w}$$
$$\text{subject to} \quad y_n(\vec{w}^T\vec{x}_n + b) \geq 1, \forall n$$



  - Solvable using Quadratic Program (QP)
  - Given solution $(\vec{w}^*, b^*)$, the learned hypothesis $g(\vec{x}) = sign({\vec{w}^*}^T\vec{x} + b^*)$

Support Vectors

- We call the points closest to the separator (candidate) support vectors
  - Since they support the separator

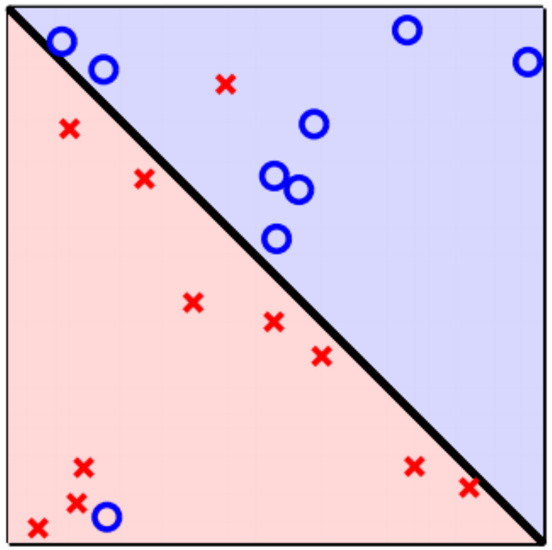


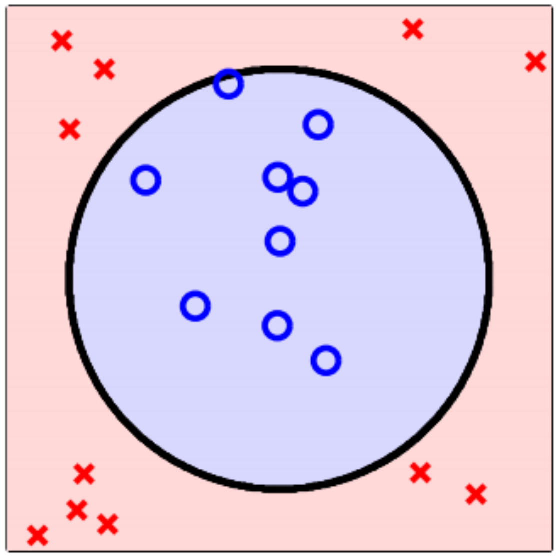
- What are the properties of support vectors?
  - They are the points that the equality holds in the constraints
    - If $\vec{x}_n$ is a support vector, $y_n(\vec{w}^T\vec{x}_n + b) = 1$
  - Removing the non-support vectors will not impact the linear separator

- Leave-One-Out Cross-Validation (LOOCV) error for SVM?
  - $E_{LOOCV} \leq \frac{\#\,\text{support vectors}}{N}$

Non-Separable Data

- Two scenarios



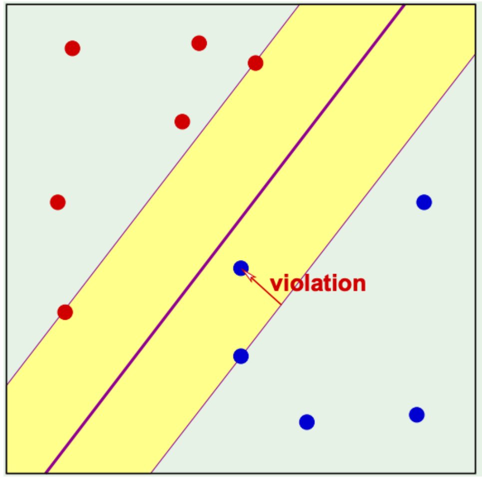


- Tolerate some noise
  - Soft-Margin SVM
- Nonlinear transform
  - Dual formulation and kernel tricks

Soft-Margin SVM

- For each point $(\vec{x}_n, y_n)$, we allow a deviation $\xi_n \geq 0$
  - The constraint becomes: $y_n(\vec{w}^T\vec{x}_n + b) \geq 1 - \xi_n$
  - We add a penalty for each deviation: Total penalty $C\sum_{n=1}^{N}\xi_n$

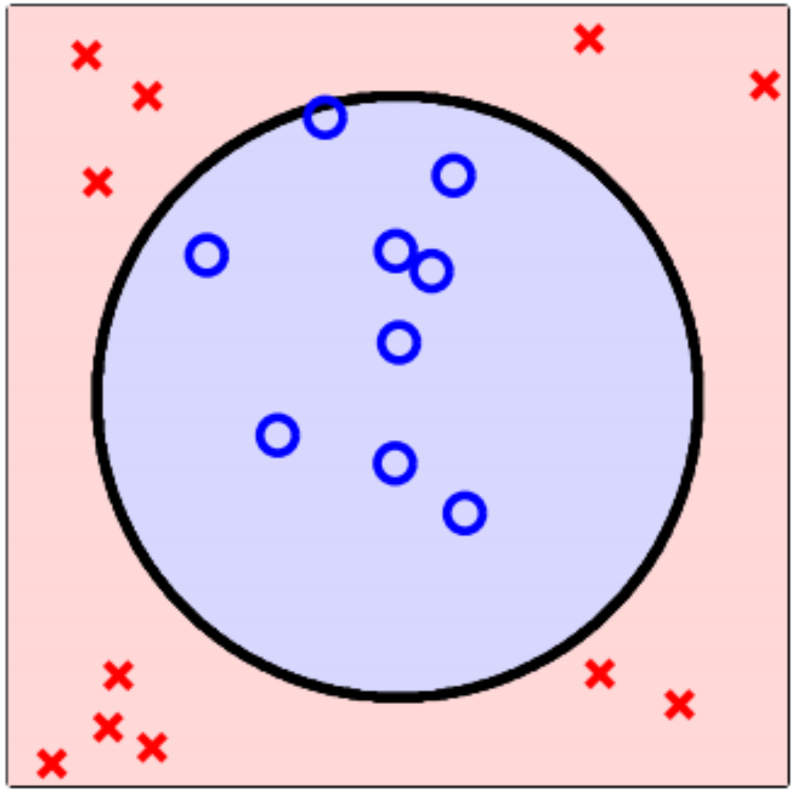
$$
\begin{aligned}
&\text{minimize}_{\vec{w},b,\vec{\xi}} && \frac{1}{2}\vec{w}^T\vec{w} + C\sum_{n=1}^{N}\xi_n \\
&\text{subject to} && y_n(\vec{w}^T\vec{x}_n + b) \geq 1 - \xi_n, \forall n \\
& && \xi_n \geq 0, \forall n
\end{aligned}
$$



Remarks:

- C is a hyper-parameter we can choose, e.g., using validation
  - Larger C => less tolerable to noise => smaller margin
- Soft-margin SVM is still a Quadratic Program, with efficient solvers

What if Tolerating Small Noises Is Not Enough



Nonlinear transform

We can apply standard nonlinear transformation procedure we talked about before

In SVM, we can combine the ideas of dual formulation and kernel tricks for the transformation

This is one of the key ingredients that makes SVM powerful

Lecture Notes Today

(Get prepared for heavier math today…)

The notes are not intended to be comprehensive. They should be accompanied by lectures and/or textbook.

Let me know if you spot errors.

Lagrangian Duality and Convex Optimization

Convex Optimization

- Standard form of convex optimization

$$\begin{aligned} &\text{minimize}_{\vec{w}} && f(\vec{w}) && && \text{Objective} \\ &\text{subject to} && g_i(\vec{w}) \le 0, && i = 1, \dots, k && \text{Inequality constraints} \\ & && h_j(\vec{w}) = 0, && j = 1, \dots, \ell && \text{Equality constraints} \end{aligned}$$

- Convex program
  - f and g_i are convex and h_j are affine
  - Special cases
    - Linear program: f, g_i, h_j are all affine
    - Quadratic program: f is quadratic; g_i and h_j are affine

Lagrangian

$$
\begin{array}{lll}
\text{minimize}_{\vec{w}} & f(\vec{w}) & \\
\text{subject to} & g_i(\vec{w}) \leq 0, & i = 1, \ldots, k \\
 & h_j(\vec{w}) = 0, & j = 1, \ldots, \ell
\end{array}
$$

- The Lagrangian of the convex program can be written as

$$L(\vec{w}, \vec{\alpha}, \vec{\beta}) = f(\vec{w}) + \sum_{i=1}^{k} \alpha_i g_i(\vec{w}) + \sum_{j=1}^{\ell} \beta_j h_j(\vec{w})$$

  - Couple each inequality constraint g_i with a dual variable α_i
  - Couple each equality constraint h_j with a dual variable β_j

- Think about the following expression

$$\max_{\vec{\alpha}, \beta; \alpha_i \geq 0} L(\vec{w}, \vec{\alpha}, \vec{\beta}) = \begin{cases} & \text{if all constraints are satisfied} \\ & \text{otherwise} \end{cases}$$

Lagrangian

$$\begin{array}{lll} \text{minimize}_{\vec{w}} & f(\vec{w}) & \\ \text{subject to} & g_i(\vec{w}) \leq 0, & i = 1, \dots, k \\ & h_j(\vec{w}) = 0, & j = 1, \dots, \ell \end{array}$$

- The Lagrangian of the convex program can be written as

$$L(\vec{w}, \vec{\alpha}, \vec{\beta}) = f(\vec{w}) + \sum_{i=1}^{k} \alpha_i g_i(\vec{w}) + \sum_{j=1}^{\ell} \beta_j h_j(\vec{w})$$

  - Couple each inequality constraint g_i with a dual variable α_i
  - Couple each equality constraint h_j with a dual variable β_j

- Think about the following expression

$$\max_{\vec{\alpha}, \beta; \alpha_i \geq 0} L(\vec{w}, \vec{\alpha}, \vec{\beta}) = \begin{cases} f(\vec{w}), & \text{if all constraints are satisfied} \\ \infty, & \text{otherwise} \end{cases}$$

  - We can rewrite the constrained optimization into unconstrained optimization

Primal-Dual Formulation

- **Primal** problem (the standard form of convex optimization)

$$\min_{\vec{w}} \max_{\vec{\alpha},\vec{\beta};\alpha_i \geq 0} L(\vec{w},\vec{\alpha},\vec{\beta})$$

- **Dual** problem

$$\max_{\vec{\alpha},\vec{\beta};\alpha_i \geq 0} \min_{\vec{w}} L(\vec{w},\vec{\alpha},\vec{\beta})$$

Reminders of definitions:

$$\begin{aligned} &\text{minimize}_{\vec{w}} && f(\vec{w}) && \\ &\text{subject to} && g_i(\vec{w}) \leq 0, && i = 1, \dots, k \\ & && h_j(\vec{w}) = 0, && j = 1, \dots, \ell \end{aligned}$$

$$L(\vec{w},\vec{\alpha},\vec{\beta}) = f(\vec{w}) + \sum_{i=1}^{k} \alpha_i g_i(\vec{w}) + \sum_{j=1}^{\ell} \beta_j h_j(\vec{w})$$

- Minimax theorem [von Neumann, 1928]

$$\min_{\vec{w}} \max_{\vec{\alpha},\vec{\beta};\alpha_i \geq 0} L(\vec{w},\vec{\alpha},\vec{\beta}) = \max_{\vec{\alpha},\vec{\beta};\alpha_i \geq 0} \min_{\vec{w}} L(\vec{w},\vec{\alpha},\vec{\beta})$$

Minimax Theorem [von Neumann, 1928]

$$\min_{\vec{w}} \max_{\vec{\alpha},\vec{\beta};\alpha_i \geq 0} L(\vec{w},\vec{\alpha},\vec{\beta}) = \max_{\vec{\alpha},\vec{\beta};\alpha_i \geq 0} \min_{\vec{w}} L(\vec{w},\vec{\alpha},\vec{\beta})$$

- Remarks
  - The solution of the primal is the same as the solution to the dual!
    - We can work on a different problem space to address the original problem
    - We'll demonstrate the usage of this in SVM, but it's also useful in other applications
  - This is an important result in many areas -- e.g., it is considered as the starting point of game theory (the two-player zero-sum game).
- Now we know the objectives of the optimal dual and the optimal primal are the same. How are the optimal solutions related?

Karush-Kuhn-Tucker (KKT) Conditions

$$L(\vec{w}, \vec{\alpha}, \vec{\beta}) = f(\vec{w}) + \sum_{i=1}^{k} \alpha_i g_i(\vec{w}) + \sum_{j=1}^{\ell} \beta_j h_j(\vec{w})$$

Primal: $\min_{\vec{w}} \max_{\vec{\alpha},\vec{\beta};\alpha_i \geq 0} L(\vec{w}, \vec{\alpha}, \vec{\beta})$

Dual: $\max_{\vec{\alpha},\vec{\beta};\alpha_i \geq 0} \min_{\vec{w}} L(\vec{w}, \vec{\alpha}, \vec{\beta})$

- The optimal solutions $(\vec{w}^*, \vec{\alpha}^*, \vec{\beta}^*)$ satisfy the following conditions
  - Stationary condition: $\nabla_{\vec{w}} L(\vec{w}, \vec{\alpha}^*, \vec{\beta}^*)|_{\vec{w}=\vec{w}^*} = \vec{0}$
  - Primal feasibility: $g_i(\vec{w}^*) \leq 0$; $h_j(\vec{w}^*) = 0$ for all (i, j)
  - Dual feasibility: $\alpha_i^* \geq 0$ for all i
  - Complementary slackness: $\alpha_i^* g_i(\vec{w}^*) = 0$ for all i

Short Break and Questions

Reminders of definitions in general convex program:

$$\begin{aligned} &\text{minimize}_{\vec{w}} \quad f(\vec{w}) \\ &\text{subject to} \quad g_i(\vec{w}) \leq 0, \qquad i = 1, \dots, k \\ &\qquad\qquad\quad h_j(\vec{w}) = 0, \qquad j = 1, \dots, \ell \end{aligned}$$

$$L(\vec{w}, \vec{\alpha}, \vec{\beta}) = f(\vec{w}) + \sum_{i=1}^{k} \alpha_i g_i(\vec{w}) + \sum_{j=1}^{\ell} \beta_j h_j(\vec{w})$$

Primal: $\min_{\vec{w}} \max_{\vec{\alpha}, \vec{\beta}; \alpha_i \geq 0} L(\vec{w}, \vec{\alpha}, \vec{\beta})$

Dual: $\max_{\vec{\alpha}, \vec{\beta}; \alpha_i \geq 0} \min_{\vec{w}} L(\vec{w}, \vec{\alpha}, \vec{\beta})$

Exercise:
Remember the weight-decay regularization:

$$\begin{aligned} &\text{minimize}_{\vec{w}} \quad E_{in}(\vec{w}) \\ &\text{subject to} \quad \vec{w}^T \vec{w} \leq C \end{aligned}$$

Use what we talked about to write the unconstrained optimization problem.

Dual SVM

Derive the Dual for Hard-Margin SVM

- Hard-margin SVM

$$\begin{aligned} &\text{minimize}_{\vec{w},b} && \frac{1}{2}\vec{w}^T\vec{w} \\ &\text{subject to} && y_n(\vec{w}^T\vec{x}_n + b) \geq 1, \forall n \end{aligned}$$

<u>Reminders of definitions in general convex program:</u>

$$\begin{aligned} &\text{minimize}_{\vec{w}} && f(\vec{w}) && \\ &\text{subject to} && g_i(\vec{w}) \leq 0, && i = 1, \dots, k \\ & && h_j(\vec{w}) = 0, && j = 1, \dots, \ell \end{aligned}$$

$$L(\vec{w}, \vec{\alpha}, \vec{\beta}) = f(\vec{w}) + \sum_{i=1}^{k} \alpha_i g_i(\vec{w}) + \sum_{j=1}^{\ell} \beta_j h_j(\vec{w})$$

Dual: $\max_{\vec{\alpha},\vec{\beta};\alpha_i \geq 0} \min_{\vec{w}} L(\vec{w}, \vec{\alpha}, \vec{\beta})$

Derive the Dual for Hard-Margin SVM

- Hard-margin SVM

$$\begin{array}{ll} \text{minimize}_{\vec{w},b} & \frac{1}{2}\vec{w}^T\vec{w} \\ \text{subject to} & y_n(\vec{w}^T\vec{x}_n + b) \geq 1, \forall n \end{array}$$

- First write down the Lagrangian

$$L(\vec{w}, b, \vec{\alpha}) = \frac{1}{2}\vec{w}^T\vec{w} + \sum_{n=1}^{N} \alpha_n \left(1 - y_n(\vec{w}^T\vec{x}_n + b)\right)$$

$$= \frac{1}{2}\vec{w}^T\vec{w} + \sum_{n=1}^{N} \alpha_n - \sum_{n=1}^{N} \alpha_n y_n(\vec{w}^T\vec{x}_n + b)$$

- Dual

$$\max_{\vec{\alpha};\alpha_i \geq 0} \min_{\vec{w},b} L(\vec{w}, b, \vec{\alpha})$$

<u>Reminders of definitions in general convex program:</u>

$$\begin{array}{lll} \text{minimize}_{\vec{w}} & f(\vec{w}) & \\ \text{subject to} & g_i(\vec{w}) \leq 0, & i = 1, \dots, k \\ & h_j(\vec{w}) = 0, & j = 1, \dots, \ell \end{array}$$

$$L(\vec{w}, \vec{\alpha}, \vec{\beta}) = f(\vec{w}) + \sum_{i=1}^{k} \alpha_i g_i(\vec{w}) + \sum_{j=1}^{\ell} \beta_j h_j(\vec{w})$$

Dual: $\max_{\vec{\alpha},\vec{\beta};\alpha_i \geq 0} \min_{\vec{w}} L(\vec{w}, \vec{\alpha}, \vec{\beta})$

- Lagrangian $L(\vec{w}, b, \vec{\alpha}) = \frac{1}{2}\vec{w}^T\vec{w} + \sum_{n=1}^{N} \alpha_n - \sum_{n=1}^{N} \alpha_n y_n(\vec{w}^T\vec{x}_n + b)$
- Dual $\max_{\vec{\alpha};\alpha_i \geq 0} \min_{\vec{w},b} L(\vec{w}, b, \vec{\alpha})$ (the variables in the dual are $\vec{\alpha}$)

- Derivations
  - Express $\vec{w}$ and b using $\vec{\alpha}$ in the dual objective $\min_{\vec{w},b} L(\vec{w}, b, \vec{\alpha})$

- Lagrangian $L(\vec{w}, b, \vec{\alpha}) = \frac{1}{2}\vec{w}^T\vec{w} + \sum_{n=1}^{N} \alpha_n - \sum_{n=1}^{N} \alpha_n y_n(\vec{w}^T\vec{x}_n + b)$
- Dual $\max_{\vec{\alpha};\alpha_i \geq 0} \min_{\vec{w},b} L(\vec{w}, b, \vec{\alpha})$ (the variables in the dual are $\vec{\alpha}$)

Dual Constraint

- Derivations
  - Express $\vec{w}$ and b using $\vec{\alpha}$ in the dual objective $\min_{\vec{w},b} L(\vec{w}, b, \vec{\alpha})$
    - Solve for $\nabla_{\vec{w},b} L(\vec{w}, b, \vec{\alpha}) = 0$
      - $\nabla_{\vec{w}} L(\vec{w}, b, \vec{\alpha}) = 0$ => $\vec{w} - \sum_{n=1}^{N} \alpha_n y_n \vec{x}_n = 0$ => $\vec{w} = \sum_{n=1}^{N} \alpha_n y_n \vec{x}_n$
      - $\nabla_{b} L(\vec{w}, b, \vec{\alpha}) = 0$ => $\sum_{n=1}^{N} \alpha_n y_n = 0$

Dual Constraint

    - Plug $\vec{w} = \sum_{n=1}^{N} \alpha_n y_n \vec{x}_n$ into $L(\vec{w}, b, \vec{\alpha})$
      - $\frac{1}{2}\vec{w}^T\vec{w} = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m \vec{x}_n^T \vec{x}_m$
      - $\sum_{n=1}^{N} \alpha_n y_n(\vec{w}^T\vec{x} + b) = \sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m \vec{x}_n^T \vec{x}_m + b\sum_{n=1}^{N} \alpha_n y_n$
  - $\min_{\vec{w},b} L(\vec{w}, b, \vec{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m \vec{x}_n^T \vec{x}_m$

Dual Objective

Dual SVM

- Dual of the hard-margin SVM

$$\begin{aligned}&\text{maximize}_{\vec{\alpha}} \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m \vec{x}_n^T \vec{x}_m \\ &\text{subject to } \sum_{n=1}^{N} \alpha_n y_n = 0 \\ &\qquad\qquad\quad \alpha_n \geq 0, \forall n\end{aligned}$$

  - The dual is still a Quadratic Program, with efficient solvers to find $\vec{\alpha}^*$

- We know that the objective of the optimal dual is the same as the optimal primal.
- Say we obtain $\vec{\alpha}^*$, how do we recover the optimal primal $(\vec{w}^*, b^*)$?
  - Apply KKT conditions

Recover $(\vec{w}^*, b^*)$ from $\vec{\alpha}^*$

- Using stationary conditions in KKT
  - $\nabla_{\vec{w}} L(\vec{w}, b^*, \vec{\alpha}^*)|_{\vec{w}=\vec{w}^*} = \vec{0}$
  - $\vec{w}^* = \sum_{n=1}^{N} \alpha_n^* y_n \vec{x}_n$
  - Since $\alpha_n^* \geq 0$, we can rewrite $\vec{w}^* = \sum_{\alpha_n^*>0} \alpha_n^* y_n \vec{x}_n$

$L(\vec{w}, b, \vec{\alpha}) = \frac{1}{2}\vec{w}^T\vec{w} + \sum_{n=1}^{N} \alpha_n - \sum_{n=1}^{N} \alpha_n y_n(\vec{w}^T\vec{x}_n + b)$

- Using complementary slackness in KKT
  - $\alpha_n^*\left(1 - y_n(\vec{x}_n^T\vec{w}^* + b^*)\right) = 0$
  - Find a $\alpha_n^* > 0$, we have $y_n(\vec{x}_n^T\vec{w}^* + b^*) = 1$
  - Since $y_n \in \{+1, -1\}$, we have $\vec{x}_n^T\vec{w}^* + b^* = y_n$
  - Therefore,
    - $b^* = y_n - \vec{x}_n^T\vec{w}^*$ (with $\vec{w}^* = \sum_{\alpha_n^*>0} \alpha_n^* y_n \vec{x}_n$)

Note that $\vec{w}^T\vec{x} = \vec{x}^T\vec{w}$.
I swapped the order to avoid two superscripts in $\vec{w}$

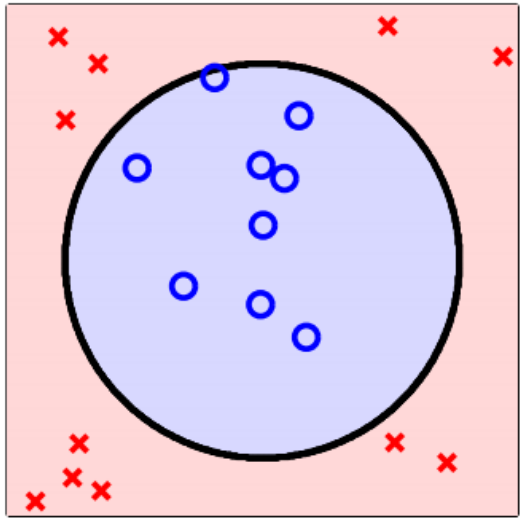
Recover $(\vec{w}^*, b^*)$ from $\vec{\alpha}^*$

- Solve the dual and find $\vec{\alpha}^*$
  - $\vec{w}^* = \sum_{\alpha_n^*>0} \alpha_n^* y_n \vec{x}_n$
  - Find a $\alpha_n^* > 0,\ b^* = y_n - \vec{x}_n^T \vec{w}^*$
  - $g(\vec{x}) = sign({\vec{w}^*}^T \vec{x} + b^*)$

- What does $\alpha_n^* > 0$ imply?
  - Complementary slackness $\alpha_n^*\left(1 - y_n(\vec{x}_n^T \vec{w}^* + b^*)\right) = 0$
  - $\alpha_n^* > 0$ => $y_n(\vec{x}_n^T \vec{w}^* + b^*) = 1$

- $\alpha_n^* > 0$ => $(\vec{x}_n, y_n)$ is the support vector
  - $\vec{w}^* = \sum_{\alpha_n^*>0} \alpha_n^* y_n \vec{x}_n$ is the linear combination of support vectors!
  - Support vector machine!

Nonlinear Transform and Kernel Tricks



Primal-Dual Formulations of Hard-Margin SVM

- Primal

$$\begin{aligned}&\text{minimize}_{\vec{w},b} && \frac{1}{2}\vec{w}^T\vec{w}\\ &\text{subject to} && y_n(\vec{w}^T\vec{x}_n + b) \geq 1, \forall n\end{aligned}$$

- Dual

$$\begin{aligned}&\text{maximize}_{\vec{\alpha}} && \textstyle\sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n\alpha_m y_n y_m \vec{x}_n^T\vec{x}_m\\ &\text{subject to} && \textstyle\sum_{n=1}^{N} \alpha_n y_n = 0\\ &&& \alpha_n \geq 0, \forall n\end{aligned}$$

Given optimal $\vec{\alpha}^*$:

- $\vec{w}^* = \sum_{\alpha_n^*>0} \alpha_n^* y_n \vec{x}_n$
- Find a $\alpha_n^* > 0,\ b^* = y_n - \vec{x}_n^T\vec{w}^*$

- Both can be efficiently solved using QP solver.
- We can infer the solution from one to the other

Nonlinear Transform: $\vec{z} = \Phi(\vec{x})$

- Primal

$$\begin{array}{ll} \text{minimize}_{\vec{w},b} & \frac{1}{2}\vec{w}^T\vec{w} \\ \text{subject to} & y_n(\vec{w}^T\vec{z}_n + b) \geq 1, \forall n \end{array}$$

Involves changing $\vec{w}$ and $\vec{z}$.
The computation grows as the dimension of the $\vec{z}$ space grows

- Dual

$$\begin{array}{ll} \text{maximize}_{\vec{\alpha}} & \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n\alpha_m y_n y_m \vec{z}_n^T\vec{z}_m \\ \text{subject to} & \sum_{n=1}^{N} \alpha_n y_n = 0 \\ & \alpha_n \geq 0, \forall n \end{array}$$

The only difference is from calculating $\vec{x}_n^T\vec{x}_m$ to $\vec{z}_n^T\vec{z}_m$

- Intuition: If we can find an efficient way to calculate $\vec{z}_n^T\vec{z}_m$, we can derive the optimal dual to infer the optimal primal.
  - Doing nonlinear transform without sacrificing much about computation.

Example: 2nd Order Polynomial Transform

We define the transform slight differently

- The $\sqrt{2}$ and the initial 1 are not in the original transform, but we include them for convenience.

- $\vec{x} = (x_1, x_2)$
- 2nd order polynomial transform
  - $\vec{z} = \Phi_2(\vec{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}\,x_1x_2, x_1^2, x_2^2)$

$$
\begin{aligned}
\vec{z}^T\vec{z}' &= 1 + 2x_1x_1' + 2x_2x_2' + 2x_1x_1'x_2x_2' + x_1^2{x_1'}^2 + x_2^2{x_2'}^2 \\
&= 1 + 2x_1x_1' + 2x_2x_2' + 2x_1x_1'x_2x_2' + (x_1x_1')^2 + (x_2x_2')^2 \\
&= (1 + x_1x_1' + x_2x_2')^2 \\
&= (1 + \vec{x}^T\vec{x}')^2
\end{aligned}
$$

- We can calculate $\vec{z}^T\vec{z}'$ from the operation in the $\vec{x}$ space!

Kernel Functions

- Define kernel function $K_\Phi(\vec{x}, \vec{x}') = \Phi(\vec{x})^T \Phi(\vec{x}')$
  - The similarity of two vectors in the projected space
- Goal: Compute $K_\Phi(\vec{x}, \vec{x}')$ without transforming $\vec{x}$ and $\vec{x}'$
- Why? This enables us to operate in the higher dimensional space without really worried about the computational overhead.

Kernel Trick: Utilize Dual and Kernel Functions

- The dual with nonlinear transform

$$\begin{aligned} &\text{maximize}_{\vec{\alpha}} \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m \vec{z}_n^T \vec{z}_m \\ &\text{subject to } \sum_{n=1}^{N} \alpha_n y_n = 0 \\ &\qquad\qquad\quad \alpha_n \geq 0, \forall n \end{aligned}$$

- Plug in the kernel function $K_\Phi(\vec{x}, \vec{x}') = \Phi(\vec{x})^T \Phi(\vec{x}')$

$$\begin{aligned} &\text{maximize}_{\vec{\alpha}} \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m K_\Phi(\vec{x}_n, \vec{x}_m) \\ &\text{subject to } \sum_{n=1}^{N} \alpha_n y_n = 0 \\ &\qquad\qquad\quad \alpha_n \geq 0, \forall n \end{aligned}$$

  - If the kernel can be computed efficiently, we can solve $\vec{\alpha}^*$ efficiently.
  - With kernel tricks, we can avoid the dependency on the dimension of $\vec{z}$

Recover $(\vec{w}^*, b^*)$ from $\vec{\alpha}^*$ with Kernel Tricks

- Note that $\vec{\alpha}^*$ is solved in the $\vec{z}$ space
  - $\vec{w}^* = \sum_{\alpha_n^*>0} \alpha_n^* y_n \Phi(\vec{x}_n)$
  - Find a $\alpha_n^* > 0,\ b^* = y_n - {\vec{w}^*}^T \Phi(\vec{x}_n)$
  - We want to avoid the transformation!

- Let's look at the hypothesis
  - $g(\vec{x}) = sign\left({\vec{w}^*}^T \Phi(\vec{x}) + b^*\right)$

$$
\begin{aligned}
{\vec{w}^*}^T \Phi(\vec{x}) &= \left(\sum_{\alpha_n^*>0} \alpha_n^* y_n \Phi(\vec{x}_n)\right)^T \Phi(\vec{x}) \\
&= \sum_{\alpha_n^*>0} \alpha_n^* y_n \Phi(\vec{x}_n)^{\mathrm{T}} \Phi(\vec{x}) \\
&= \sum_{\alpha_n^*>0} \alpha_n^* y_n K(\vec{x}_n, \vec{x})
\end{aligned}
$$

$$
\begin{aligned}
b^* &= y_n - {\vec{w}^*}^T \Phi(\vec{x}_n) \\
&= y_n - \left(\sum_{\alpha_m^*>0} \alpha_m^* y_m \Phi(\vec{x}_m)\right)^{\mathrm{T}} \Phi(\vec{x}_n) \\
&= y_n - \sum_{\alpha_m^*>0} \alpha_m^* y_m K(\vec{x}_m, \vec{x}_n)
\end{aligned}
$$

  - Still can be computed in the $\vec{x}$ space!