Machine Learning Principles

Class12 : October 16

Support Vector Machines II and SMO

Instructor: Diana Kim

Today's Lecture

1. Why soft margin SVM is needed?

1. Soft margin SVM
   - optimization (primal & dual)

3. Loss Comparison: soft SVM /logistic regression/ perceptron

4. Optimization Algorithm solving the dual problem of SVM
   (**S**equential Minimal **O**ptimization: SMO)

[1] When do we need a soft margin SVM?

In the last class, we learned a hard margin SVM.
by using a proper kernel function,
it can implicitly transform the data from its original space to high dimensional space, making the data is linearly separable.

[2] When do we need a soft margin SVM?

theoretically,
using a Gaussian kernel we can make any data is linearly sparable.

**recall Kernel Trick

[dual problem]

$$\underset{\lambda_{n=1}^{N}}{\arg\max} -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \lambda_n \cdot \lambda_m \cdot t_n \cdot t_m \cdot \kappa(x_n, x_m) + \sum_{n=1}^{N} \lambda_n$$

$$\text{subject to} \quad \lambda_n \geq 0$$

$$\sum_{n=1}^{N} \lambda_n \cdot t_n = 0$$

[SVM classifier]

$$y(x) = \sum_{n=1}^{N} \lambda_n^* t_n \kappa(x_n, x) + b$$

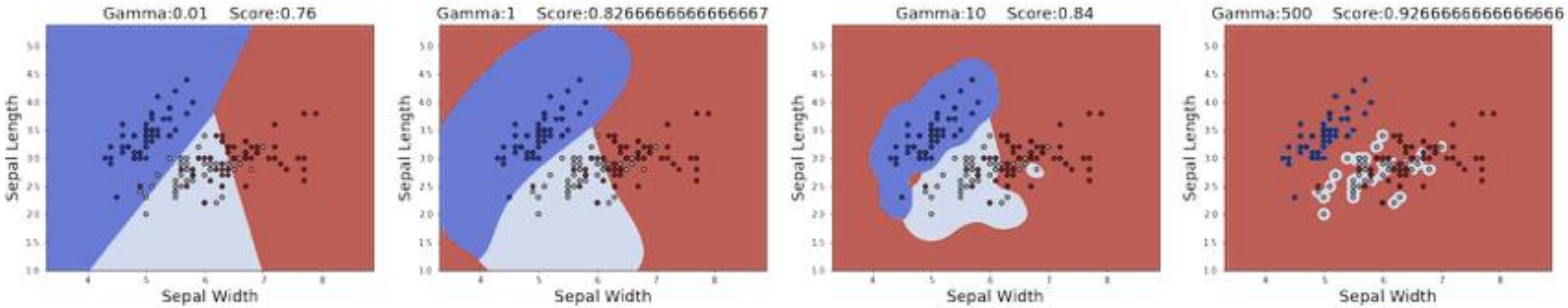
- by using a proper kernel function,
  we can compute a maximum margin classifier in a high dimensional feature space without designing a feature space directly.

**recall: Objective for Maximum Margin (Gaussian Kernel SVM)

$$\gamma = \frac{1}{\sigma^2}$$

From https://www.kaggle.com/code/gorkemgunay/understanding-parameters-of-svm

the effect of gamma on # of support vectors & decision Boundary



- small γ: some representative samples become support vectors.
- large γ : every sample become support vectors
- depending on γ, model complexity varies.

[4] When do we need a soft margin SVM?

Q: what will be like using Gaussian kernel with a small $\sigma \approx 0$ or when we have a complex data stucture?

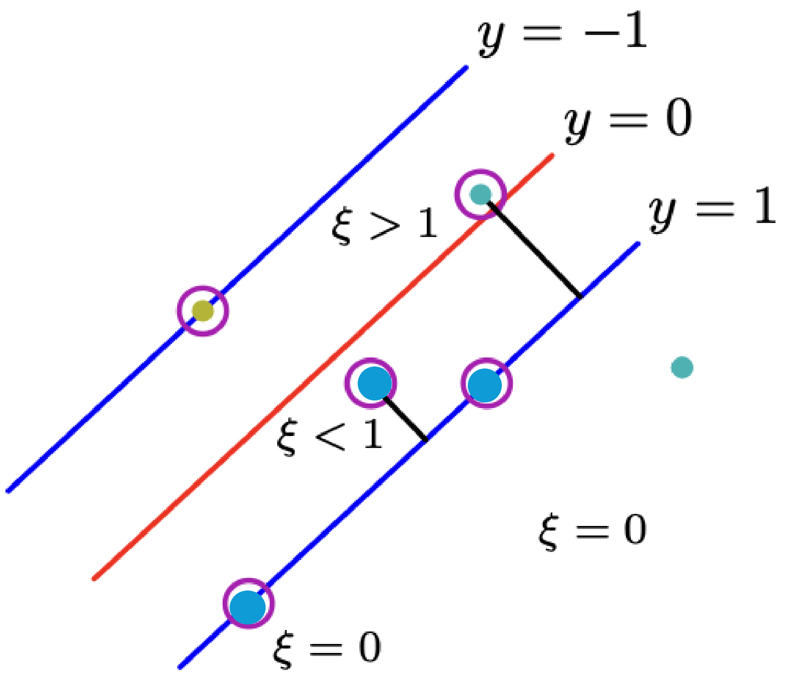
$$y(x) = \sum_{n=1}^{N} \lambda_n^* t_n \kappa(x_n, x) + b$$

Q: what will be the problem of the classifier?

[5] When do we need a soft margin SVM?

Q: What if we allow a few data points to cross the margin?

from Bishop Figure 7.3



- hard margin

$$t_n(w^t x_n + b) \geq 1 \quad \forall n$$

- soft margin

$$t_n(w^t x_n + b) \geq \underline{1 - \xi_n} \quad \text{and} \quad \xi_n \geq 0 \quad \forall n$$

relaxation of the minimum margin

- Soft Margin SVM
(primal & dual optimization problem)

[1] Objective of Soft Maximum Margin Classifier (primal)

[primal problem]

$$w*, b* = \arg\min_{w,b} C \cdot \sum_{n=1}^{N} \xi_n + \frac{1}{2}||w||^2$$

$$\text{subject to} \quad t_n(w^t x_n + b) \geq 1 - \xi_n \quad \forall n$$

$$\text{subject to} \quad \xi_n \geq 0 \quad \forall n$$

- depending on C, $\sum_{n=1}^{N} \xi_n$ (degree of relaxation) can be controlled.
- $C \rightarrow \infty$, the primal function gets close to the original hard margin SVM problem.

[2] Objective of Soft Maximum Margin Classifier (Lagrangian)

[primal problem]

$$w*, b* = \arg\min_{w,b} C \cdot \sum_{n=1}^{N} \xi_n + \frac{1}{2}||w||^2$$

$$\text{subject to} \quad t_n(w^t x_n + b) \geq 1 - \xi_n \quad \forall n$$

$$\text{subject to} \quad \xi_n \geq 0 \quad \forall n$$

[Lagrangian problem]

$$L(w, b, \xi_{n=1}^{N}, \boxed{\lambda_{n=1}^{N}, \mu_{n=1}^{N})}$$ → Lagrangian parameters

$$= C \cdot \sum_{n=1}^{N} \xi_n + \frac{1}{2}||w||^2 + \sum_{n=1}^{N} \lambda_n - \sum_{n=1}^{N} \lambda_n \cdot t_n \cdot (w^t x_n + b) - \sum_{n=1}^{N} \lambda_n \xi_n - \sum_{n=1}^{N} \mu_n \xi_n$$

**Review: Inequality Constraint Problem (KKT necessary conditions)

Let x^* be a local minimum of the problem

$$\min_x \quad f(x)$$
$$\text{s.t.} \quad g_i(x) \leq 0 \quad i = 1, \ldots, m$$

Then, there exist $\lambda_i,\ i = 1, \ldots, m$ such that

(1) $\nabla f(x*) + \sum_{i=1}^{m} \lambda_i \nabla g_m(x*) = 0$

(2) $\begin{cases} \lambda_j \geq 0 & j = 1, \ldots, r \\ \lambda_j = 0 & \forall j \notin A(x*) \end{cases}$

(3) $g(x^*) \leq 0$

- stationary condition (1)
- complementary slackness condition (2)
  $\lambda_i \cdot g_i(x^*) = 0$
- primary feasibility (3)

[1] Objective of Soft Maximum Margin Classifier (KKT conditions)

$$L(w, b, \xi_{n=1}^{N}, \lambda_{n=1}^{N}, \mu_{n=1}^{N})$$

$$= C \cdot \sum_{n=1}^{N} \xi_n + \frac{1}{2}||w||^2 + \sum_{n=1}^{N} \lambda_n - \sum_{n=1}^{N} \lambda_n \cdot t_n \cdot (w^t x_n + b) - \sum_{n=1}^{N} \lambda_n \xi_n - \sum_{n=1}^{N} \mu_n \xi_n$$

$$\left.\begin{aligned} \lambda_n^* &\geq 0 \quad \forall n \\ \mu_n^* &\geq 0 \quad \forall n \\ t_n(w*^t x_n + b*) &\geq 1 - \xi_n^* \quad \forall n \\ \xi_n^* &\geq 0 \quad \forall n \end{aligned}\right\}$$

- Langrangian value is positive
- primary feasibility

$$\left.\begin{aligned} \lambda_n^* \{t_n(w*^t x_n + b*) - 1 + \xi_n^*\} &= 0 \quad \forall n \\ \mu_n^* \xi_n^* &= 0 \quad \forall n \end{aligned}\right\}$$

- complementary slackness

[2] Objective of Soft Maximum Margin Classifier (KKT conditions)

$$L(w, b, \xi_{n=1}^{N}, \lambda_{n=1}^{N}, \mu_{n=1}^{N})$$

$$= C \cdot \sum_{n=1}^{N} \xi_n + \frac{1}{2}||w||^2 + \sum_{n=1}^{N} \lambda_n - \sum_{n=1}^{N} \lambda_n \cdot t_n \cdot (w^t x_n + b) - \sum_{n=1}^{N} \lambda_n \xi_n - \sum_{n=1}^{N} \mu_n \xi_n$$

- stationary condition

$$\nabla_w L(w, b, \xi_{n=1}^{N}, \lambda_{n=1}^{N}, \mu_{n=1}^{N}) = \vec{w*} - \sum_{n=1}^{N} \lambda_n^* \cdot t_n \cdot \vec{x_n} = 0$$

$$\nabla_b L(w, b, \xi_{n=1}^{N}, \lambda_{n=1}^{N}, \mu_{n=1}^{N}) = \sum_{n=1}^{N} \lambda_n^* \cdot t_n = 0$$

$$\nabla_{\xi_n} L(w, b, \xi_{n=1}^{N}, \lambda_{n=1}^{N}, \mu_{n=1}^{N}) = \boxed{C - \lambda_n^* - \mu_n^* = 0}$$

[3] Objective of Soft Maximum Margin Classifier (dual representation)

- [Lagrangian problem]

$$L(w, b, \xi_{n=1}^N, \lambda_{n=1}^N, \mu_{n=1}^N)$$
$$= C \cdot \sum_{n=1}^N \xi_n + \frac{1}{2}||w||^2 + \sum_{n=1}^N \lambda_n - \sum_{n=1}^N (\lambda_n \cdot t_n \cdot (w^t x_n + b) - \sum_{n=1}^N \lambda_n \xi_n - \sum_{n=1}^N \mu_n \xi_n$$

- [plug in optimal]

$$\nabla_w L(w, b, \xi_{n=1}^N, \lambda_{n=1}^N, \mu_{n=1}^N) = \vec{w*} - \sum_{n=1}^N \lambda_n^* \cdot t_n \cdot \vec{x_n} = 0$$

$$\nabla_{\xi_n} L(w, b, \xi_{n=1}^N, \lambda_{n=1}^N, \mu_{n=1}^N) = C - \lambda_n^* - \mu_n^* = 0$$

[4] Objective of Soft Maximum Margin Classifier (dual representation)

- [Lagrangian problem]

$$L(w, b, \xi_{n=1}^{N}, \lambda_{n=1}^{N}, \mu_{n=1}^{N})$$

$$\boxed{= C \cdot \sum_{n=1}^{N} \xi_n +} \frac{1}{2}||w||^2 + \sum_{n=1}^{N} \lambda_n - \sum_{n=1}^{N} (\lambda_n \cdot t_n \cdot (w^t x_n + b) - \sum_{n=1}^{N} \lambda_n \xi_n - \sum_{n=1}^{N} \mu_n \xi_n$$

$$C \cdot \sum_{n=1}^{N} \xi_n = \sum_{n=1}^{N} (\lambda_n + \mu_n) \xi_n$$

- [dual representation]

$$D(\lambda) = -\frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \lambda_n \cdot \lambda_m \cdot t_n \cdot t_m \cdot \vec{x_n}^t \vec{x_m} + \sum_{n=1}^{N} \lambda_n$$

Q: maximize? / minimize?

[5] Objective of Soft Maximum Margin Classifier (dual representation)

[dual problem]

$$\lambda *_{n=1}^{N} = \arg\max_{\lambda *} \sum_{n=1}^{N} \lambda *_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \lambda_n^* \lambda_m^* \cdot t_n \cdot t_m \cdot \kappa(x_n, x_m)$$

$$\text{subject to} \quad 0 \leq \lambda_n^* \leq C \longrightarrow \boxed{\begin{array}{rl} \lambda_n \geq 0 & \forall n \\ \mu_n \geq 0 & \forall n \\ C = \lambda_n + \mu_n & \forall n \end{array}}$$

$$\sum_{n=1}^{N} \lambda_n^* \cdot t_n = 0$$

- this is the quadratic optimization problem we need to solve!

[6] Objective for Maximum Margin (dual for soft vs. hard SVM)

- [dual problem of hard SVM]

$$\arg\max_{\lambda_{n=1}^{N}} -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \lambda_n \cdot \lambda_m \cdot t_n \cdot t_m \cdot \kappa(x_n, x_m) + \sum_{n=1}^{N} \lambda_n$$

$$\text{subject to} \quad \lambda_n \geq 0$$

$$\sum_{n=1}^{N} \lambda_n \cdot t_n = 0$$

- [dual problem of soft SVM]

$$\arg\max_{\lambda_{n=1}^{N}} -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \lambda_n \lambda_m \cdot t_n \cdot t_m \cdot \kappa(x_n, x_m) + \sum_{n=1}^{N} \lambda_n$$

$$\text{subject to} \quad 0 \leq \lambda_n \leq C$$

$$\sum_{n=1}^{N} \lambda_n \cdot t_n = 0$$

- soft SVM objective is equivalent to hard SVM, but Lagrangin is upper bounded by C for soft SVM.

[7] Objective for Maximum Margin (**S**upport **V**ector **M**achine : SVM)

$$\vec{w}* = \sum_{n=1}^{N} \lambda_n^* t_n \phi(x)$$

- compared to hard SVM, more data points are involved to define a classifier.

[SVM classifier]

$$y(x) = \sum_{n=1}^{N} \lambda_n^* t_n \phi(x)^t \phi(x) + b$$

$$\begin{cases} y(x) \geq 0 \quad x \in + \\ y(x) < 0 \quad x \in - \end{cases}$$

[8] Objective for Maximum Margin (dual solution)

when we found the dual solutions $\lambda_1, \lambda_2, \lambda_3 \dots . \lambda_N$,

$\lambda_n = 0,$

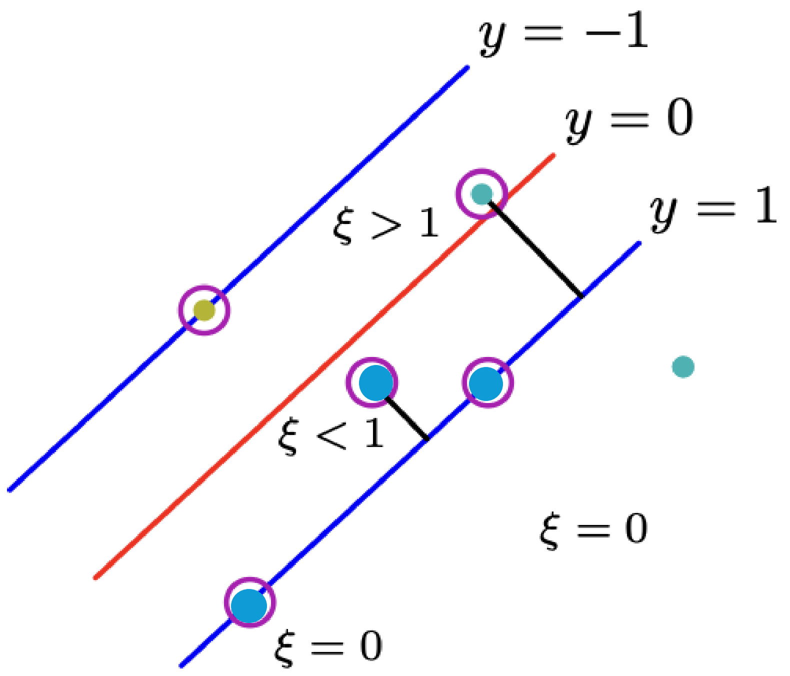
$$t_n(w *^t x_n + b) > 1 - \xi_n^*$$
$$\mu_n = C \quad \text{and} \quad \xi_n = 0$$
$$t_n(w *^t x_n + b) > 1$$

$0 < \lambda_n < C$

- the data points on the correct side but beyond the margin.

$\lambda_n = C,$

$$t_n(w *^t x_n + b) = 1 - \xi_n^*$$
$$\mu_n = 0 \quad \text{and} \quad \xi_n > 0$$
$$t_n(w *^t x_n + b) = 1 - \xi_n^* < 1$$



- the data points lie inside the margin.

[9] Objective for Maximum Margin (dual solution)

when we found the dual solutions $\lambda_1, \lambda_2, \lambda_3 \ldots.\ \lambda_N$,

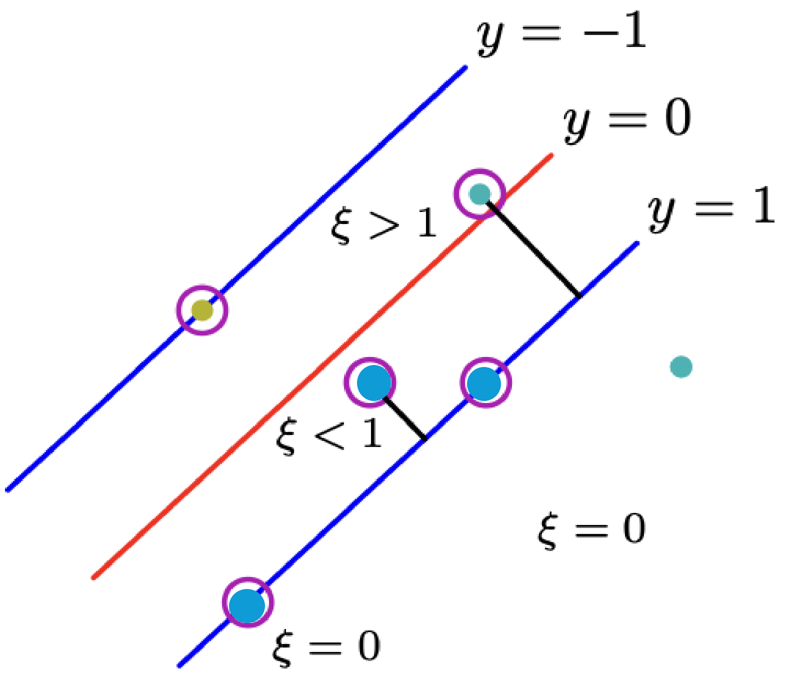
$\lambda_n = 0,$

$0 < \lambda_n < C$

$\lambda_n = C,$

$$t_n(w*^t x_n + b) = 1 - \xi_n^*$$
$$\mu_n = C - \lambda_n \quad \text{and} \quad \xi_n = 0$$
$$t_n(w*^t x_n + b) = 1$$

- the data points lie exactly the margin.



[10] Objective for Maximum Margin (dual solution)

- Q: what are the support vectors are the data vectors?

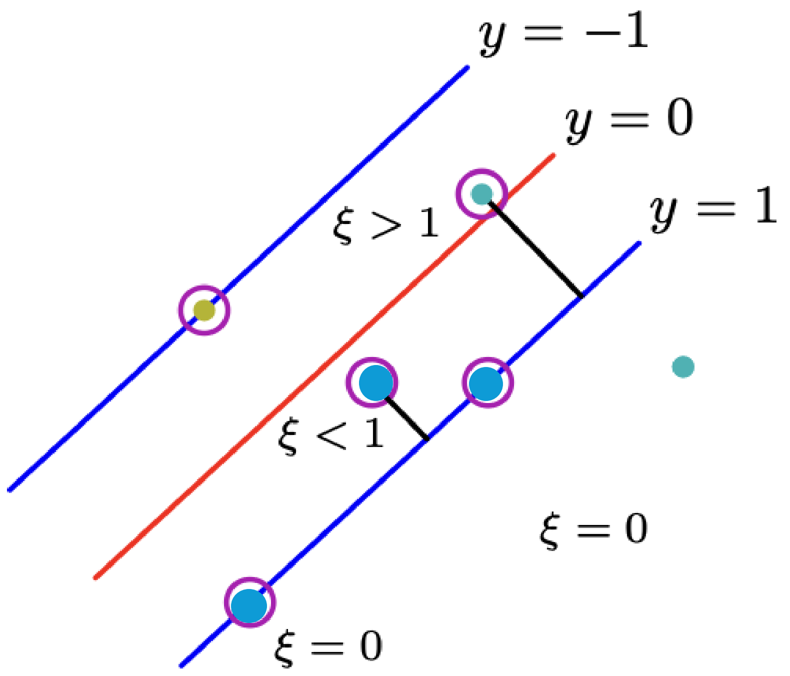
$\lambda_i = 0,$

$0 < \lambda_i < C$

$$t_n(w*^t x_n + b) = 1 - \xi_n^*$$
$$\mu_n = C - \lambda_n \quad \text{and} \quad \xi_n = 0$$
$$t_n(w*^t x_n + b) = 1$$

$\lambda_i = C,$

$$t_n(w*^t x_n + b) = 1 - \xi_n^*$$
$$\mu_n = 0 \quad \text{and} \quad \xi_n > 0$$
$$t_n(w*^t x_n + b) = 1 - \xi_n^* < 1$$



[11] Objective for Maximum Margin (dual solution)

- given optimal λ_n,
  the corredponding data (x_n, t_n) must satify conditions below.
  (by complementary slackness)

$$
\begin{cases}
\lambda_n = 0, & \longrightarrow \quad t_n(w*^t x_n + b) > 1 \\
0 < \lambda_n < C & \longrightarrow \quad t_n(w*^t x_n + b) = 1 \\
\lambda_n = C, & \longrightarrow \quad t_n(w*^t x_n + b) = 1 - \xi_n^* < 1
\end{cases}
$$

[12] Objective for Maximum Margin (convergence test)

- these conditions can be used to test convergence.

$$\begin{cases} \lambda_n = 0, & \longrightarrow \quad t_n(w*^t x_n + b) > 1 \\ 0 < \lambda_n < C & \longrightarrow \quad t_n(w*^t x_n + b) = 1 \\ \lambda_n = C, & \longrightarrow \quad t_n(w*^t x_n + b) = 1 - \xi_n^* < 1 \end{cases}$$

- Complexity Control by "C"

[1] Objective for Maximum Margin (**S**upport **V**ector **M**achine : SVM)

[primal problem]

$$w*, b* = \arg\min_{w,b} C \cdot \sum_{n=1}^{N} \xi_n + \frac{1}{2}||w||^2$$

$$\text{subject to} \quad t_n(w^t x_n + b) \geq 1 - \xi_n \quad \forall n$$

$$\text{subject to} \quad \xi_n \geq 0 \quad \forall n$$

- as C gets larger, $\sum_{n=1}^{N} \xi_n$ gets sallers: small margin / reduce train error
- as C gets smaller, $\sum_{n=1}^{N} \xi_n$ gets larger: large margin / increase train error

[2] Objective for Maximum Margin (**S**upport **V**ector **M**achine : SVM)

1. large C value (soft SVM) results in overfitting. (T/F)

2. large C value results in underfitting. (T/F)

3. when data is linearly separable, soft margin and hard margin will result in the same classifier. (T/F)

4. when data is not linearly separable then no way for a hard margin classifier converges. (T/F)

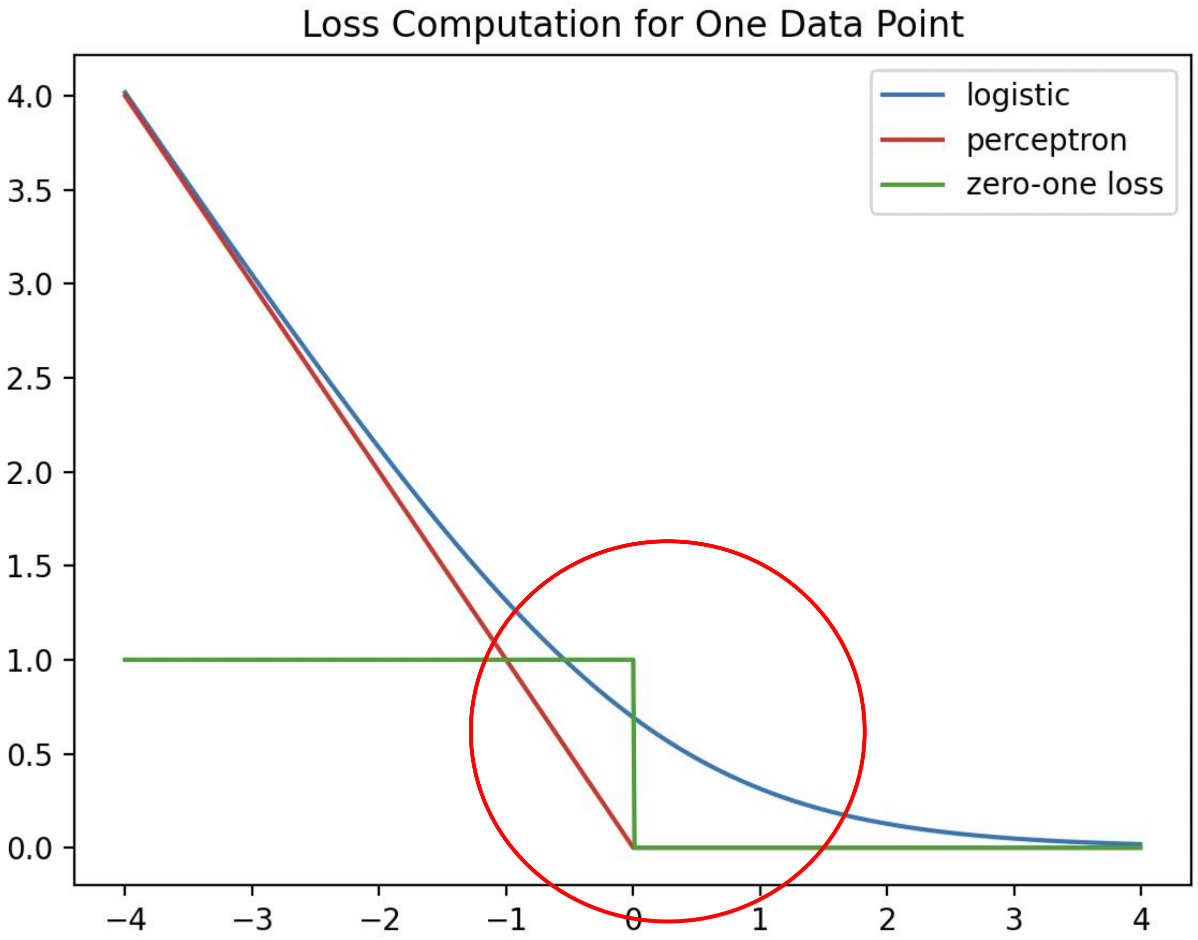
5. a hard margin SVM is sensitive to outliers. (T/F)

6. a soft margin SVM is sensitive to outliers. (T/F)

7. if C is too large, then there is a chance that the algorithm may not converge. (T/F)

- One Data Point Loss Comparison: soft SVM, perceptron, logistic regression

** recall: Logistic Sigmoid Regression vs Perceptron (loss comparison)



- perceptron does not penalize a small margin while logistic promotes a large margin.
- for $yw^tx < 0$ (misclassification), the perceptron and logistic loss behavior is asymptotically similar.

[1] One Data Point Loss Comparison (soft SVM Loss for a single point)

- Hinge Loss: $\left[1 - t(w^t x + b)\right]_+$

$$\text{soft-SVM}(x,t) = C \cdot \left[1 - t(w^t x + b)\right]_+ + ||w||^2$$

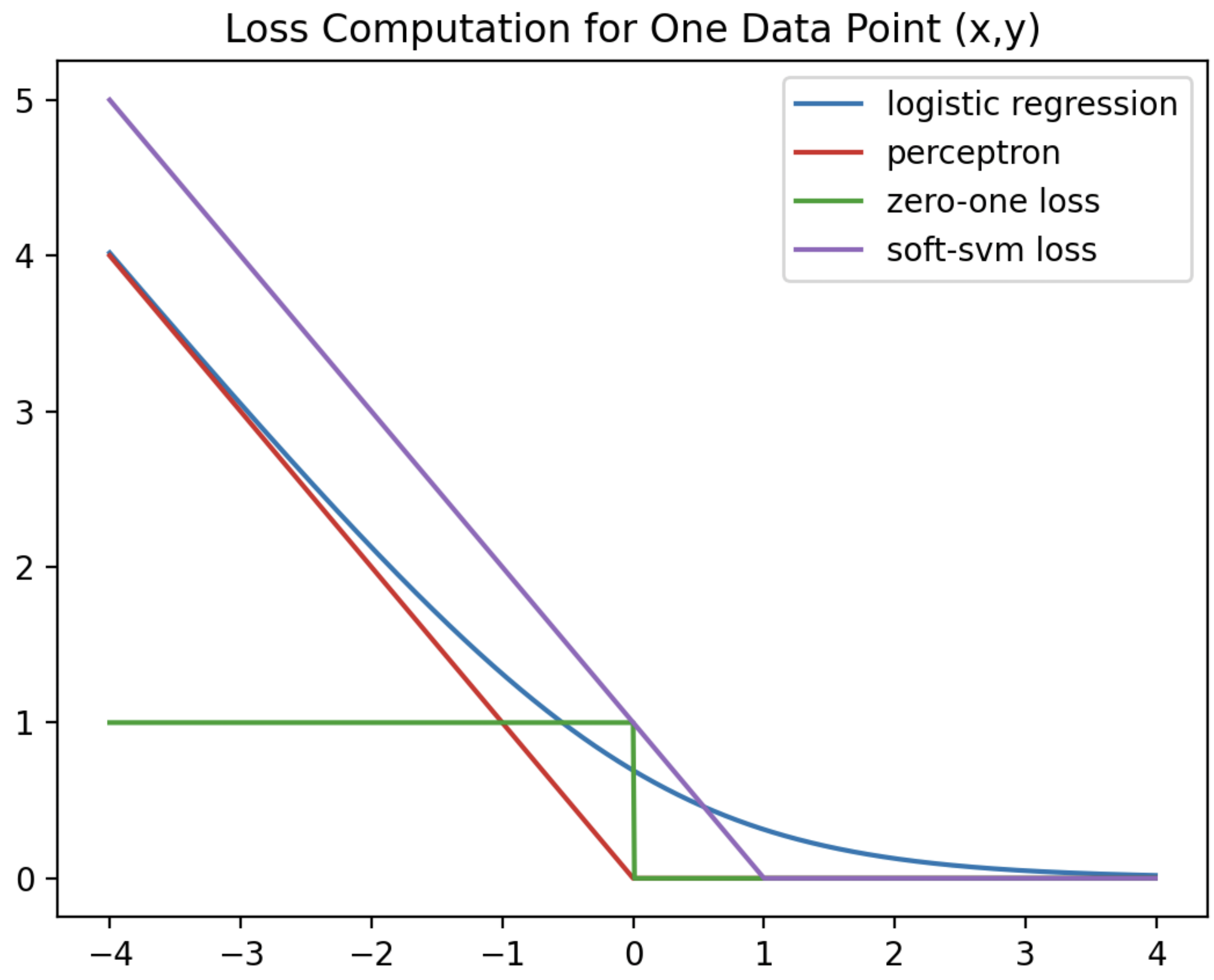
- given (x, t)
  three possible cases
  depending on w and b

$$t(w^t x + b) > 1 \quad \forall n \longrightarrow \xi = 0$$

$$t(w^t x + b) = 1 \quad \forall n \longrightarrow \xi = 0$$

$$t_n(w^t x + b) = 1 - \xi \quad \forall n \longrightarrow \xi > 0$$

[2] One Data Point Loss Comparison (soft SVM for a single point)



- In soft SVM, a hyperplane is defined by a subset of training samples while in logistic regression all training data points are involved (even though the contribution of the data points far from the hyperplane would be insignificant.)

- Solving Optimization Problem for SVM

$$\lambda *_{n=1}^{N} = \arg\max_{\lambda *} \sum_{n=1}^{N} \lambda *_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \lambda_n^* \lambda_m^* \cdot t_n \cdot t_m \cdot \kappa(x_n, x_m)$$

$$\text{subject to} \quad 0 \leq \lambda_n^* \leq C$$

$$\sum_{n=1}^{N} \lambda_n^* \cdot t_n = 0$$

[1] SMO (Sequential Minimal Optimization)

$$\lambda *_{n=1}^{N} = \arg\max_{\lambda *} \sum_{n=1}^{N} \lambda *_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \lambda_n^* \lambda_m^* \cdot t_n \cdot t_m \cdot \kappa(x_n, x_m)$$

$$\text{subject to} \quad 0 \leq \lambda_n^* \leq C$$

$$\sum_{n=1}^{N} \lambda_n^* \cdot t_n = 0$$

Q: why we don't use gradient decent algorithm?

[1] SMO (Sequential Minimal Optimization)

- SMO uses the idea of Coordinate Ascent Algorithm (iterative).
- update one coordinate at a time.

given an unconstraint problem: $\max_{\lambda} L(\lambda_1, \lambda_2, ..., \lambda_n)$

select a coordinate and find its minimum value and update the coordinate by the value while keeping other coordinates unchanged.

Loop until convergence: {

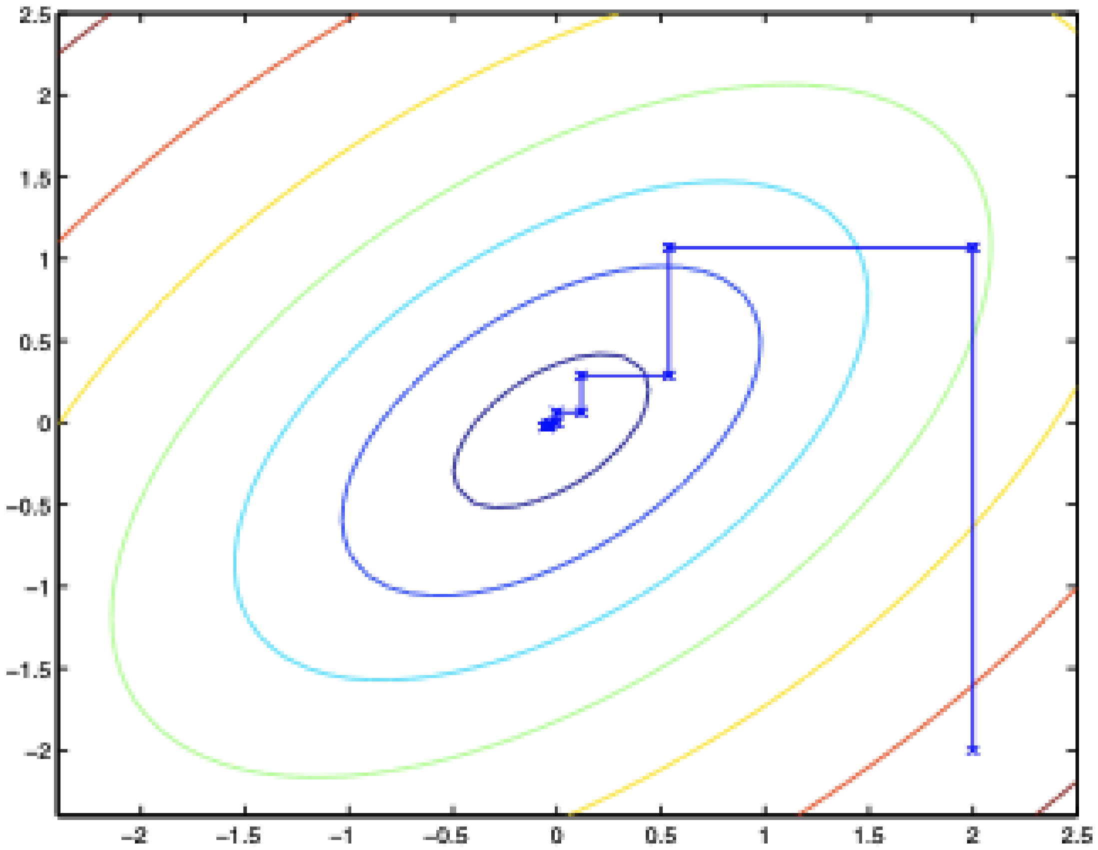
for n in $[1, 2, \ldots, N]$:

$$\lambda_n^* = \arg\max_{\lambda_n^*} L(\lambda_1, \lambda_2, ..., \lambda_n)$$

}

[2] SMO (2d: Coordinate Ascent Algorithm example)

https://see.stanford.edu/materials/aimlcs229/cs229-notes3.pdf



- Iteratively,
  the algorithm will find an optimum.

[3] SMO (applying constrained coordinate descent algorithm)

- dual problem set up given data and a kernel function
- given $\mathcal{D} : \{(x, t) : x \in \mathbb{R}^M \; and \; t \in \{-1, 1\}\}$ and $\kappa(x, x')$
  we defined a soft margin SVM dual function below.
  how can we find the optimal $\lambda_1, \lambda_2 \; \ldots, \quad \lambda_n$?

$$\lambda *_{n=1}^{N} = \arg\max_{\lambda *} \sum_{n=1}^{N} \lambda *_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \lambda_n^* \lambda_m^* \cdot t_n \cdot t_m \cdot \kappa(x_n, x_m)$$

$$\text{subject to} \quad 0 \leq \lambda_n^* \leq C$$

$$\sum_{n=1}^{N} \lambda_n^* \cdot t_n = 0$$

$$\lambda_1 t_1 + \lambda_2 t_2 = -\sum_{n=3}^{N} \lambda_n t_n = k$$

[4] SMO (applying a constrained coordinate descent algorithm)

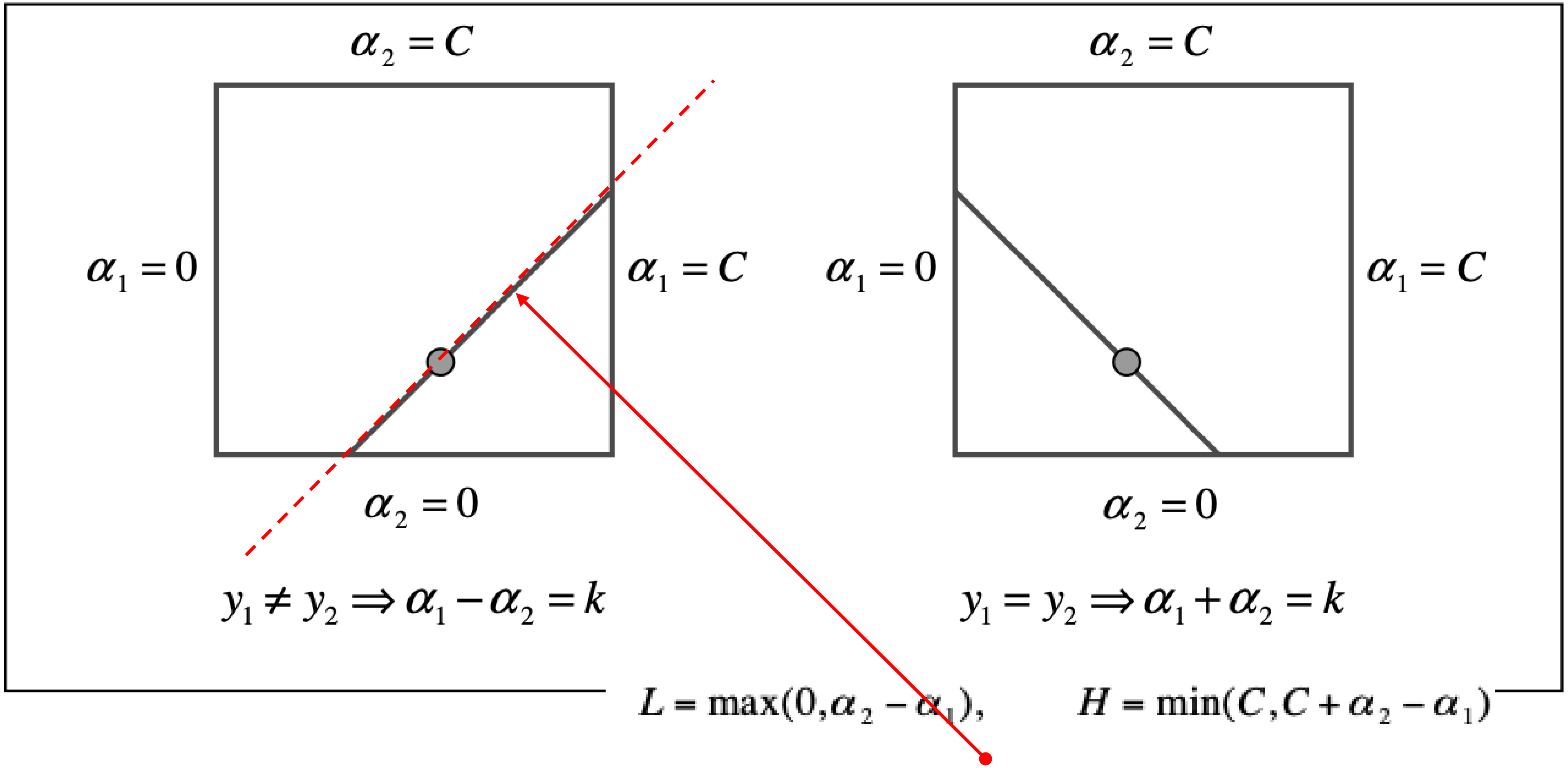
- we cannot update one coordinate λ_n because of the constraint.
- so apply the concept of coordinate ascent to solve dual problem,
  but we will update <u>two coordinates at at time.</u>

[5] SMO (applying a constrained coordinate descent algorithm)

- we are going to find a maximum along the line.($t_1\lambda_1 + t_2\lambda_2 = k$)
- still, we have another constraint,
  which is the box constraint. $0 \le \lambda_n^* \le C$

[6] SMO (setting the boundary conditions)

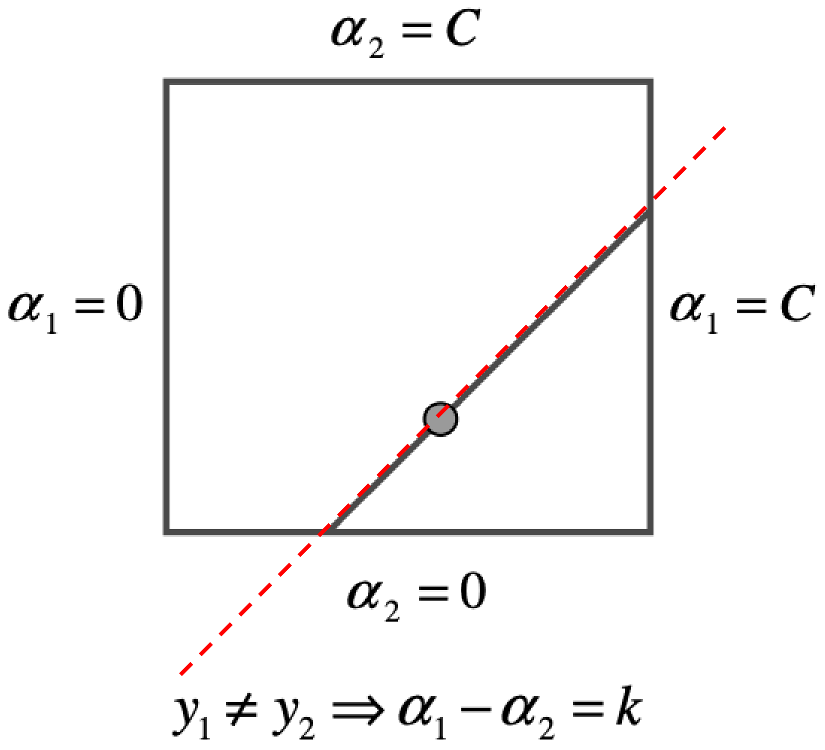
https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/



we need to find a maximum along the line.
$(y_1\alpha_1 + y_2\alpha_2 = k)$

[7] SMO (clipping)

after we compute a minimum along the line: α_1 - $\alpha_2 = k$
we need to check the solution is within the constraint bound.
otherwise, we need clipping!



$$\alpha_2^{\text{new,clipped}} = \begin{cases} H & \text{if} \quad \alpha_2^{\text{new}} \geq H; \\ \alpha_2^{\text{new}} & \text{if} \quad L < \alpha_2^{\text{new}} < H; \\ \text{L} & \text{if} \quad \alpha_2^{\text{new}} \leq L. \end{cases}$$

[8] SMO (finding the minimum)

Q: how to a minimum $\alpha_{2(new)}$ along the line : $y_1\alpha_1 + \ y_2\alpha_2 = k$

$$\alpha_{2new} = \alpha_{2old} + \frac{y_2(E_1 - E_2)}{\eta}$$

$$\eta = \kappa(x_1, x_1) + \kappa(x_2, x_2) - 2\kappa(x_1, x_2)$$

this update rule is derived by finding the α_2 minimum along the line.

$$E_1 = (\sum_{n=1}^{N} \lambda_n^* t_n \kappa(x_n, x_1) + b) - y_1$$

**classification error of data sample 1

$$E_2 = (\sum_{n=1}^{N} \lambda_n^* t_n \kappa(x_n, x_2) + b) - y_2$$

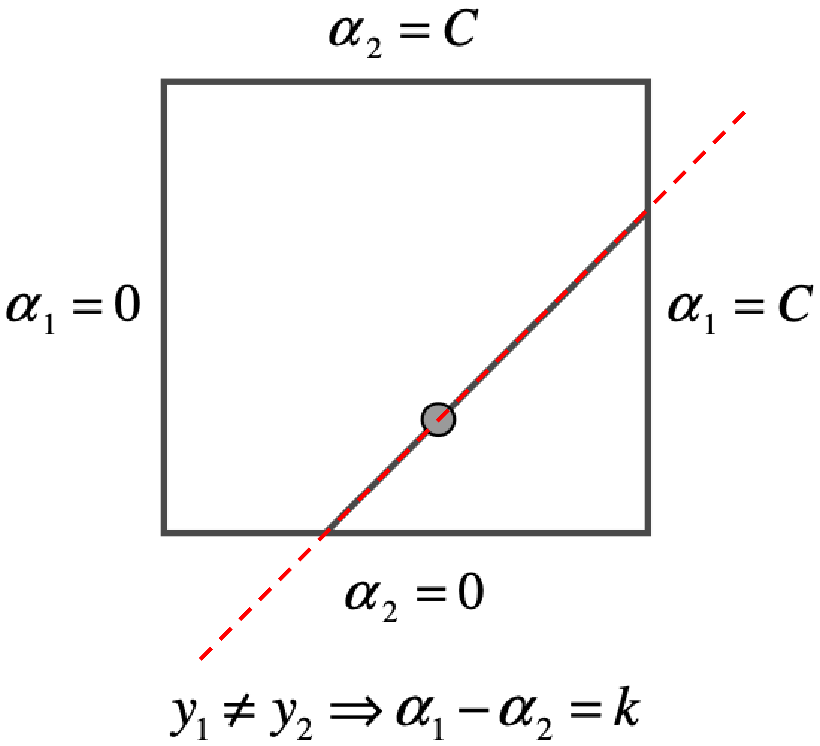
**classification error of data sample 2

for derivations:

https://dsmilab.github.io/Yuh-Jye-Lee/assets/file/teaching/2017_machine_learning/SMO_algorithm.pdf

[9] SMO (update another coordinate)

Q: once we find α_2, then we can update α_1?



[10] SMO (algorithm)

1. pick two alphas (α_1 / α_2).

2. define the range L / H for α_2.

3. compute minimum by

$$\alpha_{2new} = \alpha_{2old} + \frac{y_2(E_1 - E_2)}{\eta}$$

4. clipping by L / H

5. update α_1 by $\alpha_1 +/- \alpha_2 = k$

- repeat until all KKT conditions are satisfied for all N training samples within a preset tolerance ($10^{-3} \sim 10^{-2}$)

[11] SMO (heuristic to select two coordinates)

(1) first coordinate : choose any α that violates KKT condition
(2) second coordinate: choose a coordinate that maximize $|E_1 - E_2|$

**recall Objective for Maximum Margin (convergence test)

- these conditions can be used to test convergence.

  the tolerance (generally $10^{-3} \sim 10^{-2}$)

$$\begin{cases} \lambda_n = 0, & \longrightarrow \quad t_n(w*^t x_n + b) > 1 \\ 0 < \lambda_n < C & \longrightarrow \quad t_n(w*^t x_n + b) = 1 \\ \lambda_n = C, & \longrightarrow \quad t_n(w*^t x_n + b) = 1 - \xi_n^* < 1 \end{cases}$$