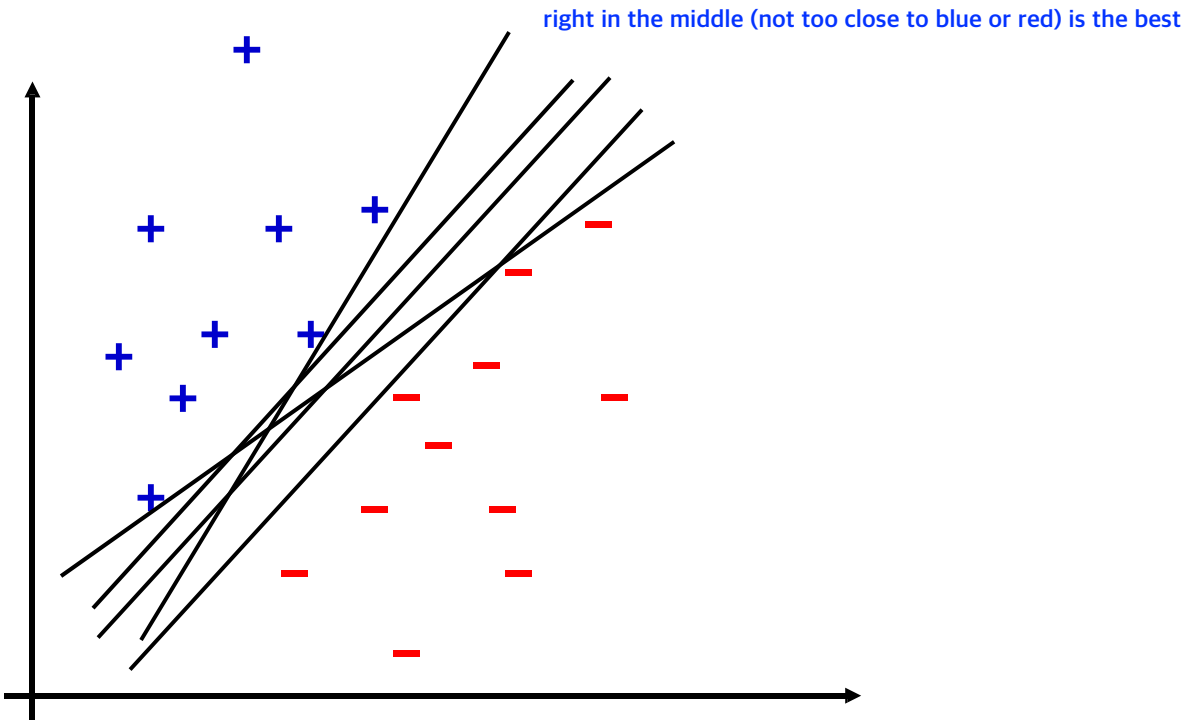
Support Vector Machines

Key concepts

- Functional and geometric margin of a classifier
- SVM objective: quadratic objective with linear constraints (turns into)
- Constrained optimization: Lagrangian
- Primal and Dual problem, the KKT conditions
- Solution characteristics of SVM
- Support vectors
- Kernel SVM

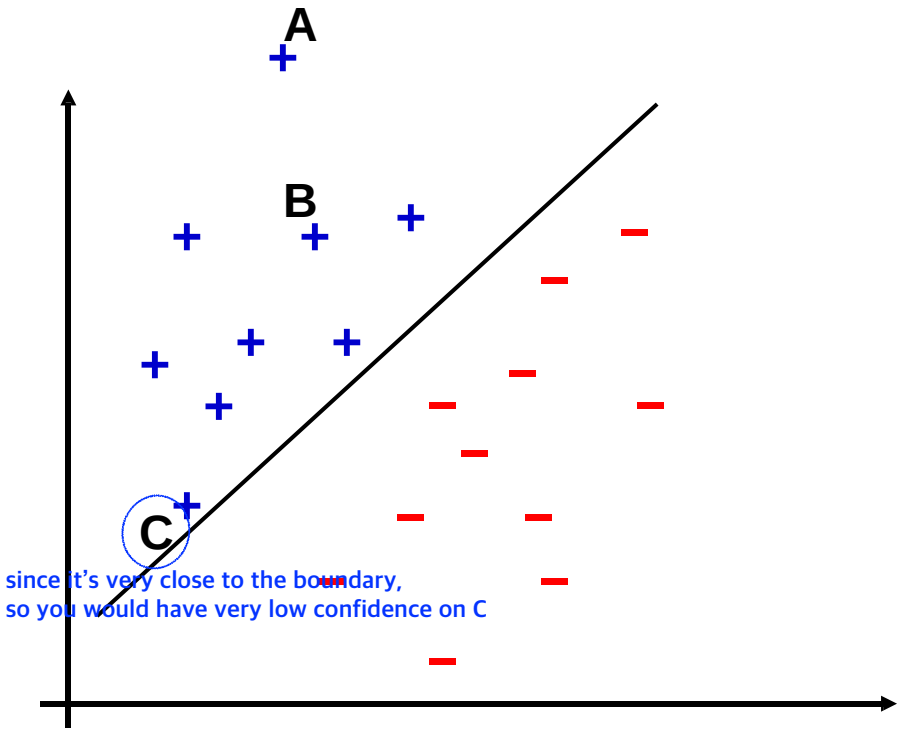
Linear Separators

- Which of the linear separators is optimal?



Intuition of Margin

- Consider points A, B, and C
- We are quite confident in our prediction for A because it is far from the decision boundary.
- In contrast, we are not so confident in our prediction for C because a slight change in the decision boundary may flip the decision.



Given a training set, we would like to make all predictions correct and confident! This leads to the concept of margin.

Functional Margin

- Given a linear classifier parameterized by $(\mathbf{w}, b)$, we define its functional margin w.r.t training example $(\mathbf{x}^i, y^i)$ as:

if the function margin is positive, it's correct prediction or on the decision boundary

$$\hat{\gamma}^i = y^i(\mathbf{w}^T\mathbf{x}^i + b)$$

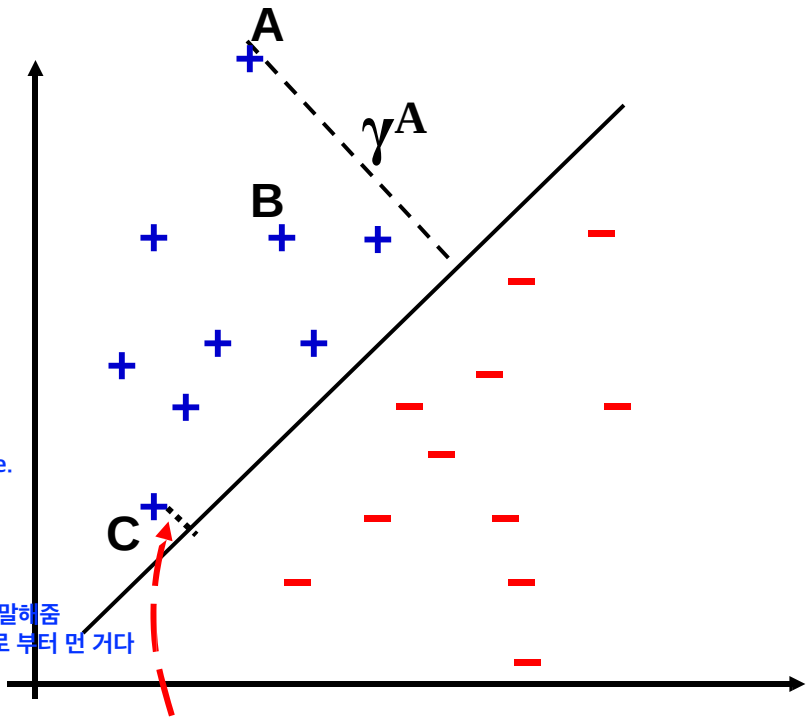
- If we rescale $(\mathbf{w}, b)$ by a factor α, functional margin gets multiplied by α
  - we can make it arbitrarily large without change anything meaningful
  - Instead, we will look at ***geometric margin***

Geometric Margin

- The geometric margin of $(\mathbf{w}, b)$ w.r.t. $\mathbf{x}^i$ is the distance from $\mathbf{x}^i$ to the decision boundary
- This distance can be computed as

$$\gamma^i = \frac{y^i(\mathbf{w}^T\mathbf{x} + b)}{\|\mathbf{w}\|}$$

margin is not distance.
It is sign(distance)
그래서
얘도 걍 w norm으로 나눈거기때문에
can be positive and negative
긍까 absolute value만 distance를 말해줌
absolute value가 크면, boundary로 부터 먼 거다



- Given training set $\boldsymbol{S} = \{(\mathbf{x}^i, y^i): i = 1, \ldots, N\}$, the geometric margin of the classifier w.r.t. $\boldsymbol{S}$ is

$$\gamma = \min_{i=1,\ldots,N} \gamma^i$$

이 마진이 클수록 좋은 classifier야~~

Points closest to the boundary are called Support vectors – we will see that these are the points that really matters

Maximum Margin Classifier

- Given a ***linearly separable*** training set $\boldsymbol{S} = \{(\mathbf{x}^i, y^i): i = 1, \ldots, N\}$, we would like to find a linear classifier with the maximum margin.
- This can be represented as an optimization problem.

maximizing the minimum of all the geometric margins

$$\max_{w,b,\gamma} \gamma$$

Nasty optimization problem! Let's make it look nicer!

$$\text{subject to: } \frac{y^i(\mathbf{w}^T\mathbf{x}^i + b)}{\|\mathbf{w}\|} \geq \gamma$$

c.f. 일반적으로 학습오차를 일반화오차가 최대로 줄어들때까지 줄이면 좋은 classifier

normalized version

- Let $\gamma' = \gamma \cdot \|w\|$, this is equivalent to

$$\max_{\mathbf{w},b,\gamma'} \frac{\gamma'}{\|\mathbf{w}\|}$$
$$\text{subject to: } y^i(\mathbf{w}^T\mathbf{x}^i + b) \geq \gamma' \quad \forall i = 1, \ldots, N$$

Maximum Margin Classifier

- Note that rescaling **w** and b (by $\frac{1}{\gamma'}$) will not change the classifier, we can thus further reformulate the optimization problem

$$\max_{\mathbf{w},b,\gamma'} \frac{\gamma'}{\|\mathbf{w}\|}$$

$$\text{subject to}: \; y^i(\mathbf{w}^T\mathbf{x}^i + b) \geq \gamma', \; i = 1, \ldots, N$$

norm = sqrt(sum of squre of every element)
그래서 always positive
그래서 이게 equivalent

$$\max_{\mathbf{w},b} \frac{1}{\|\mathbf{w}\|} \; (\text{or equivalently } \min_{\mathbf{w},b} \|\mathbf{w}\|^2)$$

$$\text{subject to}: \; y^i(\boldsymbol{w}^T\boldsymbol{x}^i + b) \geq 1, \; i = 1, \ldots, N$$

Maximizing the geometric margin is equivalent to minimizing the magnitude of **w** subject to maintaining a functional margin of at least 1

Solving the Optimization Problem

$$\min_{\mathbf{w},b} \|\mathbf{w}\|^2$$

$$\text{Subject to } y^i\left(\mathbf{w}^T\mathbf{x}^i + b\right) \geq 1, i = 1, \ldots, N$$

If all is correct has to have at least functional margin = 1

- This is a ***quadratic optimization problem*** with linear constraints.
- A well-known class of mathematical programming problems, several (non-trivial) algorithms exist.
  - One can use any of them to solve for $\mathbf{w}$ and b
- It is useful to first formulate an equivalent dual optimization problem, which serves two purposes:
  - To show that the solution for $\mathbf{w}$ can be expressed as weighted sum of subset of training examples (aka the support vectors)
  - For applying kernel trick for nonlinear svm

Aside: Constrained Optimization

라그랑지 승수법(Lagrange multiplier) :
어떤 함수(F)가 주어진 제약식(h)을 만족시키면서, 그 함수가 갖는 최대값 혹은 최소값을 찾고자할 때 사용한다.
L(x,λ) = F(x) + λ*h(x) 으로 표기하며, (x,λ)변수들로 각각 편미분 한 식이 0이 되는 값으로 해를 구한다.

- To solve the following optimization problem

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

so that

- Consider the following function known as the Lagrangian

$$\mathcal{L}(x, \alpha) = f(x) + \sum_i \alpha_i g_i(x) \quad s.t. \quad \alpha_i \geq 0$$

이거는 우리가 위에서 minimize하고싶은 애

이게 constrain

maximize할라믄 alpha must be 0

- The original optimization problem is equivalent to solving the following:

$$\min_x \max_\alpha \mathcal{L}(x, \alpha) \quad \text{subject to } \alpha_i \geq 0$$

maximize over alpha, then alpha 가 infinite여야하징

결국 constrain을 만족하는 모든 x에 대해서, 이 전체가 function of x가 된다
그래서 이게 original optimization에 equivalent가 되는그양

- By exchanging the order of min and max, we get the **dual problem**:

$$\max_\alpha \min_x \mathcal{L}(x, \alpha) \quad \text{subject to } \alpha_i \geq 0$$

Aside: Constrained Optimization

$$\text{Primal}: f^* = \min_{x} \max_{\alpha \geq 0} L(x, \alpha)$$

$$\text{Dual}: d^* = \max_{\alpha \geq 0} \min_{x} L(x, \alpha)$$

지은: 이게 x에 대해서 먼저 min을 풀고 -> solve min of function of x problem
그다음 max를 구하자낭? 그래서 듀얼인가방

Let x^* and α^* be the optimal and dual solution respectively, $f^* = d^*$ if $f(x)$ is convex and x^* and α^* satisfy the KKT conditions:

1. $\nabla L(x^*, \alpha^*) = 0$ --- zero gradient
2. $g(x^*) \leq 0$ --- primal feasibility
3. $\alpha^* \geq 0$ --- dual feasibility
4. $\alpha^* g(x^*) = 0$ --- complementary slackness

g(x*)가 < 0 면, constrain이 loose -> alpha has to be zero. 왜냠 4. 식을 봤을 때, g(x*)가 zero가 아니면 알파가 0여야지 곱하기가 0가 되자나
= 0 면, tight constrain -> alpha 가 non-zero. 같은 식으로, g(x*)가 0면, 알파가 0이 아니어도 곱하기는 걍 0 이되잔아ㅇㅇ쉽디?

Back to the Original Problem

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to}: 1 - y^i\left(\mathbf{w}^T\mathbf{x}^i + b\right) \leq 0, i = 1, \ldots, N$$

functional margin >= 1, so

The Lagrangian is

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{N} \alpha_i \left(1 - y^i\left(\mathbf{w}^T\mathbf{x}^i + b\right)\right) s.t., \alpha_i \geq 0$$

- We want to solve $\max_{\alpha \geq 0} \min_{\mathbf{w},b} \mathcal{L}(\mathbf{w}, b, \alpha)$

- Setting the gradient of $\mathcal{L}$ w.r.t. $\mathbf{w}$ and b to zero:

$$\mathbf{w} - \sum_{i=1}^{N} \alpha_i y^i \mathbf{x}^i = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{N} \alpha_i y^i \mathbf{x}^i$$

primer solution and dual solution has this connection/relation
you will converge to the same solution

$$\sum_{i=1}^{N} \alpha_i y^i = 0$$

위에거를 풀면,
이거가 be zero가 됨

The Dual Problem

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{N} \alpha_i \left(1 - y^i\left(\mathbf{w}^T\mathbf{x}^i + b\right)\right)$$

- Substitute $\mathbf{w} = \sum_{i=1}^{N} \alpha_i y^i \mathbf{x}^i$ into $\mathcal{L}$:

$$\begin{aligned} & L(\alpha) \\ &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y^i y^j < \mathbf{x}^i \cdot \mathbf{x}^j > + \sum_{i=1}^{N} \alpha_i \\ &- \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y^i y^j < \mathbf{x}^i \cdot \mathbf{x}^j > - b\sum_{i=1}^{N} \alpha_i y^i \quad = 0 \\ &= \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y^i y^j < \mathbf{x}^i \cdot \mathbf{x}^j > \end{aligned}$$

The Dual Problem

- The new objective function is in terms of α_i, known as the dual problem
- The original problem is known as the primal problem
- The objective function of the dual problem needs to be maximized!
- The dual problem is therefore:

$$\max L(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y^i y^j < \mathbf{x}^i \cdot \mathbf{x}^j >$$

$$\text{subject to} \quad \alpha_i \geq 0, i = 1, \ldots, n, \qquad \sum_{i=1}^{N} \alpha_i y^i = 0$$

Properties of α_i when we introduce the Lagrange multipliers

The result when we differentiate the original Lagrangian w.r.t. b

The Dual Problem

$$\max L(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y^i y^j < \mathbf{x}^i \cdot \mathbf{x}^j >$$

$$\text{subject to} \quad \alpha_i \geq 0, i = 1, \ldots, n, \qquad \sum_{i=1}^{N} \alpha_i y^i = 0$$

- This is also a quadratic programming (QP) problem
  - so A global maximum of α_i can always be found
- **w** can be recovered by $\mathbf{w} = \sum_{i=1}^{N} \alpha_i y^i \mathbf{x}^i$
- b can also be recovered as well (wait for a bit)

Characteristics of the Solution

- Many of the α_i are zero --- sparse solution
- **w** is a linear combination of only a small number of data points
- The KKT conditions requires that:

$$\alpha_i \geq 0, i = 1, \ldots, n$$

Dual feasibility

$$y^i \left(\sum_{j=1}^{n} \alpha_j y^j < \mathbf{x}^j \cdot \mathbf{x}^i > + b \right) \geq 1 \, , i = 1, \ldots, n$$

Primal feasibility: Functional margin ≥ 1

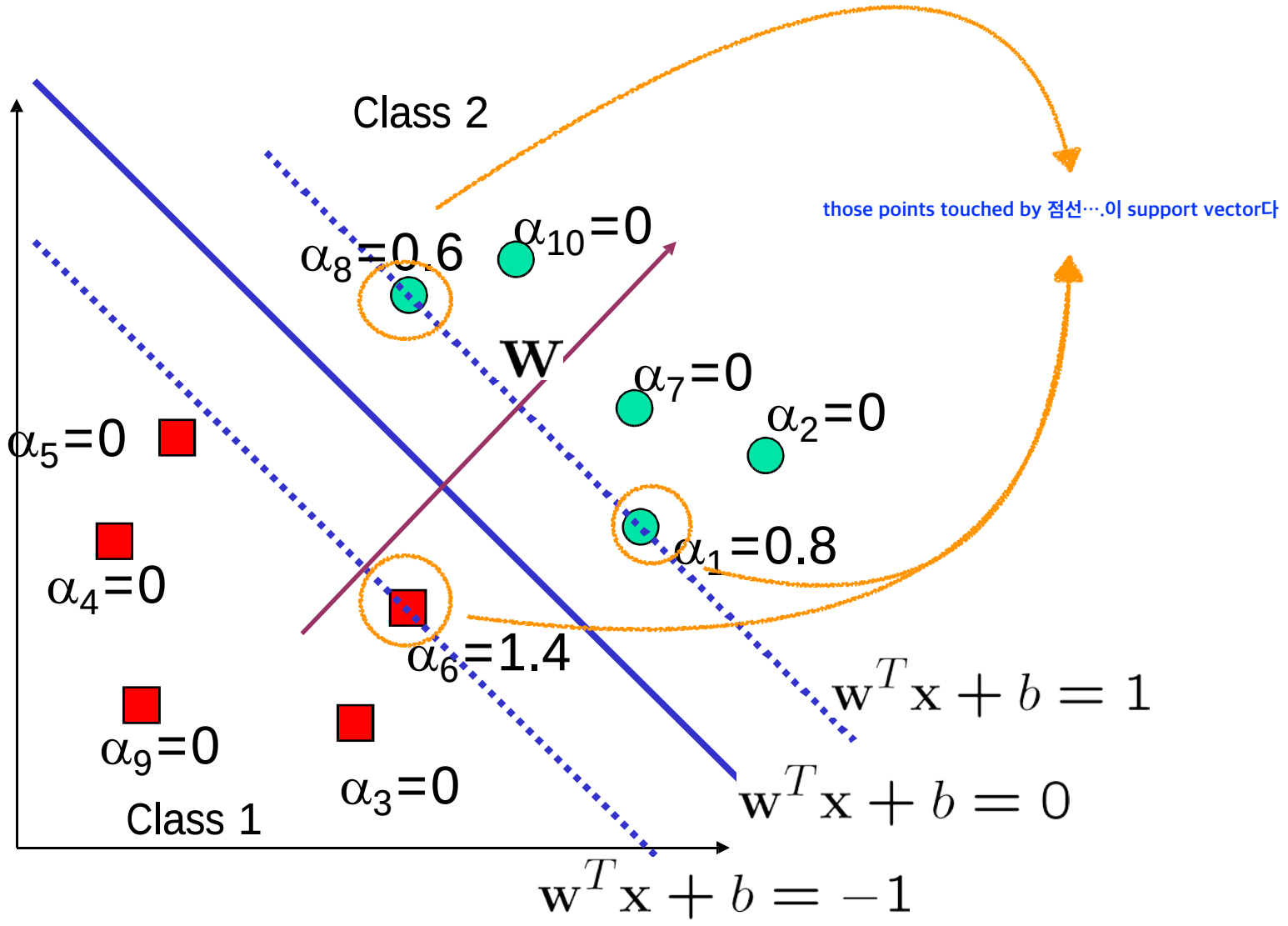
if functional margin >= 1, then alpha has to be zero

$$\alpha_i \left(y^i \left(\sum_{j=1}^{n} \alpha_j y^j < \mathbf{x}^j \cdot \mathbf{x}^i > + b \right) - 1 \right) = 0, i = 1, \ldots, n$$

긍까 only tight constrain만 non zero alpha를 가지는그야~~~!!
so what? is she saying it should be tight or what ??

Complemetary slackness: α is nonzero only when functional margin = 1

A Geometrical Interpretation



Support Vectors

- $\mathbf{x}^i$ with non-zero $\alpha's$ are called support vectors (SV)

- The decision boundary is determined only by the SV's

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y^i \mathbf{x}^i$$

- Note that we know that for support vectors the functional margin = 1

- We can use this information to solve for b

Classifying new examples

For classifying with a new input **x**

위에서 말한 정보를 이용해서 이렇게 컴퓨트 할 수 잇슴다!

- Compute $$\mathbf{w}^T\mathbf{x} + b = \sum_{i=1}^{N} \alpha_i y^i < \mathbf{x}^i \cdot \mathbf{x} > + b$$

- Note: no need to form **w** explicitly, rather, classify **x** by taking a weighted sum of **its dot products with the support vectors** (useful for generalizing from inner product to kernels)

브리프 인트로덕션 for many approaches for QP 걍 대충하고 넘어감

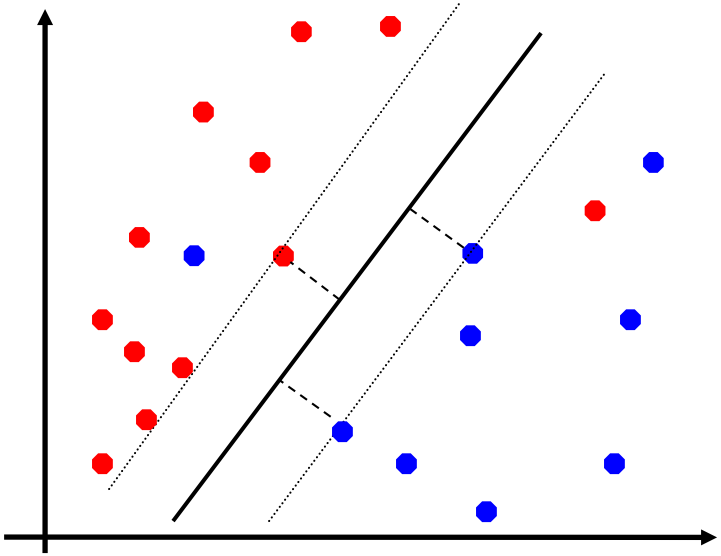
Solving the QP optimization problem

- Many approaches have been proposed for QP
  - Loqo, cplex, etc. (see http://www.numerical.rl.ac.uk/qp/qp.html)
- Early work focuses on “interior-point” methods
  - Start with an initial solution that can violate the constraints
  - Improve this solution by optimizing the objective function and/or reducing the amount of constraint violation
- Stochastic sub-gradient descent has been shown to lead to extremely efficient primal solver for large scale problems
- In practice, one can just regard the QP solver as a “black-box” without bothering how it works, but depending on the scale of the problem some solvers might be more appropriate than others

Non-separable Data

What if the data is not linearly separable?

- The solution does not exist
- i.e., the set of linear constraints are not satisfiable
- But we should still be able to find a good decision boundary



요딴식으루 되면 solution이 없다는거야~~~~
그럼 이거를 고차원으로 mapping을 해서 using kernel function
바운더리를 찾는당!

Solution:

- Project the data onto higher dimensional space
- Via kernel function

Kernel SVM

Linear SVM:

$$\max L(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y^i y^j < \mathbf{x}^i \cdot \mathbf{x}^j >$$

$$\text{subject to} \quad \alpha_i \geq 0, i = 1, ..., n, \qquad \sum_{i=1}^{N} \alpha_i y^i = 0$$

Replace dot product with kernel function

Kernel SVM:

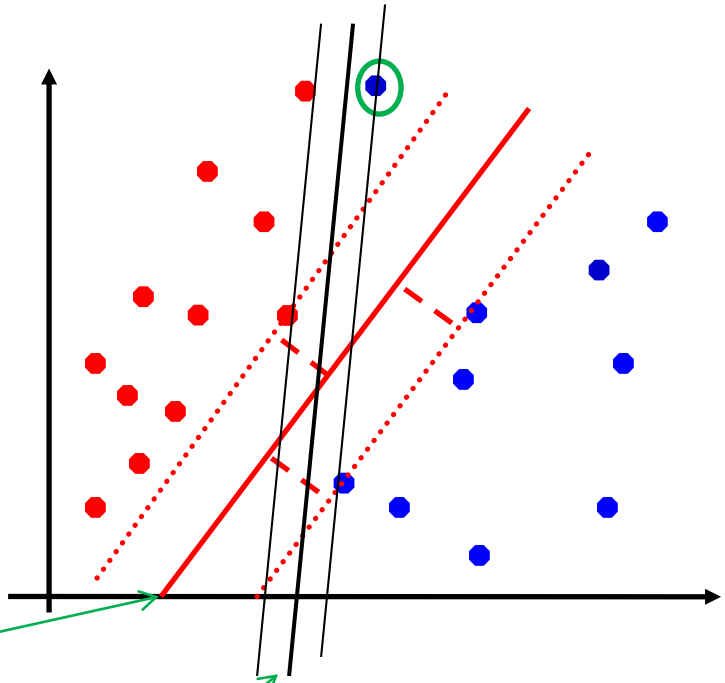
$$\max L(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y^i y^j \; K(\mathbf{x}^i, \mathbf{x}^j)$$

$$\text{subject to} \quad \alpha_i \geq 0, i = 1, ..., n, \qquad \sum_{i=1}^{N} \alpha_i y^i = 0$$

Maximum margin overfits to outliers

maximum margin이 overfit to outlier할 수 있다는그양!!!!
문제점(issue)양!!

Consider the blue point circled out. It is an outlier that is labeled as blue but really should belong to red

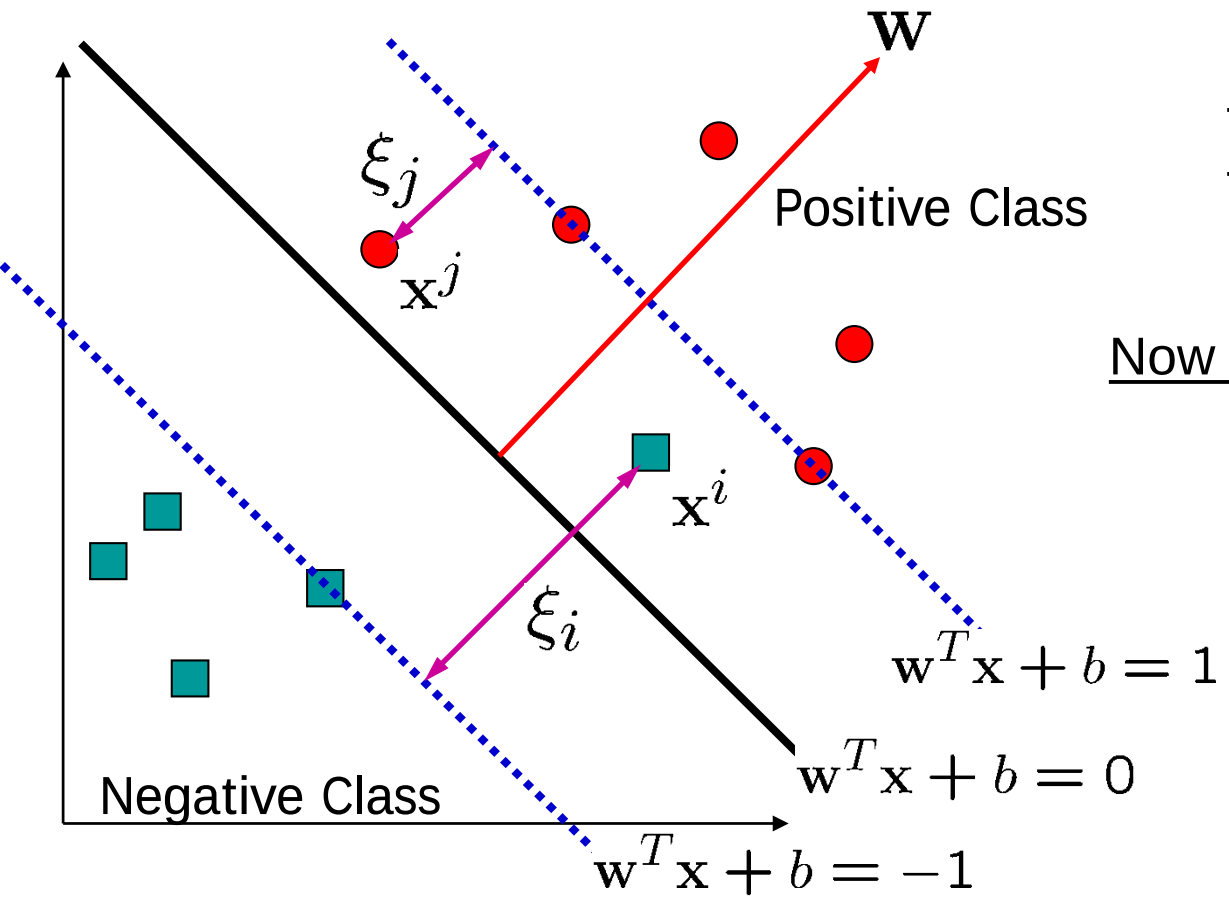


We would like to learn a boundary that ignores the outliers

But the margin will be defined by the outlier and we instead learn a boundary that overfit to the outliers

Soft Margin

- Allow functional margins to be less than 1



Originally functional margins need to satisfy:

$$y^i(\mathbf{w}^T\mathbf{x}^i+b)\geq 1$$

Now we allow it to be less than 1:

$$y^i(\mathbf{w}^T\mathbf{x}^i+b)\geq 1-\xi_i$$

$$\xi_i \geq 0$$

싸이i = ____
이 싸이는 fixed value는 아니양 function같은거래

The objective changes to:

$$\min_{\mathbf{w},b,\xi_i} \|\mathbf{w}\|^2 + c\sum_{i=1}^{N}\xi_i$$

Soft-Margin Maximization

$$\min_{\mathbf{w},b} \|\mathbf{w}\|^2$$
$$\text{subject to: } y^i(\mathbf{w}\cdot\mathbf{x}^i + b) \geq 1, \quad i = 1,\cdots,N$$

Slack variables

$$\min_{\mathbf{w},b} \|\mathbf{w}\|^2 + c\sum_{i=1}^{N}\xi_i$$
$$\text{subject to: } y^i(\mathbf{w}\cdot\mathbf{x}^i + b) \geq 1-\xi_i, \quad i = 1,\cdots,N$$
$$\xi_i \geq 0, \quad i = 1,\cdots,N$$

- This allows some functional margins < 1 (could even be < 0)
- The ξ_i 's can be viewed as the "errors" of our *fat* decision boundary
- Adding ξ_i 's to the objective function to minimize errors
- We have a tradeoff between making the decision boundary fat and minimizing the error
- Parameter ***c*** controls the tradeoff:
  - Large c: ξ_i's incur large penalty, so the optimal solution will try to avoid them
  - Small c: small cost for ξ_i's, we can sacrifice some training examples to have a large classifier margin

Soft Margin SVM: Regularized Hinge loss

$$\min_{\mathbf{w},b} \|\mathbf{w}\|^2 + c\sum_{i=1}^{N} \xi_i$$

$$\text{subject to } y^i\left(\mathbf{w}^T\mathbf{x}^i + b\right) \geq 1 - \xi_i,$$

$$\xi_i \geq 0, \forall i = 1, \ldots, N$$

w,b가 fixed면 싸이도 determined

Is equivalent to:

c: how much we focus on loss term.(heavier penelty)
c가 크면 loss에 그만큼 더 무게를 둔다는거고
c가 작으면 regularization term에 more ㅇㅇ

loss term

regularization term

$$\min_{w,b} \|\mathbf{w}\|^2 + c\sum_{i}^{N} \max\left(0, 1 - y^i\left(w^T x^i + b\right)\right)$$

글서 larger c, more overfit

λ

왜 이걸 알려주냐?
얘가 Support vector machine에 엄청 close하고
뭐 불라불라루라루라 어쩌구저쩌구 니나노~~

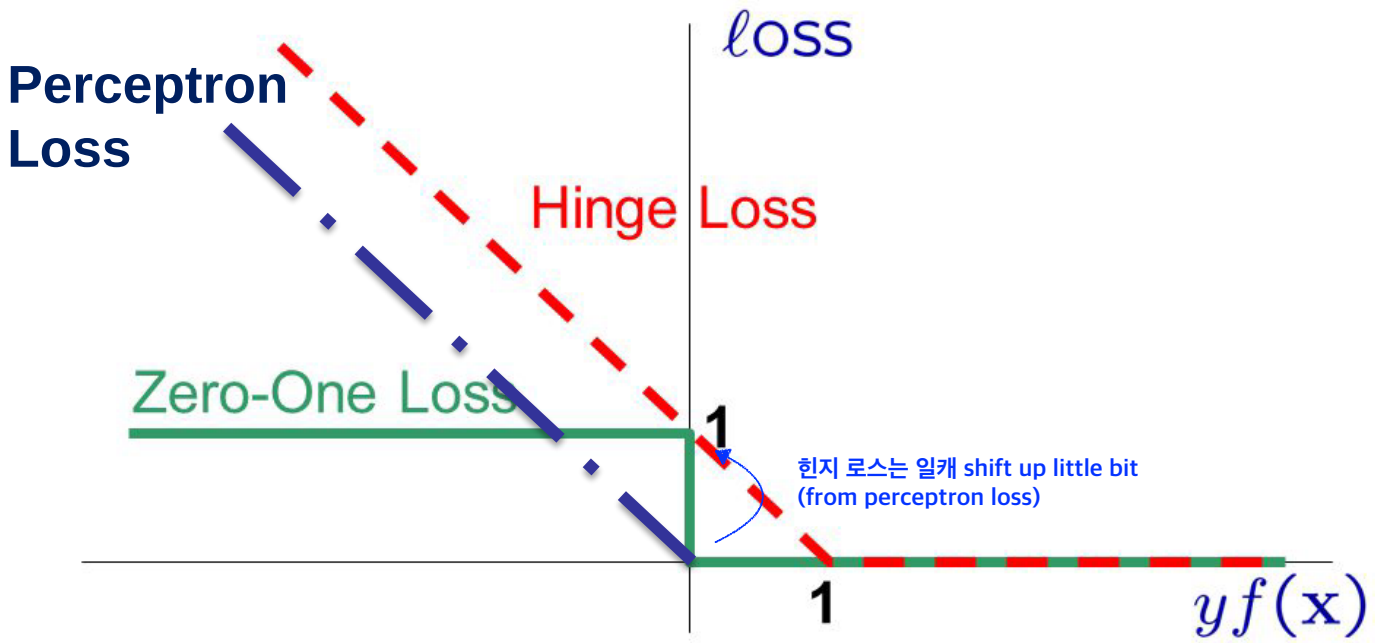
텍스트

L_2 Regularization

Hinge loss

이게 negative면 already satisfy margin 뭐시기 퓌저빌리틴ㄴ감 위에 말한거 ㅇㅇ

Different Loss functions



Solutions to soft-margin SVM

$$w = \sum_{i=1}^{N} \alpha_i y^i x^i, \quad \textbf{s.t.} \sum_{i=1}^{N} \alpha_i y^i = 0$$

No soft margin

$$w = \sum_{i=1}^{N} \alpha_i y^i x^i, \quad \textbf{s.t.} \sum_{i=1}^{N} \alpha_i y^i = 0 \ \textbf{and} \ 0 \le \alpha_i \le c$$

as a result, we limit support vector

With soft margin

- *c* effectively puts a **box constraint** on α, the weights of the support vectors
  그래서 c가 0이면 all points become support vector
  Larger C value leads to smaller number of support vector
- It limits the influence of individual support vectors (maybe outliers)
- In practice, c is a parameter to be set, similar to *k* in k-nearest neighbor
- It can be set using cross-validation

Kernel SVM with soft margin

$$\max L(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y^i y^j \; K(\mathbf{x}^i, \mathbf{x}^j)$$

subject to $0 \leq \alpha_i \leq c, \; i = 1 \ldots, N;$ $\sum_{i=1}^{N} \alpha_i y^i = 0$

Summary of SVM

- SVM aims to find the max margin linear separator
- Soft margin SVM can be interpreted as:
  - Introducing slack to the hard margin constraints – C-SVM, where C is the penalty weight for the accumulative slack
  - Minimizing $L2$ regularized hinge loss - λ-SVM, where λ is the regularization parameter
- Large C (or equivalently small λ): increased overfitting
- Small C (or equivalently large λ): decreased overfitting
- By solving the dual problem with the kernel trick, we can learn max margin separator in the mapped nonlinear space