UVA CS 4774: Machine Learning

S4: Lecture 21 Extra: (SVM Optimization and Dual basic)

Module IV Extra

Dr. Yanjun Qi

University of Virginia

Department of Computer Science

What Left in SVM?

- Support Vector Machine (SVM)
  - ✓ History of SVM
  - ✓ Large Margin Linear Classifier
  - ✓ Define Margin (M) in terms of model parameter
  - ✓ Optimization to learn model parameters (w, b)
  - ✓ Linearly Non-separable case (soft SVM)
  - ✓ Optimization with dual form
  - ✓ Nonlinear decision boundary
  - ✓ Practical Guide

$K(x, z)$

This: Kernel Support Vector Machine

Data

↓

Task

↓

Representation

↓

Score Function

↓

Search/Optimization

↓

Models, Parameters

Tabular / Text / Image/ Structured

↓

Classification / Regression / Ranking

↓

Kernel Trick Func K(xi, xj)

$$K(\mathbf{x}, z) := \Phi(\mathbf{x})^T \Phi(z)$$

↓

Margin + Hinge Loss

↓

QP with Dual form

$$w = \sum_i \alpha_i x_i y_i$$

↓

Dual Weights

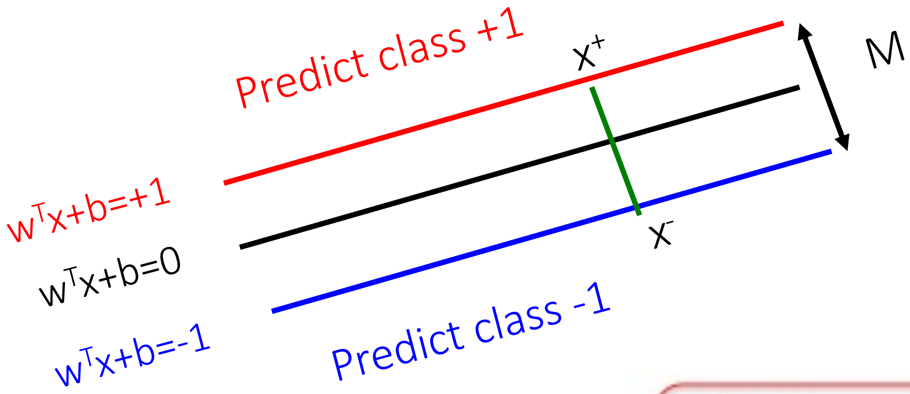
$$\underset{\mathbf{w},b}{\operatorname{argmin}} \sum_{i=1}^{p} w_i^2 + C\sum_{i=1}^{n} \varepsilon_i$$

subject to $\forall \mathbf{x}_i \in Dtrain : y_i\left(\mathbf{x}_i \cdot \mathbf{w} + b\right) \geq 1 - \varepsilon_i$

$$\max_\alpha \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i{}^T \mathbf{x}_j$$

$$\sum_i \alpha_i y_i = 0, \qquad \alpha_i \geq 0 \qquad \forall i$$

Optimization Step
i.e. learning optimal parameter for SVM



$$M = \frac{2}{\sqrt{w^T w}}$$

1. Correctly classifies all points
2. Maximizes the margin (or equivalently minimizes w^Tw)

Optimization Reformulation

1. Correctly classifies all points
2. Maximizes the margin (or equivalently minimizes w^Tw)

Min $(w^Tw)/2$

subject to the following constraints:

$\rightarrow$ pos $y_i = 1$, $w^T x_i + b \geq 1$

$y_i(w^T x_i + b) \geq 1$

For all x in class + 1

$w^Tx+b >= 1$ $y_i = 1$

A total of n constraints if we have n input samples

For all x in class - 1

$w^Tx+b <= -1$ $y_i = -1$

$\rightarrow$ neg $y_i = -1$, $w^T x_i + b \leq -1$

$y_i(w^T x_i + b) \geq 1$

Optimization Reformulation

1. Correctly classifies all points
2. Maximizes the margin (or equivalently minimizes $w^T w$)

Min $(w^T w)/2$

subject to the following constraints:

For all x in class + 1

$w^T x + b >= 1$

For all x in class - 1

$w^T x + b <= -1$

A total of n constraints if we have n input samples

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \sum_{i=1}^{p} w_i^2 / 2$$

$$\text{subject to } \forall \mathbf{x}_i \in Dtrain : y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1$$

Linearly Non separable case

- So far we assumed that a linear hyperplane can perfectly separate the points
- But this is not usually the case

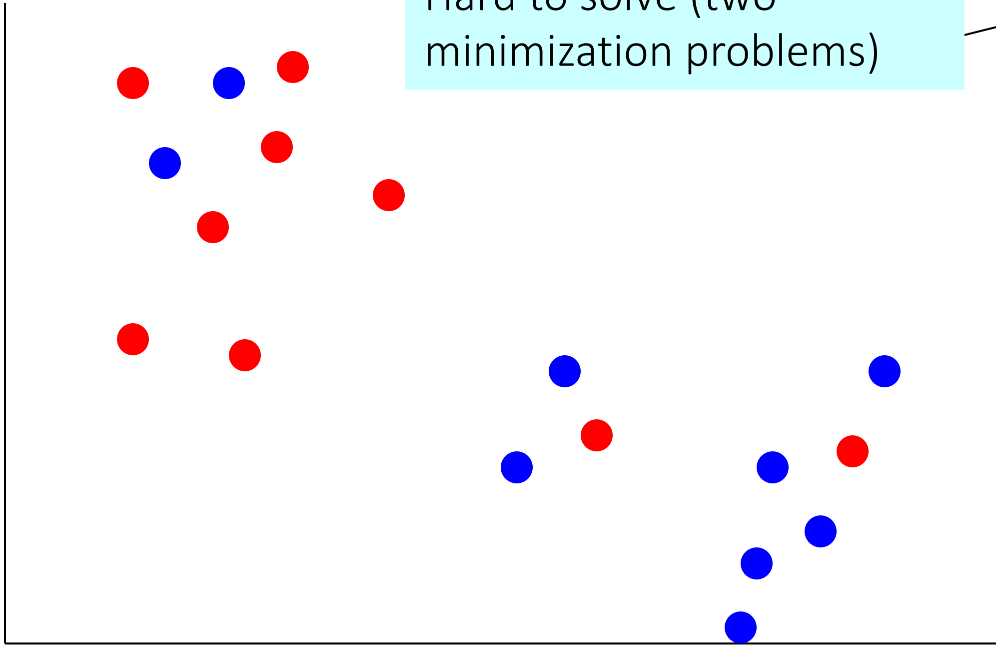
- noise, outliers

How can we convert this to a QP problem?

- Minimize training errors?

$$\begin{cases} \min w^T w/2 \\ \min \#\text{errors} \end{cases}$$

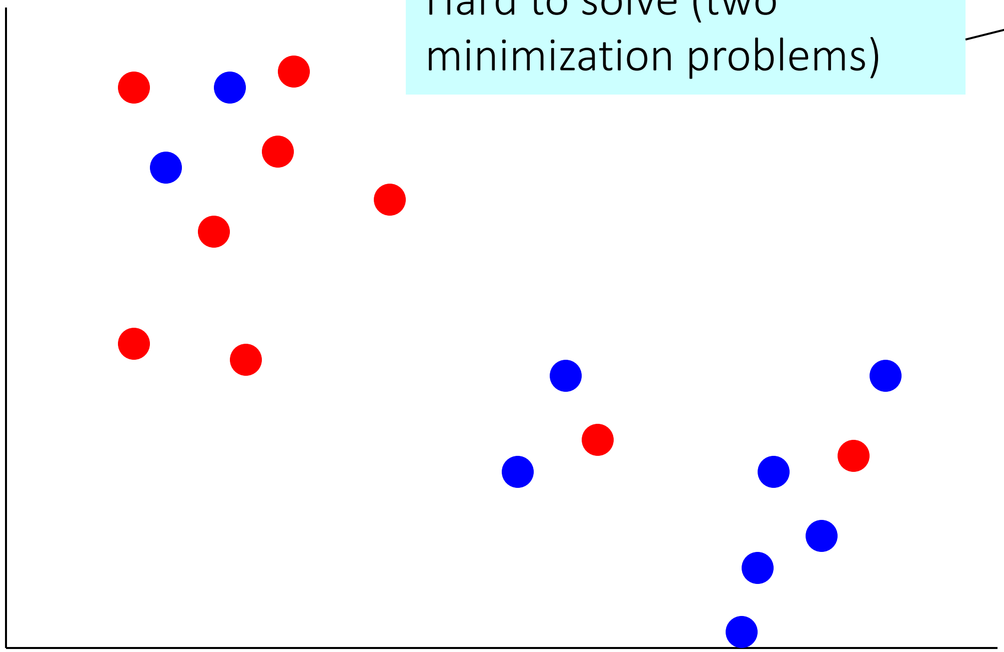
Hard to solve (two minimization problems)



Linearly Non separable case

- So far we assumed that a linear plane can perfectly separate the points
- But this is not usally the case

 - noise, outliers



Hard to solve (two minimization problems)

How can we convert this to a QP problem?

- Minimize training errors?

 min $w^T w/2$

 min #errors

- Penalize training errors:

 min $w^T w/2 + C^*$(#errors)

Hard to encode in a QP problem

$C \nearrow$ penalize errors more

SVM: $\min_{\vec{w}, b} \; w^T w + \underbrace{C}_{} (\# \text{ errors})$

Ridge Regression:

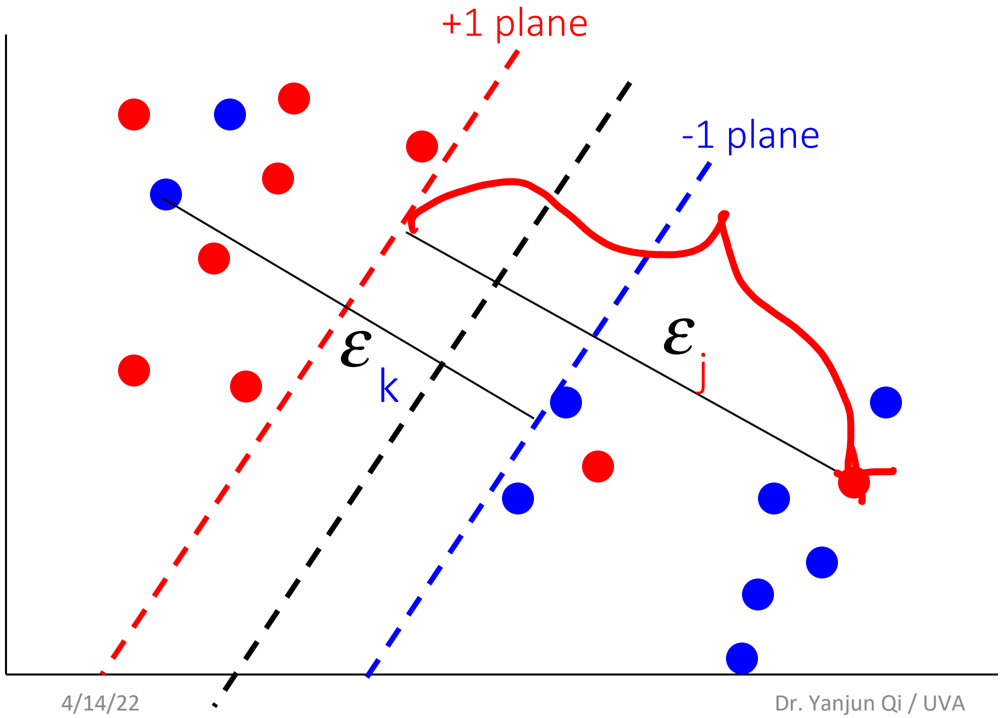
$\min_{\vec{\theta}} \; \underbrace{\lambda}_{} \theta^T \theta + J(\theta)$

Linearly Non separable case

- Instead of minimizing the number of misclassified points we can minimize the distance between these points and their correct plane

The new optimization problem is:

$$\min_w \frac{w^T w}{2} + C\sum_{i=1}^{n} \varepsilon_i$$

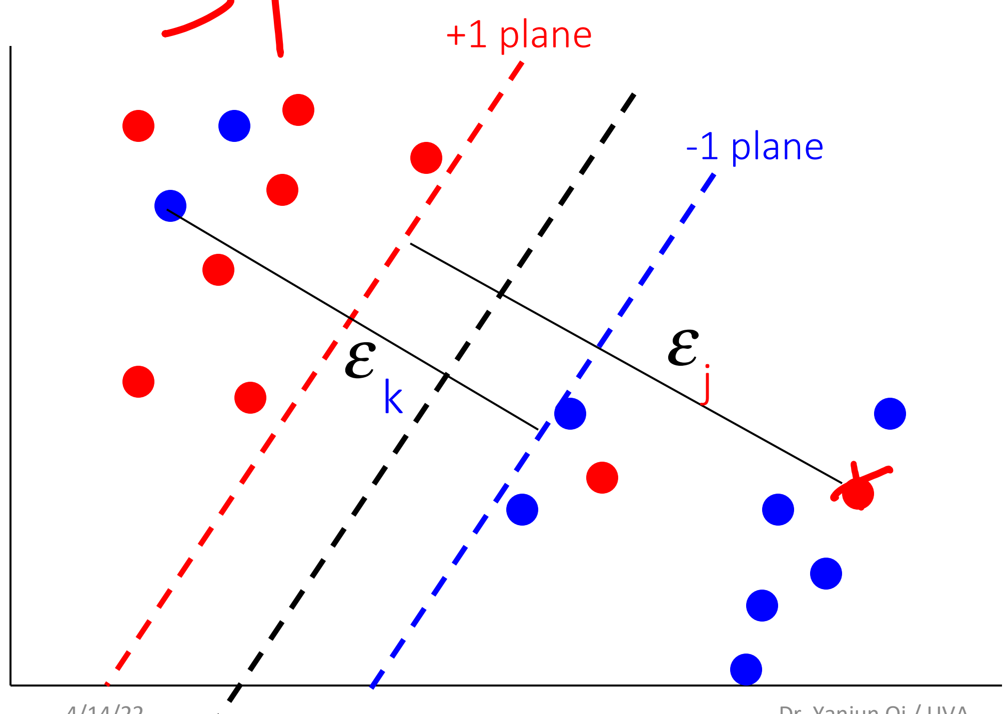


Linearly Non separable case

- Instead of minimizing the number of misclassified points we can minimize the distance between these points and their correct plane

w^Tx+b

> 1



The new optimization problem is:

$$\min_w \frac{w^T w}{2} + C\sum_{i=1}^{n} \varepsilon_i$$

subject to the following inequality constrai

For all x_i in class + 1

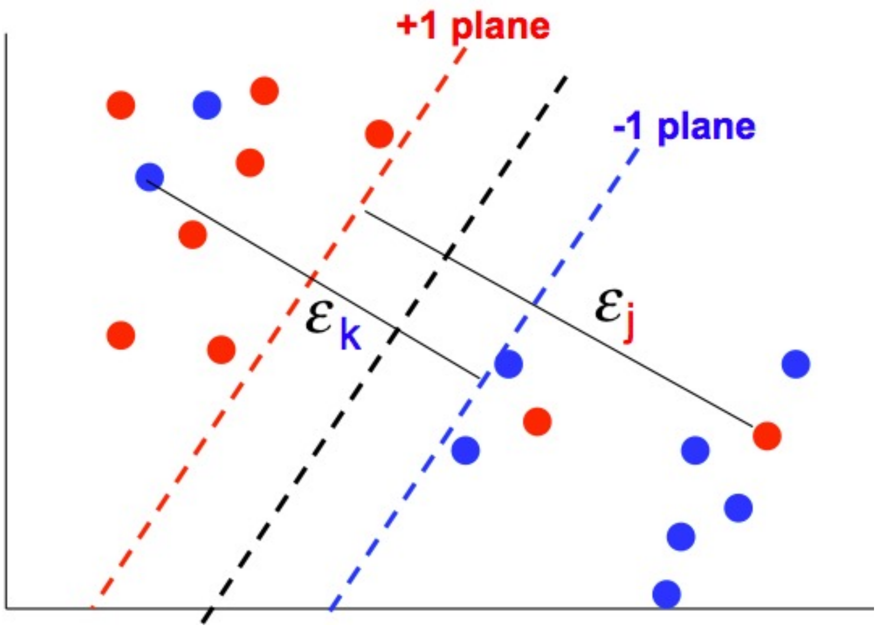
$w^T x_i + b >= 1 - \varepsilon_i$ $\quad \varepsilon_i \geq 0$

For all x_i in class - 1

$w^T x_i + b <= -1 + \varepsilon_i$

Wait. Are we missing something?

Final optimization for linearly non-separable case



The new optimization problem is:

$$\min_w \frac{w^T w}{2} + C\sum_{i=1}^{n} \varepsilon_i$$

subject to the following inequality constraints:

For all x_i in class + 1

$w^T x_i + b >= 1 - \varepsilon_i$

For all x_i in class - 1

$w^T x_i + b <= -1 + \varepsilon_i$

A total of n constraints

For all i

$$\varepsilon_i \geq 0$$

Another n constraints

Two optimization problems:

For the separable and non separable cases

$$\min_w \frac{w^T w}{2}$$

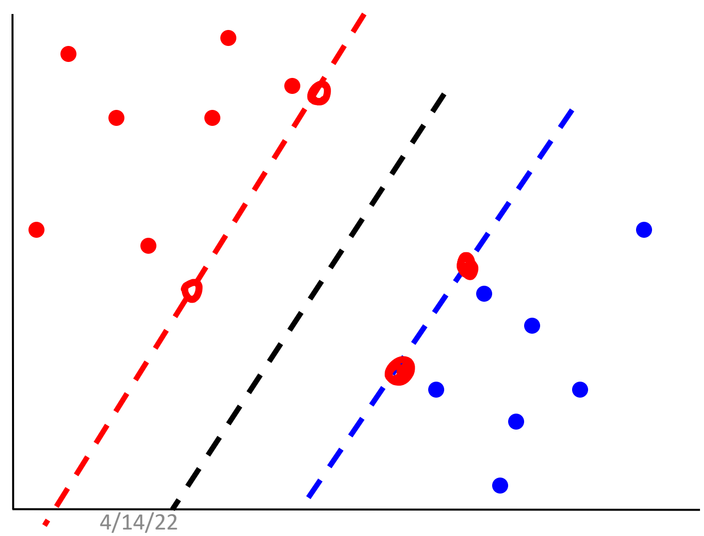
For all x in class + 1

$w^T x + b >= 1$

For all x in class - 1

$w^T x + b <= -1$

separable



$$\min_w \frac{w^T w}{2} + C\sum_{i=1}^{n} \varepsilon_i$$

For all x_i in class + 1

$w^T x_i + b >= 1 - \varepsilon_i$

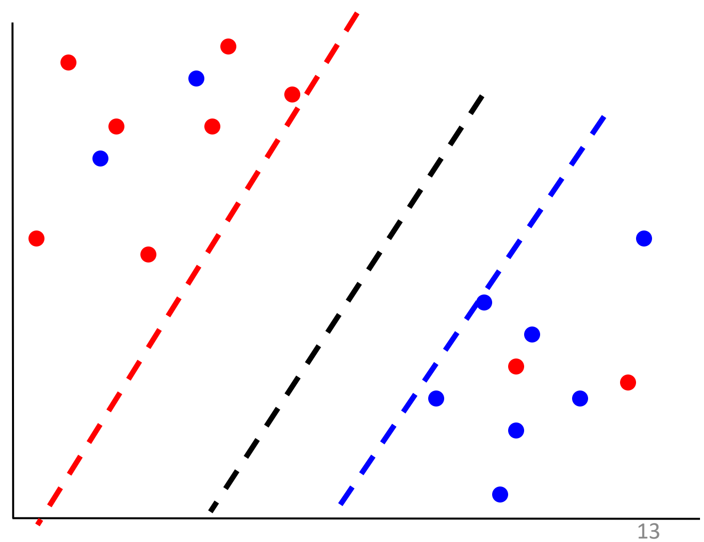
For all x_i in class - 1

$w^T x_i + b <= -1 + \varepsilon_i$

For all i

$$\varepsilon_i \geq 0$$

non separable



Model Selection, find right C

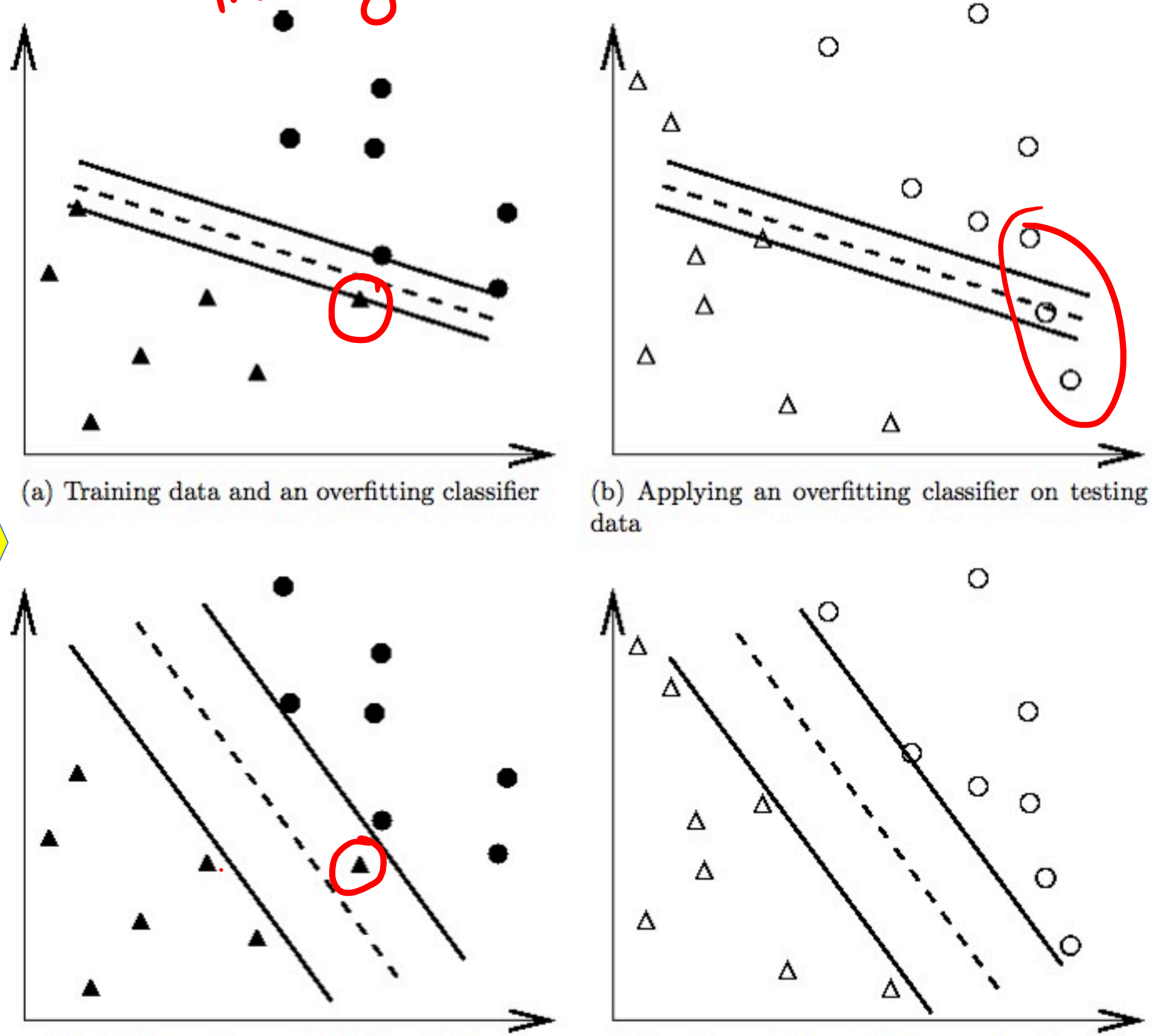
Training

Test

large C

Select the right penalty parameter C

small C



(a) Training data and an overfitting classifier

(b) Applying an overfitting classifier on testing data

(c) Training data and a better classifier

(d) Applying a better classifier on testing data

Model Selection, find right C

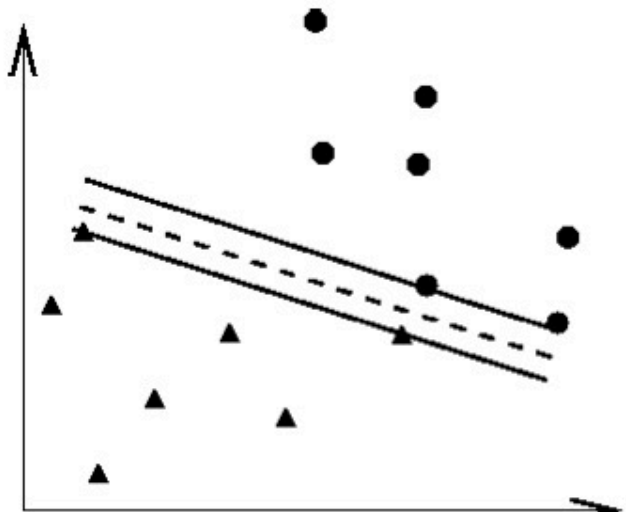
large C

A large value of C means that misclassifications are bad - resulting in smaller margins and less training error (but more expected true error).

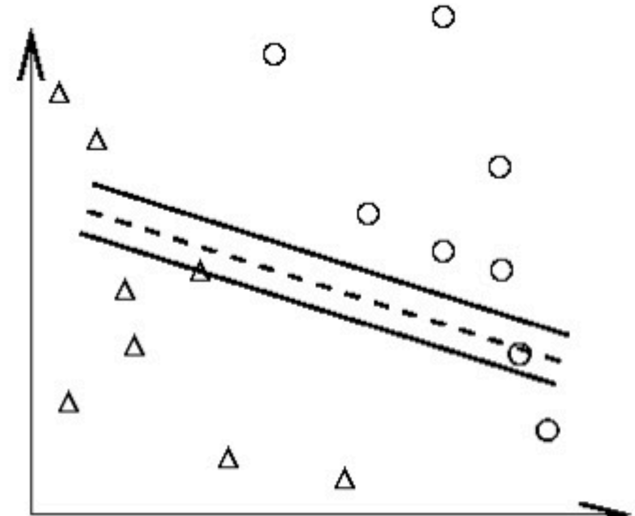
may leak

A small C results in more training error, hopefully better true error.

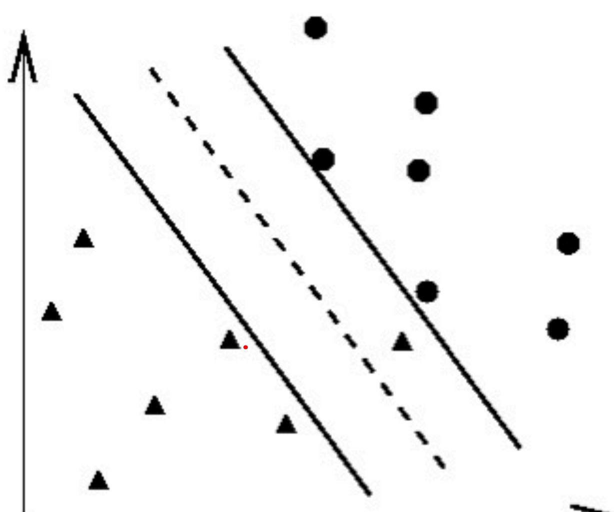
small C



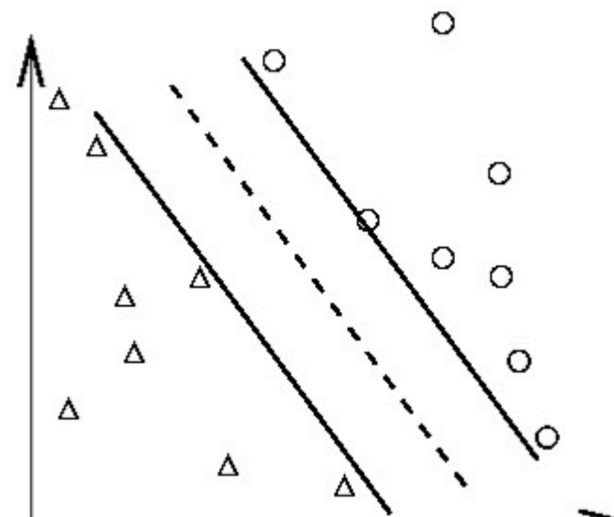
(a) Training data and an overfitting classifier



(b) Applying an overfitting classifier on testing data



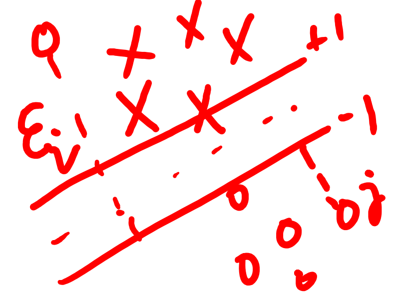
(c) Training data and a better classifier



(d) Applying a better classifier on testing data

Hinge Loss for Soft SVM

$\sum_{i=1}^{n} \varepsilon_i$



$$\sum_{i=1}^{n} \max(0, 1 - y_i f(\mathbf{x}_i))$$

$$\min_w \frac{w^T w}{2} + C\sum_{i=1}^{n} \varepsilon_i$$

For all x_i in class + 1

$w^T x_i + b >= 1 - \varepsilon_i$

For all x_i in class - 1

$w^T x_i + b <= -1 + \varepsilon_i$

For all i

$$\varepsilon_i \geq 0$$

soft

$$\underset{\mathbf{w},b}{\text{argmin}} \frac{w^T w}{2} + C\sum_{i=1}^{n} \max(0, 1 - \underbrace{y_i(\mathbf{w}^T \mathbf{x}_i + b)}_{\geq 1})$$

subject to:

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \varepsilon_i$$

$$\varepsilon_i \geq 0$$

vs. Hard SVM

$$\underset{\mathbf{w},b}{\text{argmin}} \sum_{i=1}^{p} w_i^2 / 2$$

subject to $\forall \mathbf{x}_i \in Dtrain : y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1$

What Left in SVM?

- Support Vector Machine (SVM)
  - ✓ History of SVM
  - ✓ Large Margin Linear Classifier
  - ✓ Define Margin (M) in terms of model parameter
  - ✓ Optimization to learn model parameters (w, b)
  - ✓ Linearly Non-separable case (soft SVM)
  - ✓ Optimization with dual form
  - ✓ Nonlinear decision boundary
  - ✓ Practical Guide

This: Kernel Support Vector Machine

Data

↓

Task

↓

Representation

↓

Score Function

↓

Search/Optimization

↓

Models, Parameters

Tabular / Text / Image/ Structured

↓

Classification / Regression / Ranking

↓

Kernel Trick Func K(xi, xj)

$$K(\mathbf{x}, z) := \Phi(\mathbf{x})^T \Phi(z)$$

↓

Margin + Hinge Loss

↓

QP with Dual form

↓

$$w = \sum_i \alpha_i x_i y_i$$

Dual Weights

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \sum_{i=1}^{p} w_i^2 + C\sum_{i=1}^{n} \varepsilon_i$$

$$\text{subject to } \forall \mathbf{x}_i \in Dtrain : y_i\left(\mathbf{x}_i \cdot \mathbf{w} + b\right) \geq 1 - \varepsilon_i$$

$$\max_\alpha \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x_i}^T \mathbf{x_j}$$

$$\sum_i \alpha_i y_i = 0, \qquad \alpha_i \geq 0 \qquad \forall i$$

Two optimization problems: For the separable and non separable cases

Min $(w^T w)/2$

For all x in class + 1

$w^T x+b >= 1$

For all x in class - 1

$w^T x+b <= -1$

$$\min_w \frac{w^T w}{2} + C\sum_{i=1}^{n}\varepsilon_i$$

For all x_i in class + 1

$w^T x_i+b >= 1-\varepsilon_i$

For all x_i in class - 1

$w^T x_i+b <= -1+\varepsilon_i$

For all i

$$\varepsilon_i \geq 0$$

- Instead of solving these QPs directly we will solve a dual formulation of the SVM optimization problem
- The main reason for switching to this type of representation is that it would allow us to use a neat trick that will make our lives easier (and the run time faster)

Optimization Review: Ingredients

- Objective function
- Variables
- Constraints

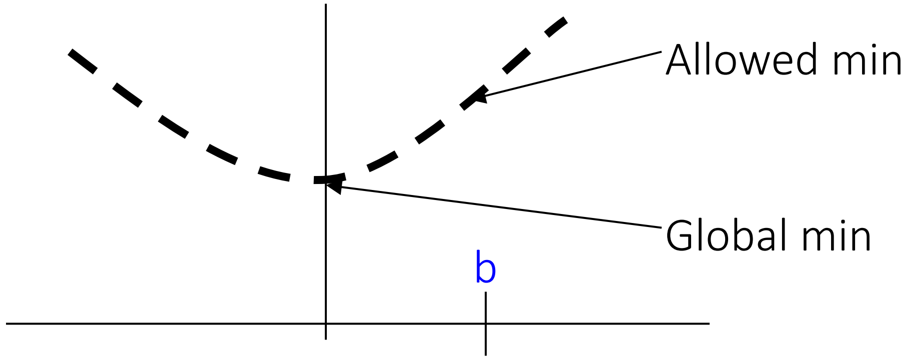
Find values of the variables
that minimize or maximize the objective function
while satisfying the constraints

Optimization Review: Constrained Optimization

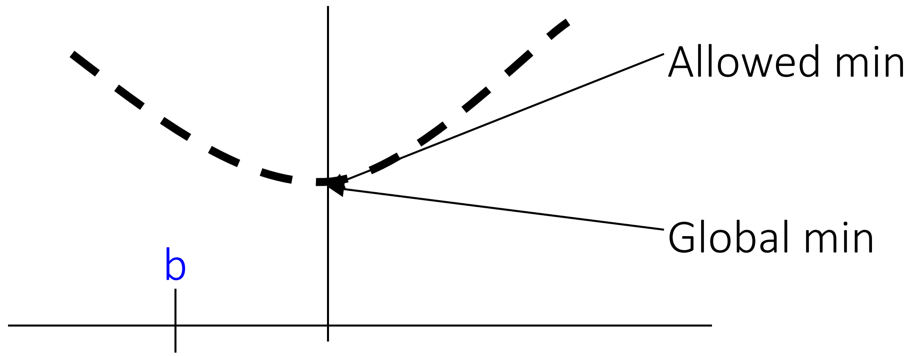
$\min_u u^2$

s.t. $u >= b$

Case 1:



Case 2:

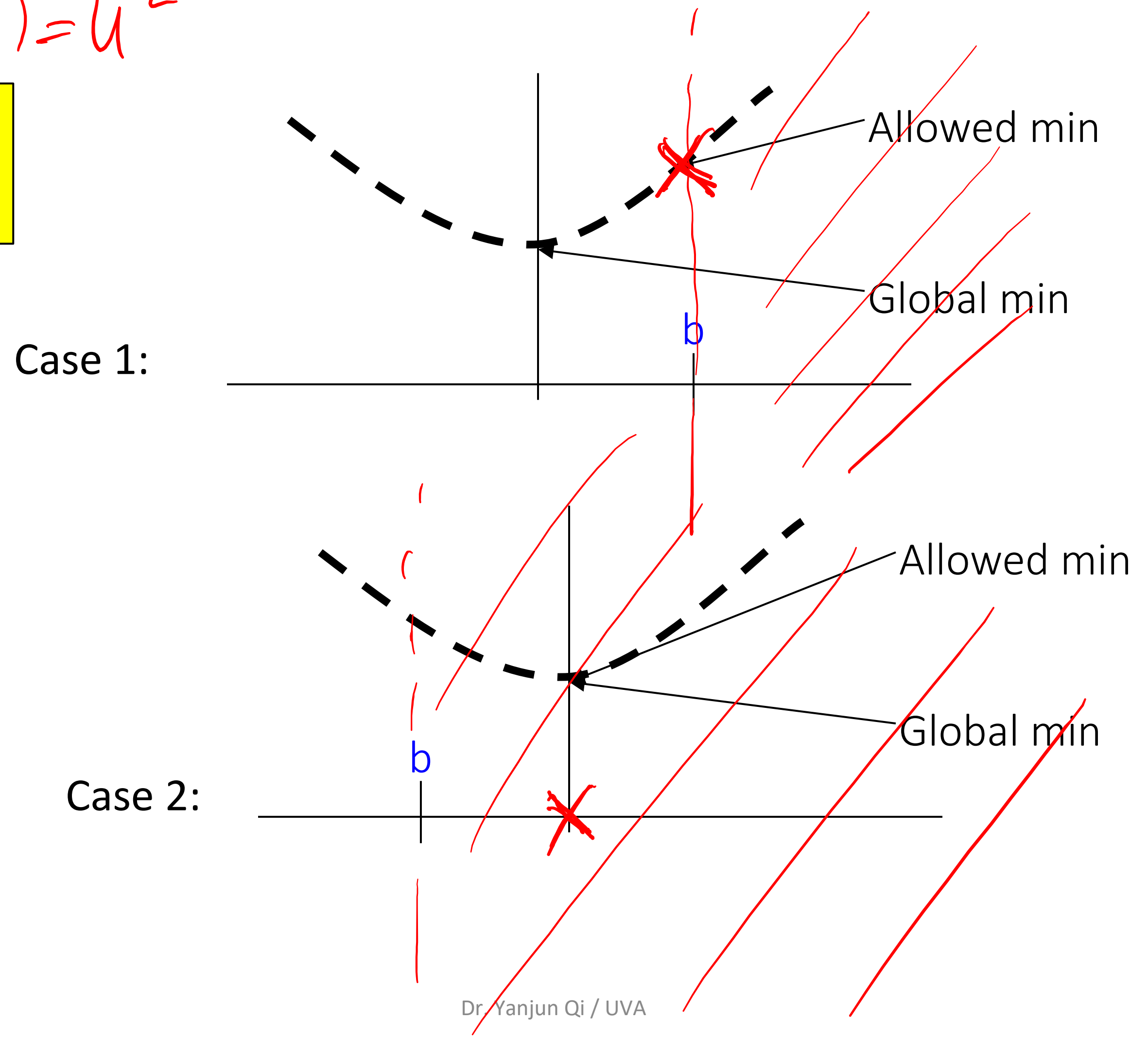


Optimization Review: Constrained Optimization

$f(u) = u^2$

$\min_u u^2$

s.t. $u >= b$



Optimization Review: Constrained Optimization

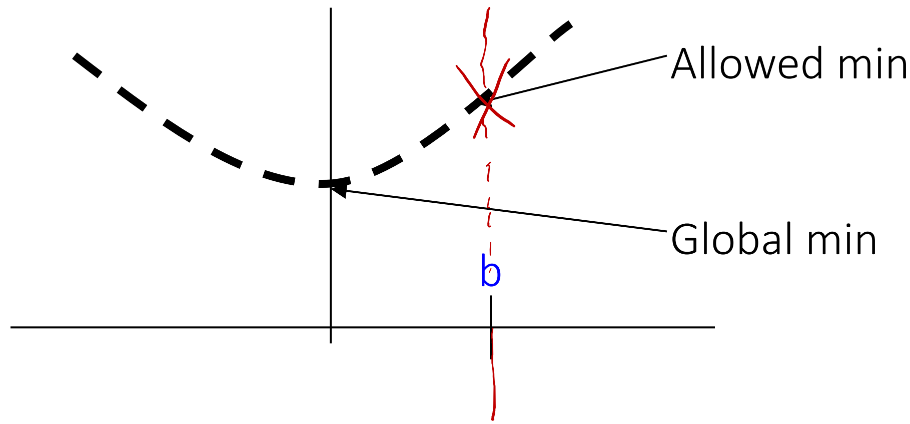
$f(u)$

$\min_u u^2$

s.t. $u >= b$

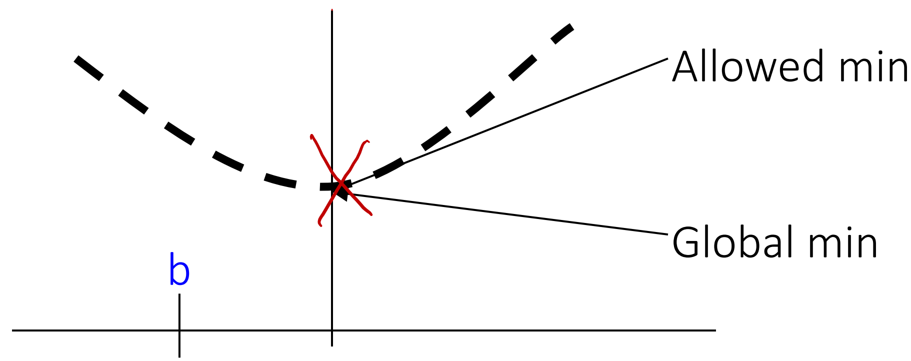
[Subject to]

Case 1:



$\begin{cases} b > 0 \\ f(u^*) = b^2 \\ u^* = b \end{cases}$

Case 2:



$\begin{cases} b < 0 \\ f(u^*) = 0 \\ u^* = 0 \end{cases}$

$\min_u u^2$

s.t. $u >= b$

$$\textcircled{0}\begin{cases} \min\limits_{u} f_0(u) = u^2 \\ \text{s.t.} \quad b - u \le 0 \end{cases}$$

primal problem

$\min_u u^2$

s.t. $u >= b$

① $\begin{cases} \min_u f_0(u) = u^2 \\ \text{s.t.} \quad b - u \leq 0 \end{cases}$

⇒ multiplier variable

② $L(\underset{1\times 1}{u}, \underset{1\times 1}{\alpha}) = u^2 + \underbrace{\alpha}_{\geq 0} \underbrace{(b - u)}_{\leq 0}$

$\min_u u^2$

s.t. $u >= b$

① $\begin{cases} \min_u f_0(u) = u^2 \\ \text{s.t.} \quad b - u \le 0 \end{cases}$

② $L(\underset{1\times 1}{u}, \underset{1\times 1}{\alpha}) = u^2 + \underbrace{\alpha}_{\geq 0} \underbrace{(b-u)}_{\leq 0}$

③ $\frac{\partial L(u,\alpha)}{\partial u} = 2u - \alpha = 0$

$u = \frac{\alpha}{2}$

$\rightarrow \arg\min_u L(u,\alpha)$

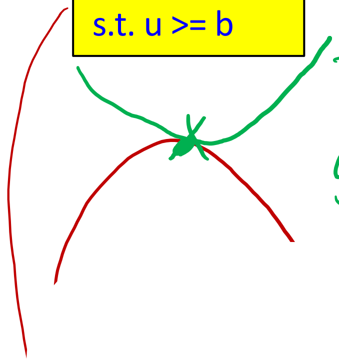
$\min_u u^2$

s.t. $u \geq b$

$$g(\alpha) = L(u, \alpha) = \frac{\alpha^2}{4} + \alpha\left(b - \frac{\alpha}{2}\right)$$

$\min_u u^2$

s.t. $u >= b$



$$g(\alpha) = L(u, \alpha) = \frac{\alpha^2}{4} + \alpha\left(b - \frac{\alpha}{2}\right)$$

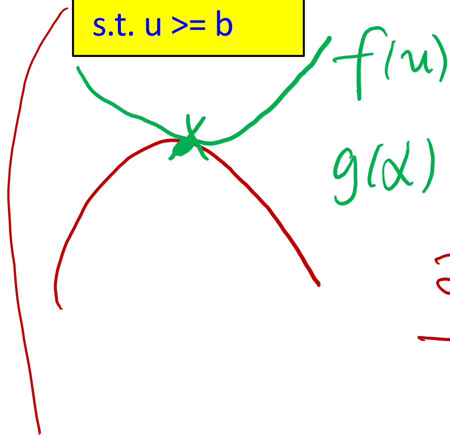
$u = \alpha/2$

$f(u)$

$$g(\alpha) = -\frac{\alpha^2}{4} + b\alpha$$

$\min_u u^2$

s.t. $u >= b$



$$g(\alpha) = L(u,\alpha) = \frac{\alpha^2}{4} + \alpha\left(b - \frac{\alpha}{2}\right)$$

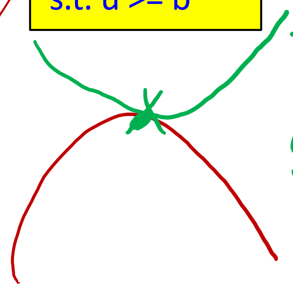
$u = \alpha/2$

$$g(\alpha) = -\frac{\alpha^2}{4} + b\alpha$$

$$\frac{\partial g(\alpha)}{\partial \alpha} = -\frac{\alpha}{2} + b = 0, \quad \alpha \geq 0$$

$\min_u u^2$

s.t. $u >= b$



$f(u)$

$$g(\alpha) = L(u,\alpha) = \frac{\alpha^2}{4} + \alpha\left(b - \frac{\alpha}{2}\right)$$

$u = \alpha/2$

$$g(\alpha) = -\frac{\alpha^2}{4} + b\alpha$$

$$\frac{\partial g(\alpha)}{\partial \alpha} = -\frac{\alpha}{2} + b = 0, \quad \alpha \geq 0$$

$\Rightarrow$ Dual

$$\begin{cases} b > 0, & \alpha = 2b, \quad g(\alpha) = b^2 \\ b < 0, & \alpha = 0, \quad g(\alpha) = 0 \end{cases}$$

$\Rightarrow$ Primal

$$\begin{cases} b > 0, & f(u) = b^2 \quad u = b \\ b < 0, & f(u) = 0 \quad u = 0 \end{cases}$$

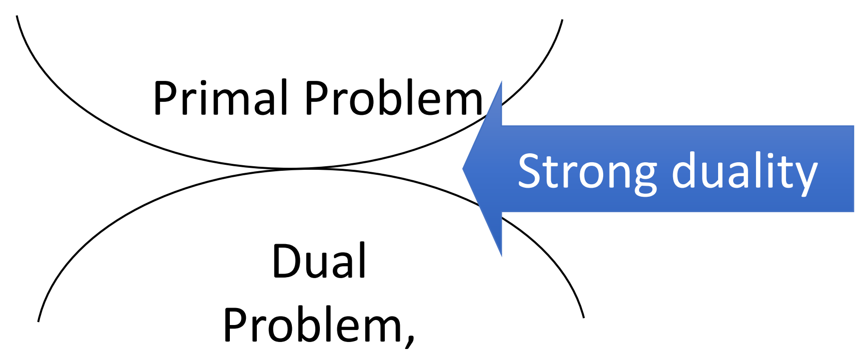
primal : $\min_{w} \max_{\alpha} L(w, \alpha)$

Dual : $\underline{\max_{\alpha} \min_{w} L(w, \alpha)}$

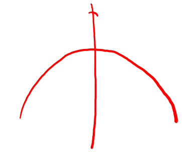
$\Rightarrow \max_{\alpha} g(\alpha)$

Optimization Review: Dual Problem (Extra)

- Solving dual problem if the dual form is easier than primal form
- Need to change primal minimization to dual maximization (OR → Need to change primal maximization to dual minimization)
- Only valid when the original optimization problem is convex/concave (strong duality)



Concrete derivation in L11Extra-SVMoptimDual



$$f(u): \begin{cases} \min \ u^2 \\ s.t. \ \ u \geq b \end{cases}$$

$$g(\alpha): \begin{cases} \max \ -\frac{\alpha^2}{4} + b\alpha = \max\left\{-\left(\cancel{\frac{\alpha}{2}}-b\right)^2 + b^2\right\} \\ s.t. \ \ \alpha \geq 0 \end{cases}$$

$$\begin{cases} \text{if}_{\,b \geq 0} \ \ b = \alpha/2, \quad u^* = b, \quad g^* = b^2 \\ \text{if}_{\,b < 0} \ \ b \neq \alpha/2, \quad \alpha^* = 0, \quad g^* = 0 \end{cases}$$

$$\Rightarrow \quad \alpha(b-u) = 0$$

KKT condition

Optimization Review: Lagrangian Duality (Extra)

- The Primal Problem

Primal:

$$\min_w \quad f_0(w)$$
$$\text{s.t.} \quad f_i(w) \le 0, \quad i = 1, \dots, k$$

"Method of Lagrange multipliers" convert to a higher-dimensional problem

The generalized Lagrangian:

$$\mathcal{L}(w,\alpha) = f_0(w) + \sum_{i=1}^{k} \alpha_i f_i(w)$$

the a's (a$_i$≥0) are called the Lagarangian multipliers

Lemma:

$$\max_{\alpha,\alpha_i \ge 0} \mathcal{L}(w,\alpha) = \begin{cases} f_0(w) & \text{if } w \text{ satisfies primal constraints} \\ \infty & \text{o/w} \end{cases}$$

A re-written Primal:

$$\min_w \max_{\alpha,\alpha_i \ge 0} \mathcal{L}(w,\alpha)$$

Optimization Review: Lagrangian Duality, cont. (Extra)

- Recall the Primal Problem:

$$\min_w \max_{\alpha,\alpha_i \geq 0} L(w,\alpha)$$

- The Dual Problem:

$$\max_{\alpha,\alpha_i \geq 0} \min_w L(w,\alpha)$$

- **Theorem (weak duality):**

$$d^* = \max_{\alpha,\alpha_i \geq 0} \min_w L(w,\alpha) \quad \leq \quad \min_w \max_{\alpha,\alpha_i \geq 0} L(w,\alpha) = p^*$$

- **Theorem (strong duality):**

  Iff there exist a saddle point of $L(w,\alpha)$

  we have $d^* = p^*$

Dual representation of the hard SVM QP

- We will start with the linearly separable case
- Instead of encoding the correct classification rule and constraint we will use Lagrange multiplies to encode it as part of the our minimization problem

Min $(w^T w)/2$

s.t.

$(w^T x_i + b) y_i >= 1$

Recall that Lagrange multipliers can be applied to turn the following problem:

$$L_{primal}(w,b,\alpha) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} - \sum_{i=1}^{N}\alpha_i\Big(y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1\Big)$$

The Dual Problem (Extra)

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

Dual formulation

- We minimize L with respect to w and b first:

$$\nabla_w L(w,b,\alpha) = w - \sum_{i=1}^{train} \alpha_i y_i x_i = 0, \quad (*)$$

$$\nabla_b L(w,b,\alpha) = \underbrace{\sum_{i=1}^{train} \alpha_i y_i = 0,} \quad (**)$$

Note that **(*)** implies: $\left[w = \sum_{i=1}^{train} \alpha_i y_i x_i \right]$ $(***)$

- Plus (***) back to L , and using (**), we have:

$$\max_{\alpha \geq 0} \; L(w,b,\alpha) = \sum_{i=1} \alpha_i - \frac{1}{2} \sum_{i,j=1} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j)$$

Summary: Dual for hard SVM (Extra)

Solving for **w** that gives maximum margin:

1. Combine objective function and constraints into new objective function, using Lagrange multipliers *alpha*$_i$

$$L_{primal} = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} - \sum_{i=1}^{N}\alpha_i\left(y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1\right)$$

2. To minimize this Lagrangian, we take derivatives of **w** and *b* and set them to 0:

Summary: Dual for hard SVM (Extra)

3. Substituting and rearranging gives the dual of the Lagrangian:

$$L_{dual} = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

which we try to maximize (not minimize).

4. Once we have the *alpha*$_i$, we can substitute into previous equations to get **w** and *b*.
5. This defines **w** and *b* as linear combinations of the training data.

$$w = \sum_{i=1}^{train} \alpha_i y_i x_i$$

Summary: Dual SVM for linearly separable case

$O(n)$ #para

w, b #para

$O(p)$

Dual formulation

$$\max_\alpha \sum_i \alpha_i - \frac{1}{2}\sum_{\mathrm{i,j}} \alpha_i \alpha_j \mathrm{y}_\mathrm{i} y_j \mathbf{x}_\mathbf{i}^T \mathbf{x}_\mathbf{j}$$

$$\sum_i \alpha_i \mathrm{y}_\mathrm{i} = 0$$

$$\alpha_i \geq 0 \qquad \forall i$$

n α_i

Min $(w^T w)/2$

subject to the following inequality constraints:

For all x in class + 1

$w^T x + b >= 1$

For all x in class - 1

$w^T x + b <= -1$

A total of n constraints if we have n input samples

Easier than original QP, more efficient algorithms exist to find a_i; e.g. SMO (see extra slides)

Dual formulation for linearly non-separable case

prima linear soft margin + Hinge

$\underbrace{\text{Hinge}}_{\{\varepsilon_i \forall i,\ \{w, b}$

$O(n+p+\cancel{1})$

Dual target function:

$$\max_\alpha \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x_i}^T \mathbf{x_j}$$

$$\sum_i \alpha_i y_i = 0$$

$$C \geq \alpha_i \geq 0, \forall i$$

Hyperparameter C should be tuned through k-folds CV

The only difference is that the \alpha are now upper bounded

$O(n)$

#para

This is very similar to the optimization problem in the linear separable case, except that there is an upper bound *C* on a_i now

Once again, efficient algorithm exist to find a_i

Prediction via Dual Weights for linear case

$f(x_{ts}) = \text{Sign}(w^T x_{ts} + b)$

$w = \sum_{i=1}^{n} \alpha_i y_i x_i$

Dual target function:

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x_i}^T \mathbf{x_j}$$

$$\sum_i \alpha_i y_i = 0$$

$$C \geq \alpha_i \geq 0, \forall i$$

Hyperparameter C should be tuned through k-folds CV

The only difference is that the \alpha are now bounded

To evaluate a new sample x_{ts} we need to compute:

$$w^T x_{ts} + b = \sum_{i \in \text{supportV}} \alpha_i y_i \mathbf{x_i}^T \mathbf{x}_{ts} + b$$

This is very similar to the optimization problem in the linear separable case, except that there is an upper bound *C* on a_i now

Once again, efficient algorithm exist to find a_i

Dual SVM – Training using Kernel Matrix

Our dual target function:

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2}\sum_{\mathrm{i,j}} \alpha_i \alpha_j \mathrm{y}_{\mathrm{i}} y_j \mathbf{x_i}^T \mathbf{x_j}$$

$$\sum_i \alpha_i \mathrm{y}_{\mathrm{i}} = 0$$

$$C \geq \alpha_i \geq 0, \forall i$$

Dot product among all training samples

$$\begin{array}{cc} & \begin{matrix} 1, & 2, & \cdots & j, & \cdots & n \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ \vdots \\ i \\ \vdots \\ n \end{matrix} & \left[\begin{matrix} & & & \vdots & & \\ & & & \vdots & & \\ & & & \vdots & & \\ & & & \vdots & & \\ \cdots & \cdots & \cdots & x_i^T x_j, & \cdots & \cdots \\ & & & \vdots & & \\ & & & \vdots & & \end{matrix}\right] \end{array} \quad n \times n$$

matrix

This: Kernel Support Vector Machine

Data
↓
Task
↓
Representation
↓
Score Function
↓
Search/Optimization
↓
Models, Parameters

Tabular / Text / Image/ Structured
↓
Classification / Regression / Ranking
↓
Kernel Trick Func K(xi, xj)

$$K(\mathbf{x}, z) := \Phi(\mathbf{x})^T \Phi(z)$$

↓
Margin + Hinge Loss
↓
QP with Dual form
↓

$$w = \sum_i \alpha_i x_i y_i$$

Dual Weights

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \sum_{i=1}^{p} w_i^2 + C\sum_{i=1}^{n} \varepsilon_i$$

$$\text{subject to } \forall \mathbf{x}_i \in Dtrain : y_i\left(\mathbf{x}_i \cdot \mathbf{w} + b\right) \geq 1 - \varepsilon_i$$

$$\max_\alpha \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x_i}^T \mathbf{x_j}$$

$$\sum_i \alpha_i y_i = 0, \qquad \alpha_i \geq 0 \qquad \forall i$$

Support vectors: non-zero a_i

- only a few a_i can be nonzero!!

$$\forall i \Rightarrow \alpha_i\left(y_i(\mathbf{w}^T\mathbf{x}_i+b)-1\right)=0, \quad i=1,\ldots,n$$

$y_i(w^T x_i+b)>1 \Rightarrow \alpha_i=0$

$y_i = 1$

$+1 \quad w^T x+b=1$

$\alpha_i>0$

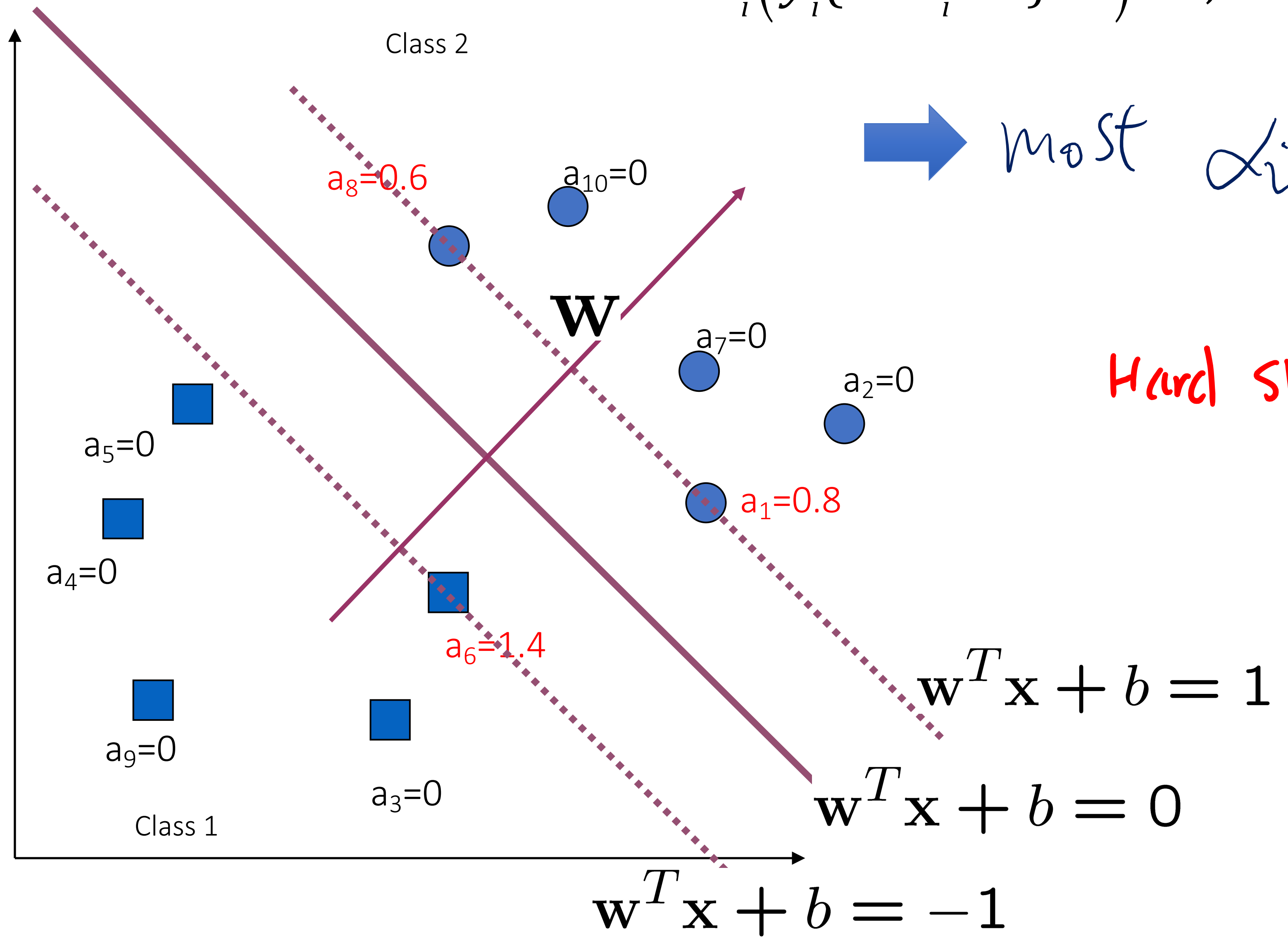
$-1 \quad w^T x+b=-1$

$<-1 \quad \alpha_i=0$

$$\alpha_i\left(y_i(\mathbf{w}\cdot\mathbf{x}_i+b)-1\right)=0,\quad i=1,\ldots,n$$

most $\alpha_i = 0$

Hard SVM



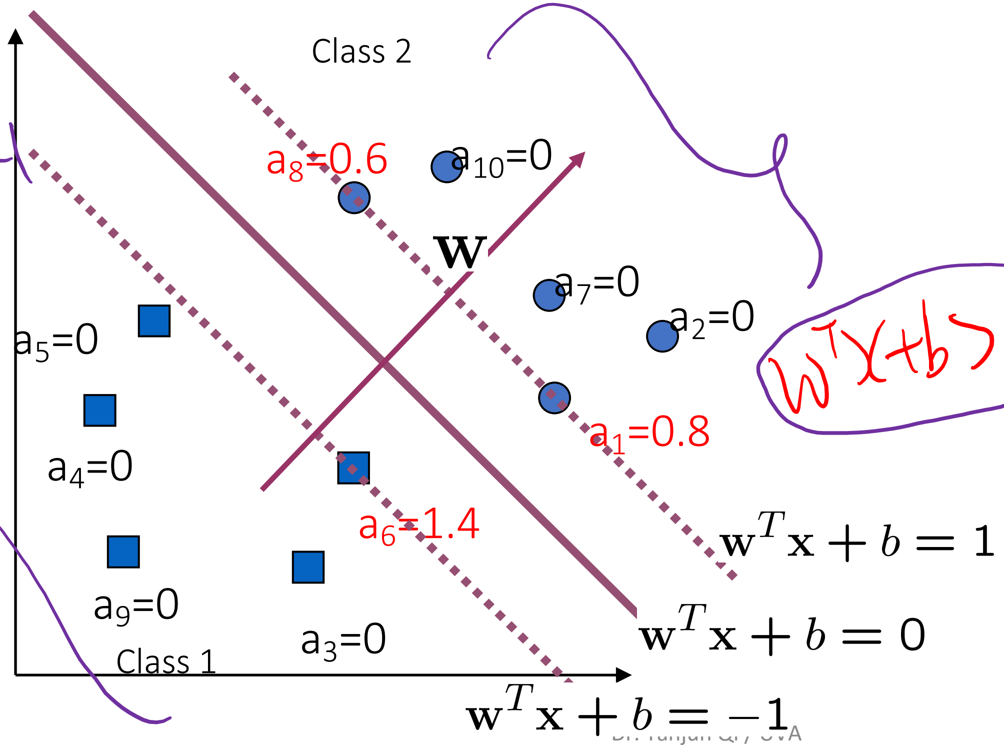
- only a few a_i can be nonzero! Ie.

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + b) = 1$$

$$\alpha_i\left(y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1\right) = 0, \quad i = 1,\ldots,n$$

for most $\alpha_i \Rightarrow \alpha_i = 0$

$w^T x + b < -1$



$w^T x + b > 1$

$\underbrace{y_i(w^T x_i + b) - 1}_{> 1}$

> 0

Support vectors: non-zero a_i

- only a few a_i can be nonzero!!

$$\alpha_i\left(y_i(\mathbf{w}\cdot\mathbf{x}_i+b)-1\right)=0,\quad i=1,\ldots,n$$

for most $\alpha_i \Rightarrow \alpha_i = 0$

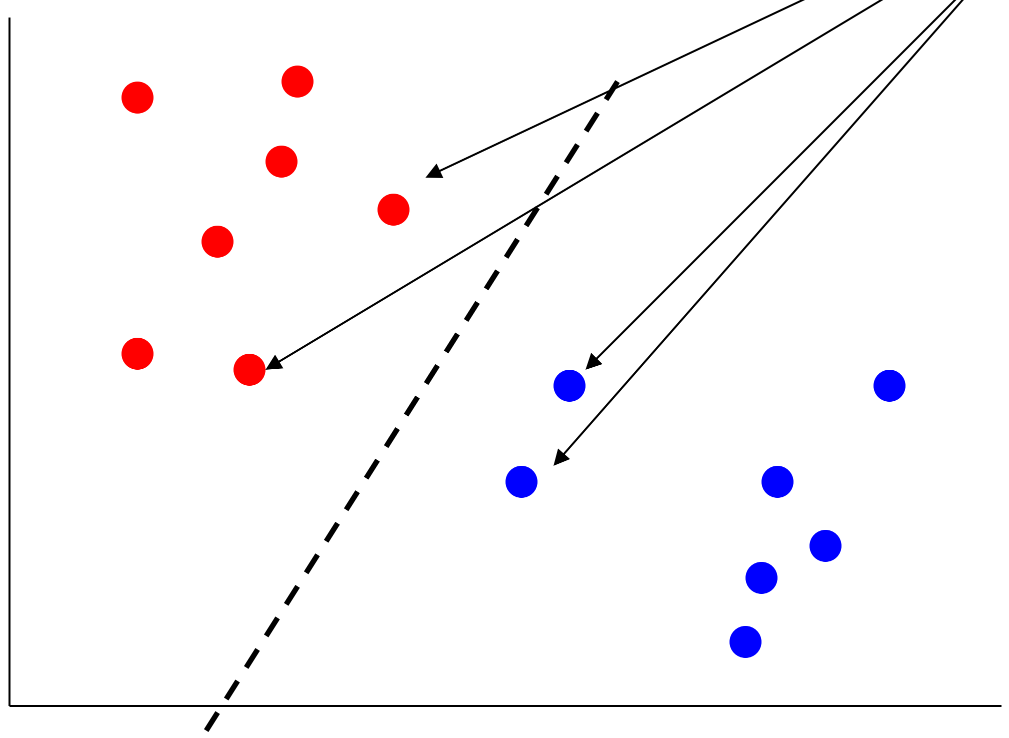
We call the training data points whose a_i's are nonzero the **support vectors** (SV)

Dual SVM - interpretation

training points

$$w = \sum_i \alpha_i x_i y_i$$

$\alpha_i \geq 0$



For α that are 0, no influence

Dual SVM– Testing

Dot product with ("all" ??) training samples

To evaluate a new sample x_{ts} we need to compute:

$$\widehat{y_{ts}} = \text{sign}(\mathrm{w}^{\mathrm{T}} x_{ts} + b) = \text{sign}\Big(\sum_{\mathrm{i=1..n}} \alpha_i \mathrm{y}_{\mathrm{i}} \mathbf{x_i}^T \mathbf{x}_{ts} + b\Big)$$

$$\Rightarrow \widehat{y_{ts}} = \text{sign}\left(\sum_{i \in SupportVectors} \alpha_i y_i \left(\mathbf{x}_i^T \mathbf{x}_{ts} \right) + b \right)$$

$\alpha_i > 0$

For \alpha$_i$ that are 0,

no influence

Support Vectors for the Soft-SVM

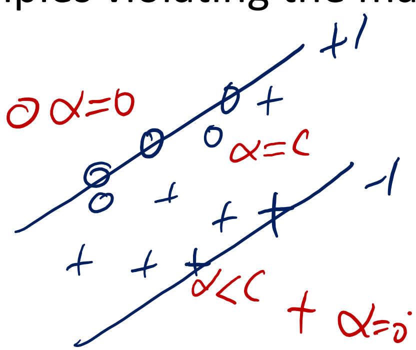
$$\alpha_i\Big(y_i(\mathbf{w}\cdot\mathbf{x}_i+b)-1+\varepsilon_i\Big)=0,\quad i=1,\ldots,n$$

- Support vectors are
  - Samples on the margin:

$$y_i\left(\mathbf{x}_i\cdot\mathbf{w}+b\right)=1,$$

$$0<\alpha_i<C$$

  - Samples violating the margin (mostly inside the margin area):



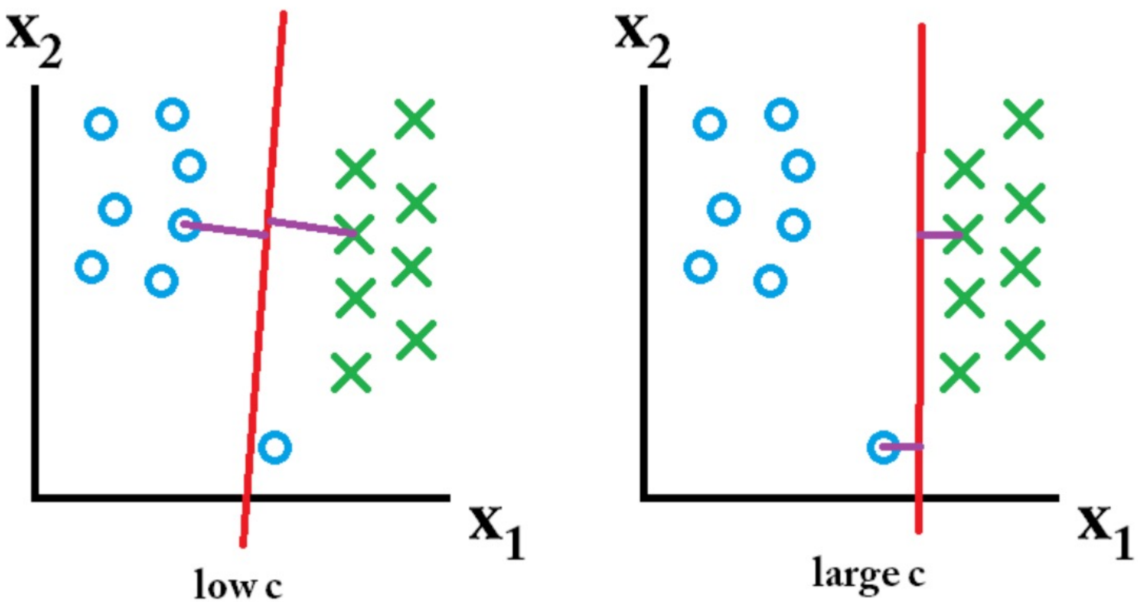
$$y_i\left(\mathbf{x}_i\cdot\mathbf{w}+b\right)<1,$$

$$\alpha_i=C$$

More in L11Extra-SVMoptimDual

Value C and Number of Support Vectors (no clear relation!!!)

C is how hard we want to punish misclassified examples in the training set.
So for large values of C, it would make sense that if misclassifed examples are severely punished, then it will choose a small margin with not many support vectors.
For small values of C, it would broaden the margin, and as a result, end up getting more points inside of the margin (if not easily separable), so there would be possibly more support vectors.
I thought that this StackExchange post described it pretty well.
As in this example, the small c has more "support vectors" than the large c



https://github.com/scikit-learn/scikit-learn/issues/7955

https://stats.stackexchange.com/questions/31066/what-is-the-influence-of-c-in-svms-with-linear-kernel

Why do SVMs work?

$X \to \phi(x)$ e.g. RBF

- ❑ If we are using huge features spaces (e.g., with kernels), how come we are not overfitting the data?
  - ✓ Number of parameters remains the same (and most are set to 0) $O(n)$, α_i $i=1,\cdots,n$
  - ✓ While we have a lot of inputs, at the end we only care about the support vectors and these are usually a small group of samples
  - ✓ The maximizing of the margin acts as a sort of regularization term leading to reduced overfitting

Summary of SVM

- ❑ Support Vector Machine (SVM)
  - ✓ History of SVM
  - ✓ Large Margin Linear Classifier
  - ✓ Define Margin (M) in terms of model parameter
  - ✓ Optimization to learn model parameters (w, b)
  - ✓ Non linearly separable case
  - ✓ Optimization with dual form
  - ✓ Nonlinear decision boundary
  - ✓ Practical Guide
    - ✓ File format / LIBSVM
    - ✓ Feature preprocsssing
    - ✓ Model selection
    - ✓ Pipeline procedure

(Recap)

KNN: $\hat{y}_{ts} = \frac{1}{k} \sum_{i \in kNeighbors\text{-}of\ X_{ts}} y_i$

find k neighbor of X_{ts} ~ $O(n \times$

SVM: $\hat{y}_{ts} = \sum_{i \in SV} \alpha_i y_i K(\vec{X}_i, \vec{X}_{ts}) + b$

para ~ $O(n)$

Logistic Regression / Linear Classifier

$\hat{y}_{ts} = \sigma(W^T X_{ts} + b)$

para ~ $O(p)$

References

- Big thanks to Prof. Ziv Bar-Joseph and Prof. Eric Xing @ CMU for allowing me to reuse some of his slides
- Elements of Statistical Learning, by Hastie, Tibshirani and Friedman
- Prof. Andrew Moore @ CMU's slides
- Tutorial slides from Dr. Tie-Yan Liu, MSR Asia
- A Practical Guide to Support Vector Classification Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, 2003-2010
- Tutorial slides from Stanford "Convex Optimization I — Boyd & Vandenberghe