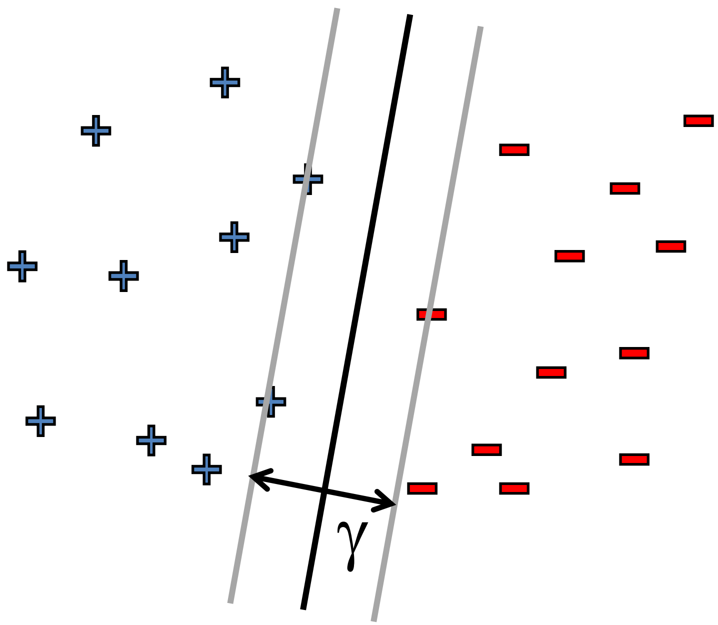
Support Vector Machines (SVMs) Recap…

Aarti Singh

Machine Learning 10-315
Oct 26, 2020

Hard-margin SVM

Data perfectly separable by a linear decision boundary



Hard margin approach

$$\min_{\mathbf{w},b} \ \mathbf{w}.\mathbf{w}$$

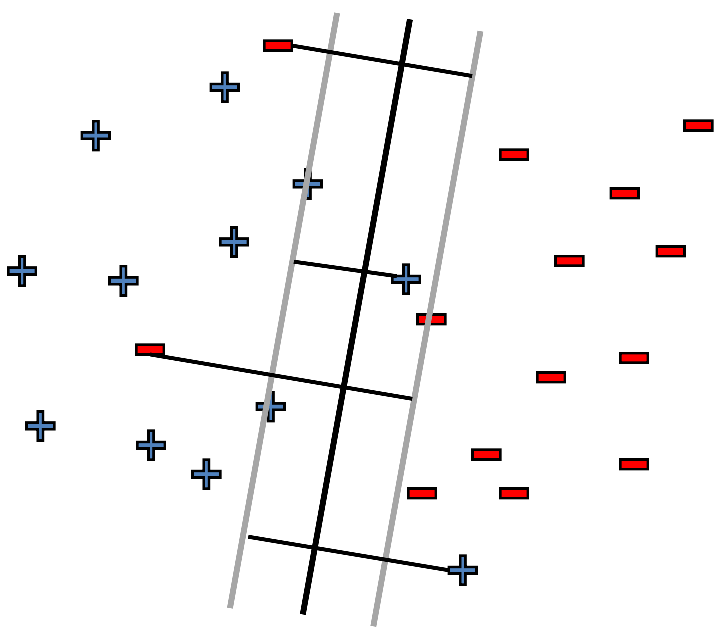
$$\text{s.t. } (\mathbf{w}.\mathbf{x}_j + b)\, y_j \geq 1 \quad \forall j$$

Solve using Quadratic Programming (QP)

Margin, $\gamma \ \alpha \ 1/\|w\|$

Soft-margin SVM

Allow "error" in classification



Soft margin approach

$$\min_{\mathbf{w},b,\{\xi_j\}} \mathbf{w}.\mathbf{w} + C \sum_j \xi_j$$
$$\text{s.t. } (\mathbf{w}.\mathbf{x}_j + b)\, y_j \geq 1-\xi_j \quad \forall j$$
$$\xi_j \geq 0 \quad \forall j$$

ξ_j - "slack" variables

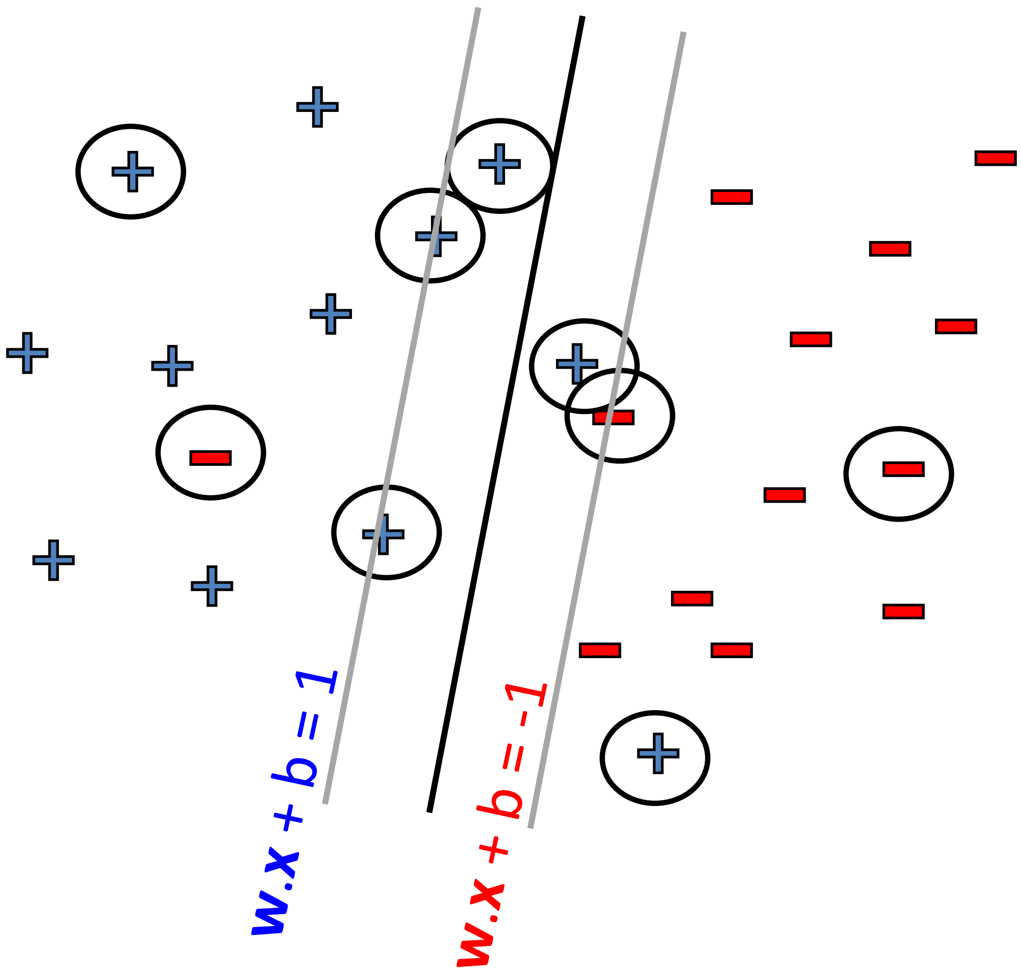
= (>1 if x_j misclassifed)

pay linear penalty if mistake

C - tradeoff parameter (chosen by cross-validation)

Still QP ☺

Slack variables – Hinge loss

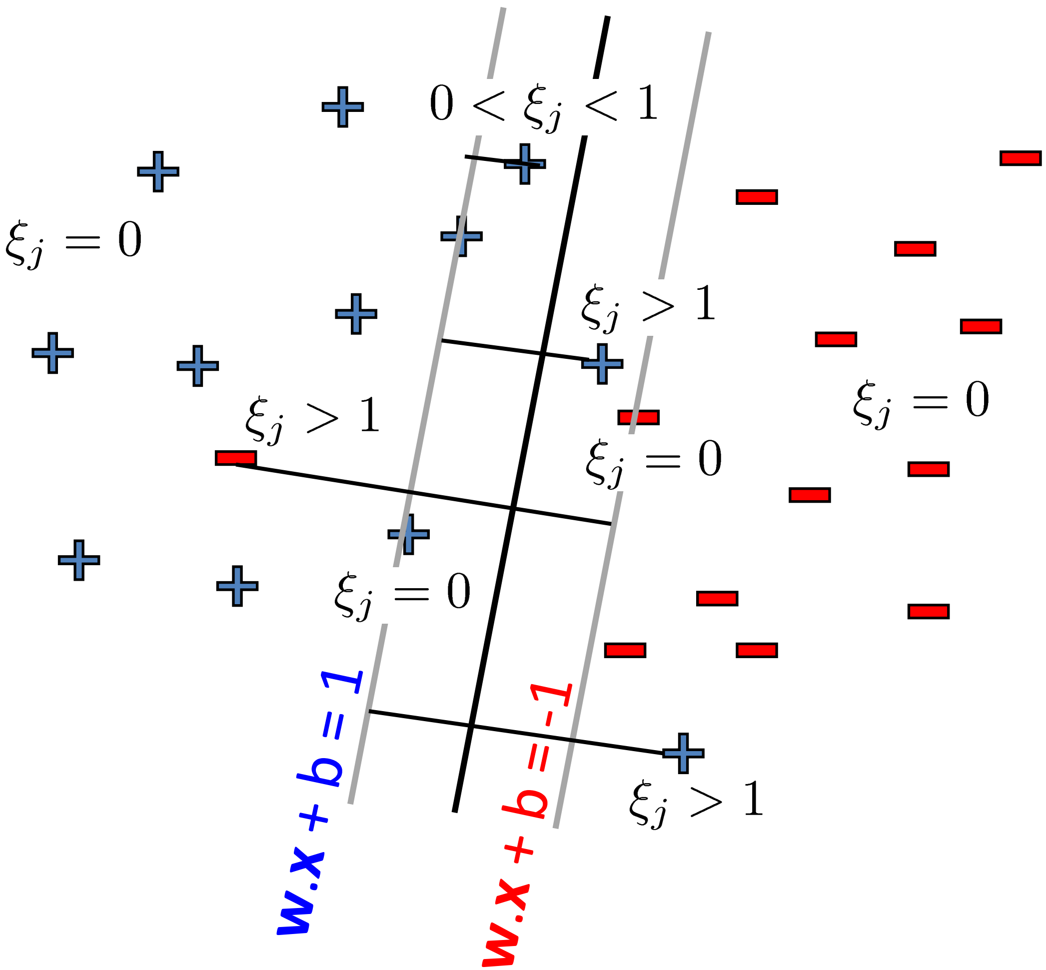


$(\mathbf{w}.\mathbf{x}_j+b)\ y_j \geq 1-\xi_j \quad \forall j$

What is the slack ξ_j for the following points?

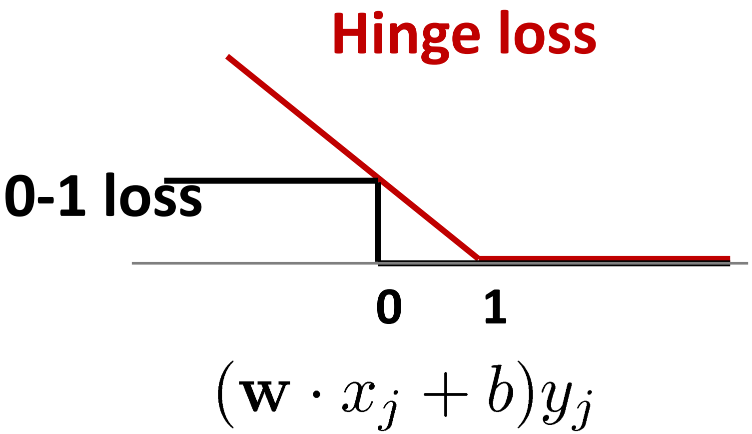
Confidence | Slack

Slack variables – Hinge loss

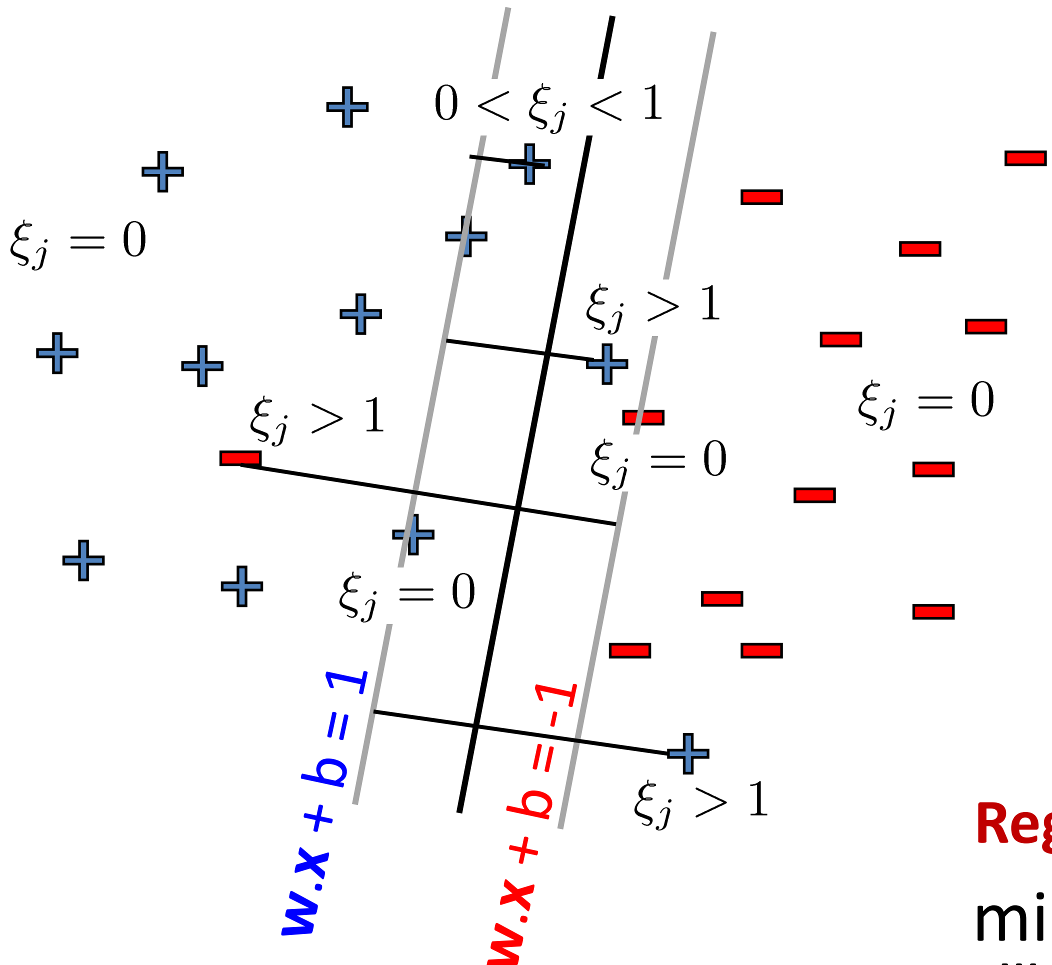


Notice that

$$\xi_j = (1 - (\mathbf{w} \cdot x_j + b) y_j))_+$$

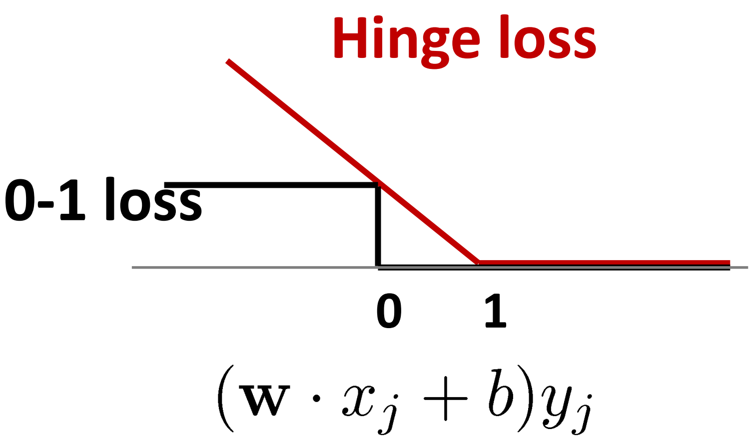


Slack variables – Hinge loss



Notice that

$$\xi_j = (1 - (\mathbf{w} \cdot x_j + b)y_j))_+$$

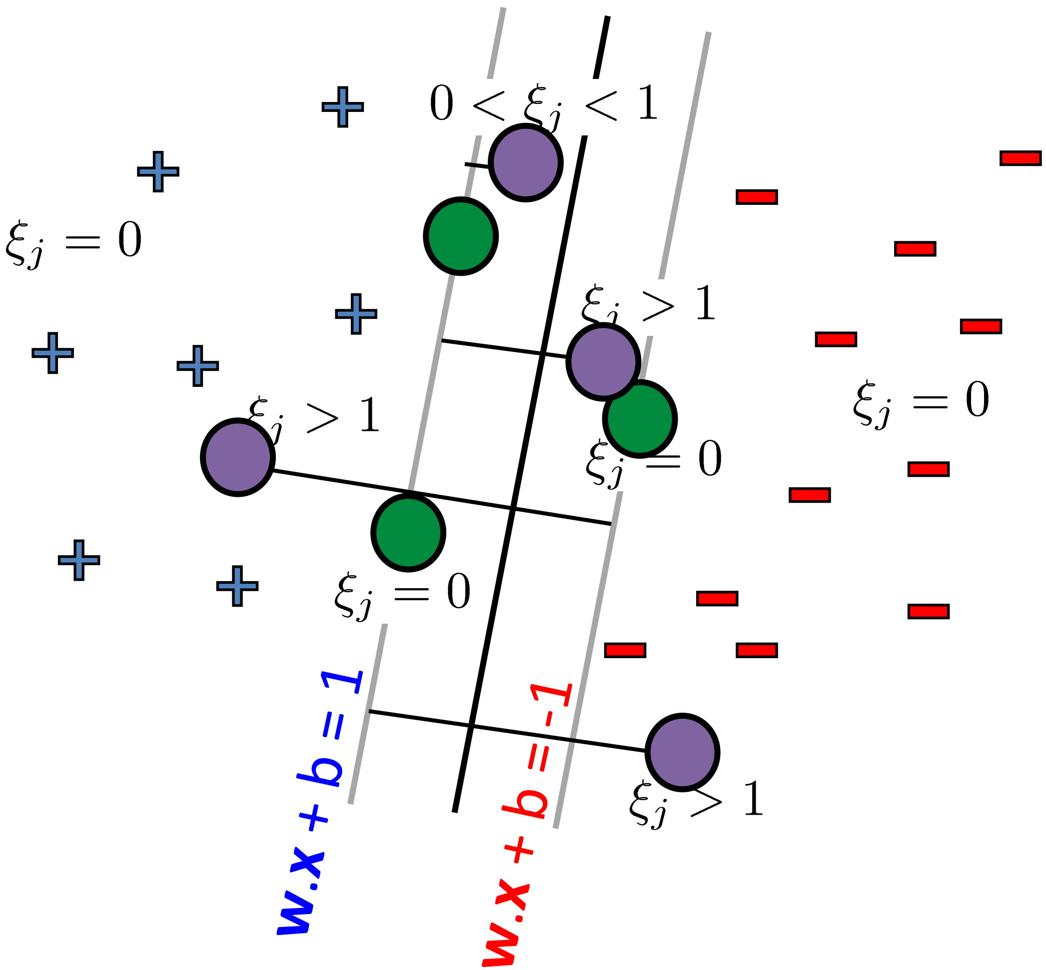


Regularized Hinge loss

$$\min_{\mathbf{w},b} \ \mathbf{w}.\mathbf{w} + C \sum_j (1-(\mathbf{w}.x_j+b)y_j)_+$$

$$\min_{\mathbf{w},b,\{\xi_j\}} \quad \mathbf{w}.\mathbf{w} + C \sum \xi_j$$
$$\text{s.t. } (\mathbf{w}.\mathbf{x}_j + b)\, y_j \geq 1 - \xi_j \quad \forall j$$
$$\xi_j \geq 0 \quad \forall j$$

pport Vectors



Margin support vectors

$\xi_j = 0$, $(\mathbf{w}.\mathbf{x}_j + b)\, y_j = 1$

(don't contribute to objective but enforce constraints on solution)

Correctly classified but on margin

Non-margin support vectors

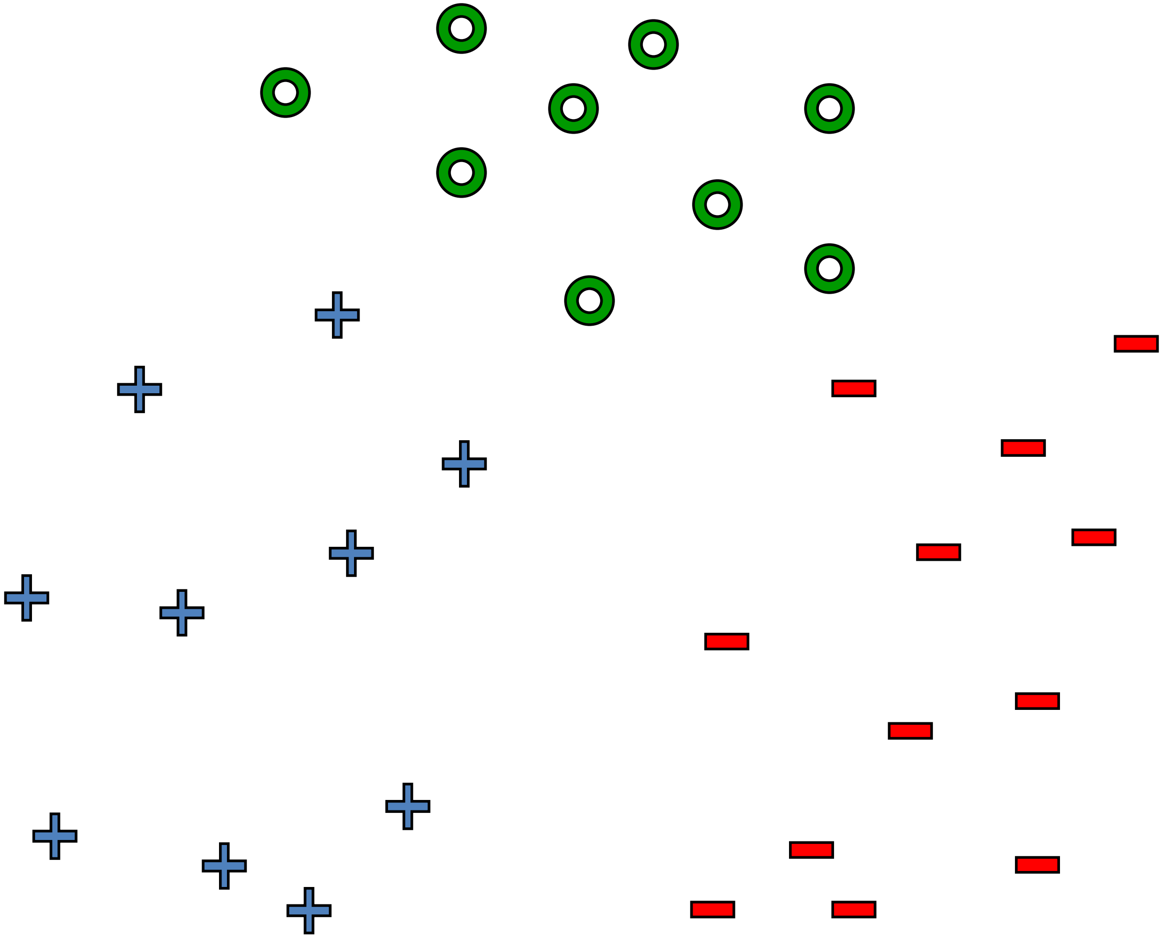
$\xi_j > 0$

(contribute to both objective and constraints)

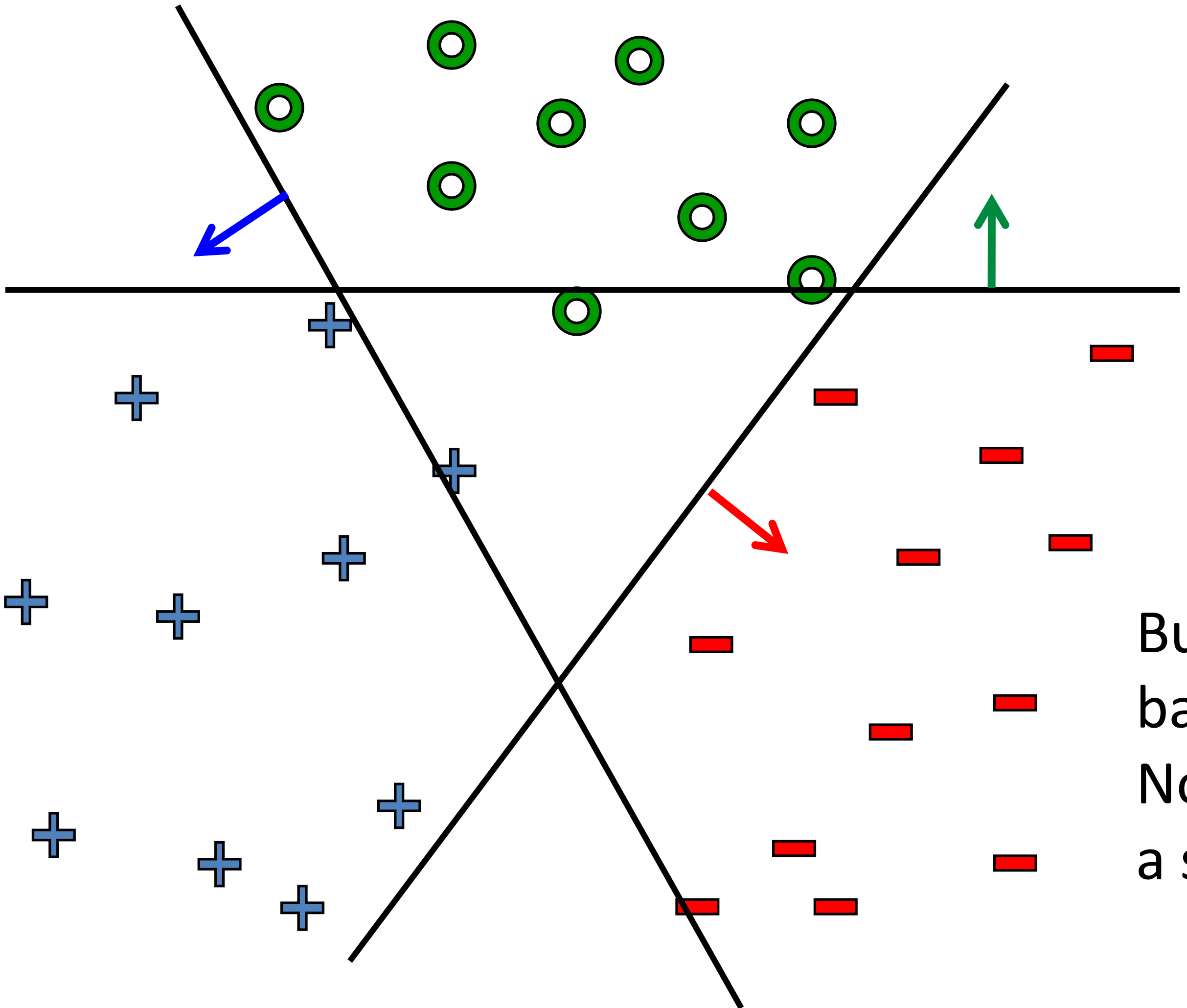
$1 > \xi_j > 0$ Correctly classified but inside margin

$\xi_j > 1$ Incorrectly classified

What about multiple classes?



One vs. rest



Learn 3 classifiers separately:

Class k vs. rest

$(\mathbf{w}_k, b_k)_{k=1,2,3}$

$$y = \arg\max_k \mathbf{w}_k.x + b_k$$

But $\mathbf{w}_k$s may not be based on the same scale.

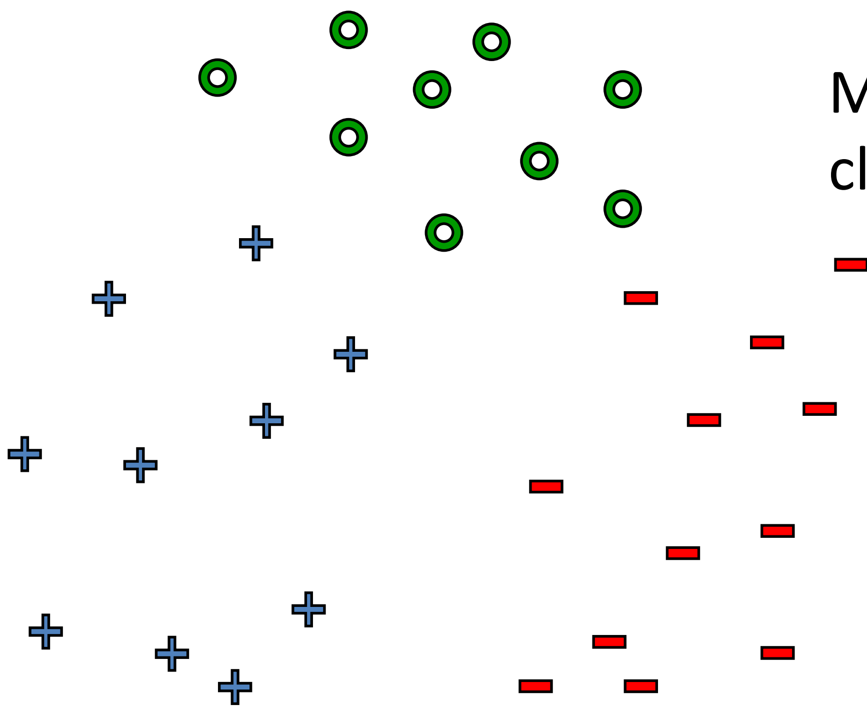
Note: $(a\mathbf{w}).x + (ab)$ is also a solution

Learn 1 classifier: Multi-class SVM

Simultaneously learn 3 sets of weights

min $_{\{w^{(y)}\}, \{b^{(y)}\}}$ $\sum_y \mathbf{w}^{(y)}.\mathbf{w}^{(y)}$

$$\mathbf{w}^{(y_j)}.\mathbf{x}_j + b^{(y_j)} \geq \mathbf{w}^{(y')}.\mathbf{x}_j + b^{(y')} + 1, \ \forall y' \neq y_j, \ \forall j$$



Margin - gap between correct class and nearest other class

y = arg max$_k$ $\mathbf{w}^{(k)}$.x + b$^{(k)}$

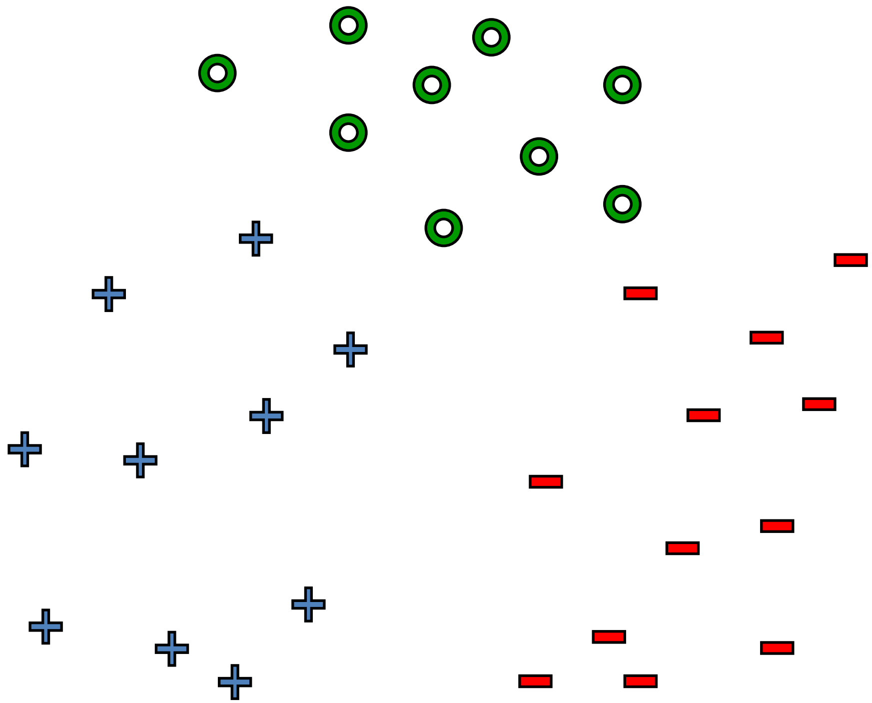
Learn 1 classifier: Multi-class SVM

Simultaneously learn 3 sets of weights

minimize $\sum_y \mathbf{w}^{(y)}.\mathbf{w}^{(y)} + C\sum_j \sum_{y\neq y_j} \xi_j^{(y)}$ over $\{w^{(y)}\}$, $\{b^{(y)}\}$, $\{\xi_j^{(y)}\}$

$$\mathbf{w}^{(y_j)}.\mathbf{x}_j + b^{(y_j)} \geq \mathbf{w}^{(y)}.\mathbf{x}_j + b^{(y)} + 1 - \xi_j^{(y)}, \ \forall y \neq y_j, \ \forall j$$

$$\xi_j^{(y)} \geq 0 \quad , \ \forall y \neq y_j, \ \forall j$$



$y = \arg\max \mathbf{w}^{(k)}.x + b^{(k)}$

Joint optimization: $\mathbf{w}_k$s have the same scale.

Support Vector Machines - Dual formulation

Aarti Singh

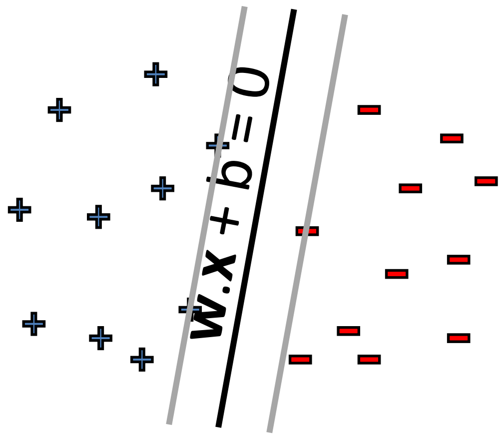
Machine Learning 10-315
Oct 26, 2020

MACHINE LEARNING DEPARTMENT
Carnegie Mellon. School of Computer Science

SVM – linearly separable case

n training points $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$

d features $\mathbf{x}_j$ is a d-dimensional vector



- Primal problem: $\text{minimize}_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}.\mathbf{w}$

$$\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j \geq 1, \ \forall j$$

w – weights on features (d-dim problem)

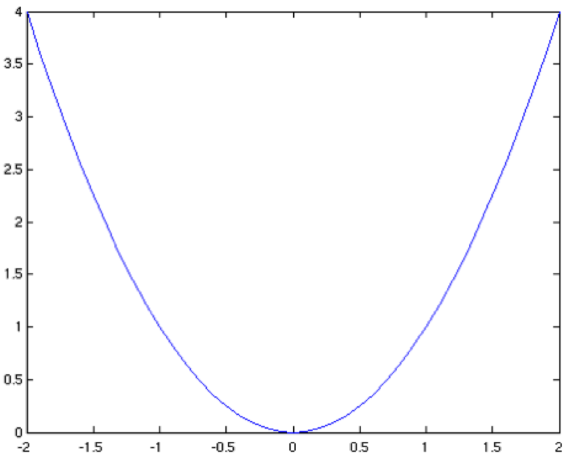
- Convex quadratic program – quadratic objective, linear constraints
- But expensive to solve if d is very large
- Often solved in dual form (n-dim problem)

Detour - Constrained Optimization

$$\min_x \ x^2 \quad \text{s.t.} \quad x \geq b$$

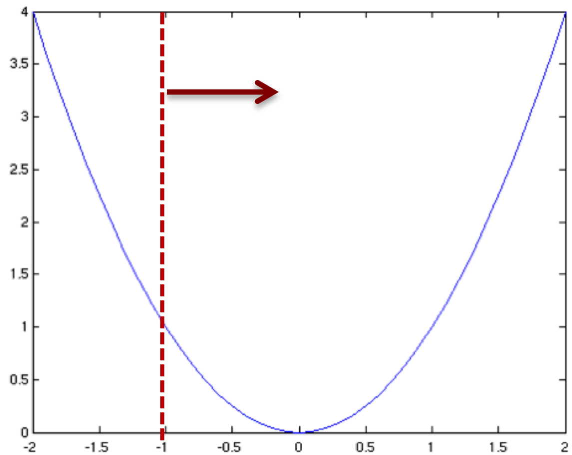
$$x^* = \max(b, 0)$$

$$\min_x \ x^2$$



$$x^* = 0$$

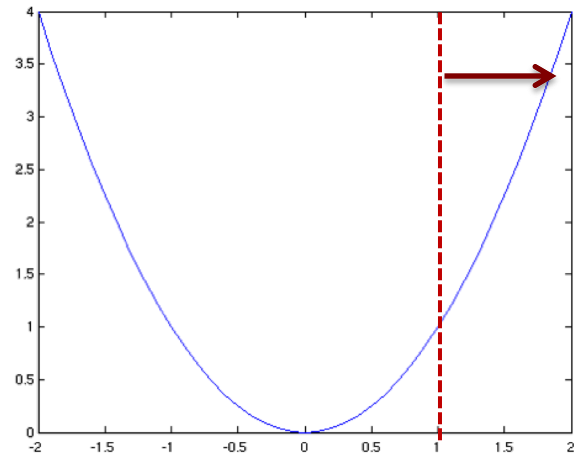
$$\min_x \ x^2 \quad \text{s.t.} \quad x \geq -1$$



$$x^* = 0$$

Constraint inactive

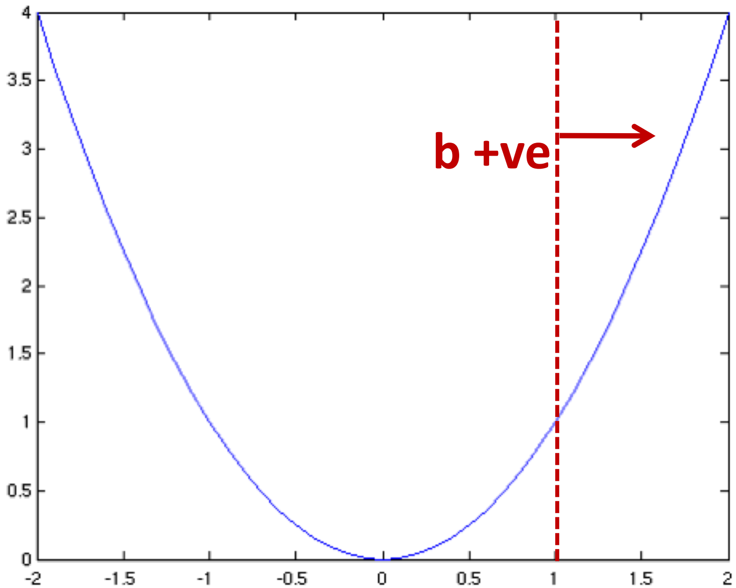
$$\min_x \ x^2 \quad \text{s.t.} \quad x \geq 1$$



$$x^* = 1$$

Constraint active (tight)

Constrained Optimization



$x^* = b$

$$\min_x \; x^2$$
$$\text{s.t.} \quad x \geq b$$

Equivalent unconstrained optimization:

$\min_x \; x^2 + I(x-b)$

Replace with lower bound ($\alpha >= 0$)

$x^2 + I(x-b) \;>=\; x^2 - \alpha(x-b)$

Primal and Dual Problems

Notice that

Primal problem: p* = $\min_x \; x^2$ $\quad$ = $\quad \min_x \max_{\alpha \geq 0} L(x, \alpha)$

s.t. $\; x \geq b$

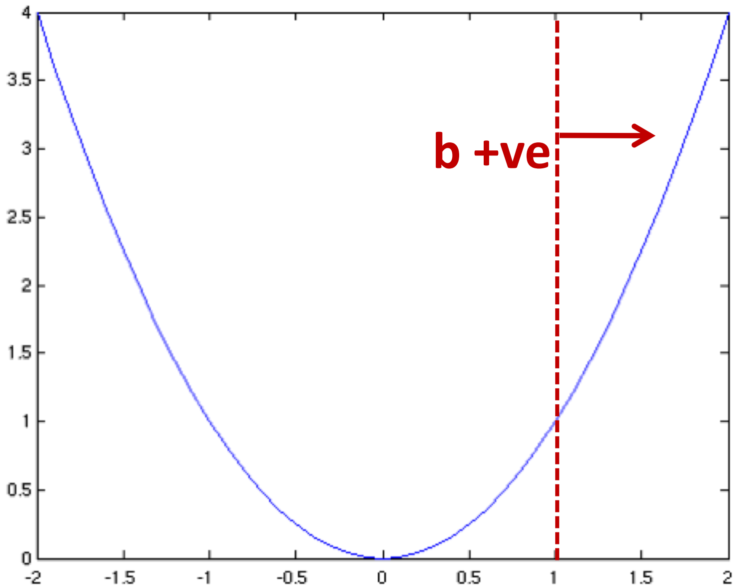
Why? $\quad L(x, \alpha) = x^2 - \alpha(x - b)$

$$\max_{\alpha \geq 0} L(x, \alpha) = x^2 - \min_{\alpha \geq 0} \alpha(x - b) \;\; =$$

Dual problem: d* = $\max_\alpha \; d(\alpha)$ = $\max_\alpha \min_x L(x, \alpha)$

s.t. $\; \alpha \geq 0$ $\qquad$ s.t. $\; \alpha \geq 0$

Constrained Optimization – Dual Problem



$x^* = b$

Primal problem:

$$\min_x \; x^2$$
$$\text{s.t.} \quad x \geq b$$

Moving the constraint to objective function

Lagrangian:

$$L(x, \alpha) = x^2 - \alpha(x - b)$$
$$\text{s.t.} \quad \alpha \geq 0$$

Dual problem:

$$\max_\alpha \; d(\alpha) \rightarrow \min_x L(x, \alpha)$$
$$\text{s.t.} \quad \alpha \geq 0$$

α = 0 constraint is inactive

α > 0 constraint is active

Connection between Primal and Dual

Primal problem: p* = $\min_x \; x^2$
s.t. $x \geq b$

= $\min_x \max_{\alpha \geq 0} L(x, \alpha)$

Dual problem: d* = $\max_\alpha \; d(\alpha)$
s.t. $\alpha \geq 0$

= $\max_\alpha \min_x L(x, \alpha)$
s.t. $\alpha \geq 0$

- **Dual problem (maximization) is always concave even if primal is not convex**

  Why? Pointwise infimum of concave functions is concave.
  [Pointwise supremum of convex functions is convex.]

  $$L(x, \alpha) = x^2 - \alpha(x - b)$$

- **As many dual variables α as constraints, helpful if fewer constraints than dimension of primal variable x**

Connection between Primal and Dual

Primal problem: p* = $\min_x \; x^2$ s.t. $x \geq b$

Dual problem: d* = $\max_\alpha \; d(\alpha)$ s.t. $\alpha \geq 0$

- **Weak duality:** The dual solution d* lower bounds the primal solution p* i.e. d* ≤ p*

To see this, recall $L(x, \alpha) = x^2 - \alpha(x - b)$

For every feasible x' (i.e. x' ≥ b) and feasible α' (i.e. α' ≥ 0) , notice that

$$d(\alpha) = \min_x L(x, \alpha) \leq x'^2 - \alpha'(x'-b) \leq x'^2$$

Since above holds true for every feasible x', we have $d(\alpha) \leq x^{*2} = p^*$

Connection between Primal and Dual

Primal problem: p* = $\min_x \; x^2$
s.t. $x \geq b$

Dual problem: d* = $\max_\alpha \; d(\alpha)$
s.t. $\alpha \geq 0$

- **Weak duality:** The dual solution d* lower bounds the primal solution p* i.e. d* ≤ p*
- **Strong duality:** d* = p* holds often for many problems of interest e.g. if the primal is a feasible convex objective with linear constraints

Connection between Primal and Dual

What does strong duality say about α^* (the α that achieved optimal value of dual) and x^* (the x that achieves optimal value of primal problem)?

Whenever strong duality holds, the following conditions (known as KKT conditions) are true for α^* and x^*:

- 1. $\nabla L(x^*, \alpha^*) = 0$ i.e. Gradient of Lagrangian at x^* and α^* is zero.
- 2. $x^* \geq b$ i.e. x^* is primal feasible
- 3. $\alpha^* \geq 0$ i.e. α^* is dual feasible
- 4. $\alpha^*(x^* - b) = 0$ (called as complementary slackness)

We use the first one to relate x^* and α^*. We use the last one (complimentary slackness) to argue that $\alpha^* = 0$ if constraint is inactive and $\alpha^* > 0$ if constraint is active and tight.

Solving the dual

Solving:

$$\max_\alpha \min_x \underbrace{x^2 - \alpha(x-b)}_{L(x,\alpha)}$$

$$\text{s.t.} \quad \alpha \geq 0$$

Solving the dual

Solving:

$$\overbrace{}^{L(x,\alpha)}$$

$$\max_\alpha \min_x \ x^2 - \alpha(x-b)$$
$$\text{s.t.} \quad \alpha \geq 0$$

<u>Find the dual:</u> Optimization over x is unconstrained.

$$\frac{\partial L}{\partial x} = 2x - \alpha = 0 \Rightarrow x^* = \frac{\alpha}{2}$$

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

<u>Solve</u>: Now need to maximize $L(x^*,\alpha)$ over $\alpha \geq 0$
Solve unconstrained problem to get α' and then take $\max(\alpha',0)$

$$\frac{\partial}{\partial \alpha} L(x^*, \alpha) = -\frac{\alpha}{2} + b \ \Rightarrow \alpha' = 2b$$

$$\Rightarrow \alpha^* = \max(2b, 0) \qquad \Rightarrow x^* = \frac{\alpha^*}{2} = \max(b, 0)$$

α = 0 constraint is inactive, α > 0 constraint is active (tight)

Dual SVM – linearly separable case

n training points, d features $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ where x_i is a d-dimensional vector

- Primal problem:

$$\text{minimize}_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}.\mathbf{w}$$
$$\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j \geq 1, \ \forall j$$

w - weights on features (d-dim problem)

- Dual problem (derivation):

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}.\mathbf{w} - \sum_j \alpha_j \left[\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j - 1\right]$$
$$\alpha_j \geq 0, \ \forall j$$

α - weights on training pts (n-dim problem)

Dual SVM – linearly separable case

- Dual problem:

$$\max_\alpha \min_{\mathbf{w},b} L(\mathbf{w},b,\alpha) = \frac{1}{2}\mathbf{w}.\mathbf{w} - \sum_j \alpha_j \left[\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j - 1\right]$$

$$\alpha_j \geq 0, \ \forall j$$

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \qquad \Rightarrow \mathbf{w} = \sum_j \alpha_j y_j \mathbf{x}_j$$

$$\frac{\partial L}{\partial b} = 0 \qquad \Rightarrow \sum_j \alpha_j y_j = 0$$

If we can solve for αs (dual problem), then we have a solution for **w**,b (primal problem)

Dual SVM – linearly separable case

- Dual problem:

$$\max_\alpha \min_{\mathbf{w},b} L(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}.\mathbf{w} - \sum_j \alpha_j \left[\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j - 1\right]$$

$$\alpha_j \geq 0, \ \forall j$$

$$\Rightarrow \mathbf{w} = \sum_j \alpha_j y_j \mathbf{x}_j \qquad \Rightarrow \sum_j \alpha_j y_j = 0$$

Dual SVM – linearly separable case

$$\text{maximize}_\alpha \quad \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i . \mathbf{x}_j$$

$$\sum_i \alpha_i y_i = 0$$

$$\alpha_i \geq 0$$

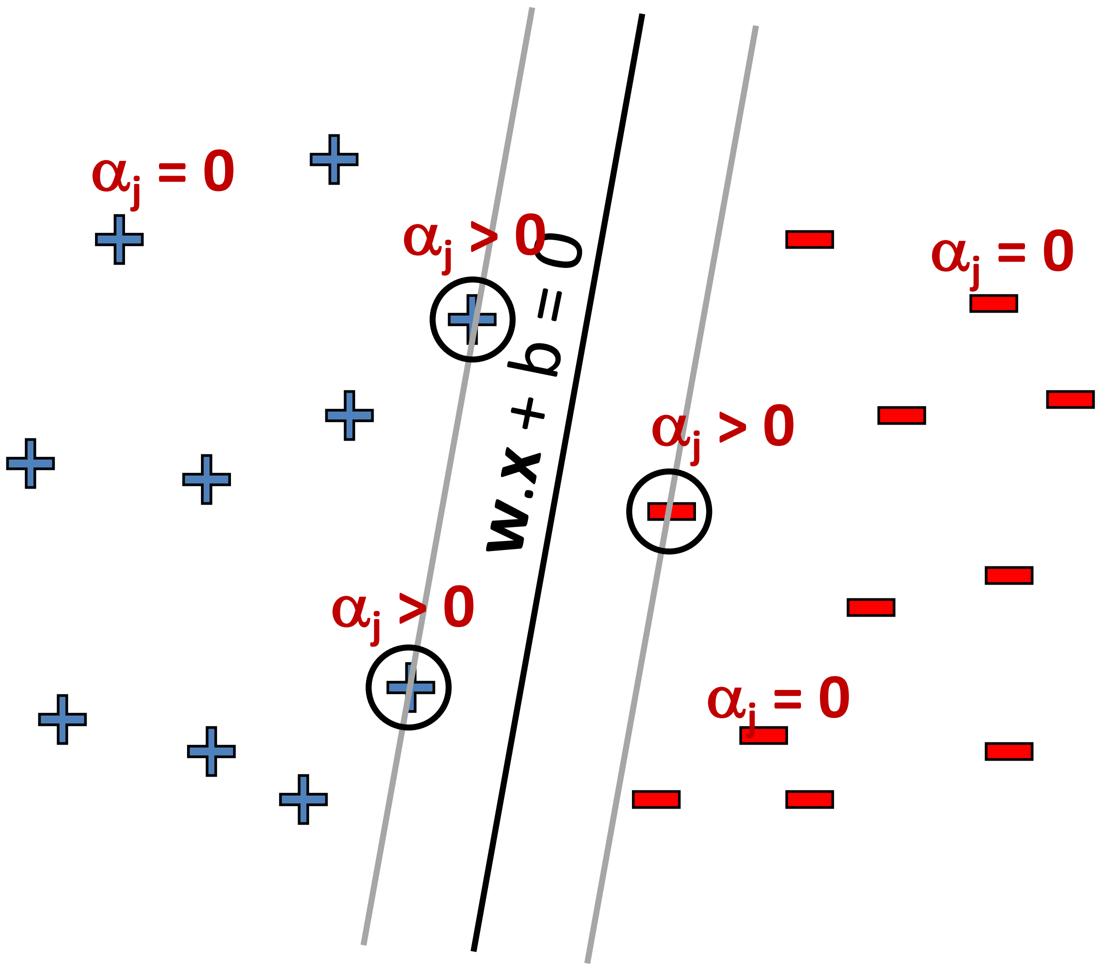
Dual problem is also QP

Solution gives α_js →

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

What about b?

Dual SVM: Sparsity of dual solution



$$\mathbf{w} = \sum_j \alpha_j y_j \mathbf{x}_j$$

Only few α_js can be non-zero : where constraint is active and tight

$$(\mathbf{w}.\mathbf{x}_j + b)y_j = 1$$

Support vectors – training points j whose α_js are non-zero

Dual SVM – linearly separable case

$$\text{maximize}_\alpha \quad \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j \mathbf{x}_i.\mathbf{x}_j$$

$$\sum_i \alpha_i y_i = 0$$

$$\alpha_i \geq 0$$

Dual problem is also QP

Solution gives α_js →

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

$$b = y_k - \mathbf{w}.\mathbf{x}_k$$

for any k where $\alpha_k > 0$

Use any one of support vectors with α_k>0 to compute b since constraint is tight (w.x_k + b)y_k = 1

Dual SVM – non-separable case

- Primal problem:

$$\text{minimize}_{\mathbf{w},b,\{\xi_j\}} \frac{1}{2}\mathbf{w}.\mathbf{w} + C\sum_j \xi_j$$
$$\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j \geq 1 - \xi_j, \ \ \forall j$$
$$\xi_j \geq 0, \ \ \forall j$$

α_j
μ_j

Lagrange Multipliers

- Dual problem:

$$\max_{\alpha,\mu} \min_{\mathbf{w},b,\{\xi_j\}} L(\mathbf{w}, b, \xi, \alpha, \mu)$$
$$s.t. \alpha_j \geq 0 \quad \forall j$$
$$\mu_j \geq 0 \quad \forall j$$

HW3!

Dual SVM – non-separable case

maximize$_\alpha$ $\sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i . \mathbf{x}_j$

$\sum_i \alpha_i y_i = 0$

$C \geq \alpha_i \geq 0$

comes from $\frac{\partial L}{\partial \xi} = 0$

Intuition:

If C→∞, recover hard-margin SVM

Dual problem is also QP

Solution gives α_js →

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

$$b = y_k - \mathbf{w}.\mathbf{x}_k$$

for any k where $C > \alpha_k > 0$

So why solve the dual SVM?

- There are some quadratic programming algorithms that can solve the dual faster than the primal, (specially in high dimensions d>>n)

- But, more importantly, the "**kernel trick**"!!!