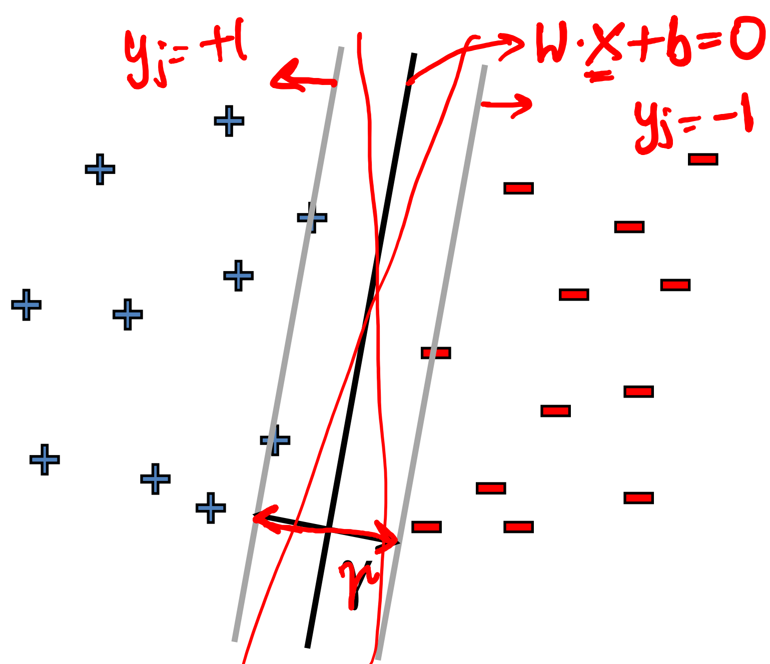
Support Vector Machines (SVMs) Recap…

Aarti Singh

Machine Learning 10-315
Oct 26, 2020

Hard-margin SVM

Data perfectly separable by a linear decision boundary



Hard margin approach

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

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

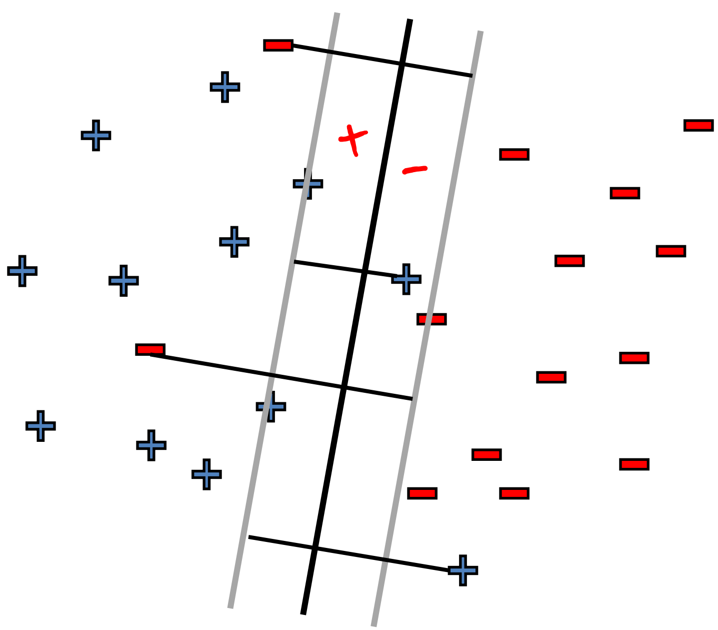
Solve using Quadratic Programming (QP)

$\boxed{\text{Margin, } \gamma \quad \alpha \quad 1/\|w\|}$

Soft-margin SVM

penalty for misclassified

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

$C \rightarrow \infty$

C - tradeoff parameter (chosen by cross-validation)

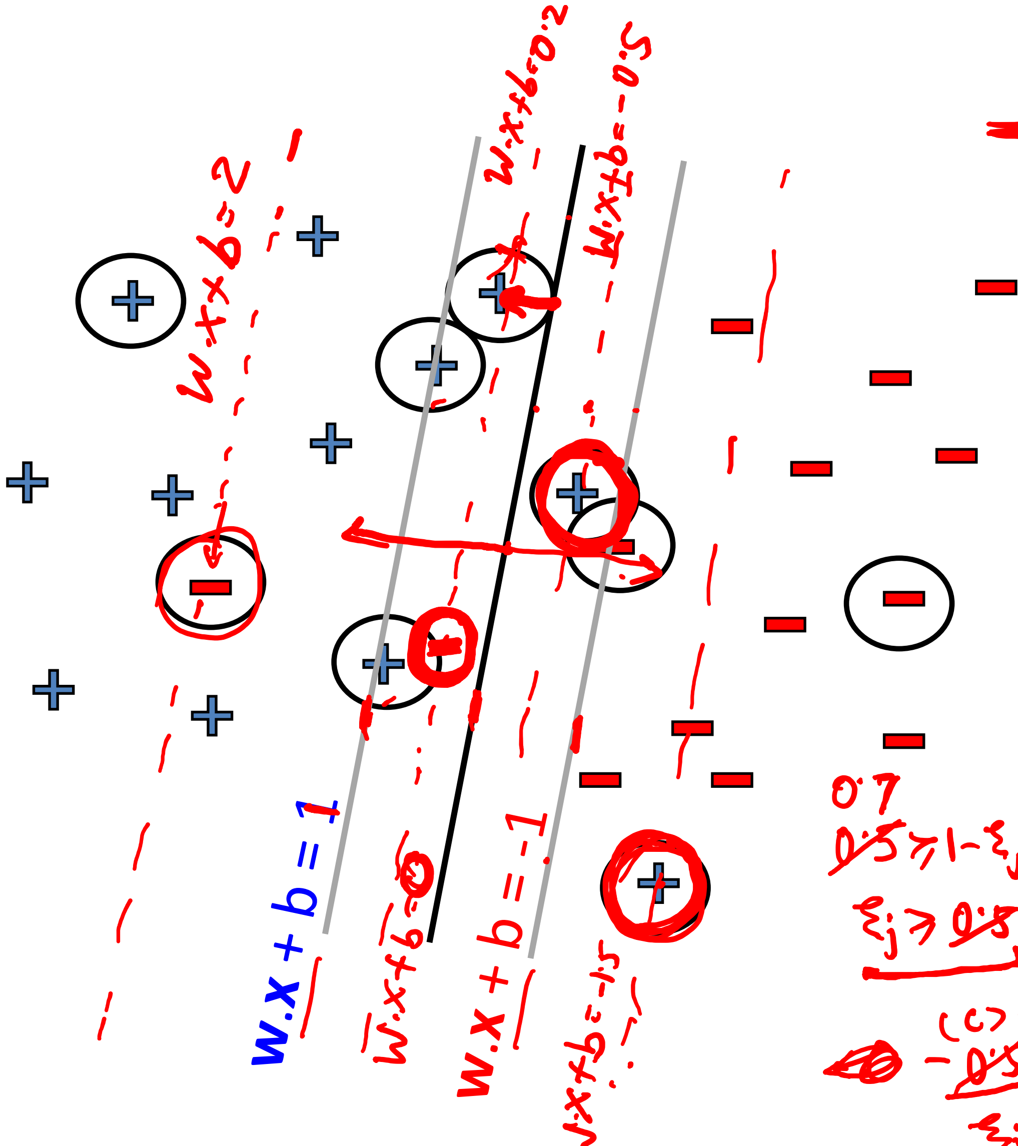
Still QP ☺

Slack variables – Hinge loss

$$\min_{w,b,\xi} \|w\|^2 + C\sum_j \xi_j$$

$$\xi_j \geq 0$$

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



What is the slack ξ_j for the following points?

Confidence	Slack
1	0
>1	0
0 – 1 (c)	0 – 1 (1 – c)
< 0 (c)	> 1 (1 – c)
– 0.2	1.2

0.7

$0.5 \geq 1-\xi_j$

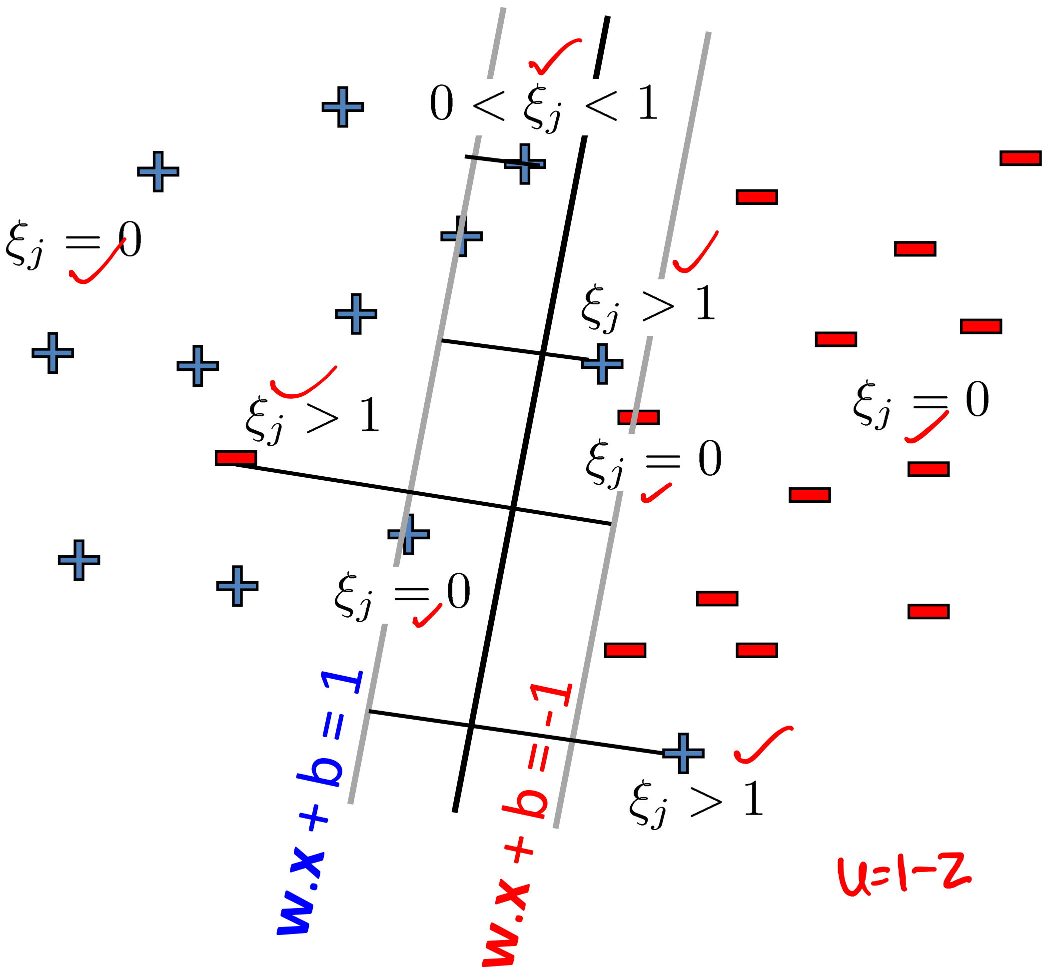
$\xi_j \geq \cancel{0.5}\ 0.3$

(c) –0.2

$-\cancel{0.5} \geq 1-\xi_j$

$\xi_j \geq 1-(-\cancel{0.5}) = \cancel{1.5}\ 1.2$ (0.2)

Slack variables – Hinge loss



Notice that

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

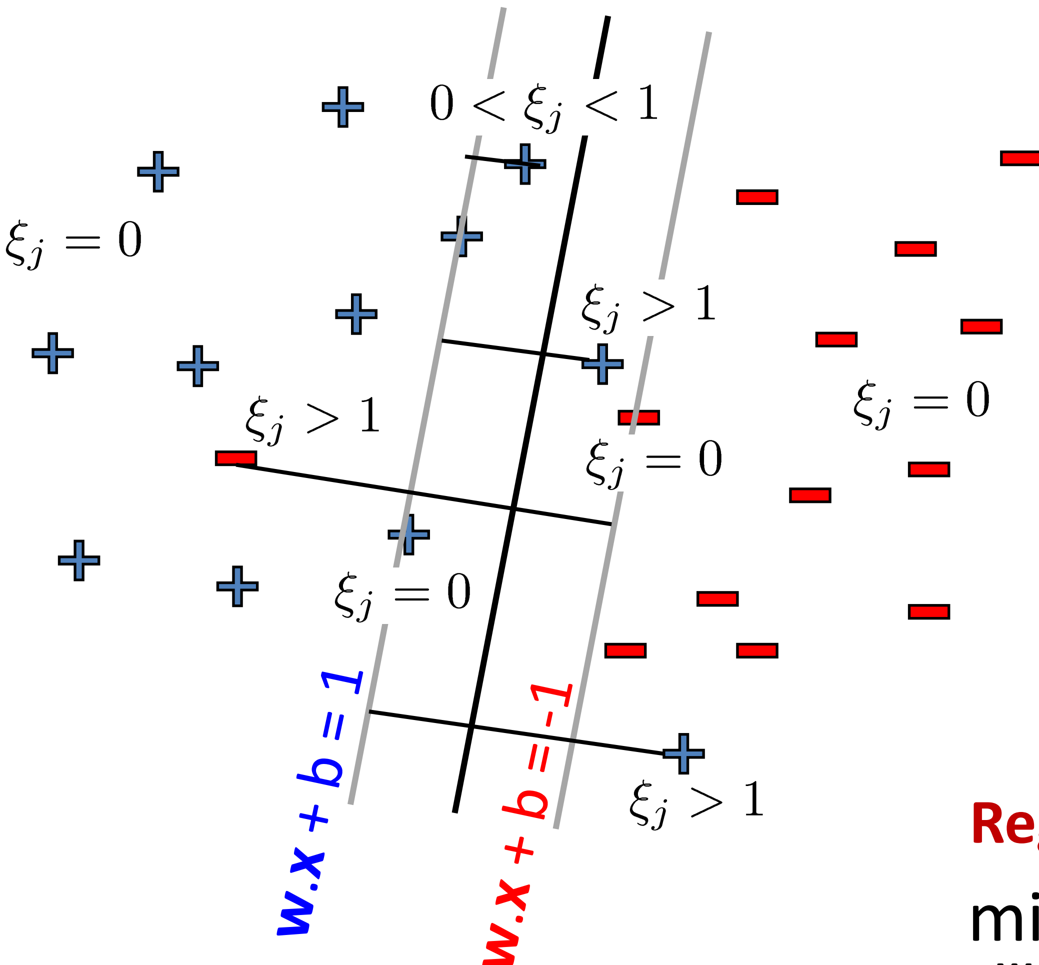
$$= \begin{cases} 1-(\mathbf{w}\cdot x_j + b)y_j & \text{if } c \le 1 \\ 0 & \text{if } c > 1 \end{cases}$$

$$(1-z)_+ = \begin{cases} 1-z & \text{if } z \le 1 \\ 0 & \text{if } z > 1 \end{cases}$$

$$u_+ = \begin{cases} u & \text{if } u \ge 0 \\ 0 & \text{o.w.} \end{cases}$$

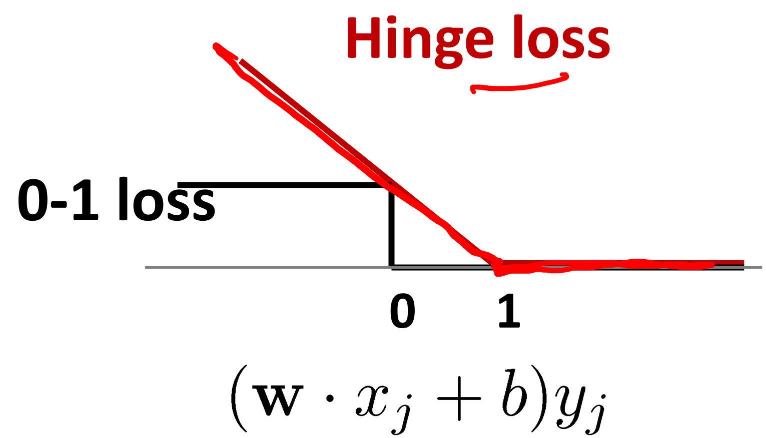
$u = 1 - z$

Slack variables – Hinge loss



Notice that

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



Regularized Hinge loss

loss

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

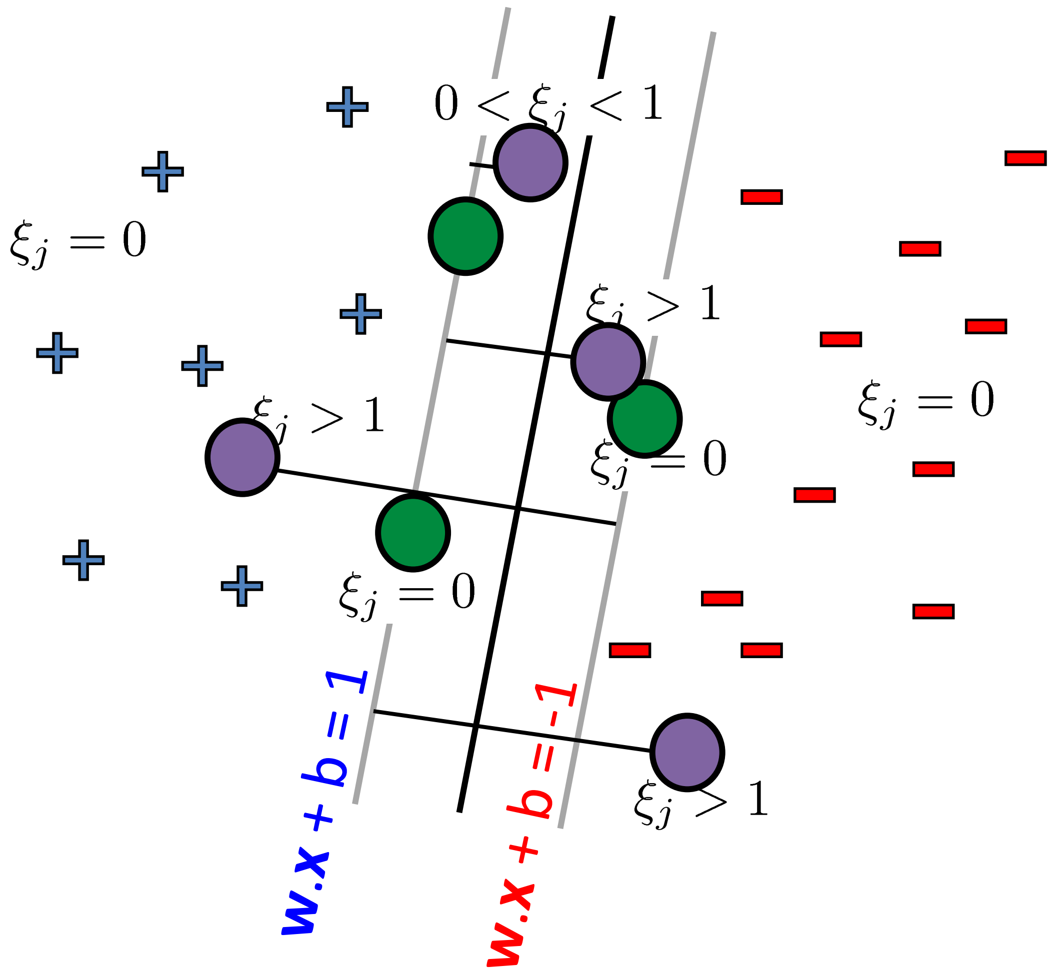
$w \cdot w + C \sum_j \xi_j$

$$\min_{\mathbf{w},b,\{\xi_j\}} \quad \mathbf{w}.\mathbf{w} + C \sum \xi_j$$

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

$$\xi_j \geq 0 \quad \forall j$$

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

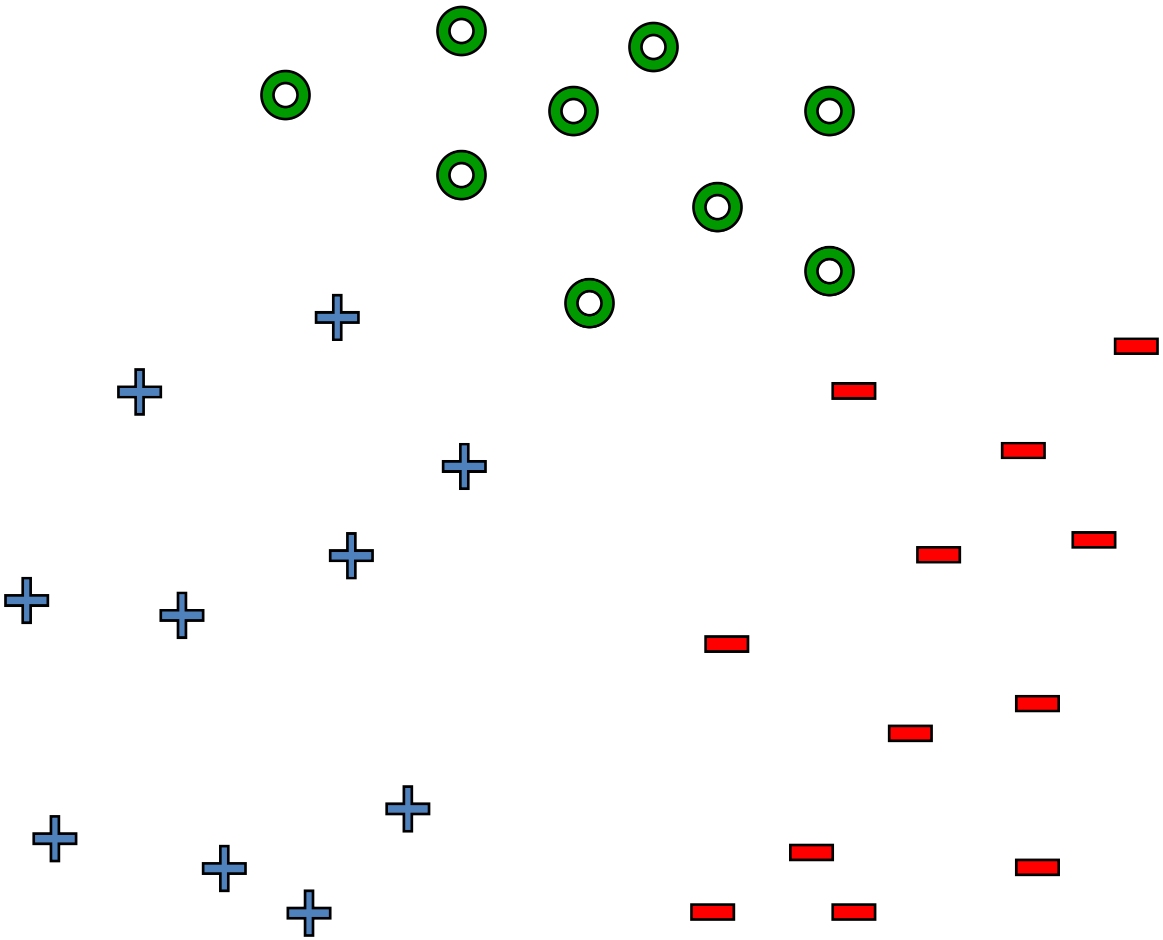
$\xi_j > 0$

(contribute to both objective and constraints)

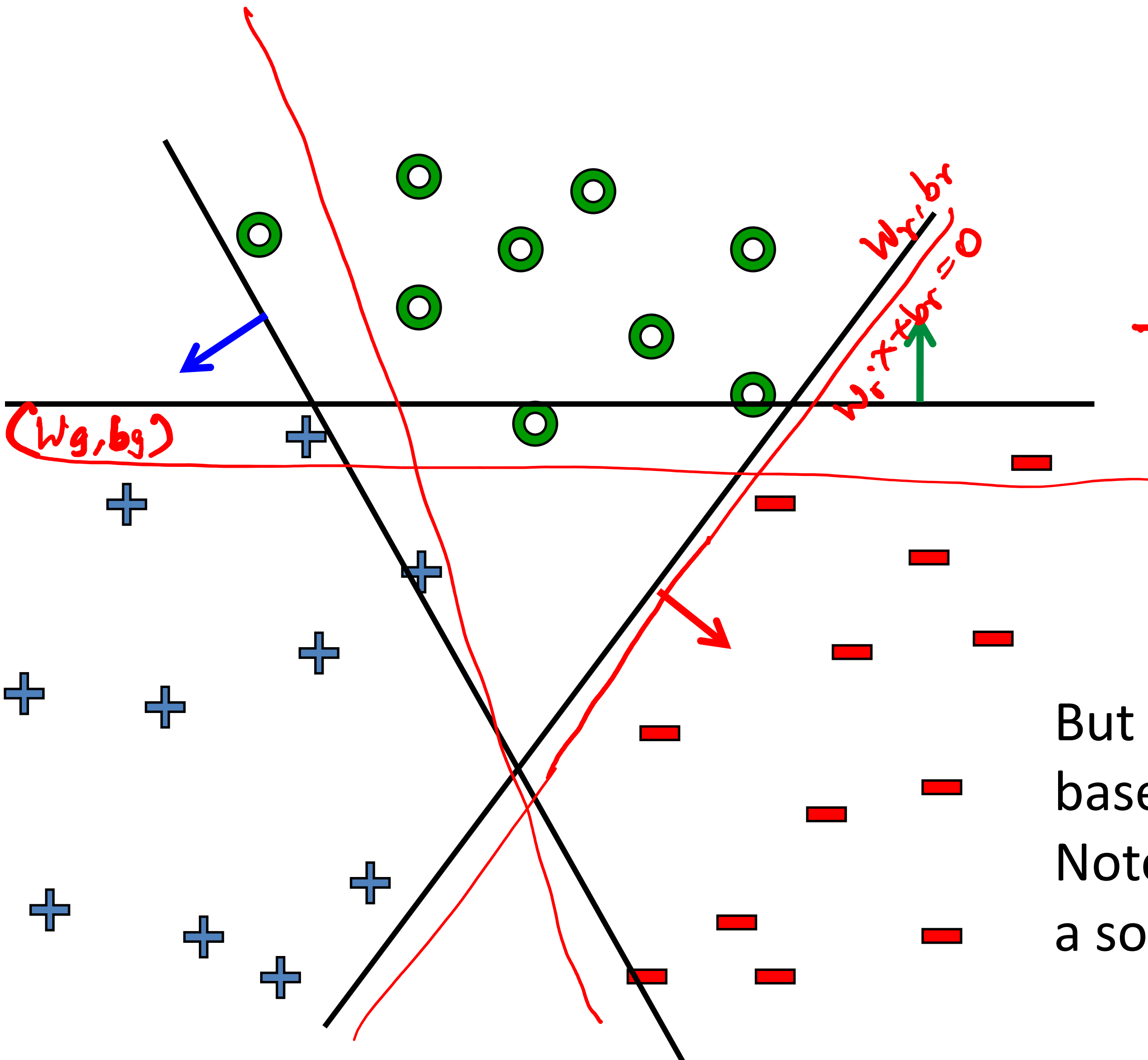
$1 > \xi_j > 0$ Correctly classified but inside margin

$\xi_j > 1$ Incorrectly classified

What about multiple classes?



One vs. rest



Learn 3 classifiers separately:

Class k vs. rest

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

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

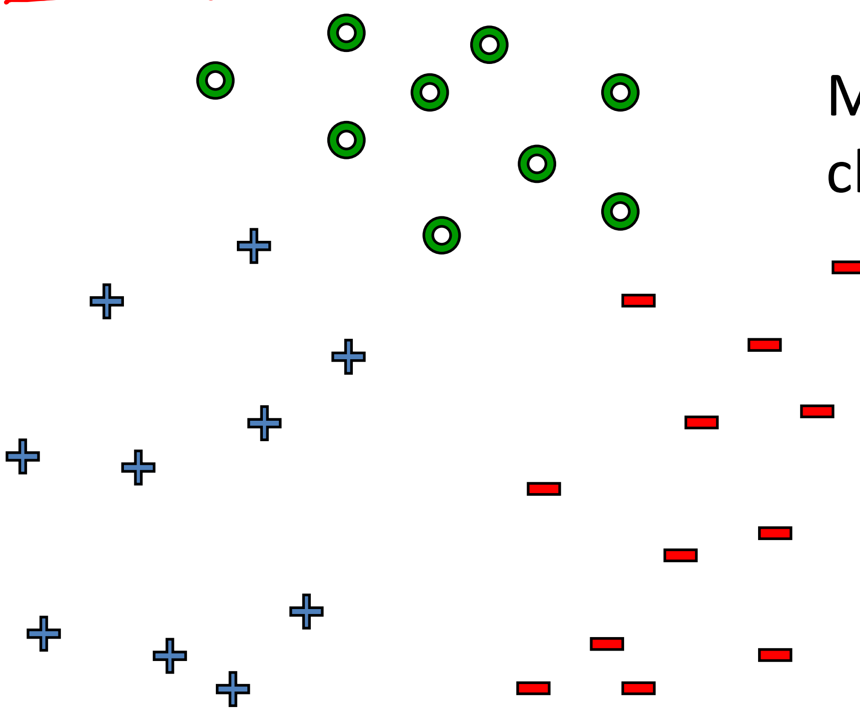
But $\mathbf{w}_k$s may not be based on the same scale. Note: $(a\mathbf{w}).x + (ab)$ is also a solution

Learn 1 classifier: Multi-class SVM

Simultaneously learn 3 sets of weights

$$\min_{\{w^{(y)}\},\ \{b^{(y)}\}} \quad \textstyle\sum_y \mathbf{w}^{(y)}.\mathbf{w}^{(y)}$$

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



Margin - gap between correct class and nearest other class

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

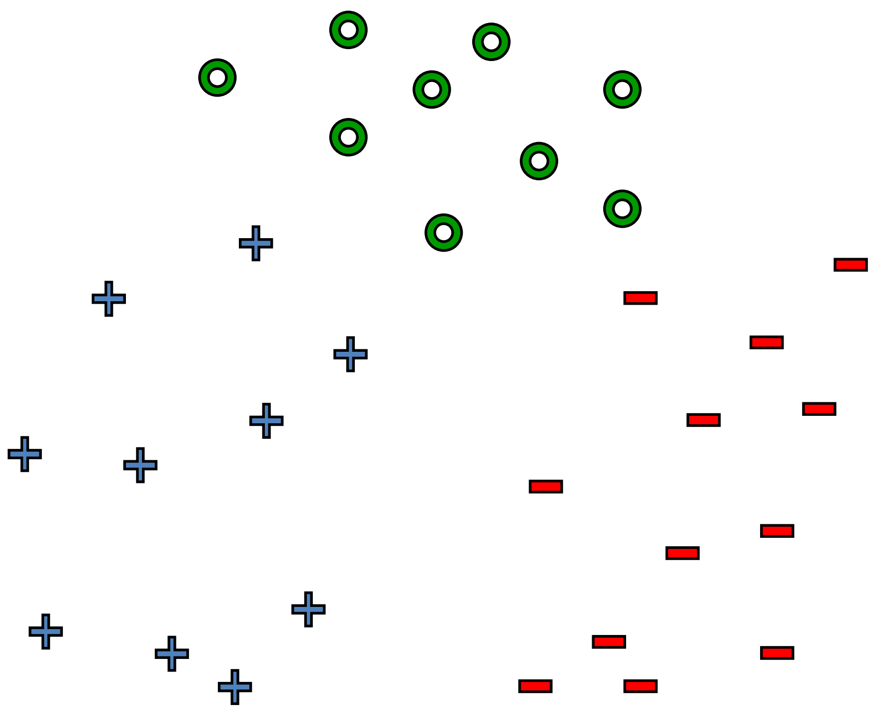
Learn 1 classifier: Multi-class SVM

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 $\{\mathbf{w}^{(y)}\}$, $\{b^{(y)}\}$, $\{\xi_j^{(y)}\}$

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

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



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

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

Support Vector Machines - Dual formulation

Aarti Singh

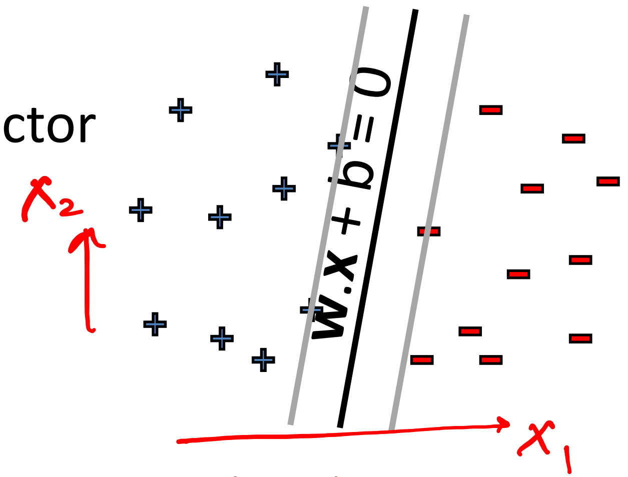
Machine Learning 10-315

Oct 26, 2020

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)

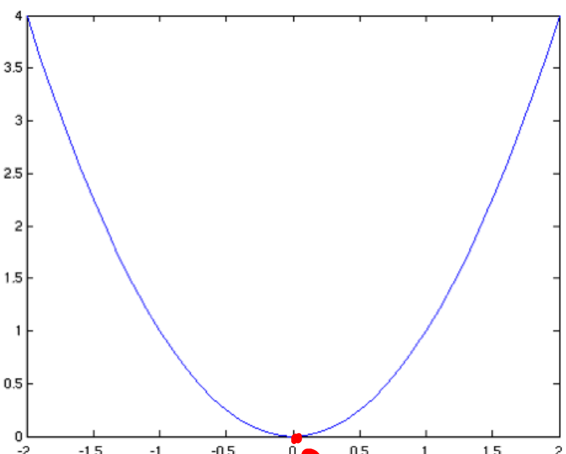
\# training points

Detour - Constrained Optimization

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

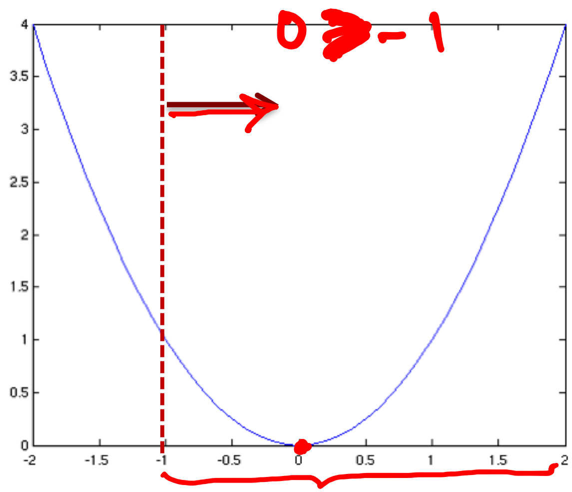
$$x^* = \max(b, 0)$$

$\min_x \; x^2$



$x^* = 0$

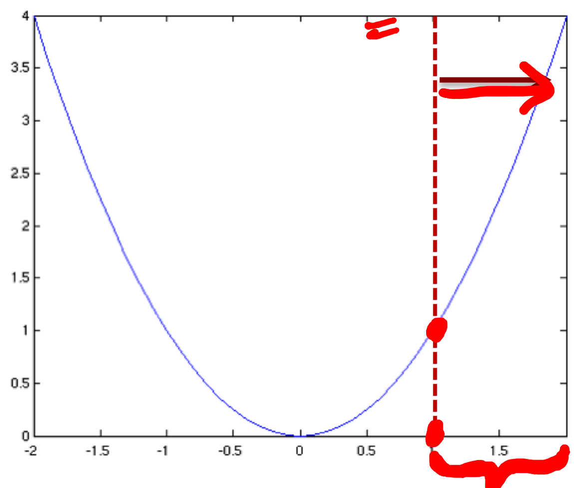
$\min_x \; x^2$ s.t. $x \geq -1$



$x^* = 0$

Constraint inactive

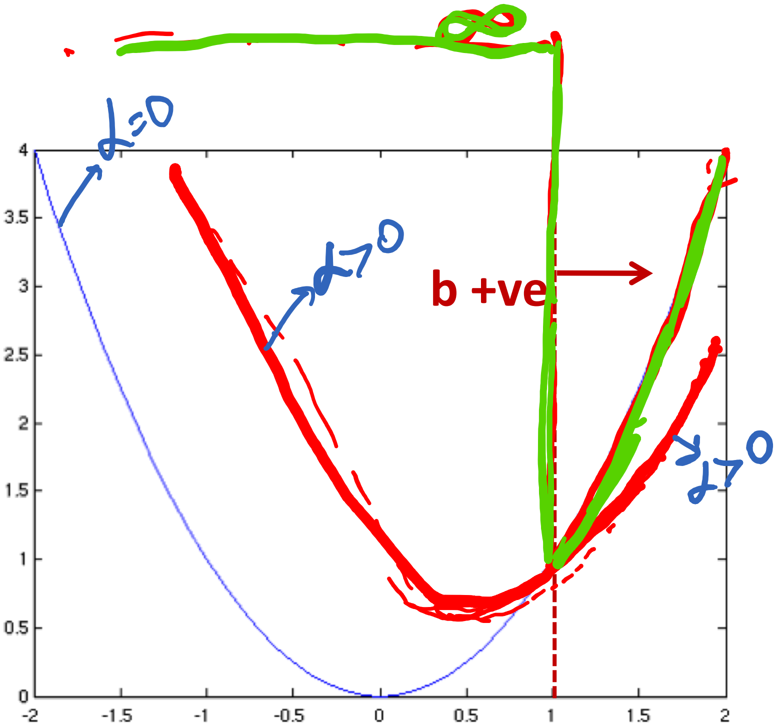
$\min_x \; x^2$ s.t. $x \geq 1$



$x^* = 1$

Constraint active (tight)

Constrained Optimization



$x^* = b$

$x < 2$

$-x > -2$

$-x+2 \geq 0$

$$\min_x \; x^2$$
$$\text{s.t.} \quad x \geq b \rightarrow x-b \geq 0$$

Equivalent unconstrained optimization:

$$\min_x \; x^2 + \underbrace{\begin{cases} \infty & x < b \\ 0 & x \geq b \end{cases}}_{} \quad I(x-b)$$

$$\min_x \; x^2 + \underbrace{I(x-b)}_{}$$

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

$\underbrace{x^2 + I(x-b)}_{} >= \underbrace{x^2 - \alpha(x-b)}_{\alpha>0 \quad x<b}$

Primal and Dual Problems

$L(x,\alpha) = x^2 - \alpha(x-b)$ $\alpha \geq 0$

Lagrangian $-\alpha'(x-c)$ Lagrange multiplier

Notice that

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

s.t. $x \geq b$

$x-b \geq 0$ $x-c \geq 0$

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

$$\max_{\alpha \geq 0} L(x,\alpha) = x^2 - \min_{\alpha \geq 0} \alpha(x-b) = \begin{cases} x^2 & x \geq b \\ \infty & x < b \end{cases}$$

∞ $-ve$

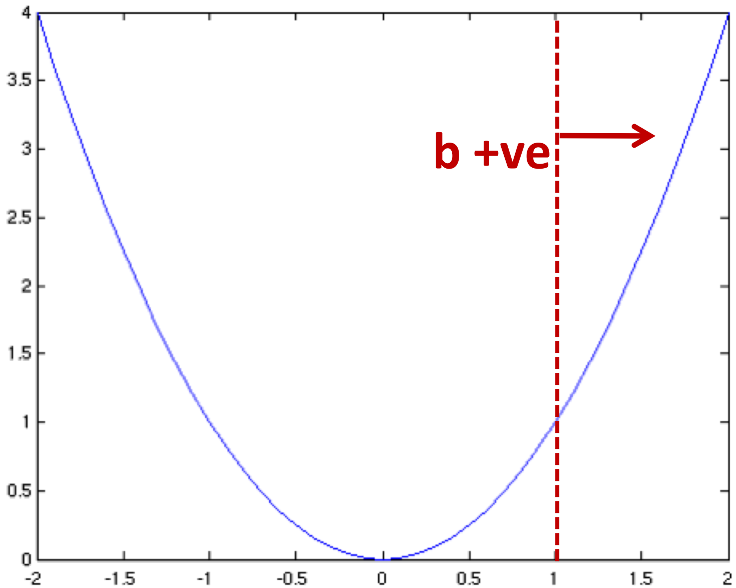
Dual problem: d* = $\max_\alpha \; d(\alpha)$ = $\max_\alpha \min_x L(x,\alpha)$

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

hard margin SVM –

$(w \cdot x_j + b) y_j \geq 1 \quad \forall j$

Constrained Optimization – Dual Problem



$$x^* = b$$

α = 0 constraint is inactive

α > 0 constraint is active

Primal problem:

$$\min_x \; x^2$$
$$\text{s.t.} \quad x \geq b \;\rightarrow\; \alpha \geq 0$$

$x - b \geq 0$

Moving the constraint to objective function

Lagrangian:

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

Dual problem:

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

Connection between Primal and Dual

Primal problem: p* = $\min_x \; x^2$

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

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

Dual problem: d* = $\max_\alpha \; d(\alpha)$

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

$= \max_\alpha \underbrace{\min_x L(x, \alpha)}_{d(\alpha)}$

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

- **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.]

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

- **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. $x \ge b$

x^*

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

- **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

$L(x', \alpha)$

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

+ve +ve

Since above holds true for every feasible x', we have d(α) ≤ 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