

Support Vector Machines, SVMs

Machine Learning – 10701/15781

Carlos Guestrin

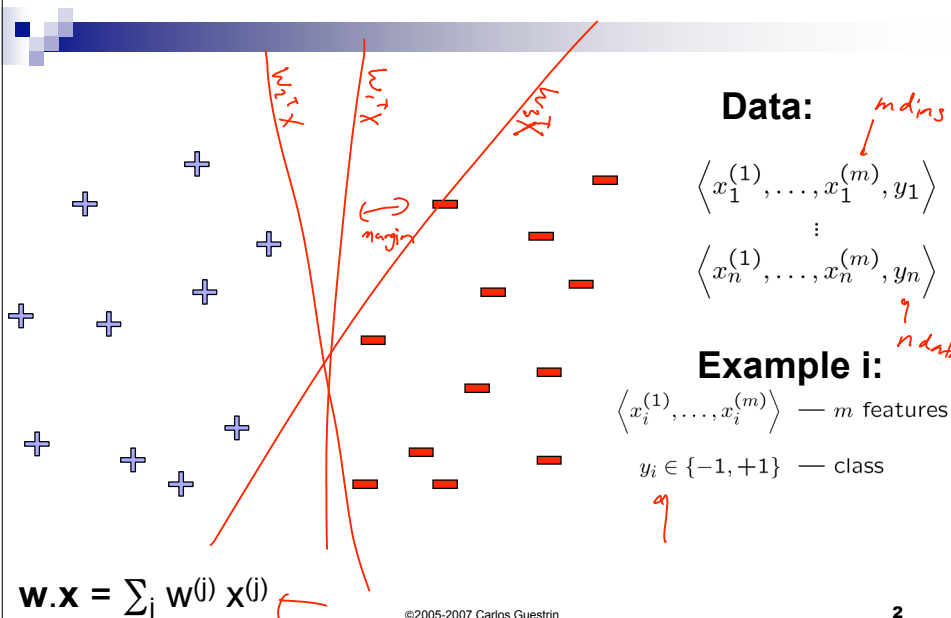
Carnegie Mellon University

October 17th, 2007

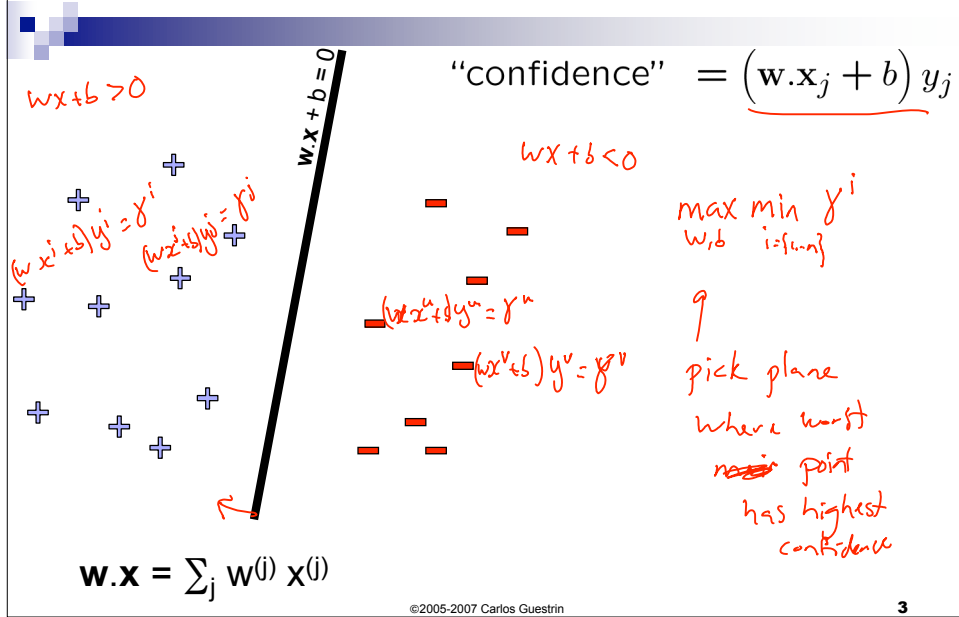
©2005-2007 Carlos Guestrin

1

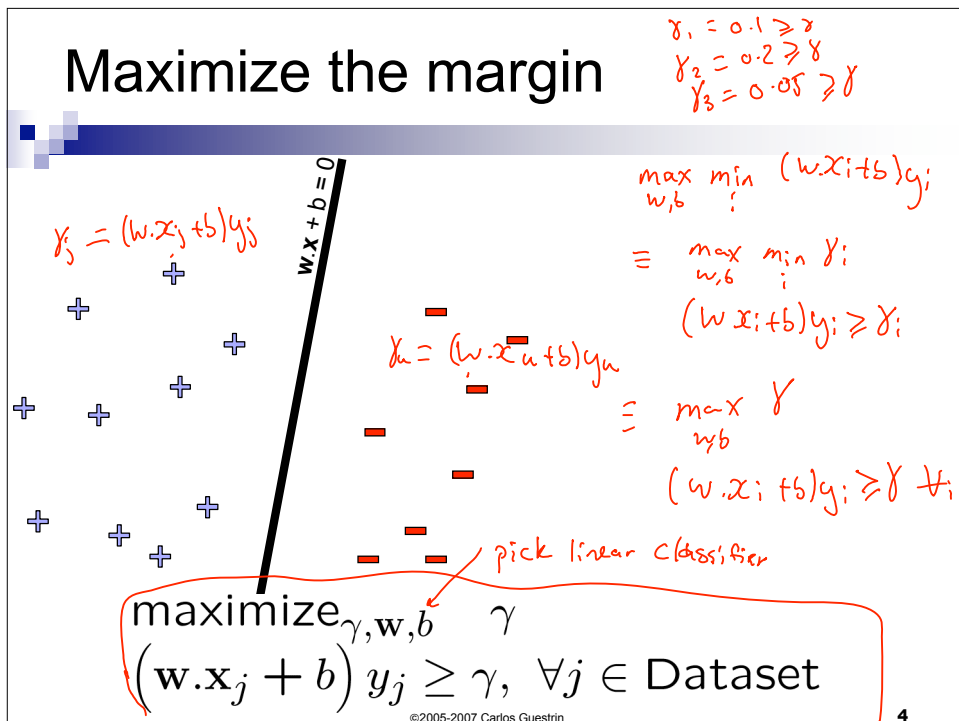
Linear classifiers – Which line is better?



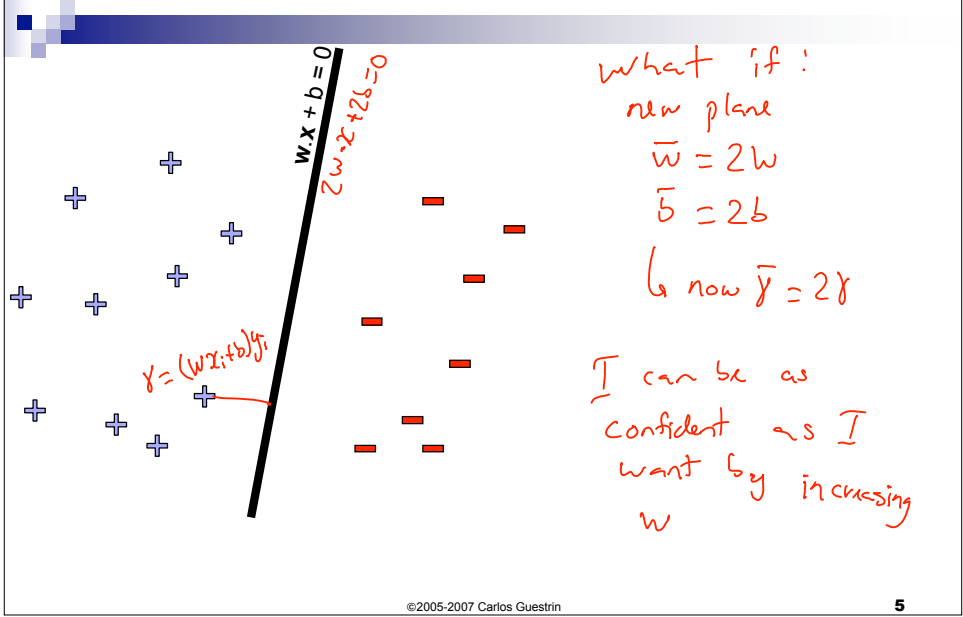
Pick the one with the largest margin!



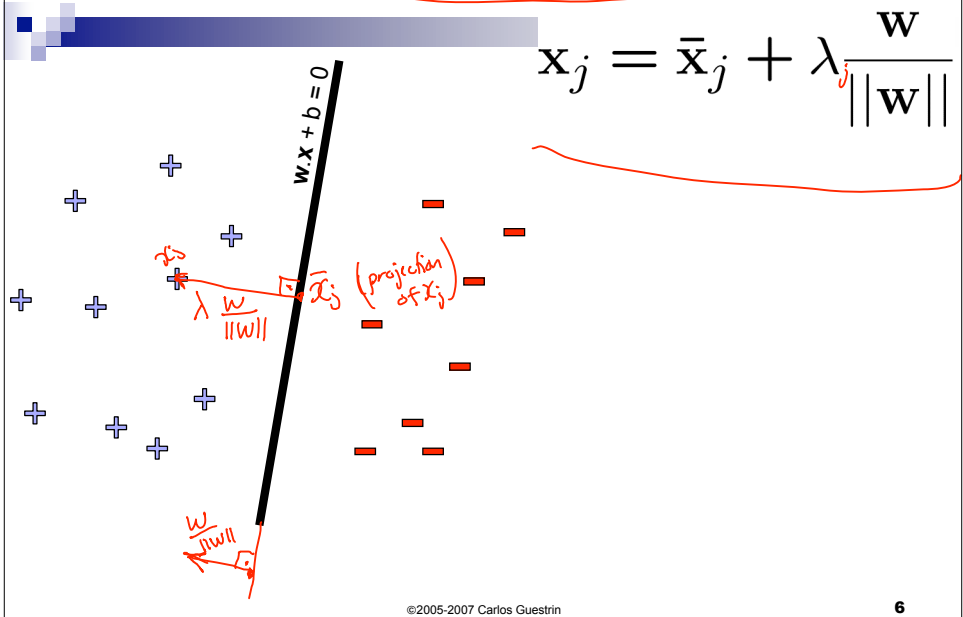
Maximize the margin

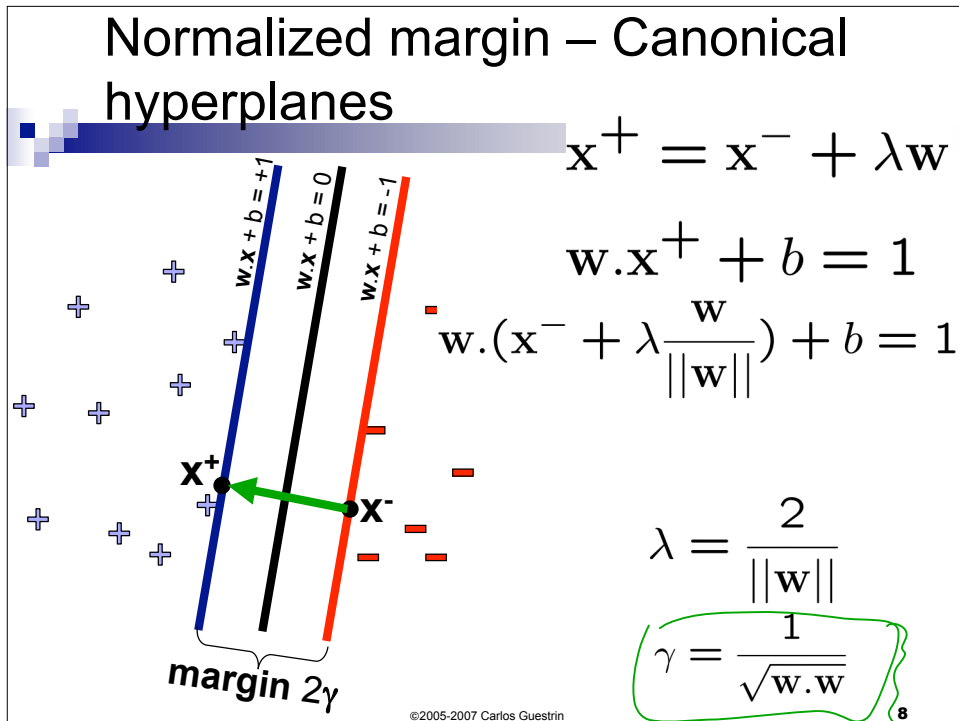
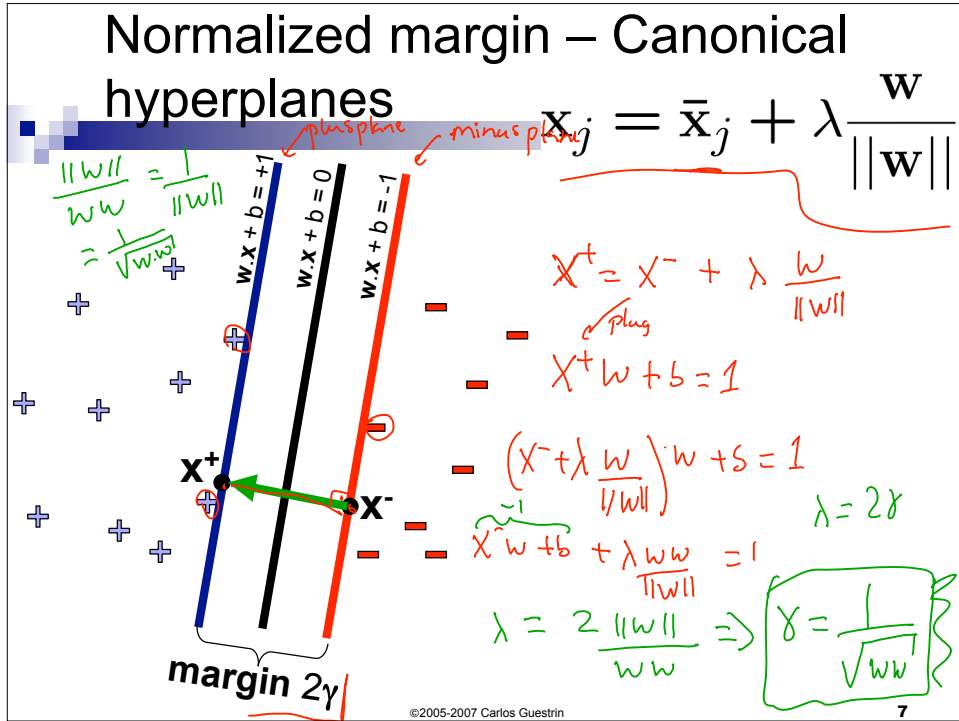


But there are a many planes...



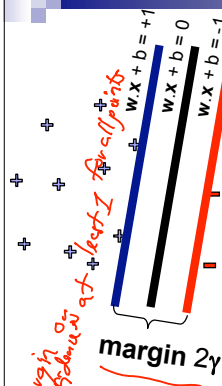
Review: Normal to a plane





Margin maximization using canonical hyperplanes

$$\gamma = \frac{1}{\sqrt{w \cdot w}}$$



maximize γ, w, b γ , $\gamma \leq 1$
 $(w \cdot x_j + b) y_j \geq \gamma, \forall j \in \text{Dataset}$

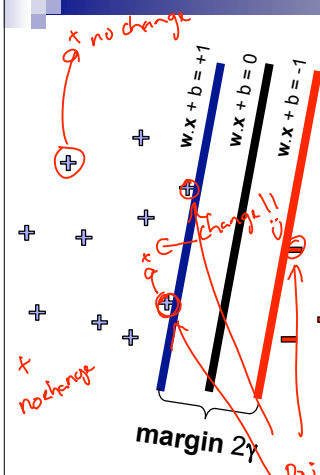
$\max \frac{1}{\sqrt{w \cdot w}}$
 $(w \cdot x_j + b) y_j \geq \gamma, \gamma \leq 1$
 $\equiv \min w \cdot w$
 $(w \cdot x_j + b) y_j \geq \gamma; \gamma \leq 1$

$\max \frac{1}{\sqrt{w \cdot w}}$
 $\equiv \min \sqrt{w \cdot w}$
 $\equiv \min w \cdot w$
 is monotonic function

minimize w, b $w \cdot w$
 $(w \cdot x_j + b) y_j \geq 1, \forall j \in \text{Dataset}$

maximum achieved when $\gamma = 1$

Support vector machines (SVMs)



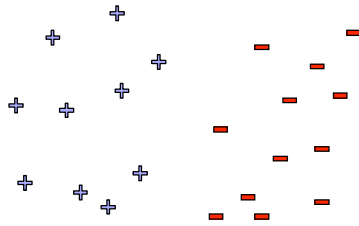
minimize w, b $w \cdot w$
 $(w \cdot x_j + b) y_j \geq 1, \forall j$

- Solve efficiently by quadratic programming (QP)
 - Well-studied solution algorithms
- Hyperplane defined by support vectors

points where $(w \cdot x_j + b) y_j = 1$ (support vectors)

What if the data is not linearly separable?

Use features of features of features of features....

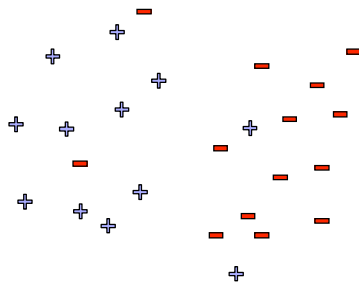


©2005-2007 Carlos Guestrin

11

What if the data is still not linearly separable?

$$\text{minimize}_{\mathbf{w}, b} \quad \mathbf{w} \cdot \mathbf{w} \\ (\mathbf{w} \cdot \mathbf{x}_j + b) y_j \geq 1 \quad , \forall j$$



- Minimize $\mathbf{w} \cdot \mathbf{w}$ and number of training mistakes
 - Tradeoff two criteria?
- Tradeoff #(mistakes) and $\mathbf{w} \cdot \mathbf{w}$
 - 0/1 loss
 - Slack penalty C
 - Not QP anymore
 - Also doesn't distinguish near misses and really bad mistakes

©2005-2007 Carlos Guestrin

12

Slack variables – Hinge loss

$$\text{minimize}_{w,b} \quad w \cdot w$$

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

- If margin ≥ 1 , don't care
- If margin < 1 , pay linear penalty

©2005-2007 Carlos Guestrin 13

Side note: What's the difference between SVMs and logistic regression?

SVM:

$$\text{minimize}_{w,b} \quad w \cdot w + C \sum_j \xi_j$$

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

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

Logistic regression:

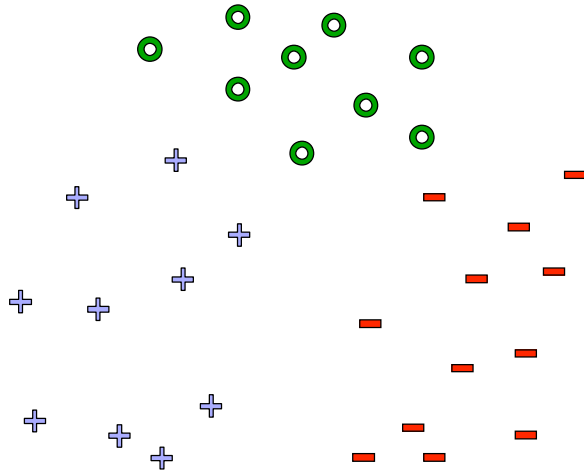
$$P(Y = 1 | x, w) = \frac{1}{1 + e^{-(\mathbf{w} \cdot \mathbf{x} + b)}}$$

Log loss:

$$-\ln P(Y = 1 | x, w) = \ln(1 + e^{-(\mathbf{w} \cdot \mathbf{x} + b)})$$

©2005-2007 Carlos Guestrin 14

What about multiple classes?

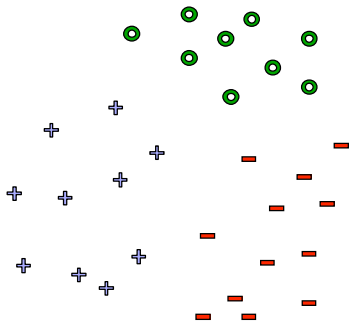


©2005-2007 Carlos Guestrin

15

One against All

Learn 3 classifiers:

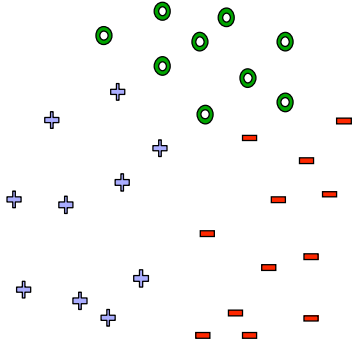


©2005-2007 Carlos Guestrin

16

Learn 1 classifier: Multiclass SVM

Simultaneously learn 3 sets of weights



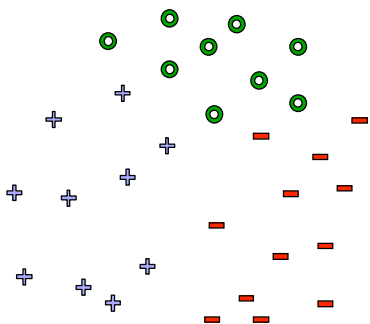
$$\mathbf{w}^{(y_j)} \cdot \mathbf{x}_j + b^{(y_j)} \geq \mathbf{w}^{(y')} \cdot \mathbf{x}_j + b^{(y')} + 1, \quad \forall y' \neq y_j, \quad \forall j$$

©2005-2007 Carlos Guestrin

17

Learn 1 classifier: Multiclass SVM

$$\begin{aligned} & \text{minimize}_{\mathbf{w}, b} \quad \sum_y \mathbf{w}^{(y)} \cdot \mathbf{w}^{(y)} + C \sum_j \xi_j \\ & \mathbf{w}^{(y_j)} \cdot \mathbf{x}_j + b^{(y_j)} \geq \mathbf{w}^{(y')} \cdot \mathbf{x}_j + b^{(y')} + 1 - \xi_j, \quad \forall y' \neq y_j, \quad \forall j \\ & \xi_j \geq 0, \quad \forall j \end{aligned}$$



©2005-2007 Carlos Guestrin

18

What you need to know

- Maximizing margin
- Derivation of SVM formulation
- Slack variables and hinge loss
- Relationship between SVMs and logistic regression
 - 0/1 loss
 - Hinge loss
 - Log loss
- Tackling multiple class
 - One against All
 - Multiclass SVMs

©2005-2007 Carlos Guestrin

19

SVMs, Duality and the Kernel Trick

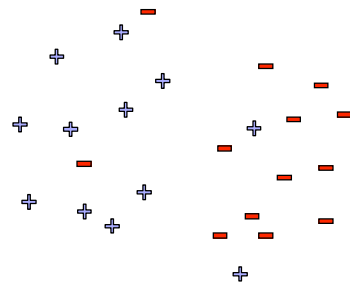
Machine Learning – 10701/15781
Carlos Guestrin
Carnegie Mellon University

October 17th, 2007

©2005-2007 Carlos Guestrin

20

SVMs reminder



$$\begin{aligned} \text{minimize}_{\mathbf{w}, b} \quad & \mathbf{w} \cdot \mathbf{w} + C \sum_j \xi_j \\ & (\mathbf{w} \cdot \mathbf{x}_j + b) y_j \geq 1 - \xi_j, \quad \forall j \\ & \xi_j \geq 0, \quad \forall j \end{aligned}$$

©2005-2007 Carlos Guestrin

21

Today's lecture



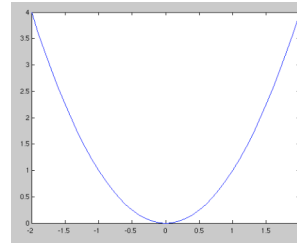
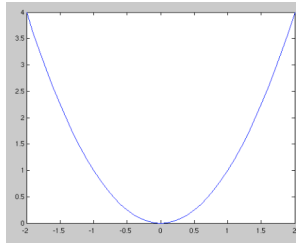
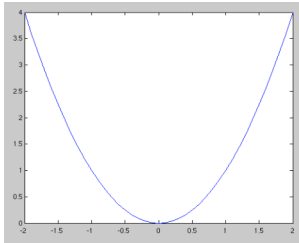
- Learn one of the most interesting and exciting recent advancements in machine learning
 - The “kernel trick”
 - High dimensional feature spaces at no extra cost!
- But first, a detour
 - Constrained optimization!

©2005-2007 Carlos Guestrin

22

Constrained optimization

$$\begin{aligned} \min_x \quad & x^2 \\ \text{s.t.} \quad & x \geq b \end{aligned}$$



©2005-2007 Carlos Guestrin

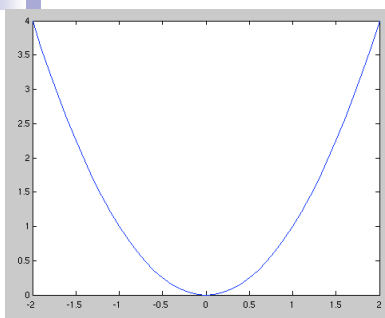
23

Lagrange multipliers – Dual variables

$$\begin{aligned} \min_x \quad & x^2 \\ \text{s.t.} \quad & x \geq b \end{aligned}$$

Moving the constraint to objective function
Lagrangian:

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



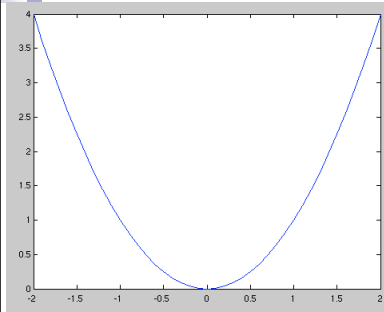
Solve:

$$\begin{aligned} \min_x \max_{\alpha} \quad & L(x, \alpha) \\ \text{s.t.} \quad & \alpha \geq 0 \end{aligned}$$

©2005-2007 Carlos Guestrin

24

Lagrange multipliers – Dual variables



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

©2005-2007 Carlos Guestrin

25

Dual SVM derivation (1) – the linearly separable case

$$\begin{aligned} &\text{minimize}_{\mathbf{w}, b} \quad \frac{1}{2} \mathbf{w} \cdot \mathbf{w} \\ &(\mathbf{w} \cdot \mathbf{x}_j + b) y_j \geq 1, \quad \forall j \end{aligned}$$

©2005-2007 Carlos Guestrin

26

Dual SVM derivation (2) – the linearly separable case

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

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

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

$$\text{minimize}_{\mathbf{w}} \frac{1}{2} \mathbf{w} \cdot \mathbf{w}$$

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

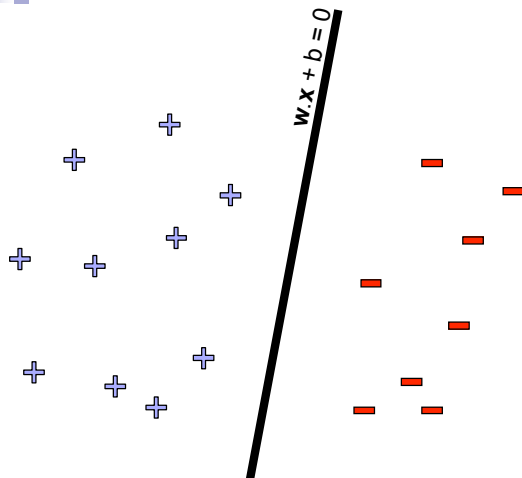
$$b = y_k - \mathbf{w} \cdot \mathbf{x}_k$$

for any k where $\alpha_k > 0$

©2005-2007 Carlos Guestrin

27

Dual SVM interpretation



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

©2005-2007 Carlos Guestrin

28

Dual SVM formulation – the linearly separable case

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

$$\sum_i \alpha_i y_i = 0$$

$$\alpha_i \geq 0$$

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

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

for any k where $\alpha_k > 0$

©2005-2007 Carlos Guestrin

29

Dual SVM derivation – the non-separable case

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

©2005-2007 Carlos Guestrin

30

Dual SVM formulation – the non-separable case

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

$$\sum_i \alpha_i y_i = 0$$

$$C \geq \alpha_i \geq 0$$

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

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

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

©2005-2007 Carlos Guestrin

31

Why did we learn about the dual SVM?

- There are some quadratic programming algorithms that can solve the dual faster than the primal
- But, more importantly, the “**kernel trick**”!!!
 - Another little detour...

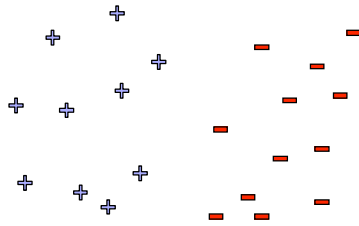
©2005-2007 Carlos Guestrin

32

Reminder from last time: What if the data is not linearly separable?

Use features of features of features of features....

$$\Phi(\mathbf{x}) : \mathbb{R}^m \mapsto F$$



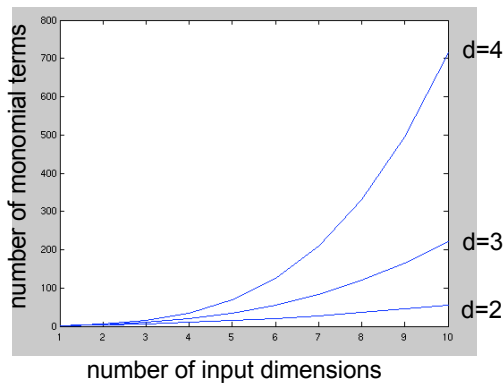
Feature space can get really large really quickly!

©2005-2007 Carlos Guestrin

Higher order polynomials

$$\text{num. terms} = \binom{d + m - 1}{d} = \frac{(d + m - 1)!}{d!(m - 1)!}$$

m – input features
d – degree of polynomial



grows fast!
d = 6, m = 100
about 1.6 billion terms

©2005-2007 Carlos Guestrin

34

Dual formulation only depends on dot-products, not on \mathbf{w} !

$$\begin{aligned} \text{maximize}_{\alpha} \quad & \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ & \sum_i \alpha_i y_i = 0 \\ & C \geq \alpha_i \geq 0 \end{aligned}$$

$$\begin{aligned} \text{maximize}_{\alpha} \quad & \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ & K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \\ & \sum_i \alpha_i y_i = 0 \\ & C \geq \alpha_i \geq 0 \end{aligned}$$

©2005-2007 Carlos Guestrin

35

Dot-product of polynomials

$$\Phi(\mathbf{u}) \cdot \Phi(\mathbf{v}) = \text{polynomials of degree } d$$

©2005-2007 Carlos Guestrin

36

Finally: the “kernel trick”!

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

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$$

$$\sum_i \alpha_i y_i = 0$$

$$C \geq \alpha_i \geq 0$$

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

$$b = y_k - \mathbf{w} \cdot \Phi(\mathbf{x}_k)$$

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

- Never represent features explicitly
 - Compute dot products in closed form
- Constant-time high-dimensional dot-products for many classes of features
- Very interesting theory – Reproducing Kernel Hilbert Spaces
 - Not covered in detail in 10701/15781, more in 10702

©2005-2007 Carlos Guestrin

37

Polynomial kernels

- All monomials of degree d in $O(d)$ operations:
 $\Phi(\mathbf{u}) \cdot \Phi(\mathbf{v}) = (\mathbf{u} \cdot \mathbf{v})^d = \text{polynomials of degree } d$
- How about all monomials of degree up to d ?
 - Solution 0:
 - Better solution:

©2005-2007 Carlos Guestrin

38

Common kernels

- Polynomials of degree d

$$K(\mathbf{u}, \mathbf{v}) = (\mathbf{u} \cdot \mathbf{v})^d$$

- Polynomials of degree up to d

- Gaussian: $K(\mathbf{u}, \mathbf{v}) = (\mathbf{u} \cdot \mathbf{v} + 1)^d$

- RBF: $K(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{\|\mathbf{u} - \mathbf{v}\|^2}{2\sigma^2}\right)$

$$K(\mathbf{u}, \mathbf{v}) = \tanh(\eta \mathbf{u} \cdot \mathbf{v} + \nu)$$

©2005-2007 Carlos Guestrin

39

Overfitting?

- Huge feature space with kernels, what about overfitting???
- Maximizing margin leads to sparse set of support vectors
- Some interesting theory says that SVMs search for simple hypothesis with large margin
- Often robust to overfitting

©2005-2007 Carlos Guestrin

40

What about at classification time

- For a new input \mathbf{x} , if we need to represent $\Phi(\mathbf{x})$, we are in trouble!
- Recall classifier: $\text{sign}(\mathbf{w} \cdot \Phi(\mathbf{x}) + b)$
- Using kernels we are cool!

$$K(\mathbf{u}, \mathbf{v}) = \Phi(\mathbf{u}) \cdot \Phi(\mathbf{v})$$

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

$$b = y_k - \mathbf{w} \cdot \Phi(\mathbf{x}_k)$$

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

©2005-2007 Carlos Guestrin

41

SVMs with kernels

- Choose a set of features and kernel function
- Solve dual problem to obtain support vectors α_i
- At classification time, compute:

$$\mathbf{w} \cdot \Phi(\mathbf{x}) = \sum_i \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i)$$

$$b = y_k - \sum_i \alpha_i y_i K(\mathbf{x}_k, \mathbf{x}_i)$$

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

Classify as

$$\text{sign}(\mathbf{w} \cdot \Phi(\mathbf{x}) + b)$$

©2005-2007 Carlos Guestrin

42

What's the difference between SVMs and Logistic Regression?

	SVMs	Logistic Regression
Loss function		
High dimensional features with kernels		

©2005-2007 Carlos Guestrin

43

Kernels in logistic regression

$$P(Y = 1 | x, \mathbf{w}) = \frac{1}{1 + e^{-(\mathbf{w} \cdot \Phi(\mathbf{x}) + b)}}$$

- Define weights in terms of support vectors:

$$\mathbf{w} = \sum_i \alpha_i \Phi(\mathbf{x}_i)$$

$$\begin{aligned} P(Y = 1 | x, \mathbf{w}) &= \frac{1}{1 + e^{-(\sum_i \alpha_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}) + b)}} \\ &= \frac{1}{1 + e^{-(\sum_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b)}} \end{aligned}$$

- Derive simple gradient descent rule on α_i

©2005-2007 Carlos Guestrin

44

What's the difference between SVMs and Logistic Regression? (Revisited)

	SVMs	Logistic Regression
Loss function	Hinge loss	Log-loss
High dimensional features with kernels	Yes!	Yes!

©2005-2007 Carlos Guestrin

45

What you need to know

- Dual SVM formulation
 - How it's derived
- The kernel trick
- Derive polynomial kernel
- Common kernels
- Kernelized logistic regression
- Differences between SVMs and logistic regression

©2005-2007 Carlos Guestrin

46