Soft-margin SVM

Solves 2 problems:

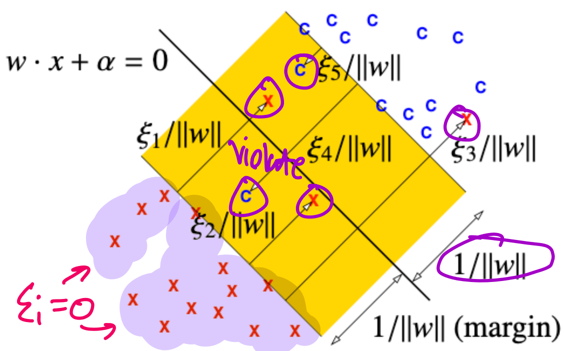
① hard-margin SVMs need linearly separable data

② hard-margin SVMs are sensitive to outliers

idea: let some points violate margin ← "slack variables"

modified constraint: $y_i(X_i \cdot w + \alpha) \geq 1 - \xi_i$ nonzero ξ_i iff X_i violates the margin

$\xi_i \geq 0$



Same as hard-margin SVM

same margin length

$C \to \infty$, slacks $\to 0$
↳ hard-margin SVM

Redefined margin: $\frac{1}{\|\vec{w}\|}$ (consistent w/ hard-margin SVMs)

To prevent abuse of slacks, add a loss term to objective function

new problem: find $\min_{\vec{w}, \alpha, \xi_i} \|\vec{w}\|^2 + C\sum_{i=1}^{n} \xi_i$ ← form of regularization

if $C < 0$, rewarding slack

s.t. $y_i(X_i \cdot w + \alpha) \geq 1 - \xi_i$, $\xi_i \geq 0$, $C > 0$

quadratic program in $d+n+1$ dimensions, $2n$ constraints

w, ξ, α

1 constraint for each sample point

↳ quadratic objective function

↳ linear inequality constraint

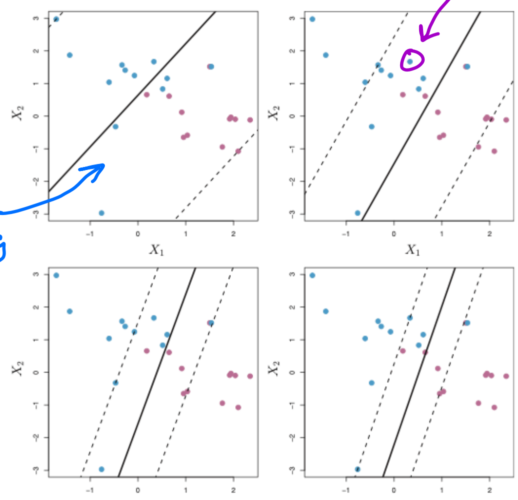
	Small C	Big C
desire	– maximizing margin $1/\|\vec{w}\|$	– want to keep slack terms zero or small
danger	– underfitting (misclassified lots of training data)	– overfitting (careful test/validation)
outliers	– less sensitive to outliers	– more sensitive to outliers
boundaries	– flatter decision boundary	– more sinusoidal/curvy decision boundaries

how do you pick C?

– cross validation

Small C

Classified correctly, still violating margin



almost all points violating margin

big C

$C \to \infty$ then soft-margin becomes hard margin

---

features

how do we do nonlinear decision boundaries?

$\to$ make nonlinear features that lift points to a higher-dimension

high-d linear classifier $\longrightarrow$ low-d nonlinear classifier

example 1: parabolic lifting map

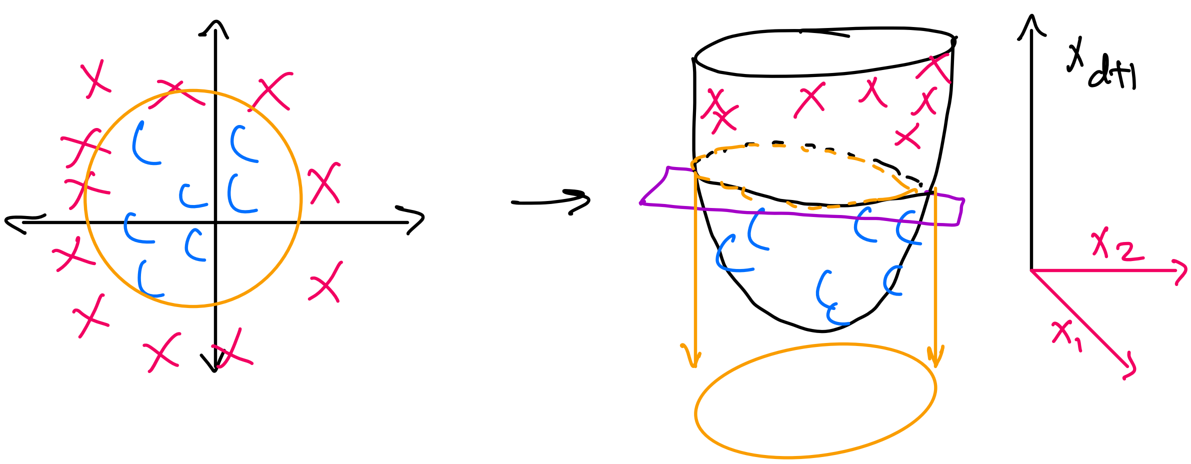
$$\Phi : \mathbb{R}^d \to \mathbb{R}^{d+1}$$

"lifts" $\vec{x}$ onto paraboloid

$$x_{d+1} = \|\vec{x}\|^2$$

$$\Phi(x) = \underbrace{\begin{bmatrix} \vec{x} \\ \|x\|^2 \end{bmatrix}}_{d+1 \text{ components}}$$

[If linear classifier in Φ-space, it induces a spherical classifier in x-space]



theorem: lifted points $\Phi(x_1) \dots \Phi(x_n)$ are linearly separable in Φ-space iff $x_1 \dots x_n$ are linearly separable by a hyper-sphere in the x-space

(sphere might have ∞ radius $\rightarrow$ plane-like)

Proof: Consider h-plane in $\mathbb{R}^d$ w/ center c & radius ρ

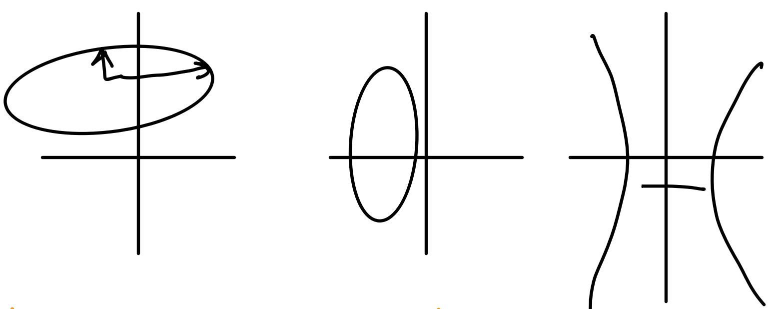
Points inside: $\|x-c\|^2 < \rho^2$

$$\Rightarrow \|x\|^2 - 2c \cdot x + \|c\|^2 \le \rho^2 \quad \Phi(x) \text{ (lifted vector)}$$

Vectorize w/ lifted points

$$\Longrightarrow \underbrace{\underbrace{\begin{bmatrix} -2c^T & 1 \end{bmatrix}}_{\text{normal vector in lifted space}} \begin{bmatrix} x \\ \|x\|^2 \end{bmatrix} \Big\} d+1}_{\text{point is on same side of hyperplane in lifted space}} < \underbrace{\rho^2 - \|c\|^2}_{\text{point inside sphere}}$$

example of axis-aligned ellipsoid/hyperboloid decision boundaries (general quadratic features)



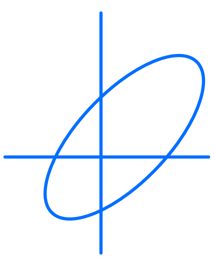
In 3d, these shapes correspond to:

general formula: $Ax_1^2 + Bx_2^2 + Cx_3^2 + Dx_1 + Ex_2 + Fx_3 + \alpha = 0$

$\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^{d+1}$

$$\Phi(x) = \begin{bmatrix} x_1^2 \\ \vdots \\ x_d^2 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} \quad \rightarrow \text{hyperplane} = \overbrace{\begin{bmatrix} A & B & C & D & E & F \end{bmatrix}}^{\vec{w}^T} \Phi(x) + \alpha = 0$$

example 3: ellipsoid/hyperboloid (not axis-aligned)



(order 2 polynomial decision fns)

3D formula: $Ax_1^2 + Bx_2^2 + Cx_3^2 + Dx_1x_2 + Ex_2x_3 + Fx_3x_1 + Gx_1 + Hx_2 + Ix_3 + \alpha = 0$

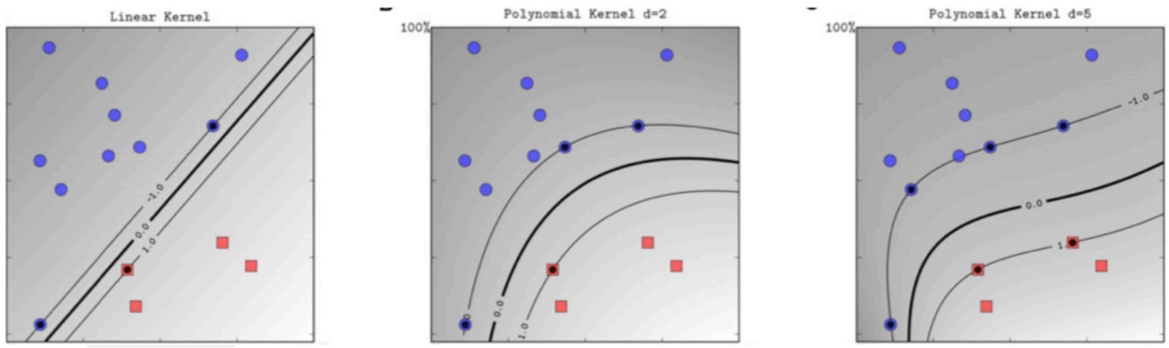
$\rightarrow \Phi(x): \mathbb{R}^d \rightarrow \mathbb{R}^{(d^2+3d)/2}$

quadric: Isosurface defined by this equation example 4: decision function is degree-p polynomial

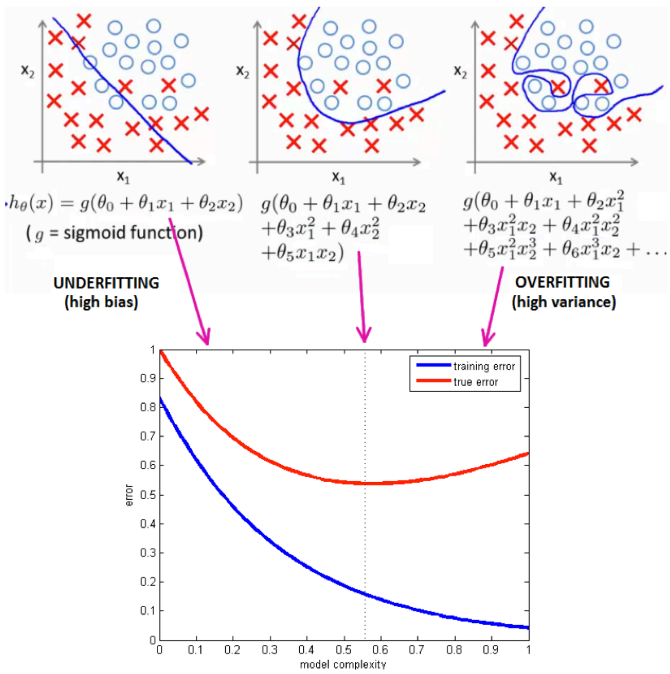
e.g. cubic in $\mathbb{R}^2$

$$\Phi(x) = [x_1^3 \quad x_1^2 x_2 \quad x_1 x_2^2 \quad x_2^3 \quad x_2^2 \quad x_1^2 \quad x_1 x_2 \quad x_2^2 x_1 \quad x_1 \quad x_2]^T$$

$$\Phi(x): \mathbb{R}^d \to \mathbb{R}^{O(d^p)}$$



p=1 p=2



example: edge detector

edge detector: algo for approximating grayscale/color gradients in images

image derivatives:
- tap filter
- Sobel filter
- oriented Gaussian derivative filters

paper: Maji Malik 2009

collect line orientations in local histograms (each having 12 orientation bins per region), use histograms as features instead of raw pixels