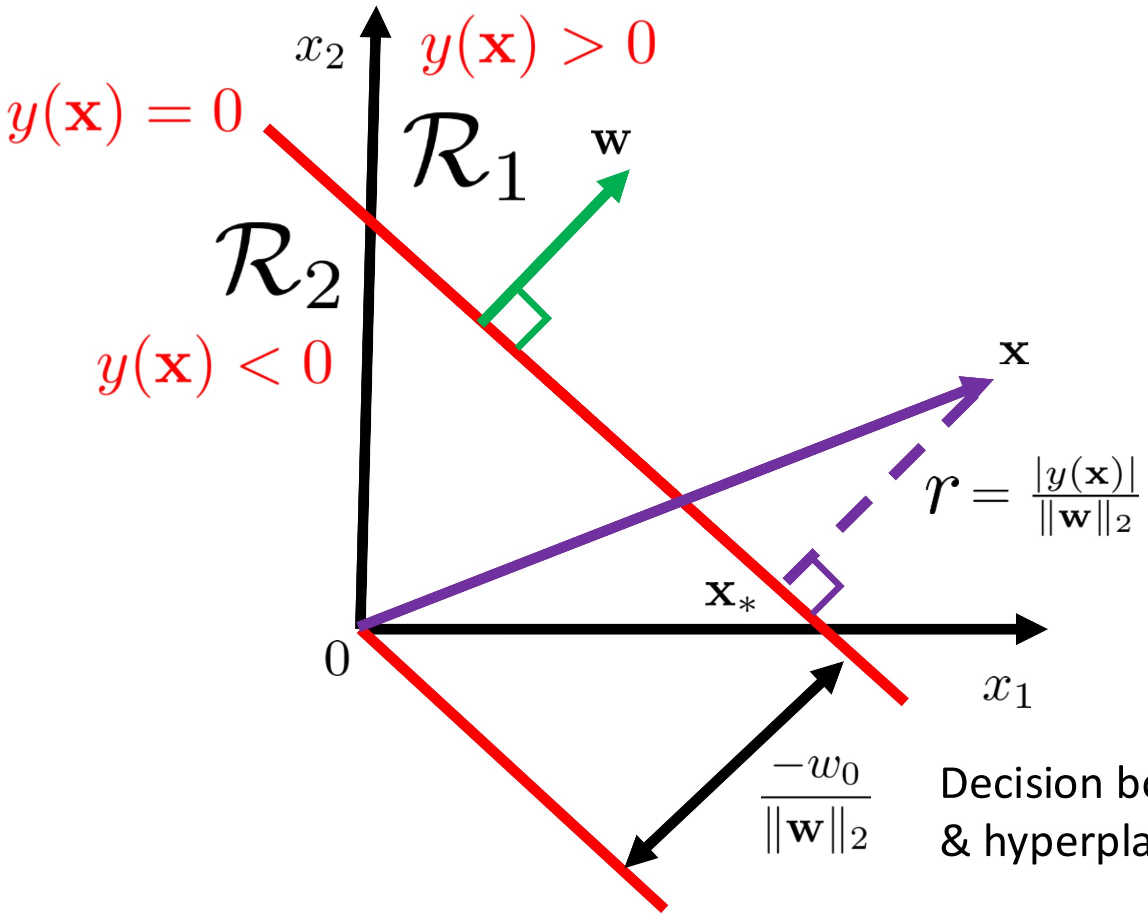
Support Vector Machine & Kernels

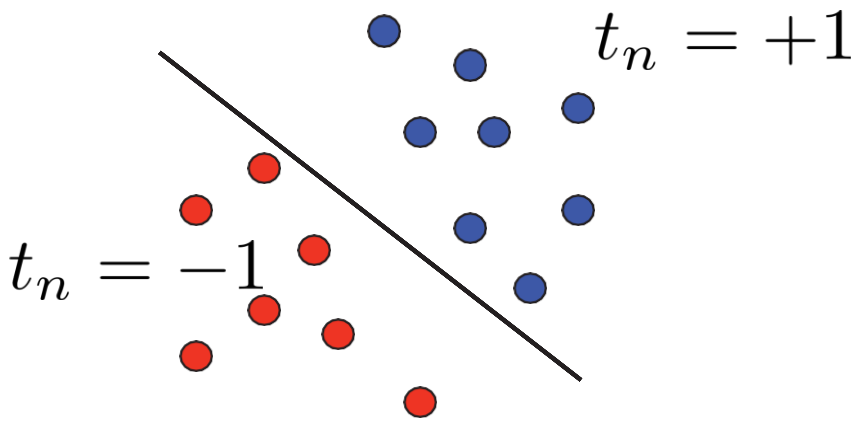
Recap

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$$



Decision boundary
& hyperplane

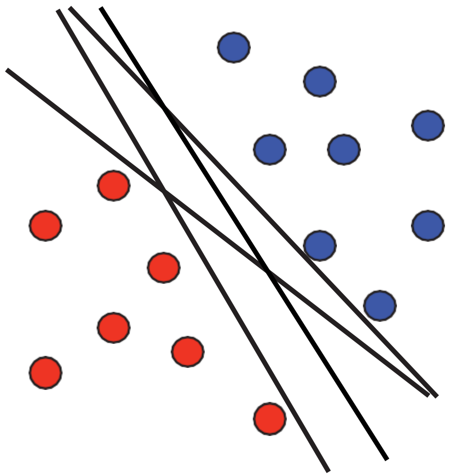
$$\mathbf{w}^T\phi(\mathbf{x}_n) \cdot t_n > 0$$

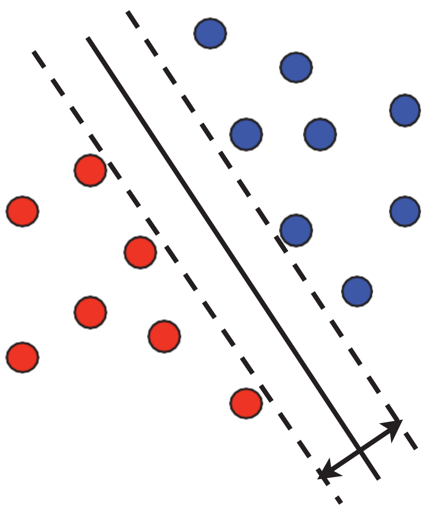


Linearly separable

Linearly Separable & Margin

Perceptron is guaranteed to find some linear separator





Which of these is optimal?

The separator that maximizes the margin

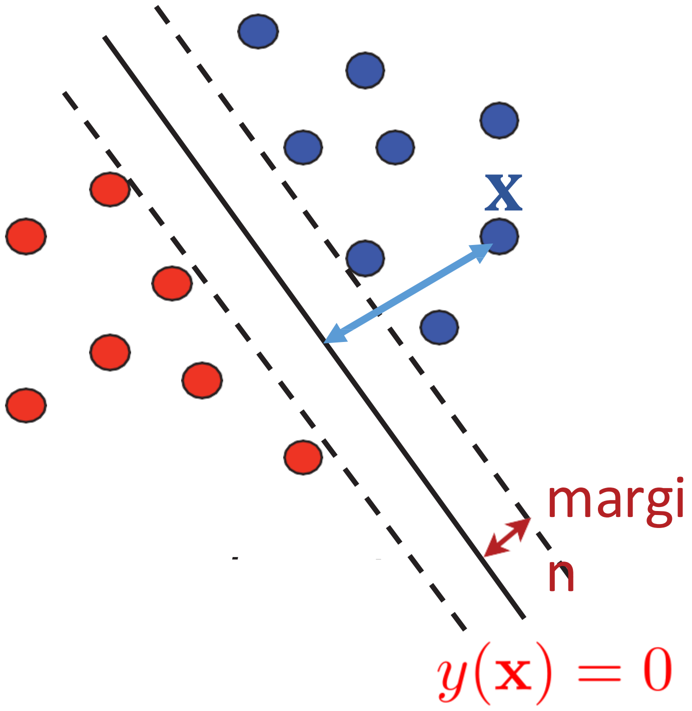
Margin: the smallest distance between the decision boundary and any data point

Hard-Margin SVM

Support Vector Machine (SVM)

Assume data are linearly separable

$$y(\mathbf{x}_n) \cdot t_n > 0$$



The distance between any data point $\mathbf{x}$ and the hyperplane is

$$\frac{|y(\mathbf{x})|}{\|\mathbf{w}\|_2}$$

The margin is the smallest distance

$$\min_n \frac{|y(\mathbf{x}_n)|}{\|\mathbf{w}\|_2}$$

$$= \min_n \frac{t_n \cdot y(\mathbf{x}_n)}{\|\mathbf{w}\|_2}$$

SVM Formulation

margin $= \min_n \frac{t_n \cdot y(\mathbf{x}_n)}{\|\mathbf{w}\|_2}$; we aim to maximize the margin

Support Vector Machine (SVM): $\max_{\mathbf{w}} \min_n \frac{t_n \cdot y(\mathbf{x}_n)}{\|\mathbf{w}\|_2}$

Challenge to solve!

SVM Formulation

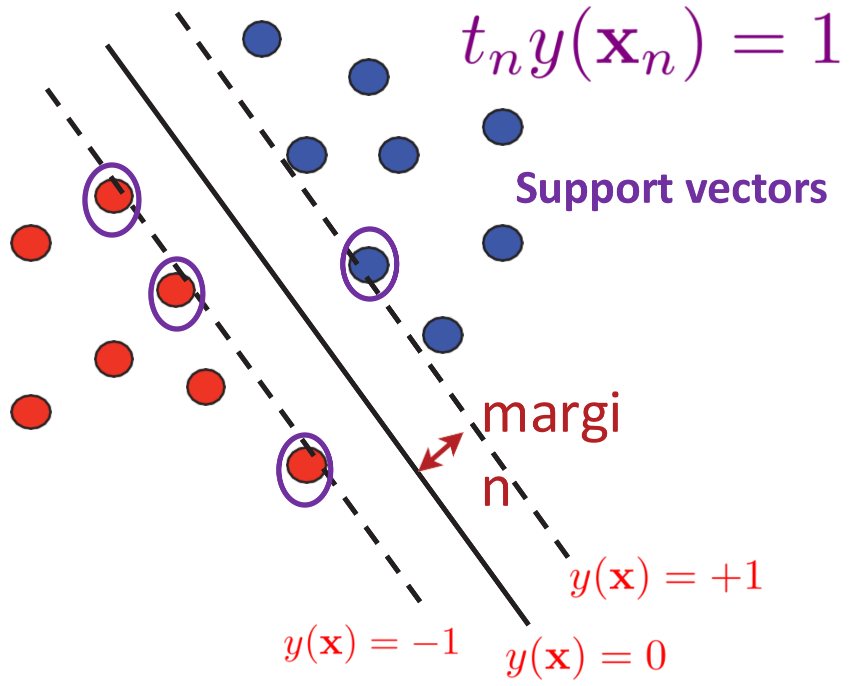
Support Vector Machine (SVM): $\max_{\mathbf{w}} \min_{n} \frac{t_n \cdot y(\mathbf{x}_n)}{\|\mathbf{w}\|_2}$

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \min_{n} \frac{t_n \cdot y(\mathbf{x}_n)}{\|\mathbf{w}\|_2}$$

$$= \arg\max_{\mathbf{w}} \frac{1}{\|\mathbf{w}\|_2} \quad \text{s.t.} \quad \min_{n} t_n y(\mathbf{x}_n) = 1$$

$$= \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 \quad \text{s.t.} \quad \min_{n} t_n y(\mathbf{x}_n) = 1$$

$$= \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 \quad \text{s.t.} \quad t_n y(\mathbf{x}_n) \geq 1, \forall n$$



Quadratic Programming (QP)

Hard Margin SVM: $$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 \qquad \text{s.t. } t_n y(\mathbf{x}_n) \geq 1, \forall n$$

Quadratic optimization problem subject to linear constraints

A unique minimum

d variables $O(d^3)$

Inefficient for high-dim data

Lagrangian Duality

Hard Margin SVM: $\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 \quad \text{s.t. } t_n y(\mathbf{x}_n) \geq 1, \forall n$

$$t_n y(\mathbf{x}_n) \geq 1 \Longrightarrow 1 - t_n y(\mathbf{x}_n) \leq 0 \qquad y(\mathbf{x}_n) = \mathbf{w}^T \mathbf{x}_n + w_0$$

$$\min_{\mathbf{w}, w_0} \mathcal{L}(\mathbf{w}, w_0; \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|_2^2 + \sum_{n=1}^{N} \overset{\geq 0}{a_n} \cdot (1 - t_n(\mathbf{w}^T \mathbf{x}_n + w_0))$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \Longrightarrow \mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n$$

w is a linear combination of the training data

$$\frac{\partial \mathcal{L}}{\partial w_0} = 0 \Longrightarrow 0 = \sum_{n=1}^{N} a_n t_n$$

Representer Theorem

Dual Representation (QP Problem)

Primal: Hard Margin SVM: $\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 \quad \text{s.t. } t_n y(\mathbf{x}_n) \geq 1, \forall n$

Dual:

$$\max_{\mathbf{a}} \tilde{\mathcal{L}}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \boxed{\langle \mathbf{x}_n, \mathbf{x}_m \rangle}$$

Inner product

$$\text{s.t. } a_n \geq 0, \forall n$$

$$\sum_{n=1}^{N} a_n t_n = 0$$

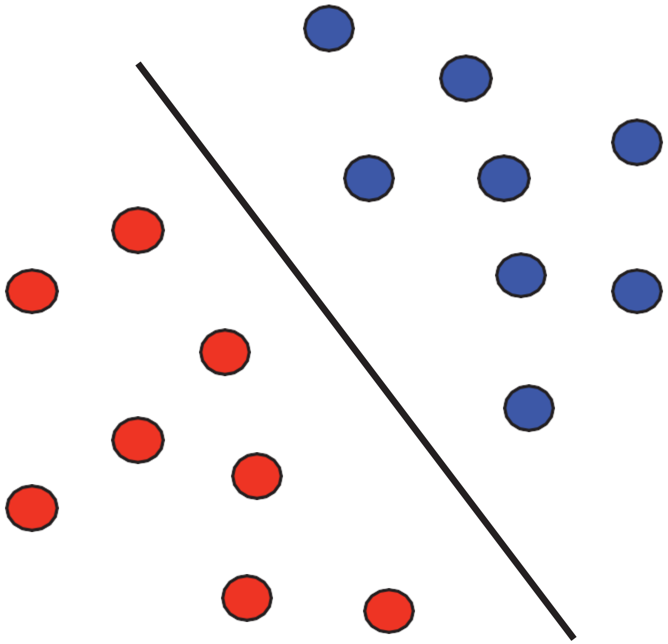
N variables $O(N^3)$

$N \ll d$ Efficient for high-dim data

Only support vectors (which is small) have non-zero a's

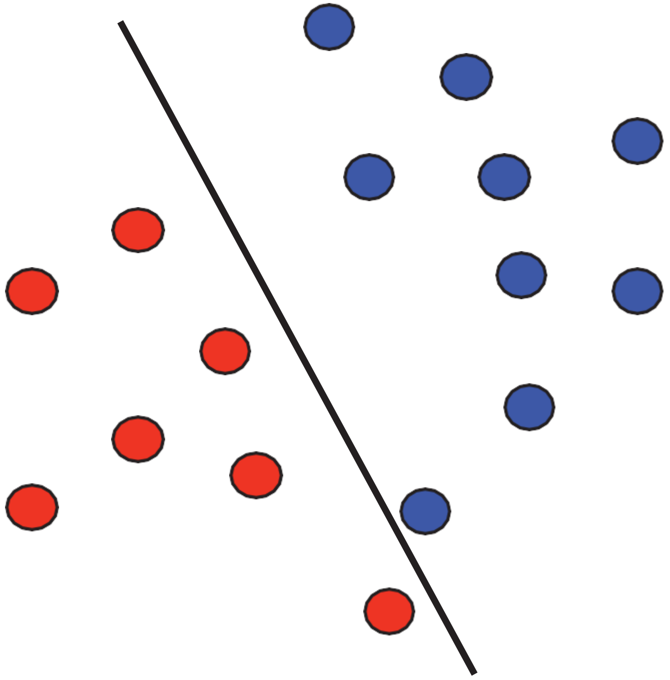
Linearly Separable Again

Data points can be linearly separated



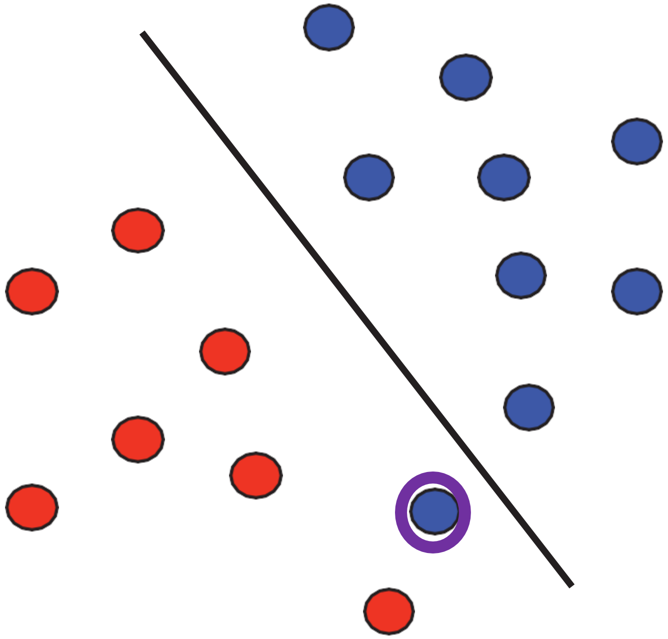
A large margin

Data points can be linearly separated



A very narrow margin

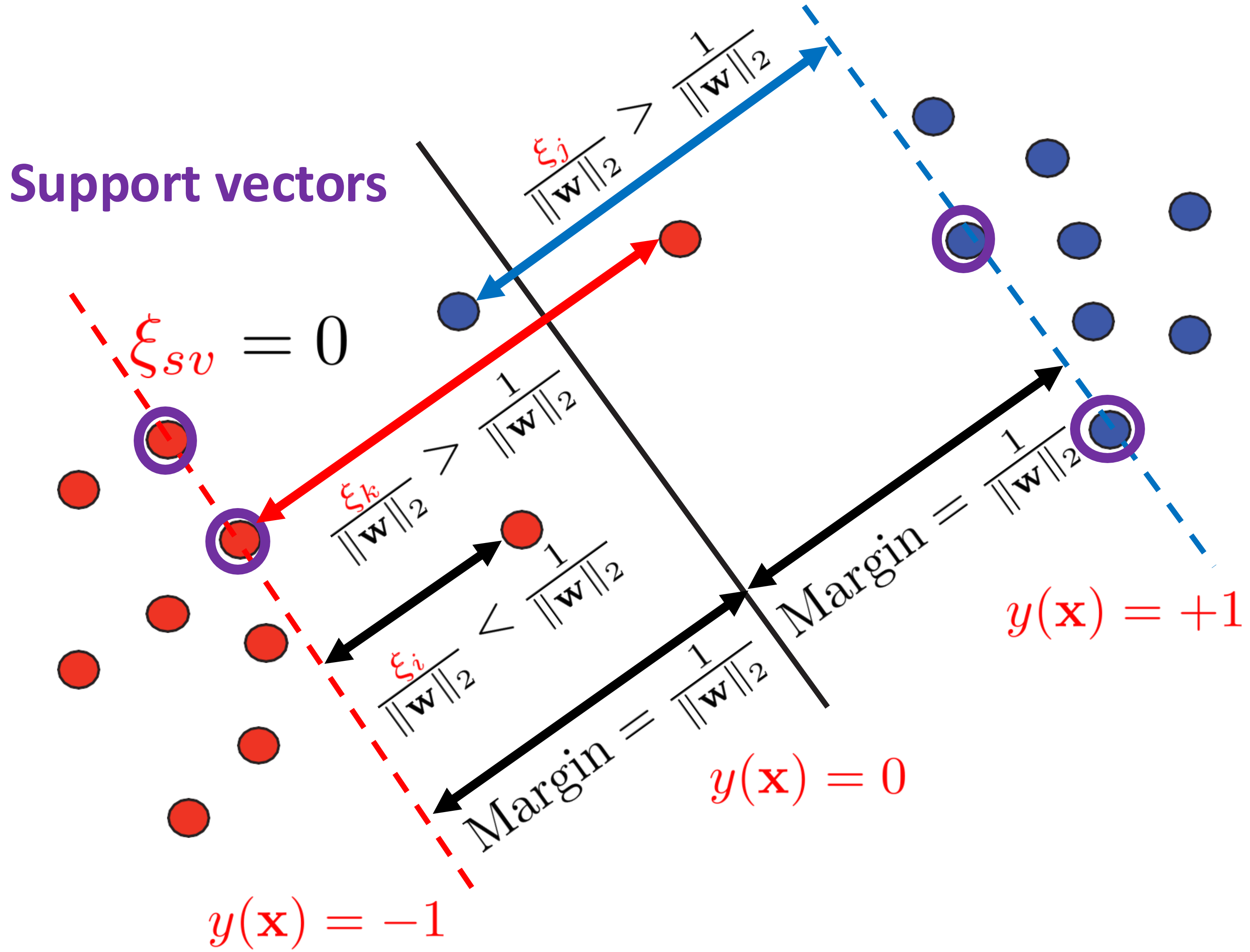
Possibly the large margin solution is better



Even one constraint violated

Soft-Margin SVM

Introduce Slack Variables



Slack variable $\xi_n \geq 0, \forall n$

ξ =0: Support vectors

0 < ξ ≤ 1 points are between margin and **correct** side of boundary, but margin violation

Small penalty

ξ > 1 points are **misclassified**

Large penalty

ξ_n indicates penalty

Soft-Margin SVM: Relaxation

Hard Margin SVM: $\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 \qquad \text{s.t. } t_n y(\mathbf{x}_n) \geq 1, \forall n$

Soft Margin SVM:
$$\min_{\mathbf{w}} \min_{\{\xi_n\}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_n [\xi_n]_+ \qquad [\xi_n]_+ = \max\{\xi_n, 0\}$$
$$\text{s.t. } t_n y(\mathbf{x}_n) \geq 1 - \xi_n, \forall n$$

Large C makes constraints hard to ignore => **narrow** margin

$C = \infty \Longrightarrow \forall \xi_n = 0$ Hard margin SVM

Small C makes allows constraints to be ignored => ***large*** margin

$C = 0 \Longrightarrow \forall \xi_n \geq 0$ Ignore the data distribution!

Equivalent Formulation using Hinge Loss

Soft Margin: $$\min_{\mathbf{w},\{\xi_n\}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_n [\xi_n]_+ \quad \text{s.t. } t_n y(\mathbf{x}_j) \geq 1 - \xi_n, \forall n$$

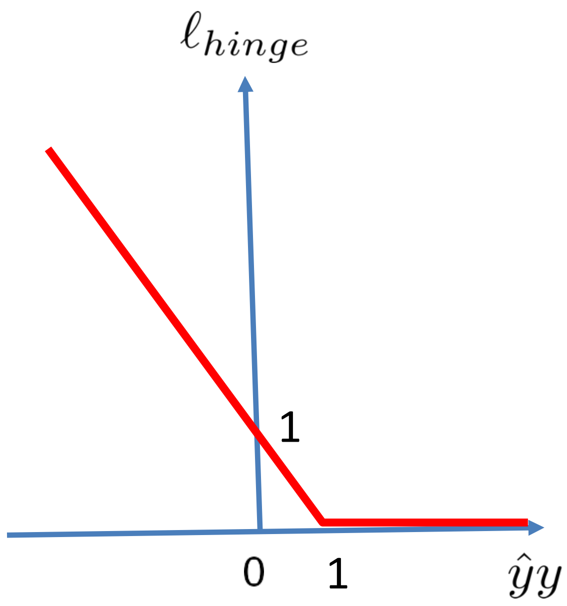
Unconstrained optimization

$$\min_{\mathbf{w}} \ \frac{1}{2}\|\mathbf{w}\|_2^2 \ + \ C\sum_n \boxed{[1 - t_n y(\mathbf{x}_n)]_+}$$

Hinge loss $\quad \ell_{hinge}(y, \hat{y}) = [1 - \hat{y}y]_+$

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_n \ell_{hinge}(y(\mathbf{x}_n), t_n)$$

Regularization **Empirical loss**



Property of Hinge Loss

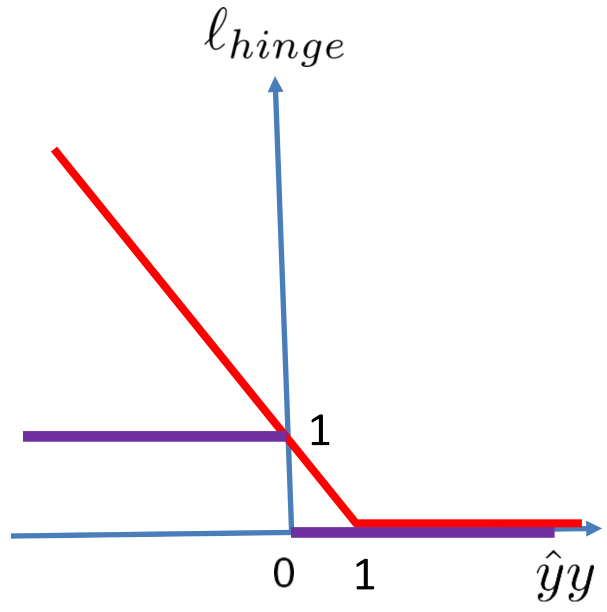
$$\ell_{hinge}(y(\mathbf{x}_n), t_n) = [1 - t_n(\mathbf{w}^T\mathbf{x}_n + w_0)]_+$$

An approximation to the 0-1 loss

$$\ell_{0-1}(x) = \begin{cases} 0, & \text{if } x \geq 0; \\ 1, & \text{if } x < 0. \end{cases}$$

Non-differentiable (subgradient)

$$\frac{\partial \ell_{hinge}(y(\mathbf{x}_n, t_n))}{\partial \mathbf{w}} = \begin{cases} -t_n\mathbf{x}_n, & \text{if } t_n y(\mathbf{x}_n) < 1; \\ 0, & \text{if } t_n y(\mathbf{x}_n) > 1; \\ [0, -t_n\mathbf{x}_n], & \text{if } t_n y(\mathbf{x}_n) = 1. \end{cases}$$



Sub-gradient Descent for Soft Margin SVM

$$\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_n \ell_{hinge}(y(\mathbf{x}_n), t_n)$$

$$= \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_n [1 - t_n(\mathbf{w}^T\mathbf{x}_n + w_0)]_+$$

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta\nabla_{\mathbf{w}_t}\mathcal{L}(\mathbf{w}^{(t)})$$

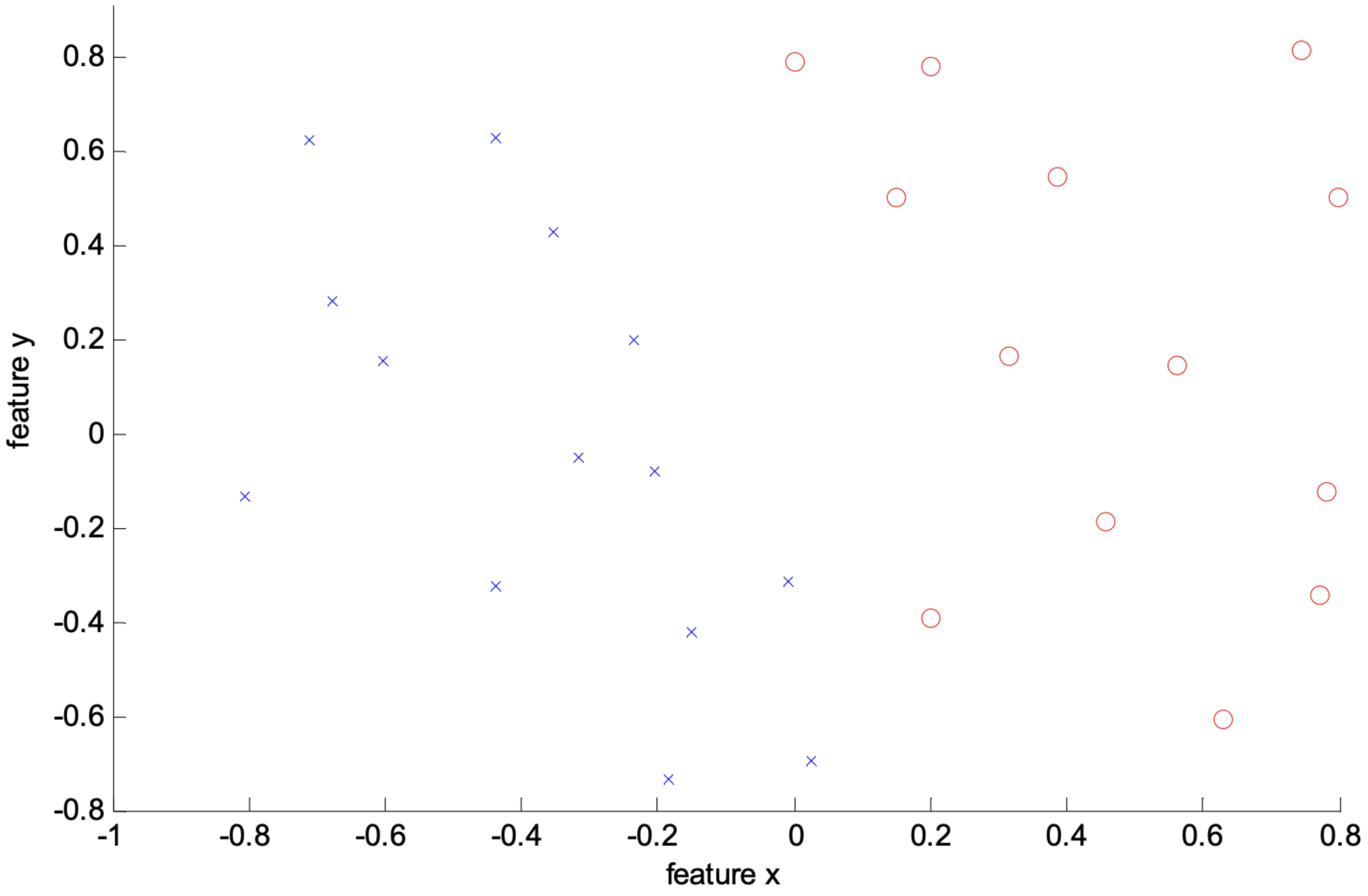
$$= \mathbf{w}^{(t)} - \eta\Big(\mathbf{w}^{(t)} + C\sum_n \frac{\partial \ell_{hinge}}{\partial \mathbf{w}^{(t)}}\Big)$$

Ideally: $t_n y(\mathbf{x}_n) \geq 1, \forall n$

$$= (1-\eta)\mathbf{w}^{(t)} + \begin{cases} C\sum_n t_n\mathbf{x}_n, & \text{if } t_n y(\mathbf{x}_n) < 1; \\ 0, & \text{otherwise} \end{cases}$$

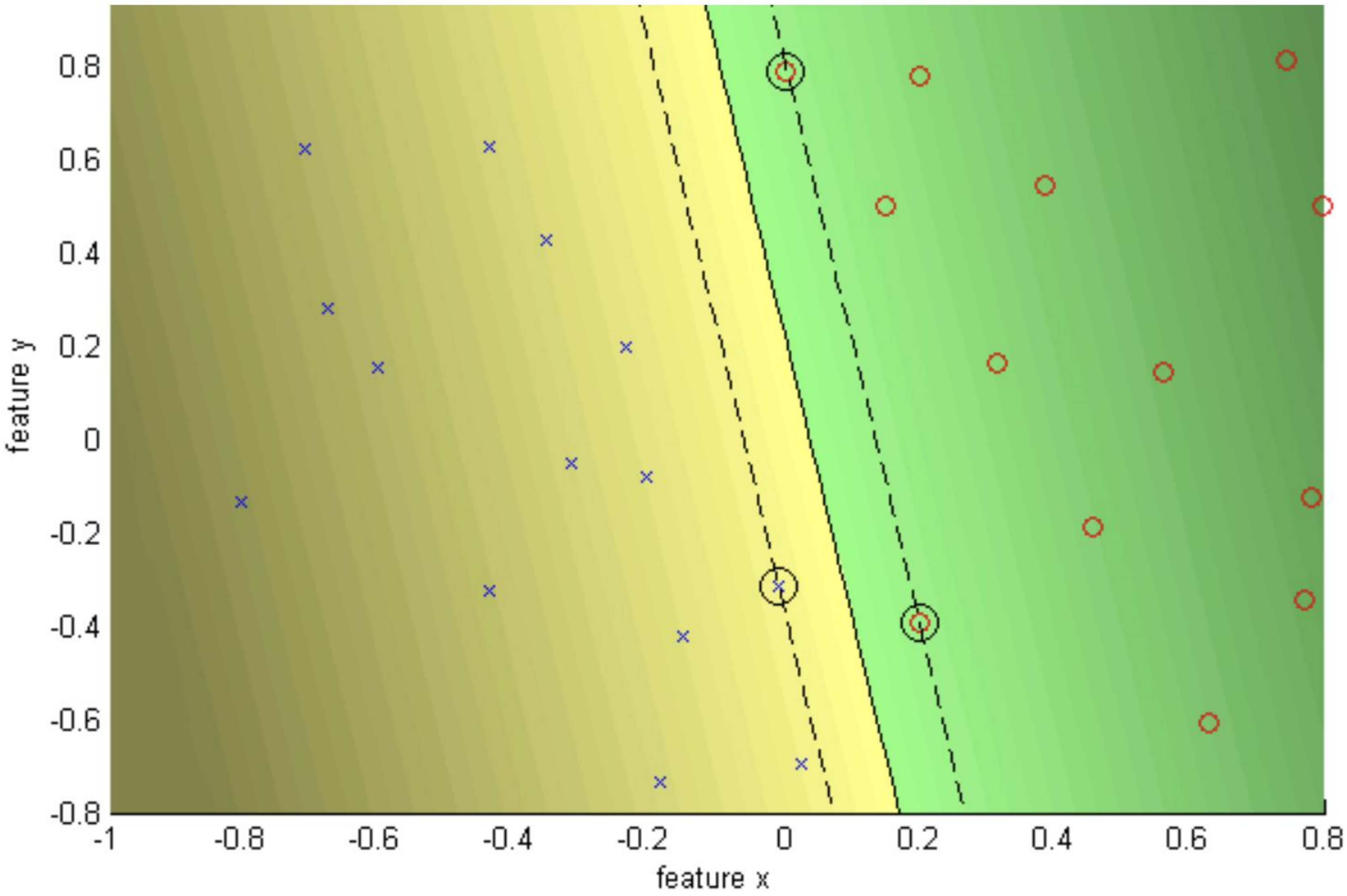
Focus on small margin or misclassified points

Example

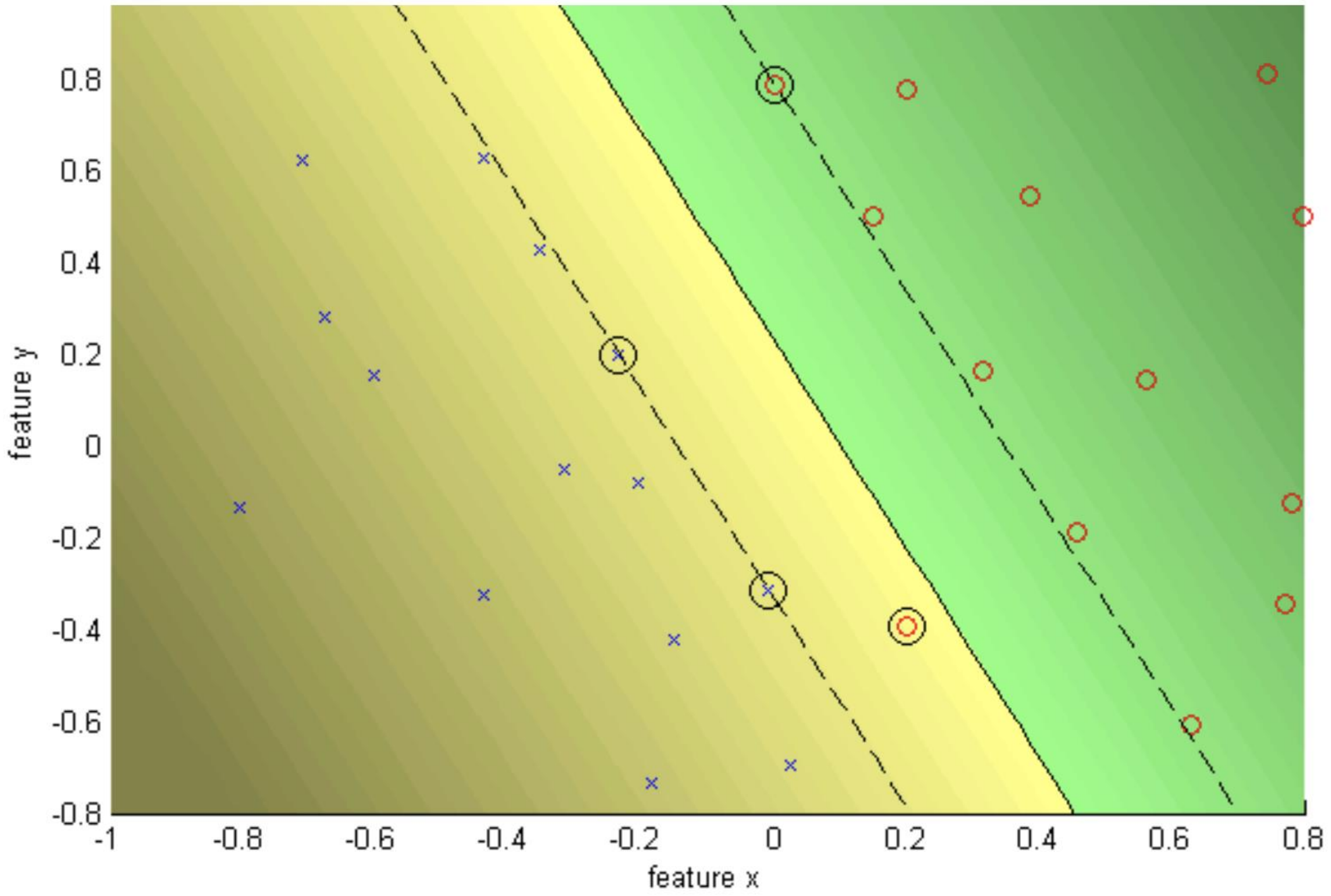


0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
feature y
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
feature x

Hard margin: C = Infinity



Soft margin: C = 10



Dual Representation

Soft Margin: $\min_{\mathbf{w},\{\xi_n\}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_n [\xi_n]_+ \quad \text{s.t. } t_n y(\mathbf{x}_j) \geq 1 - \xi_n, \forall n$

Dual of hard-margin SVM

$$\max_{\mathbf{a}} \tilde{\mathcal{L}}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \langle \mathbf{x}_n, \mathbf{x}_m \rangle$$

$$\text{s.t. } a_n \geq 0, \forall n$$

$$\sum_{n=1}^{N} a_n t_n = 0$$

Dual Representation

Soft Margin: $\min_{\mathbf{w},\{\xi_n\}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_n [\xi_n]_+ \quad \text{s.t. } t_n y(\mathbf{x}_j) \geq 1 - \xi_n, \forall n$

Dual of soft-margin SVM

$$\max_{\mathbf{a}} \tilde{\mathcal{L}}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \langle \mathbf{x}_n, \mathbf{x}_m \rangle$$

$$\text{s.t. } 0 \leq a_n \leq C, \forall n$$

$$\sum_{n=1}^{N} a_n t_n = 0$$

Prime and Dual for Prediction

Primal version of classifier

$$y(\mathbf{x}_t) = \mathbf{w}^T \mathbf{x}_t + w_0$$

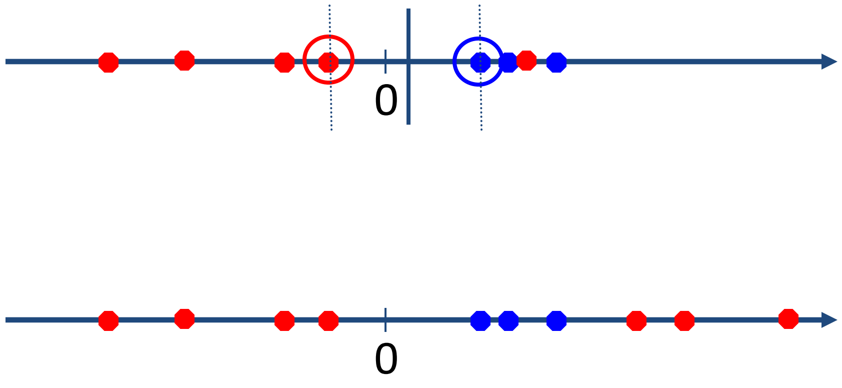
Dual version of classifier

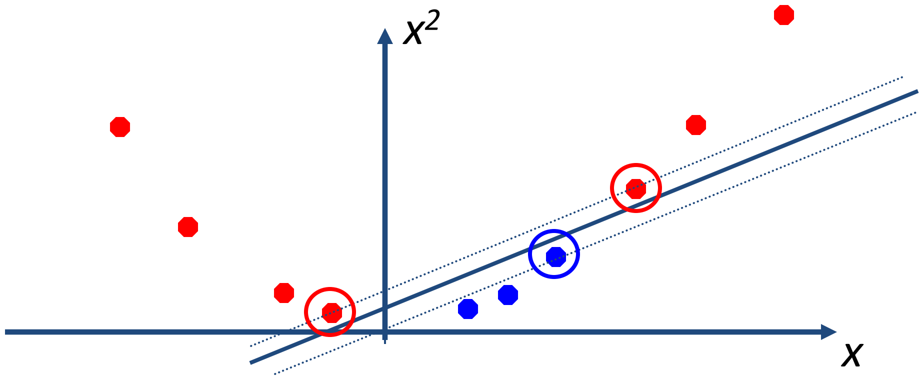
$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n \qquad y(\mathbf{x}_t) = \sum_{n=1}^{N} a_n t_n \boxed{\langle \mathbf{x}_n, \mathbf{x}_t \rangle} + w_0$$

Remember: only support vectors have non-zero a's

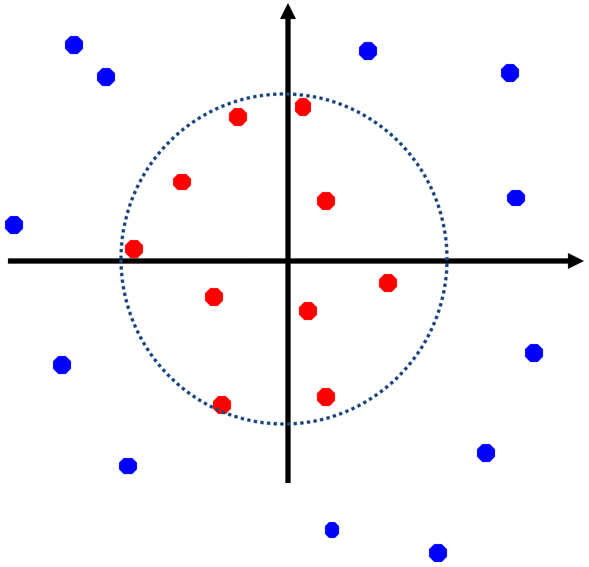
Linear Separators are IMPOSSIBLE

Introduce slack variables



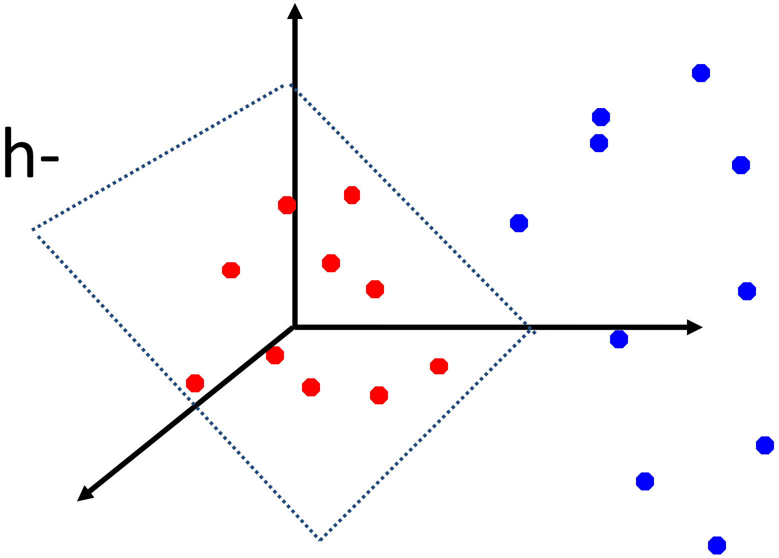


Data is POSSIBLY linearly separable in high-dim space



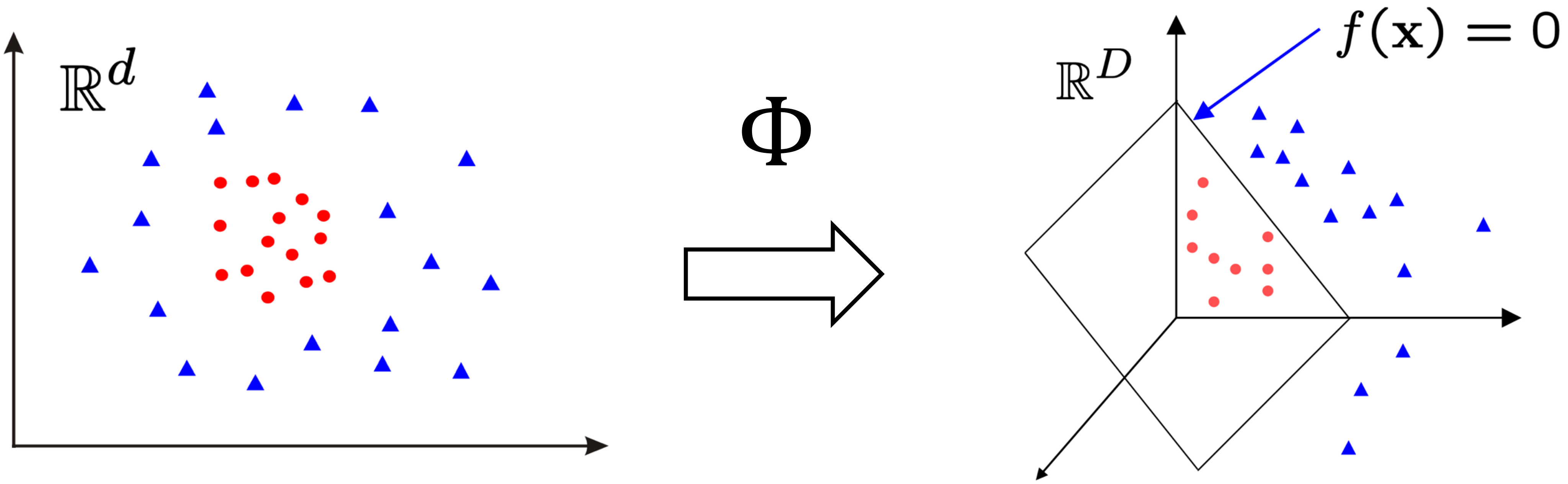
How can we map/handle data into high-dim space?

Kernels



Kernels

Map Low-Dim Data into High-dim Feature Space



Feature map $\Phi : \mathbf{x} \in \mathbb{R}^d \rightarrow \Phi(\mathbf{x}) \in \mathbb{R}^D \quad D > d$

Primal Soft-Margin SVM in High-Dim Space

Learning $$\min_{\mathbf{w}\in\mathbb{R}^D} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_n [1 - t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + w_0)]_+$$

Prediction $$y(\mathbf{x}_t) = \mathbf{w}^T\phi(\mathbf{x}_t) + w_0$$

1. Simply map x to Φ(x) where data is separable

2. Solve for **w** in the high D-dim space

3. Make predictions in the D-dim space

However, if D >> d there are many more parameters to learn for w

In some cases, possibly require infinite dimensional space

Dual Soft-Margin SVM in High-Dim Space

Learning

$$\max_{\mathbf{a}\in\mathbb{R}^N} \tilde{\mathcal{L}}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \langle \phi(\mathbf{x}_n), \phi(\mathbf{x}_m)\rangle \qquad \text{s.t. } a_n \geq 0, \forall n$$

$$= \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \qquad \sum_{n=1}^{N} a_n t_n = 0$$

Prediction

$$y(\mathbf{x}_n) = \sum_{n=1}^{N} a_n t_n \langle \phi(\mathbf{x}_n), \phi(\mathbf{x}_t)\rangle + w_0 = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}_n, \mathbf{x}_t) + w_0$$

1. Φ(x) occurs in **pairs**, i.e., inner product $\langle \Phi(x_i), \Phi(x_j)\rangle$
2. Solve for **a** in the same **N**-dim space
3. Write $\langle \Phi(x_i), \Phi(x_j)\rangle = k(x_i, x_j)$. => this is known as a Kernel

Classifier can be learnt and applied **without explicitly computing Φ(x)**

Only need to define/use a kernel k

Kernel Example

$$\phi : \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2 \to \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{pmatrix} \in \mathbb{R}^3$$

$$k(\mathbf{x}, \mathbf{z}) = ?$$

$$\begin{aligned} \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle &= (x_1^2, x_2^2, \sqrt{2}x_1x_2) \begin{pmatrix} z_1^2 \\ z_2^2 \\ \sqrt{2}z_1z_2 \end{pmatrix} \\ &= (x_1z_1 + x_2z_2)^2 \end{aligned}$$

$$k(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle)^2$$

$$k(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + c)^2$$

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \in \mathbb{R}^3$$

$$\phi(\mathbf{x}) = ?$$

$$\phi(\mathbf{x}) = \begin{pmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \\ \sqrt{2c}x_1 \\ \sqrt{2c}x_2 \\ \sqrt{2c}x_3 \\ c \end{pmatrix}$$

Representative Kernels

- **Linear** kernels $k(x_i, x_j) = \langle x_i, x_j \rangle$
- **Polynomial** kernels $k(x_i, x_j) = \langle 1 + x_i, x_j \rangle^d$ for any *d*>0
  - Contains ***all polynomials*** terms up to degree *d*
- **Gaussian** kernels $k(x_i, x_j) = \exp(-||x_i - x_j||^d / 2\sigma^2)$ for σ>0
  - ***Infinite*** dimensional feature space (Hint: Taylor series expansion)
- **Kernel (Gram) matrix K**
  - ***Semi-definite*** matrix $\mathbf{v}^T \mathbf{K} \mathbf{v} \geq 0, \forall \mathbf{v}$
  - Computed ***once*** and stored ***offline***

$$\mathbf{K} = \begin{bmatrix} k_{1,1} & k_{1,2} & \cdots & k_{1,N} \\ k_{2,1} & k_{2,2} & \cdots & k_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ k_{N,1} & k_{N,2} & \cdots & k_{N,N} \end{bmatrix} \in \mathbb{R}^{N \times N}$$

SVM Classifier with Gaussian Kernel

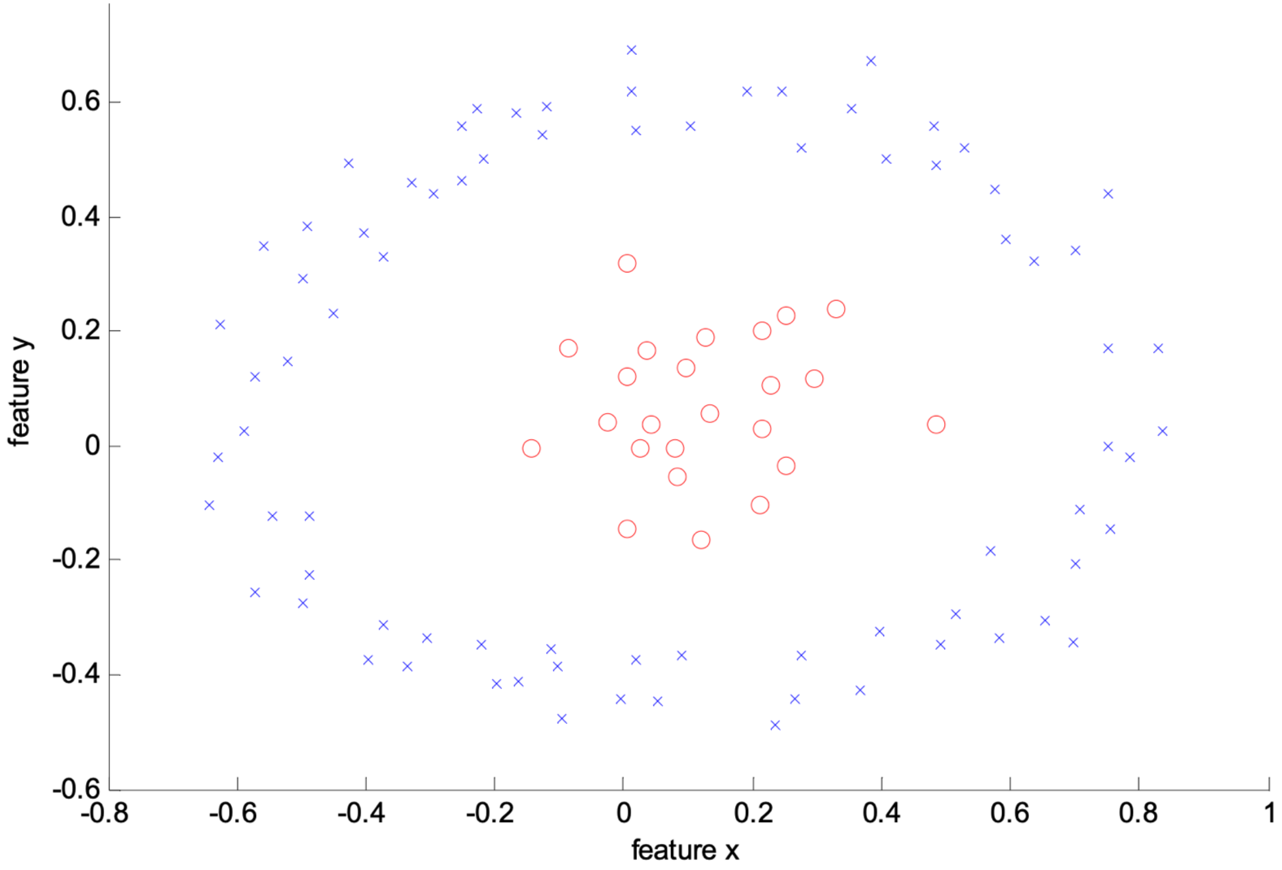
$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}_n, \phi(\mathbf{x}_t)) + w_0$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 / 2\sigma^2\right)$$

Radial Basis Function (RBF) Kernel SVM

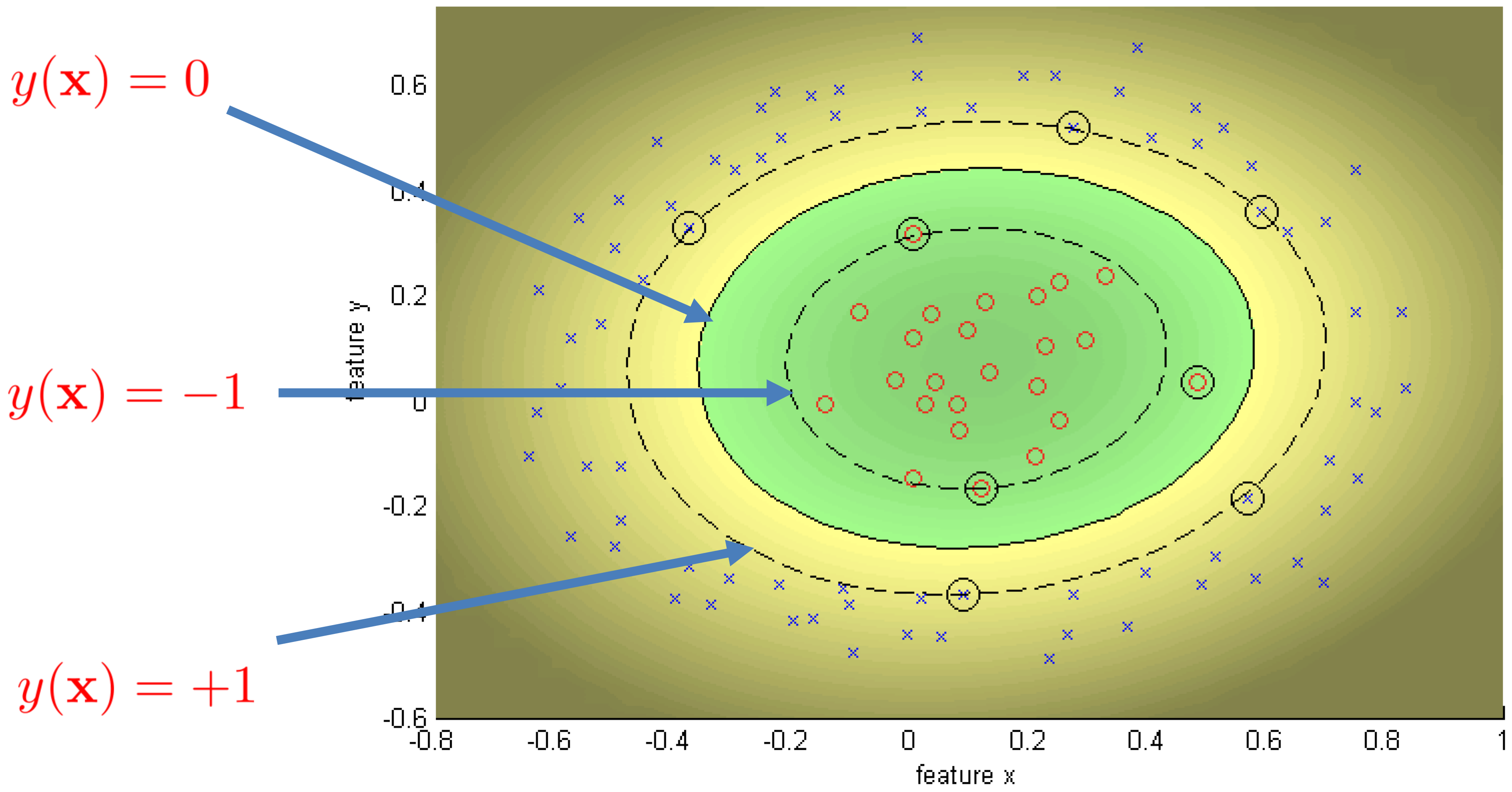
$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n \exp\left(-\|\mathbf{x} - \mathbf{x}_n\|_2^2 / 2\sigma^2\right) + w_0$$

RBF Kernel SVM Example



Data are not linearly separable in original feature space

RBF Kernel SVM Example (C=100, $\sigma = 1.0$)



Data are separable via RBF Kernel

Summary

- Support vector machine (SVM): maximal margin classifier

- Hard-margin SVM
  - Prime: QP problem, solve for **#features** variables
  - Dual: QP problem, solve for **#samples** variables, efficient for high-dim data

- Soft-margin SVM: Handle a few outliers
  - Prime & Dual (can be rewritten using hinge loss)
  - Hinge loss (approximate 0-1 loss; non-differentiable)

- Kernels: Handle non-linearly separable data
  - Map data from the original space to a linearly separable high-dim space
  - Linear kernel; polynomial kernel; Gaussian/RBF kernel
  - Kernel matrix: semi-definite; computed and stored offline

Acknowledgement

Some slides are adapted from **Andrew Zisserman**
https://www.robots.ox.ac.uk/~az/lectures/ml/lect2.pdf
& Shusen Wang
https://github.com/wangshusen/DeepLearning