MACHINE LEARNING FOR SIGNAL PROCESSING

12-2-2025

Sriram Ganapathy
LEAP lab, Electrical Engineering, Indian Institute of Science,
sriramg@iisc.ac.in

Viveka Salinamakki, Varada R.
LEAP lab, Electrical Engineering, Indian Institute of Science

http://leap.ee.iisc.ac.in/sriram/teaching/MLSP25/

STORY SO FAR

EM algorithm

Generative Modeling

Gaussian Modeling

Gaussian Mixture Modeling

Classification Problem

Decision Theory

Function Modeling

Linear Models for Regression and Classification

Kernel Machines & Max-margin classifiers

Gradient Descent Algorithm

Discriminative Modeling

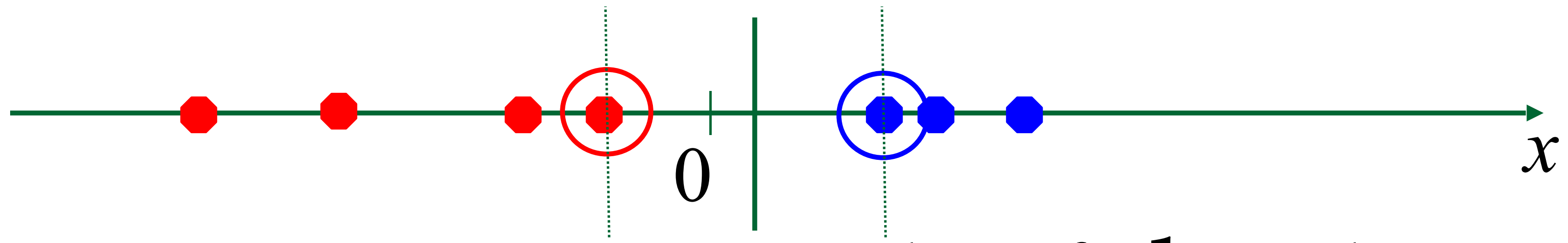
Data Representations PCA, LDA

LINEAR REGRESSION REVISITED

- Primal and dual forms
- Solution in dual space
- Kernels

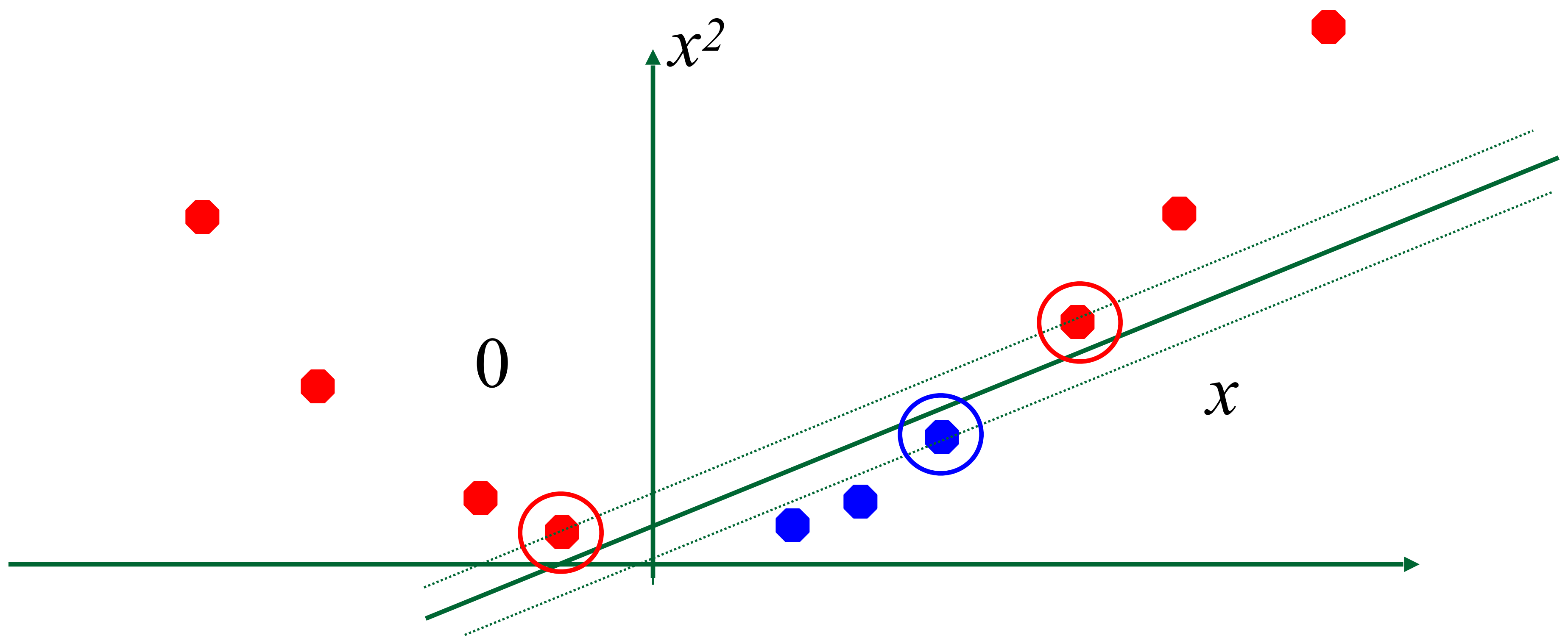
Kernel Machines

- Datasets that are linearly separable with some noise work out great:



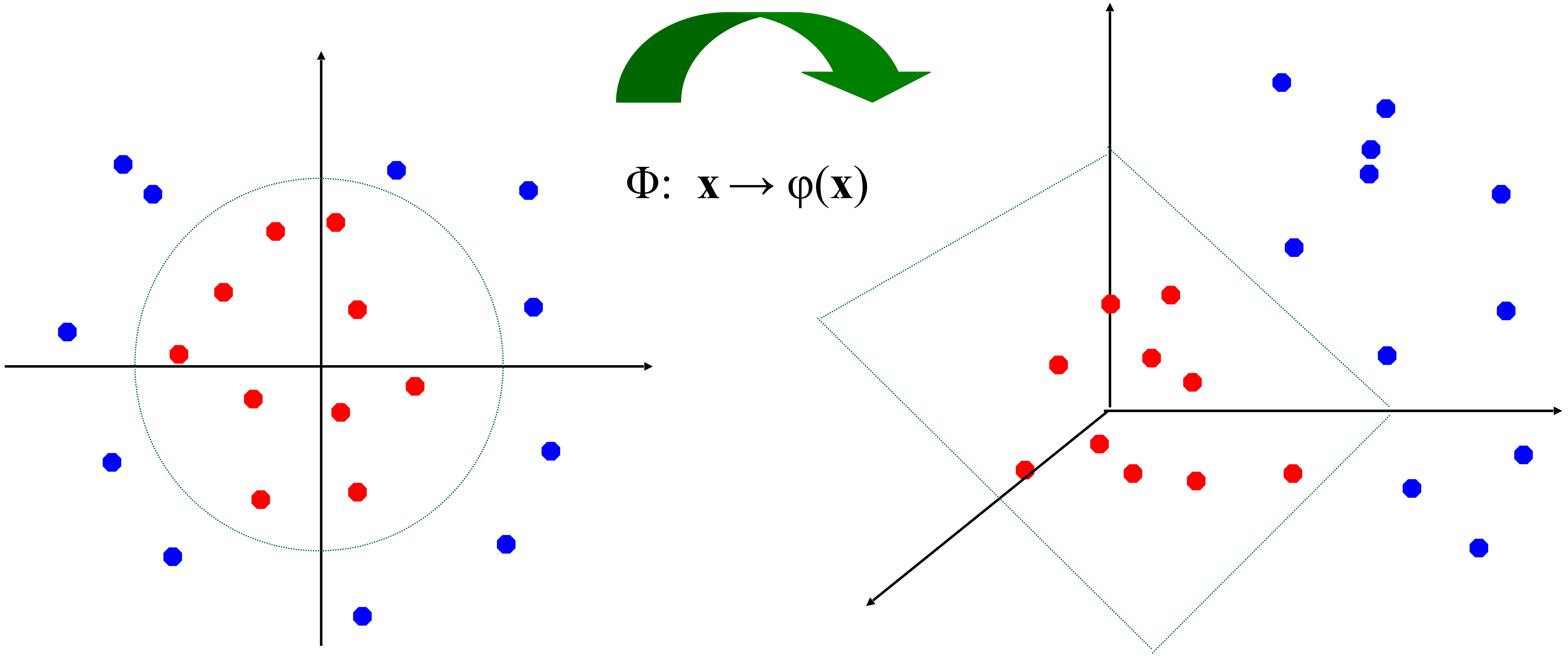
- But what are we going to do if the dataset is just too hard?

- How about… mapping data to a higher-dimensional space:



Kernel Trick

- General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable:



The "Kernel Trick"

- The linear classifier relies on dot product between vectors $k(x_i,x_j)=\mathbf{x}_i^T\mathbf{x}_j$
- If every data point is mapped into high-dimensional space via some transformation Φ: $\mathbf{x} \rightarrow \phi(\mathbf{x})$, the dot product becomes:

$$k(\mathbf{x}_i,\mathbf{x}_j)= \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$$

- A *kernel function* is some function that corresponds to an inner product in some expanded feature space.
- Example:

2-dimensional vectors x=$[x_1 \;\; x_2]$; let $k(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^T\mathbf{x}_j)^2$,

Need to show that $K(\mathbf{x}_i,\mathbf{x}_j)= \boldsymbol{\phi}(\mathbf{x}_i)^T\boldsymbol{\phi}(\mathbf{x}_j)$:

$$k(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^T\mathbf{x}_j)^2,$$

$$= 1+ x_{i1}^2x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2}+ x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$$

$$= [1 \;\; x_{i1}^2 \;\; \sqrt{2}\, x_{i1}x_{i2} \;\; x_{i2}^2 \;\; \sqrt{2}x_{i1} \;\; \sqrt{2}x_{i2}]^T [1 \;\; x_{j1}^2 \;\; \sqrt{2}\, x_{j1}x_{j2} \;\; x_{j2}^2 \;\; \sqrt{2}x_{j1} \;\; \sqrt{2}x_{j2}]$$

$$= \boldsymbol{\phi}(\mathbf{x}_i)^T\boldsymbol{\phi}(\mathbf{x}_j), \quad \text{where } \boldsymbol{\phi}(\mathbf{x}) = [1 \;\; x_1^2 \;\; \sqrt{2}\, x_1x_2 \;\; x_2^2 \;\; \sqrt{2}x_1 \;\; \sqrt{2}x_2]$$

KERNELS

- For many functions $k(\mathbf{x_i},\mathbf{x_j})$ checking that

  $k(\mathbf{x_i},\mathbf{x_j}) = \phi(\mathbf{x_i})^T\phi(\mathbf{x_j})$ can be cumbersome.
- Mercer's theorem: Every semi-positive definite symmetric function is a kernel
  - Semi-positive definite symmetric functions correspond to a semi-positive definite symmetric Gram matrix:

K =

$k(\mathbf{x}_1,\mathbf{x}_1)$	$k(\mathbf{x}_1,\mathbf{x}_2)$	$k(\mathbf{x}_1,\mathbf{x}_3)$	…	$k(\mathbf{x}_1,\mathbf{x}_N)$
$k(\mathbf{x}_2,\mathbf{x}_1)$	$k(\mathbf{x}_2,\mathbf{x}_2)$	$k(\mathbf{x}_2,\mathbf{x}_3)$		$k(\mathbf{x}_2,\mathbf{x}_N)$
…	…	…	…	…
$k(\mathbf{x}_N,\mathbf{x}_1)$	$k(\mathbf{x}_N,\mathbf{x}_2)$	$k(\mathbf{x}_N,\mathbf{x}_3)$	…	$k(\mathbf{x}_N,\mathbf{x}_N)$

EXAMPLES OF KERNEL FUNCTIONS

- Linear: $k(\mathbf{x}_i,\mathbf{x}_j) = \mathbf{x}_i{}^T\mathbf{x}_j$
- Polynomial of power p: $k(\mathbf{x}_i,\mathbf{x}_j) = (1 + \mathbf{x}_i{}^T\mathbf{x}_j)^p$
- Gaussian (radial-basis function network):

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp \frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{\sigma^2}$$

- Sigmoid: $k(\mathbf{x}_i,\mathbf{x}_j) = \tanh(\beta_0\mathbf{x}_i{}^T\mathbf{x}_j + \beta_1)$

Properties of Kernel Functions

$$\begin{aligned}
k(\mathbf{x},\mathbf{x}') &= ck_1(\mathbf{x},\mathbf{x}') \\
k(\mathbf{x},\mathbf{x}') &= f(\mathbf{x})k_1(\mathbf{x},\mathbf{x}')f(\mathbf{x}') \\
k(\mathbf{x},\mathbf{x}') &= q\left(k_1(\mathbf{x},\mathbf{x}')\right) \\
k(\mathbf{x},\mathbf{x}') &= \exp\left(k_1(\mathbf{x},\mathbf{x}')\right) \\
k(\mathbf{x},\mathbf{x}') &= k_1(\mathbf{x},\mathbf{x}') + k_2(\mathbf{x},\mathbf{x}') \\
k(\mathbf{x},\mathbf{x}') &= k_1(\mathbf{x},\mathbf{x}')k_2(\mathbf{x},\mathbf{x}') \\
k(\mathbf{x},\mathbf{x}') &= k_3\left(\boldsymbol{\phi}(\mathbf{x}),\boldsymbol{\phi}(\mathbf{x}')\right) \\
k(\mathbf{x},\mathbf{x}') &= \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}' \\
k(\mathbf{x},\mathbf{x}') &= k_a(\mathbf{x}_a,\mathbf{x}'_a) + k_b(\mathbf{x}_b,\mathbf{x}'_b) \\
k(\mathbf{x},\mathbf{x}') &= k_a(\mathbf{x}_a,\mathbf{x}'_a)k_b(\mathbf{x}_b,\mathbf{x}'_b)
\end{aligned}$$

Non-linear Kernel Function

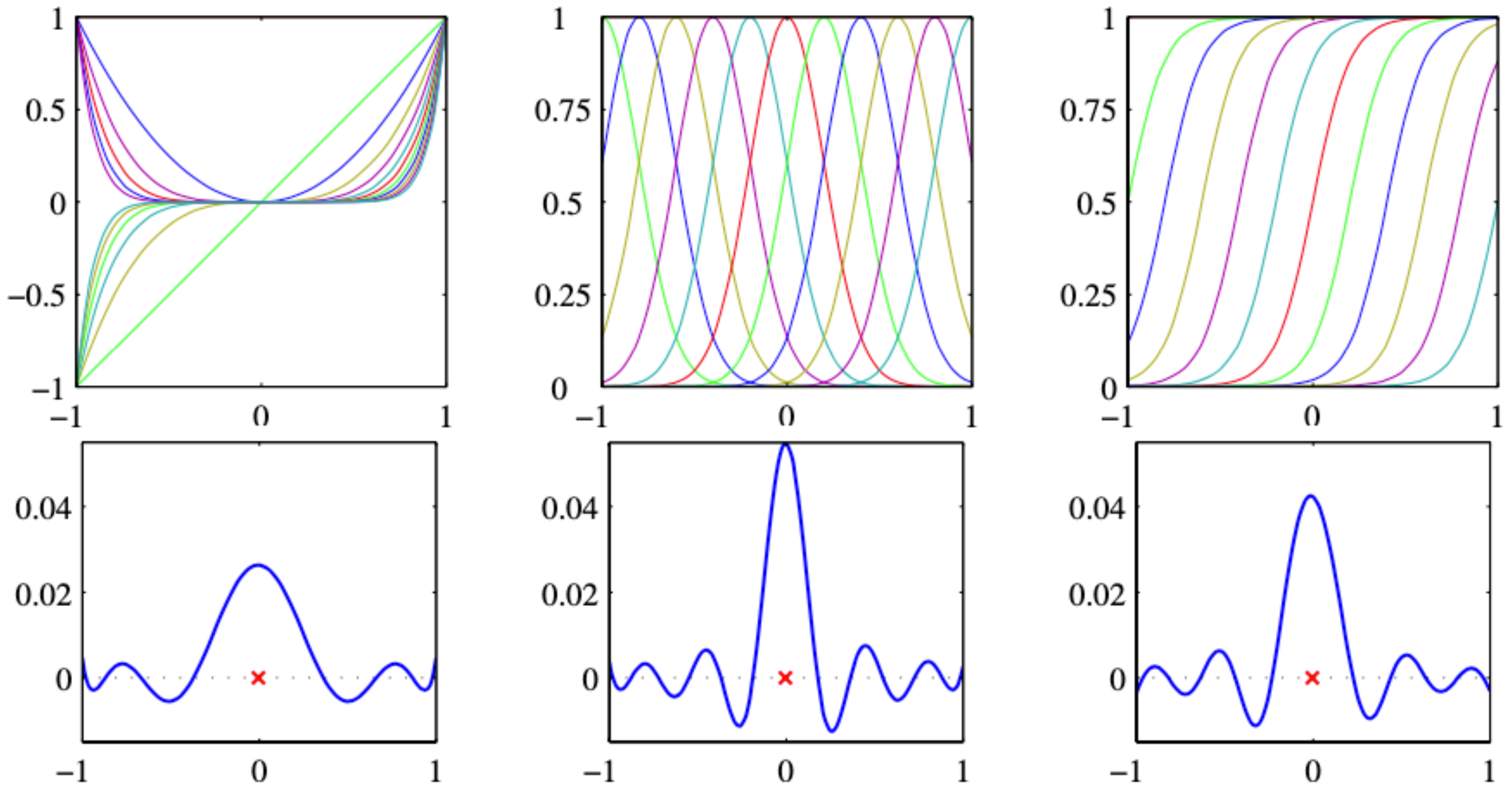
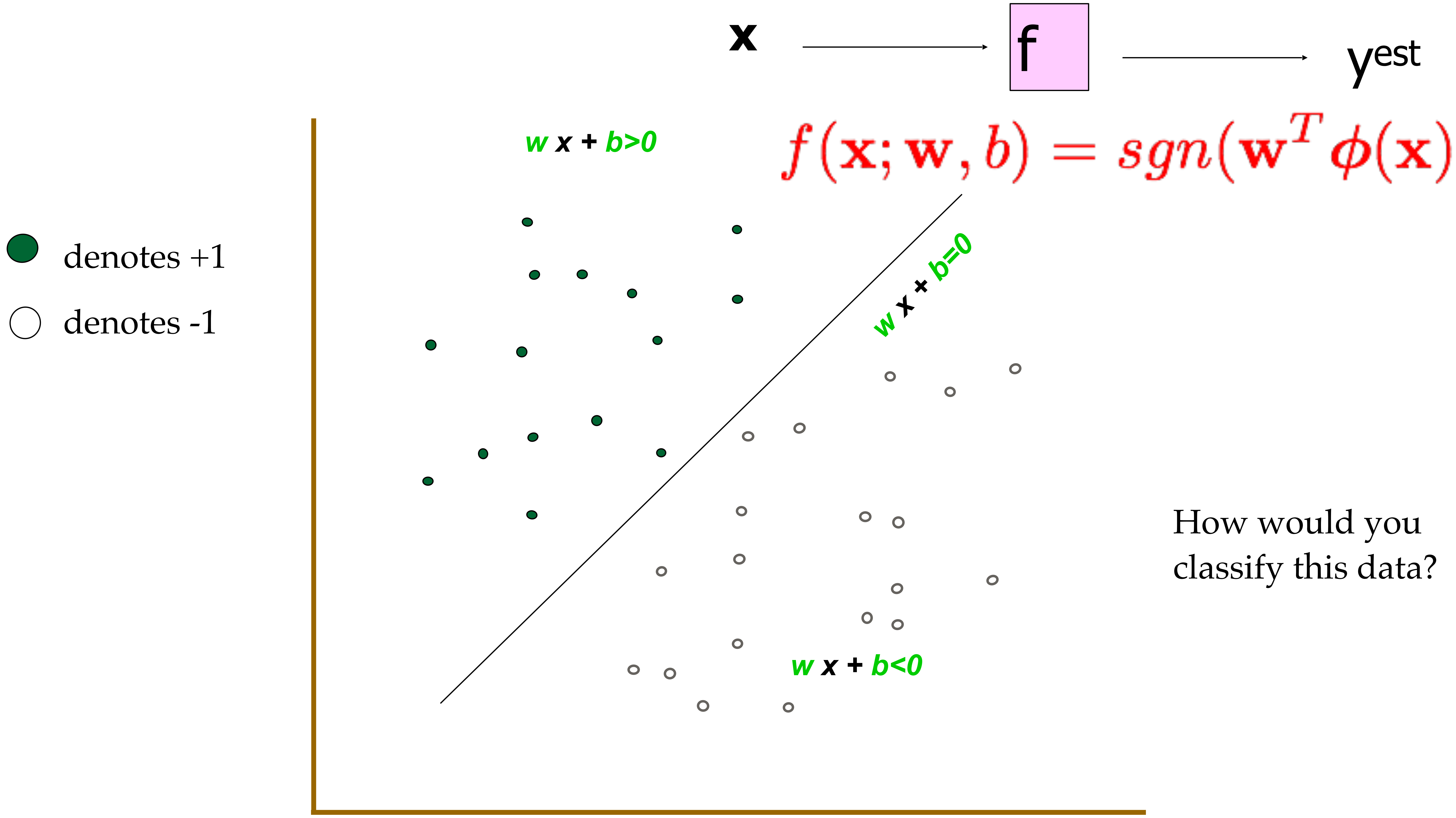


Figure 6.1 Illustration of the construction of kernel functions starting from a corresponding set of basis functions. In each column the lower plot shows the kernel function $k(x, x')$ defined by (6.10) plotted as a function of x for $x' = 0$, while the upper plot shows the corresponding basis functions given by polynomials (left column), 'Gaussians' (centre column), and logistic sigmoids (right column).

Linear Classifiers



$$f(\mathbf{x}; \mathbf{w}, b) = sgn(\mathbf{w}^T \phi(\mathbf{x}) + b)$$

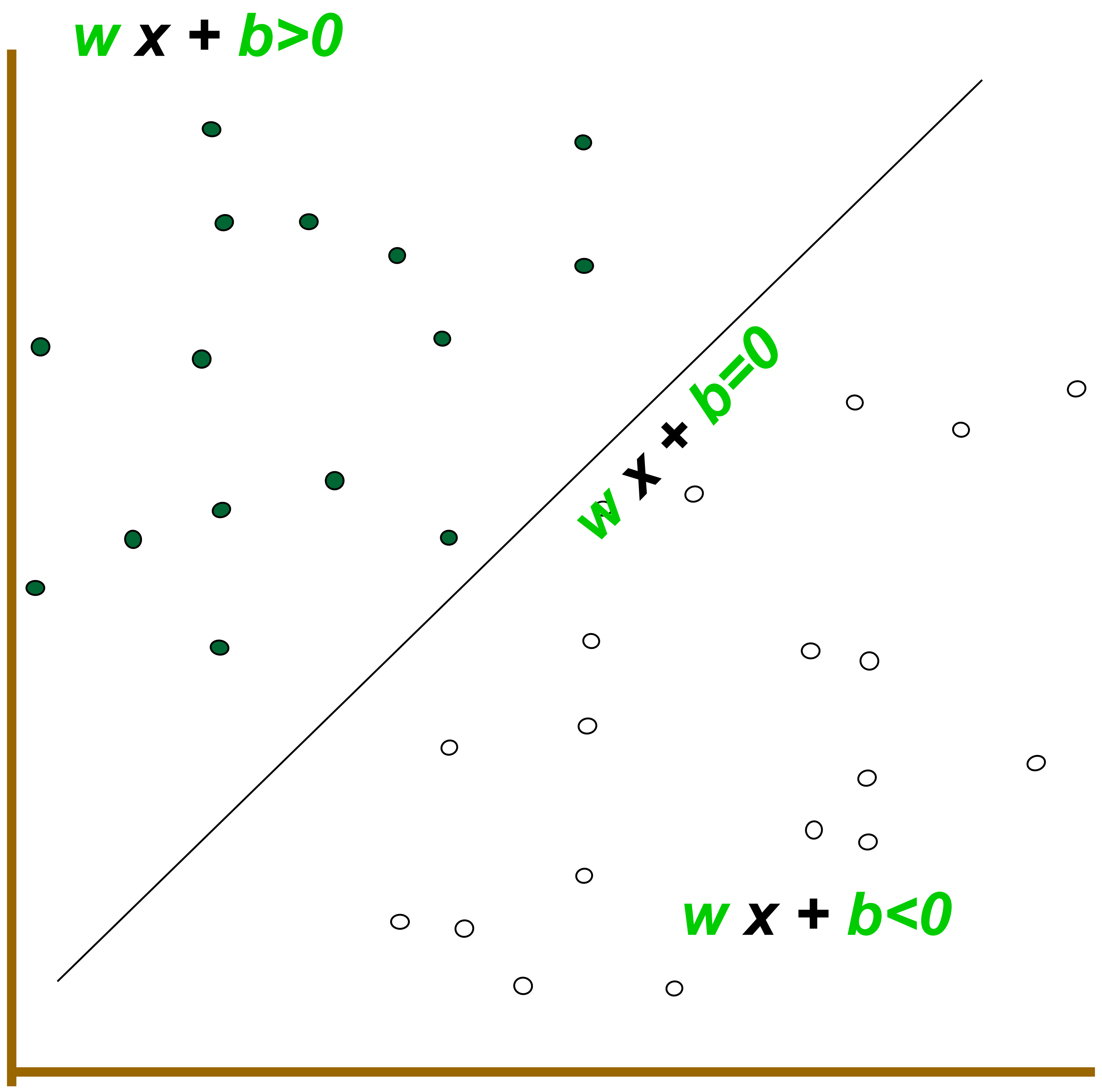
How would you classify this data?

Linear Classifiers

$$\mathbf{x} \longrightarrow \boxed{f} \longrightarrow y^{est}$$

$$f(\mathbf{x}; \mathbf{w}, b) = sgn(\mathbf{w}^T \phi(\mathbf{x}) + b)$$

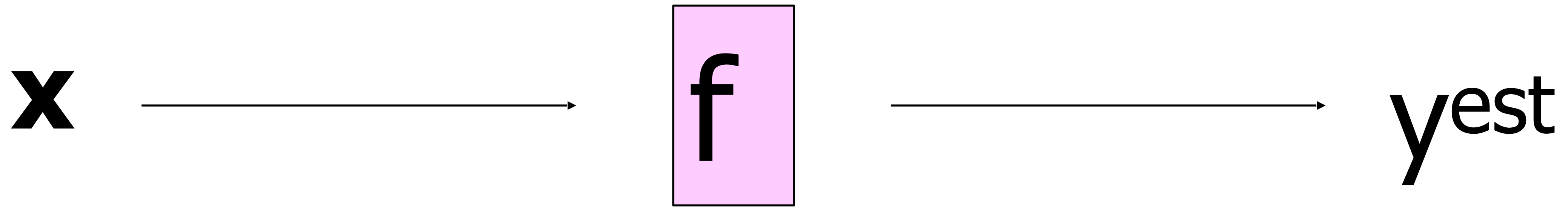
- denotes +1
- denotes -1



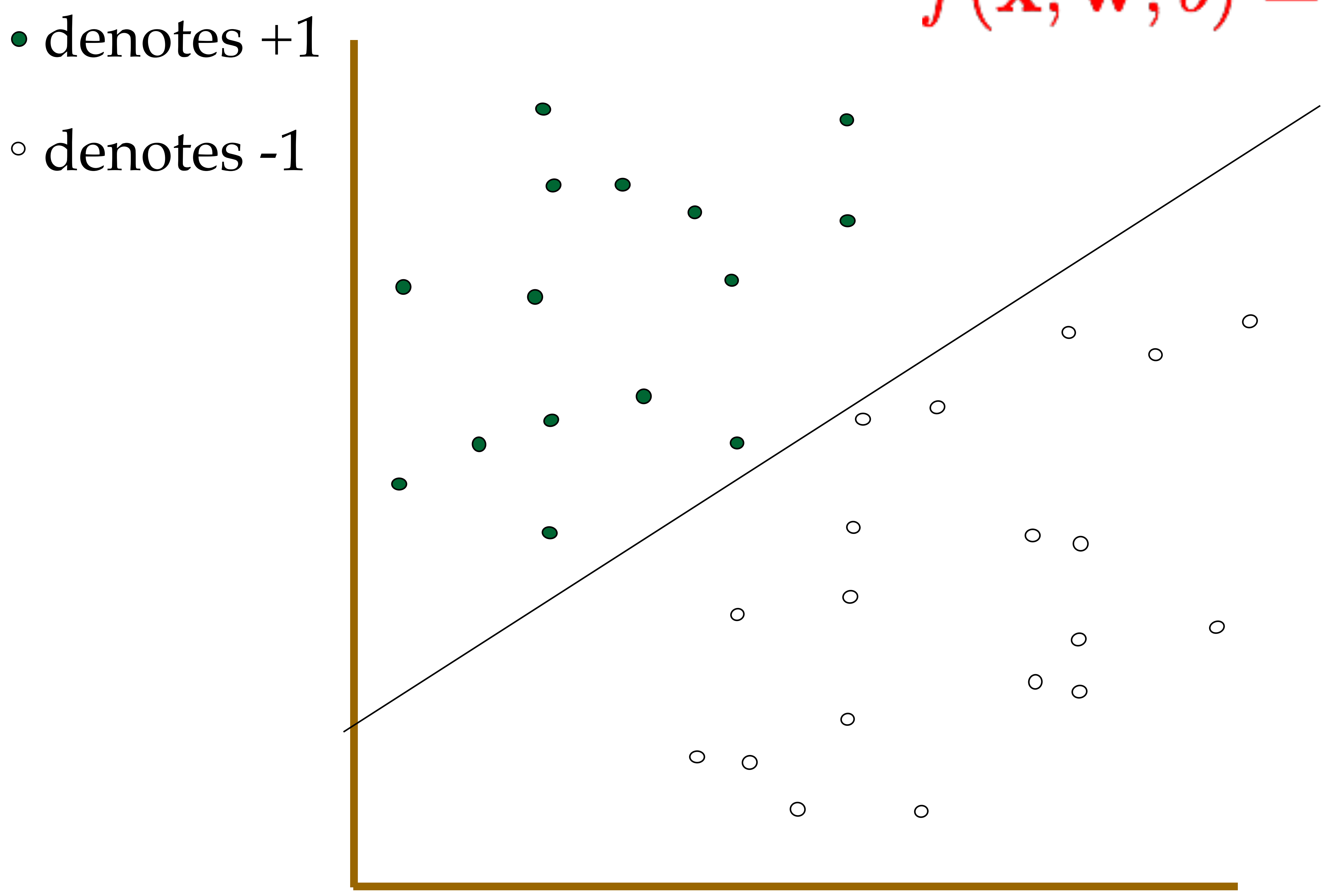
How would you classify this data?

"SVM and applications", Mingyue Tan. Univ of British Columbia

LINEAR CLASSIFIERS



$$f(\mathbf{x}; \mathbf{w}, b) = sgn(\mathbf{w}^T \phi(\mathbf{x}) + b)$$



How would you classify this data?

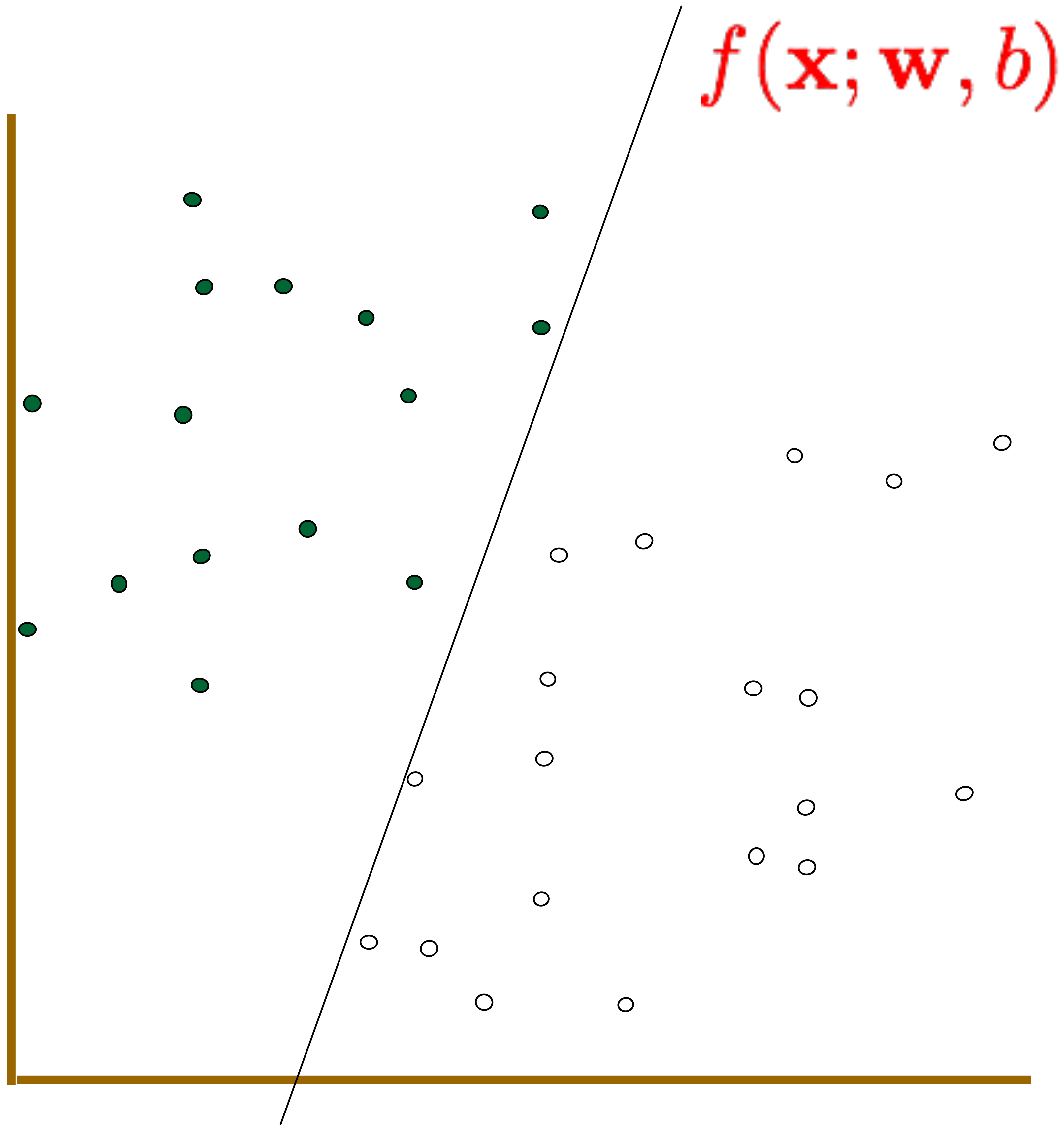
“SVM and applications”, Mingyue Tan. Univ of British Columbia

Linear Classifiers

x ⟶ f ⟶ yest

$$f(\mathbf{x}; \mathbf{w}, b) = sgn(\mathbf{w}^T \phi(\mathbf{x}) + b)$$

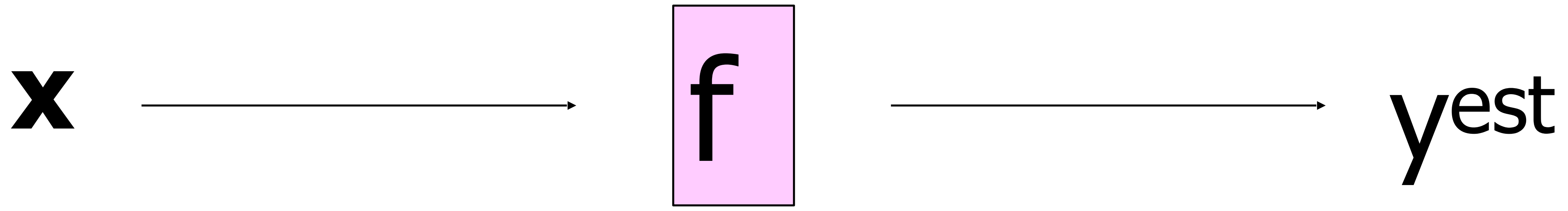
- denotes +1
- denotes -1

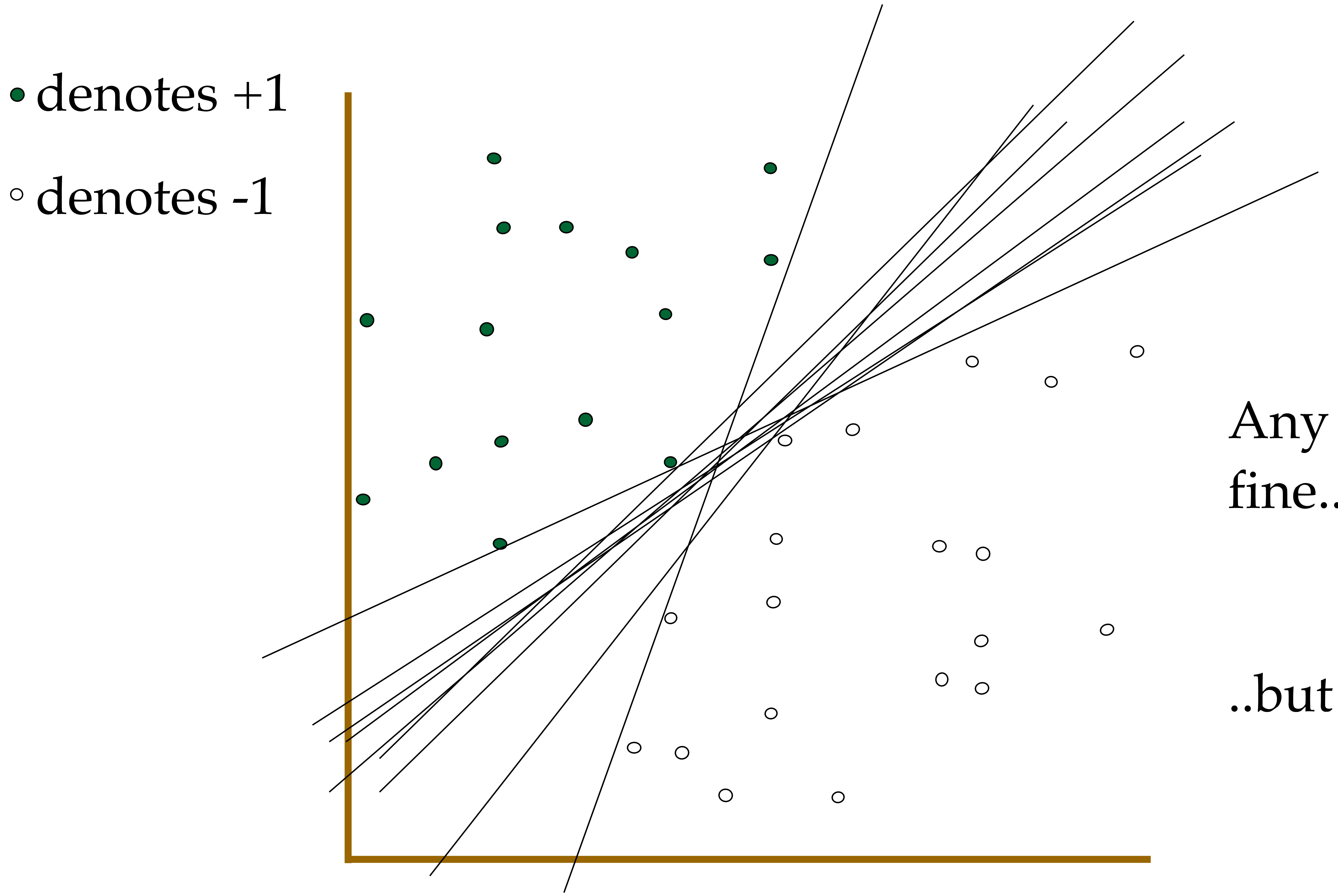


How would you classify this data?

"SVM and applications", Mingyue Tan. Univ of British Columbia

Linear Classifiers





Any of these would be fine..

..but which is best?

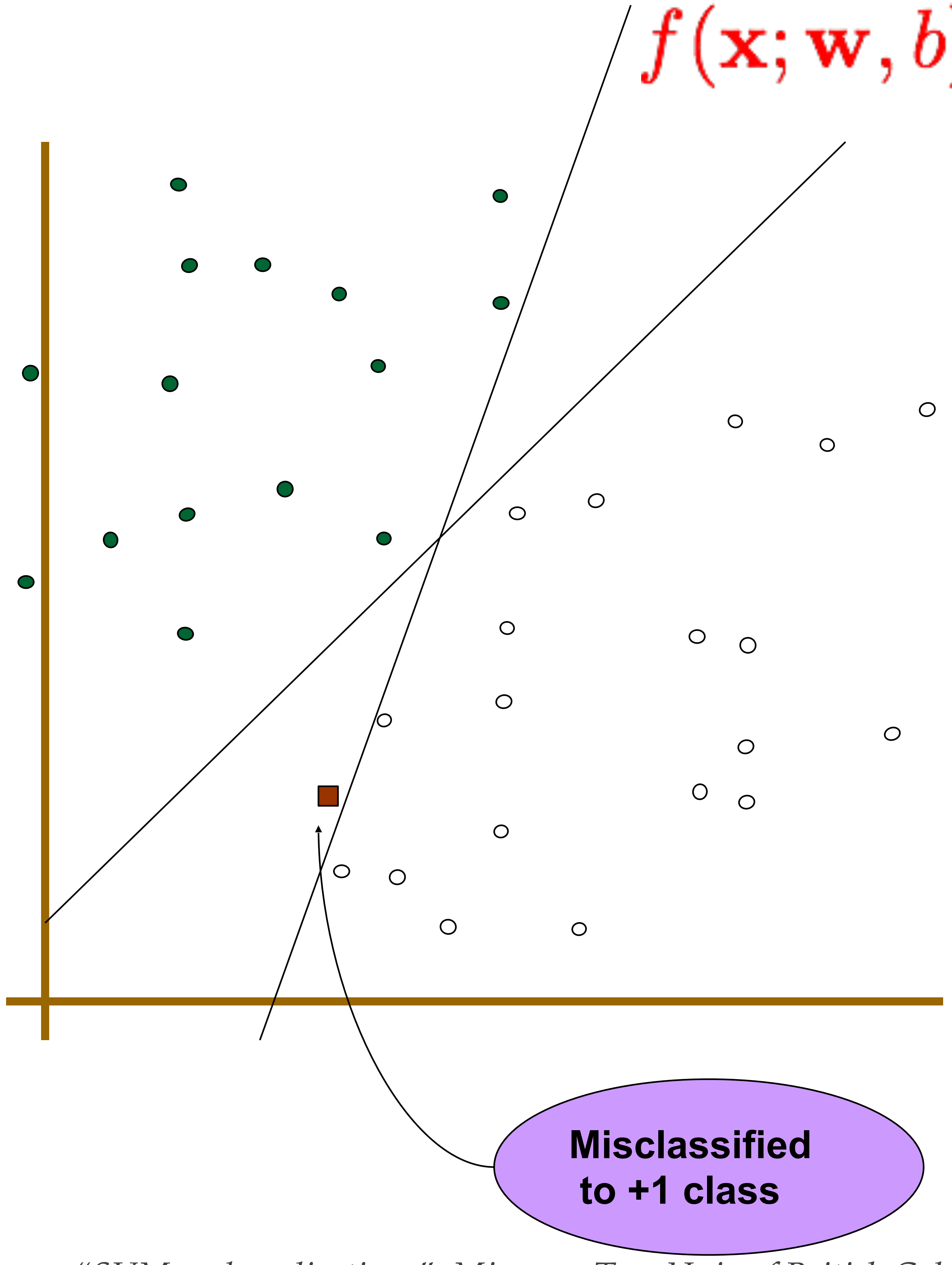
"SVM and applications", Mingyue Tan. Univ of British Columbia

LINEAR CLASSIFIERS

$$\mathbf{x} \longrightarrow \boxed{f} \longrightarrow y^{est}$$

$$f(\mathbf{x}; \mathbf{w}, b) = sgn(\mathbf{w}^T \phi(\mathbf{x}) + b)$$

- ● denotes +1
- ○ denotes -1



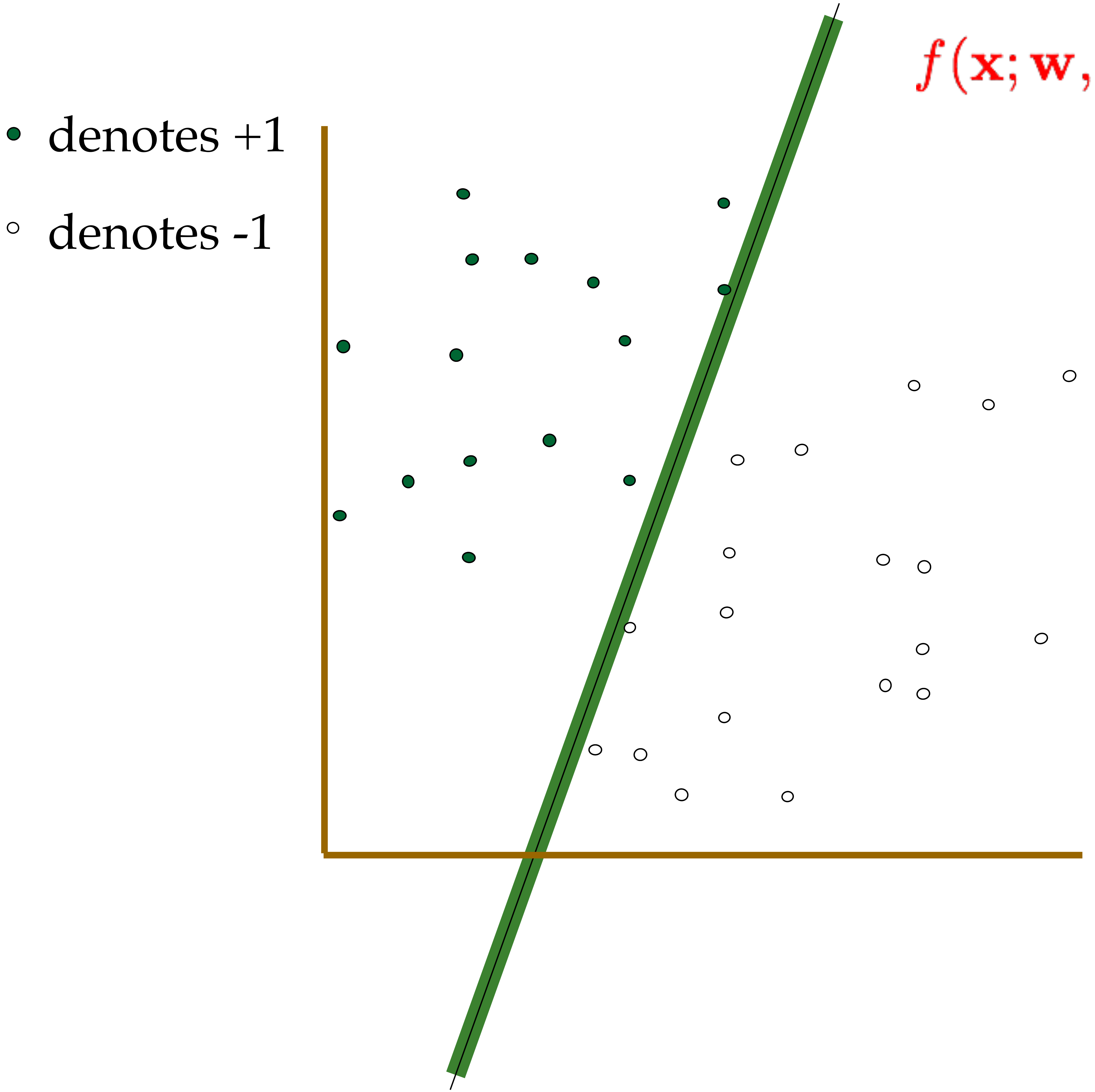
How would you classify this data?

"SVM and applications", Mingyue Tan. Univ of British Columbia

MAX-MARGIN CLASSIFIERS

$\mathbf{x} \longrightarrow \boxed{f} \longrightarrow$ yest

$$f(\mathbf{x}; \mathbf{w}, b) = sgn(\mathbf{w}^T \phi(\mathbf{x}) + b)$$



Define the **margin** of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

SVM Formulation

- Goal - **1) Correctly classify all training data**

$$\left.\begin{array}{ll}\mathbf{w}^T\phi(\mathbf{x}_n)+b\geq 1 & if \quad t_n=+1\\ \mathbf{w}^T\phi(\mathbf{x}_n)+b\leq 1 & if \quad t_n=-1\end{array}\right\}$$

$$t_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b)\geq 1$$

2) Define the Margin

$$\frac{1}{||\mathbf{w}||}min_n\ [t_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b)]$$

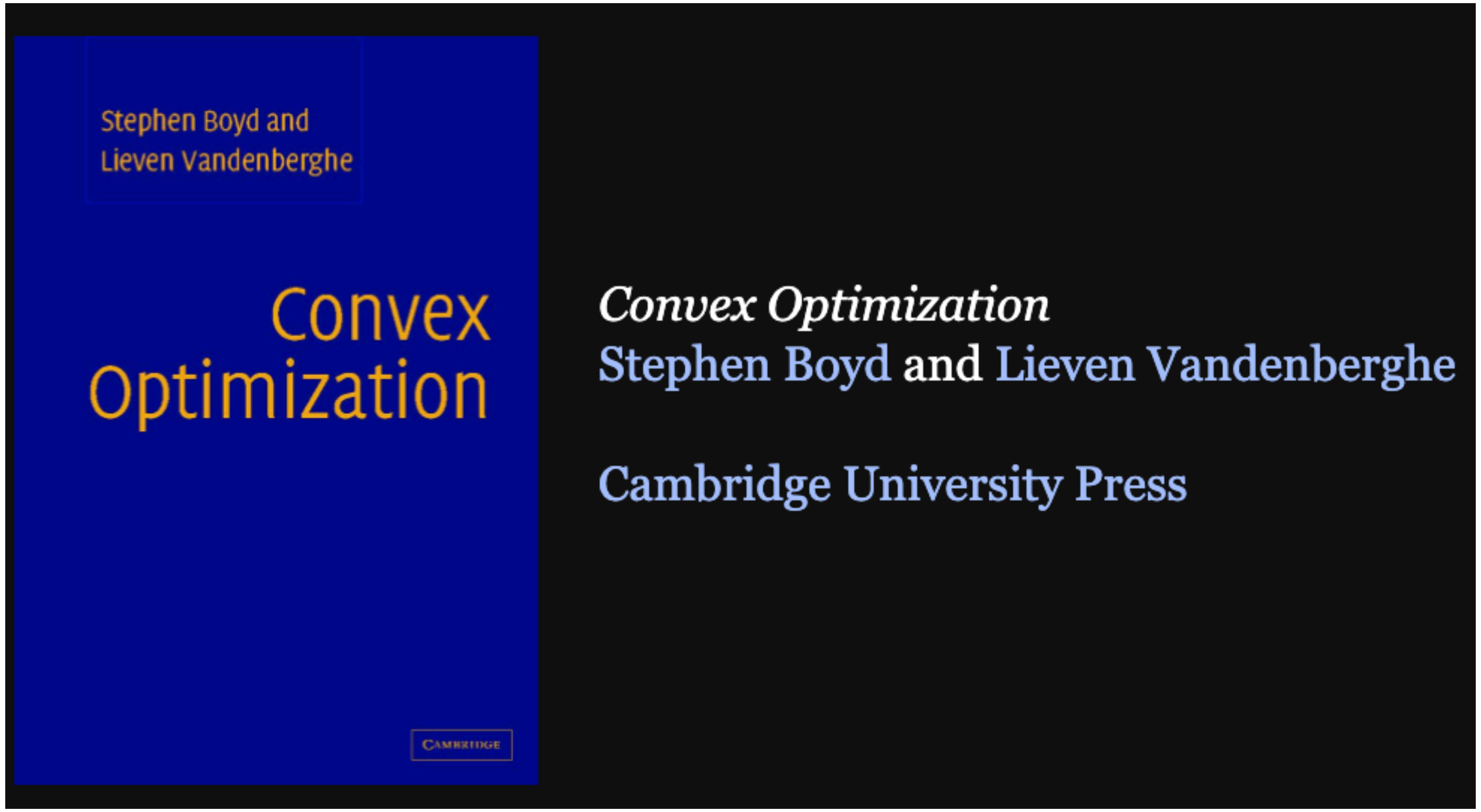
3) Maximize the Margin

$$argmax_{\mathbf{w},b}\left\{\frac{1}{||\mathbf{w}||}min_n\ [t_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b)]\right\}$$

- Equivalently written as

$$argmin_{\mathbf{w},b}\ \frac{1}{2}||\mathbf{w}||^2 \text{such that} \quad t_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b)\geq 1$$

Constrained Optimization Basics



https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf

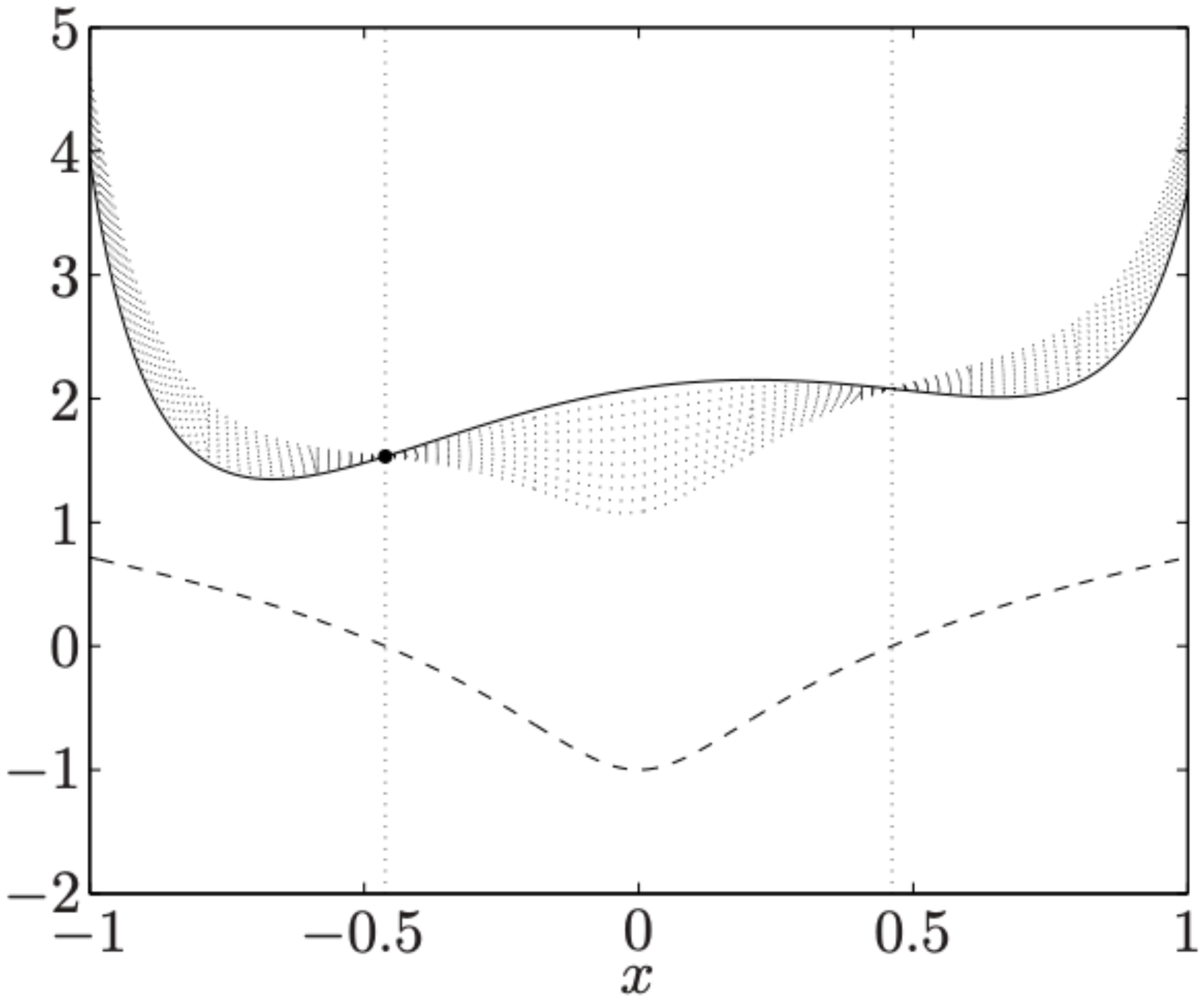


Figure 5.1 *Lower bound from a dual feasible point.* The solid curve shows the objective function f_0, and the dashed curve shows the constraint function f_1. The feasible set is the interval $[-0.46, 0.46]$, which is indicated by the two dotted vertical lines. The optimal point and value are $x^\star = -0.46$, $p^\star = 1.54$ (shown as a circle). The dotted curves show $L(x, \lambda)$ for $\lambda = 0.1,\ 0.2, \ldots, 1.0$. Each of these has a minimum value smaller than $p^\star$, since on the feasible set (and for $\lambda \geq 0$) we have $L(x, \lambda) \leq f_0(x)$.

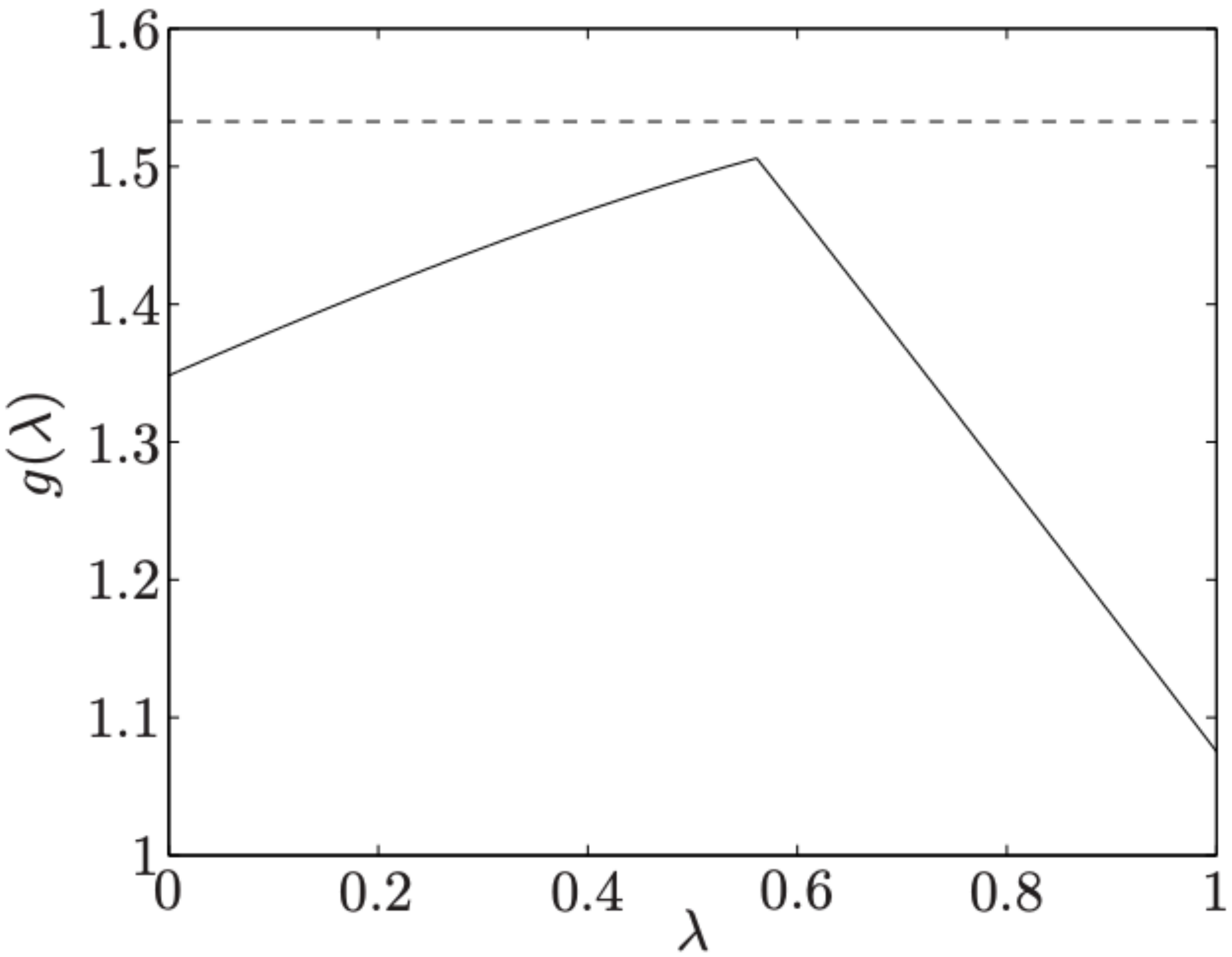


Figure 5.2 The dual function g for the problem in figure 5.1. Neither f_0 nor f_1 is convex, but the dual function is concave. The horizontal dashed line shows $p^\star$, the optimal value of the problem.

Solving the Optimization Problem

- Need to optimize a *quadratic* function subject to *linear* constraints.
- Quadratic optimization problems are a well-known class of mathematical programming problems, and many (rather intricate) algorithms exist for solving them.
- The solution involves constructing a *dual problem* where a *Lagrange multiplier* a_n is associated with every constraint in the primary problem:
- The dual problem in this case is maximized

Find $\{a_1, .., a_N\}$ such that

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

and $\sum_n a_n t_n = 0, \quad a_n \geq 0$

Solving the Optimization Problem

- The solution has the form:

$$\mathbf{w} = \sum_{n=1}^{N} a_n \phi(\mathbf{x}_n)$$

- Each non-zero a_n indicates that corresponding $\mathbf{x_n}$ is a support vector. Let S denote the set of support vectors.

$$b = y(\mathbf{x}_n) - \sum_{m \in S} a_m k(\mathbf{x}_m, \mathbf{x}_n)$$

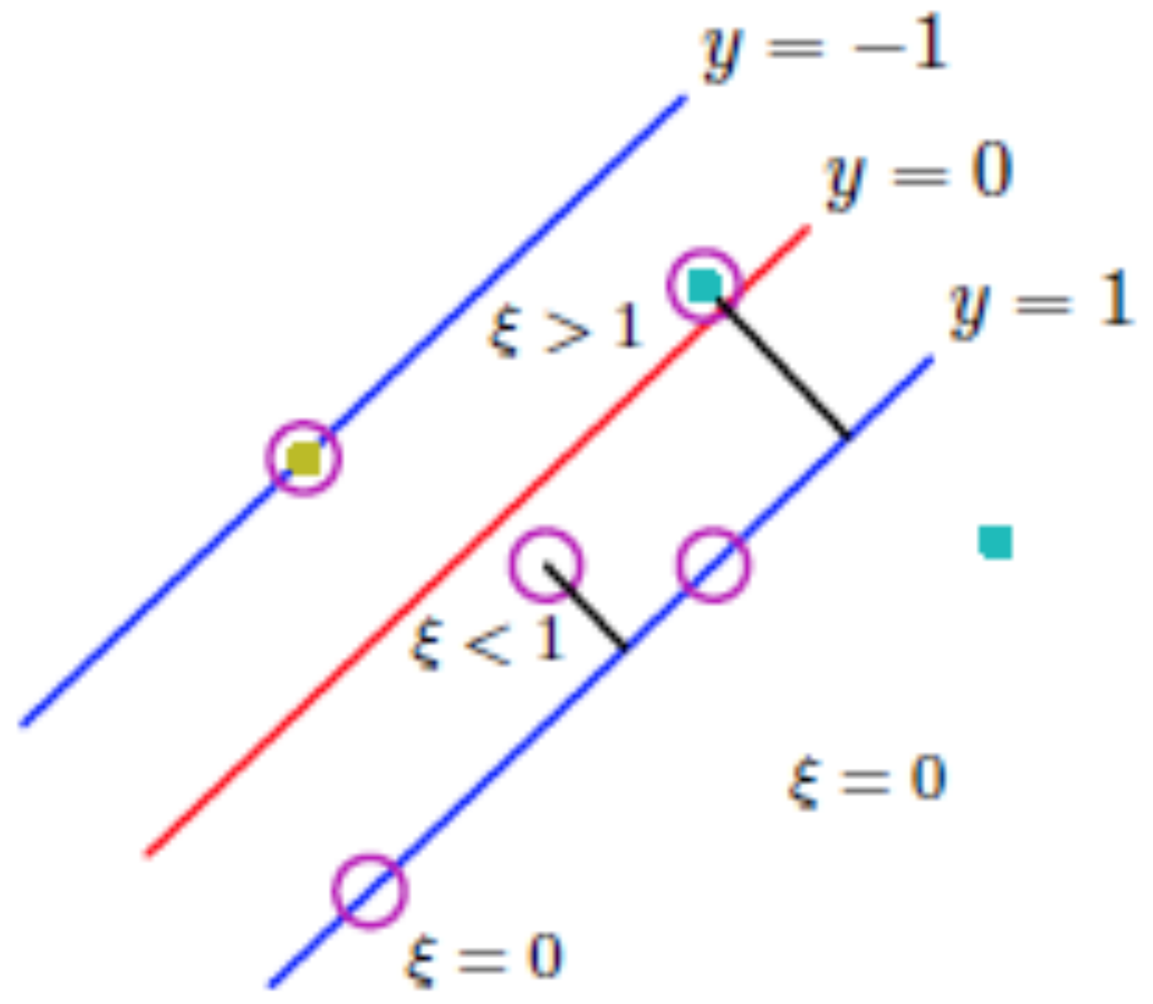
- And the classifying function will have the form:

$$y(\mathbf{x}) = \sum_{n \in S} a_n k(\mathbf{x}_n, \mathbf{x}) + b$$

Overlapping class boundaries

- The classes are not linearly separable - Introducing slack variables ζ_n
- Slack variables are non-negative $\zeta_n \geq 0$
- They are defined using

$$t_n y(\mathbf{x}_n) \geq 1 - \zeta_n$$



- The upper bound on mis-classification

$$\sum_n \zeta_n$$

- The cost function to be optimized in this case

$$C \sum_n \zeta_n + \frac{1}{2} \mathbf{w}^T \mathbf{w}$$

SVM Formulation - overlapping classes

- Formulation very similar to previous case except for additional constraints

$$0 \leq a_n \leq C$$

- Solved using the dual formulation - sequential minimal optimization algorithm
- Final classifier is based on the sign of

$$y(\mathbf{x}) = \sum_{n \in S} a_n k(\mathbf{x}_n, \mathbf{x}) + b$$

THANK YOU

Sriram Ganapathy and TA team
LEAP lab, C328, EE, IISc
sriramg@iisc.ac.in