CS57300
PURDUE UNIVERSITY

OCTOBER 13, 2021

DATA MINING

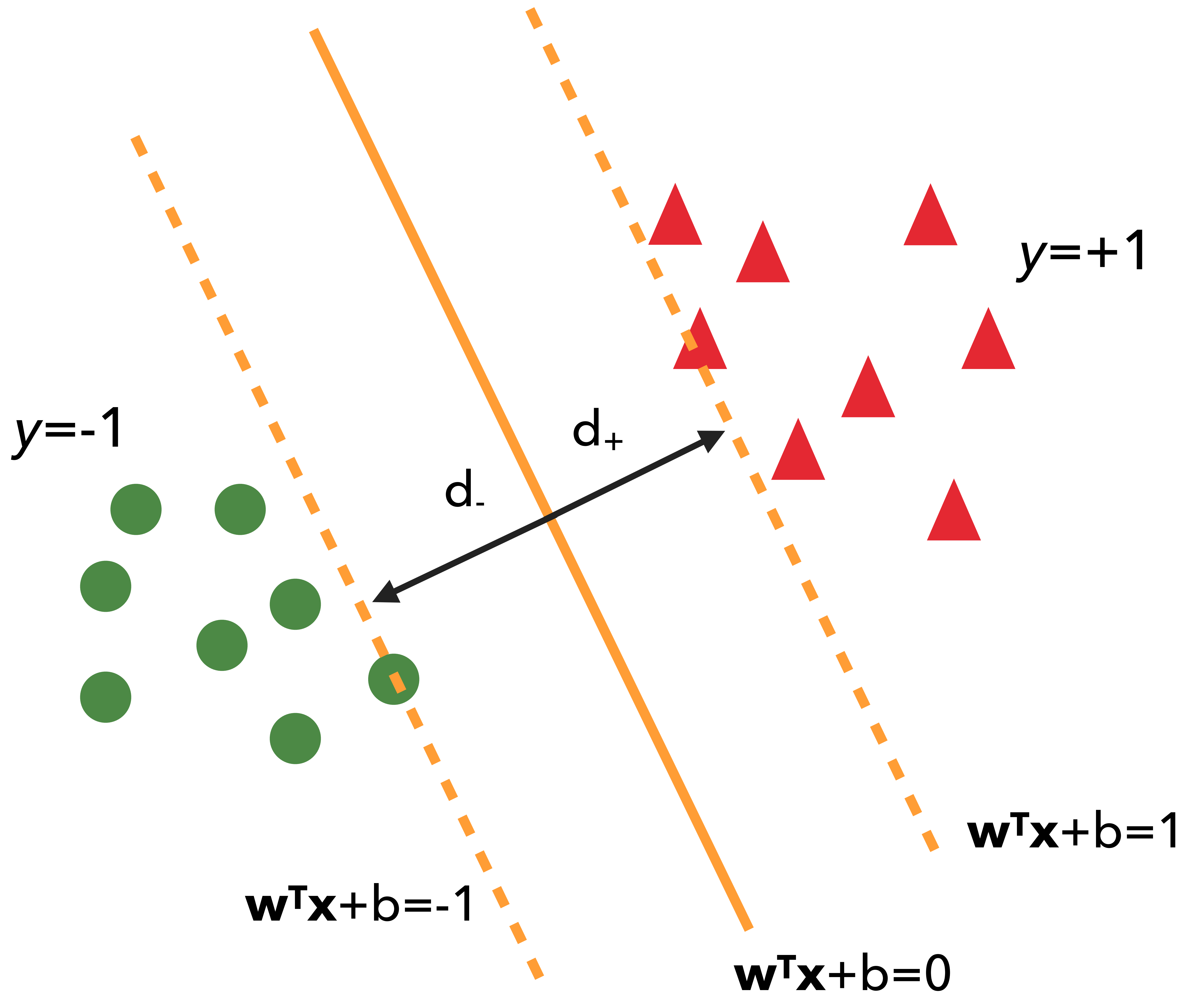
ANNOUNCEMENT

- Midterm exam:
  - October 25 - October 26: 24 hour exam, exam questions will be released on BrightSpace at 10am EST October 25, and you need to submit the answers to BrightSpace before 10am EST October 26.
  - **Closed-book, closed-note**; non-programmable calculator allowed
  - Question type: Multiple choice, T/F, short questions, application of data mining algorithms
  - You need to type all your answers (template will be provided)
  - You need to sign a honor statement
  - We will randomly sample 10% of the students and ask them to go through their answers with TAs after the exam

SVM: RECAP

SVM: KNOWLEDGE REPRESENTATION AND SCORING FUNCTION

- Linear SVM: $y = sign\left[\sum_{i=1}^{m} w_i x_i + b\right]$
- Margin = $d_+ + d_- = 2/||\mathbf{w}||$
- Optimization problem
  - max $2/||\mathbf{w}||$
  - subject to
    $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \forall i \in \{1,2,\ldots,N\}$



SVM LEARNING

- Equivalent to minimize $||\mathbf{w}||^2/2$ subject to

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \forall i \in \{1,2,\ldots,N\}$$

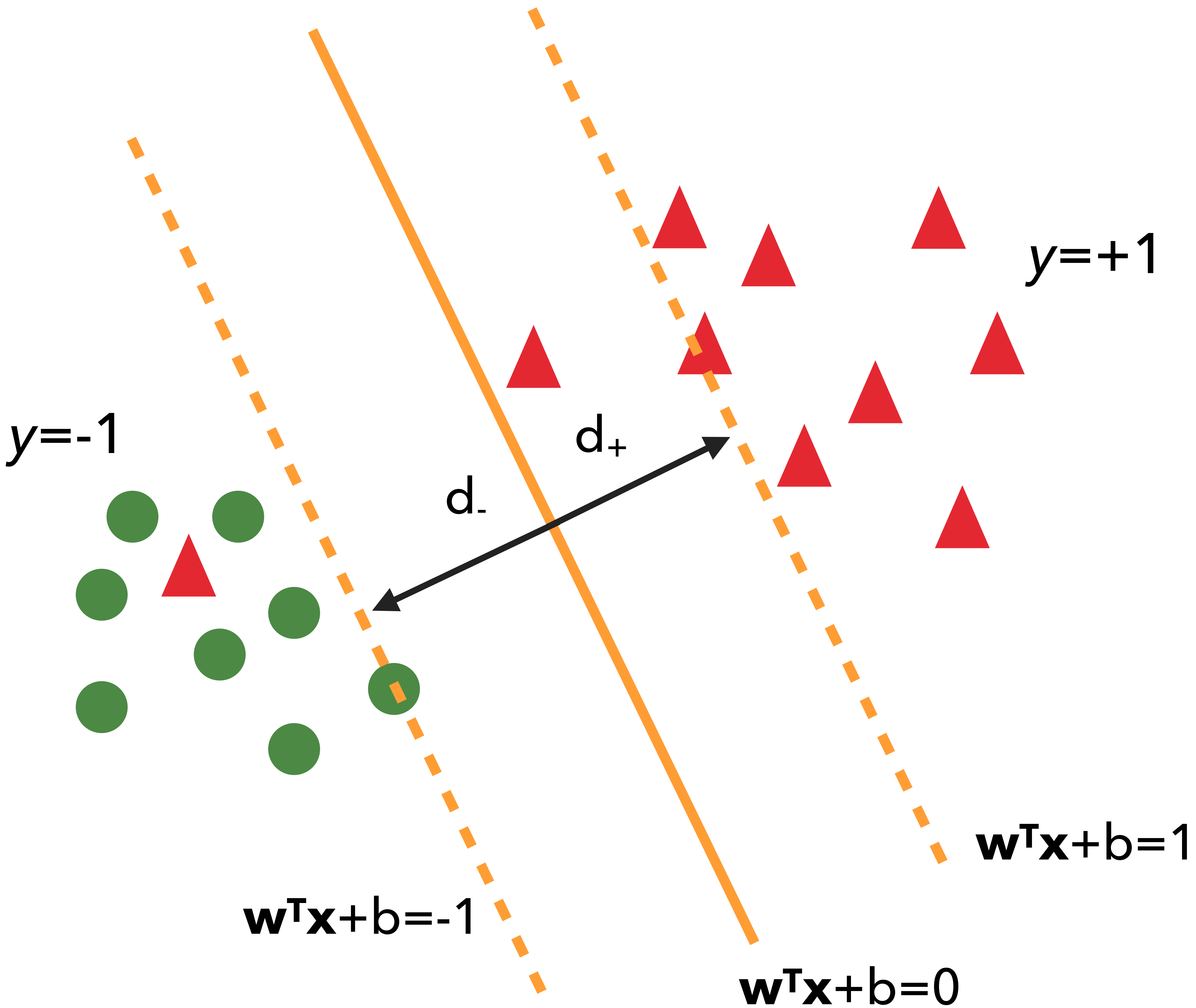
- This is a **quadratic optimization** problem subject to linear constraints, there is a unique minimum
- Lagrangian function $L(\mathbf{w}, b, \lambda_i) = \frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{N} \lambda_i (1 - y_i(\mathbf{w}^T \mathbf{x}_i + b))$

WHAT ABOUT LINEARLY NON-SEPARABLE DATA?

- Introduce slack variables $\varepsilon_i \geq 0$ such that:

  $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \varepsilon_i, \forall i \in \{1,2,\ldots,N\}$

- ε_i measures the amount of error
  - When $0 < \varepsilon_i \leq 1$, data is between the margin, but classified correctly
  - When $\varepsilon_i > 1$, data is misclassified



"SOFT" MARGIN OPTIMIZATION

- With slack variables the score function is:

$$\min_{\mathbf{w},\xi} ||\mathbf{w}||^2 + C\sum_i^N \xi_i$$

- And new constraints:

$$y_i(x_i \cdot w + b) - (1 - \xi_i) \geq 0 \;\; \forall i$$

- If ξ are sufficiently large, then every constraint can be satisfied
- C is regularization parameter
  - Small C means constraints can be ignored in order to find large margin
  - Large C means constraints cannot be ignored and result is small margin (C=∞ enforces hard margin)

SVM OPTIMIZATION

- Constraint can be rewritten as:

$$y_i f(x_i) \geq 1 - \xi_i \quad \forall i$$

- Together with $\xi_i \geq 0$, is equivalent to:

$$\xi_i = max\Big(0, 1 - y_i f(x_i)\Big)$$

- Hence we can use the following score in unconstrained optimization:

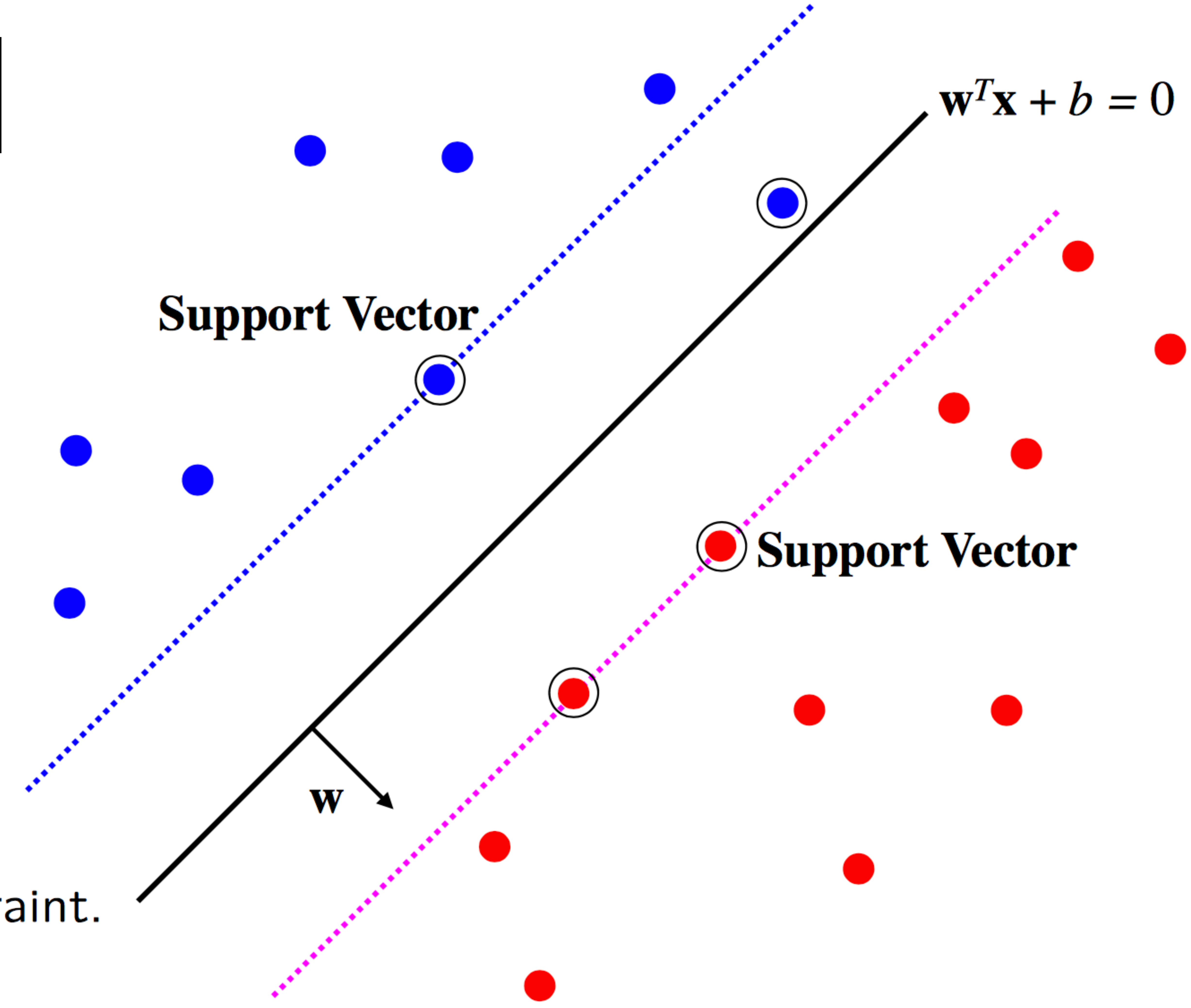
$$\underset{\mathbf{w}}{min}||\mathbf{w}||^2 + C\sum_{i}^{N}\Big[max\Big(0, 1 - y_i f(x_i)\Big)\Big]$$

NEW OBJECTIVE

$$\min_{\mathbf{w}} ||\mathbf{w}||^2 + C \sum_{i}^{N} \left[\underbrace{\boxed{max\Big(0, 1 - y_i f(x_i)\Big)}}_{\text{Hinge Loss}} \right]$$

Points are in three categories:

1. $y_i f(x_i) > 1$
   Point is outside margin.
   No contribution to loss
2. $y_i f(x_i) = 1$
   Point is on margin.
   No contribution to loss.
   As in hard margin case.
3. $y_i f(x_i) < 1$
   Point violates margin constraint.
   Contributes to loss



REWRITE THE OBJECTIVE FUNCTION

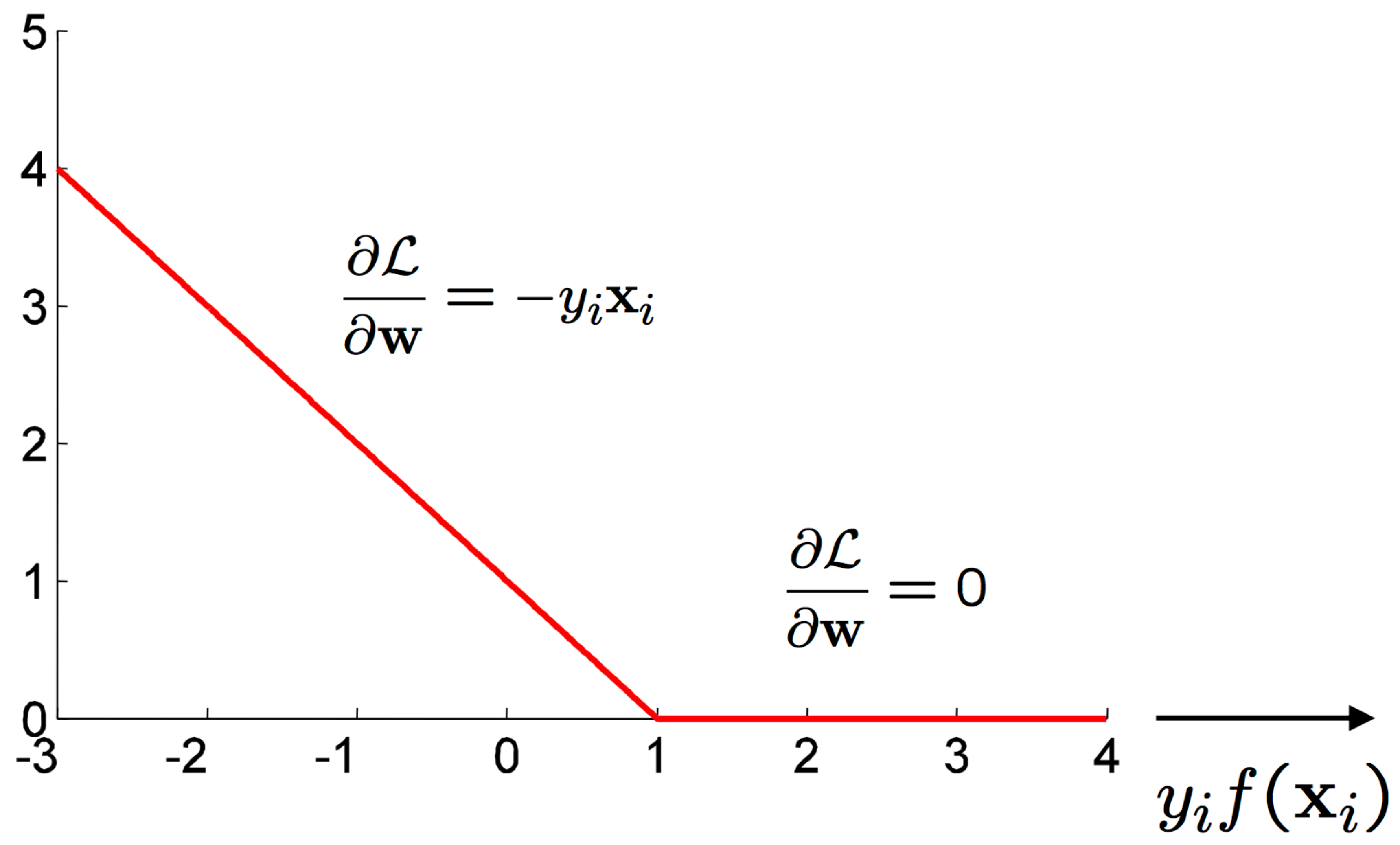
$$\min_{\mathbf{w}} ||\mathbf{w}||^2 + C \sum_{i}^{N} \left[max\Big(0, 1 - y_i f(x_i) \Big) \right]$$

SVM OPTIMIZATION WITH SUB-GRADIENT

- Rewrite optimization problem:

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{i}^{N} \left(\frac{\lambda}{2} ||\mathbf{w}||^2 + \left[max\left(0, 1 - y_i f(x_i)\right)\right]\right)$$

- Now $\lambda = \dfrac{2}{N \cdot C}$, becomes the regularization parameter
- Hinge loss is not differentiable however– so must use sub-gradient for optimization



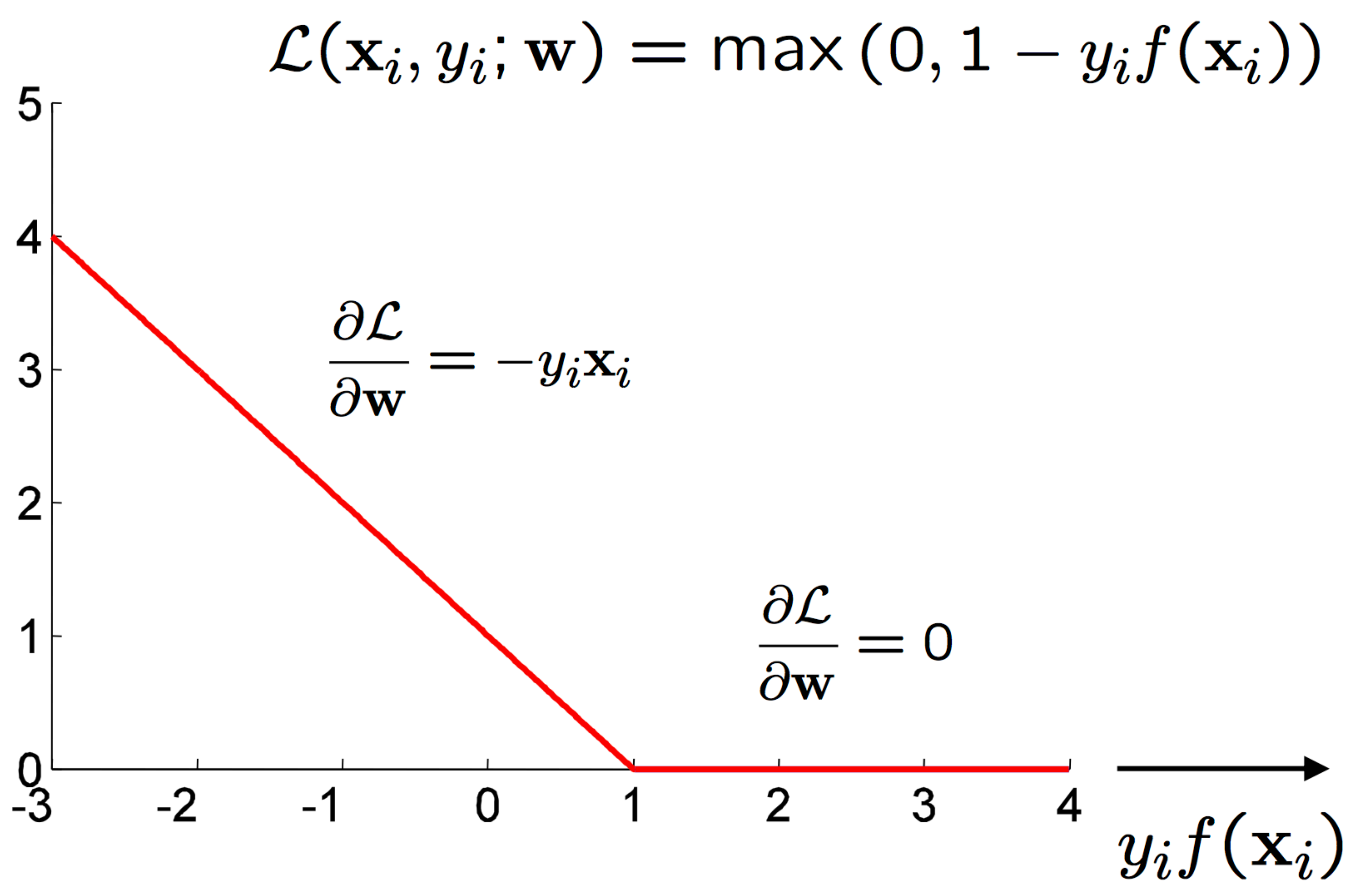
SUB-GRADIENT DESCENT

- Iterative update for SVM weights using sub-gradient descent:

$$\mathbf{w}_{t+1} \quad \leftarrow \quad \mathbf{w}_t - \eta \frac{1}{N} \sum_i^N (\lambda \mathbf{w}_t + \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{x}_i, y_i; \mathbf{w}_t))$$

- where η is the learning rate as per usual
- Each iteration cycles through the data:

$$\begin{aligned} \mathbf{w}_{t+1} \quad &\leftarrow \quad \mathbf{w}_t - \frac{\eta}{N}(\lambda \mathbf{w}_t - y_i \mathbf{x}_i) && \text{if } y_i f(\mathbf{x}_i) < 1 \\ &\leftarrow \quad \mathbf{w}_t - \frac{\eta}{N} \lambda \mathbf{w}_t && \text{otherwise} \end{aligned}$$



SVM EXAMPLE

$$BC = +1 \quad \text{if} \quad \left[\mathbf{w}^T \mathbf{x}\right] > 0$$
$$BC = -1 \quad \text{otherwise}$$

Intercept	Age>40	Income=high	Student=yes	Credit=fair	BuysComp?
1	0	1	0	1	-1
1	0	1	0	0	-1
1	0	1	0	1	+1
1	1	0	0	1	+1
1	1	0	1	1	+1
1	1	0	1	0	-1
1	0	0	1	0	+1
1	0	0	0	1	-1
1	0	0	1	1	+1
1	1	0	1	1	+1
1	0	0	1	0	+1
1	0	0	0	0	+1
1	0	1	1	1	+1
1	1	0	0	0	-1

$$\mathbf{x} = [Int, A, I, S, CR]$$
$$\mathbf{w} = [w_0, w_A, w_I, w_S, w_{CR}]$$

SVM parameters = w

- Score function: margin + hinge loss on errors
- Estimate **w** to maximize margin, while minimizing errors

SVM LEARNING

- Score function: soft margin (includes hinge loss on errors)

$$\min_{\mathbf{w}} \frac{1}{N}\sum_{i}^{N}\left(\frac{\lambda}{2}||\mathbf{w}||^2 + \left[max\left(0, 1 - y_i f(x_i)\right)\right]\right)$$

- Estimate **w** by minimizing objective function using gradient descent

Gradient descent:

Start at some **w**, e.g., **w**=[0,0,0,0,0]

Make predictions given current **w**: $\forall i \;\; \widehat{y}_i = \mathbf{w}^T \mathbf{x_i}$

Calculate gradient for each parameter:

$$\forall j \;\; \nabla_j = \frac{1}{N}\left[\sum_{i=1}^{n}(\lambda w_j - \nabla_{ji})\right]$$

$$\text{where } \nabla_{ji} = y_i x_{ij} \;\; \text{if } y_i \widehat{y}_i < 1; \;\; 0 \;\; \text{otherwise}$$

Move parameters in direction of gradient: $\forall j \;\; w_j^{new} = w_j - \eta \nabla_j$

Repeat until stopping criteria is met

SVM PREDICTION

Intercept	Age>40	Income=high	Student=yes	Credit=fair	BuysComp?
1	0	1	0	1	-1
1	0	1	0	0	-1
1	0	1	0	1	+1
1	1	0	0	1	+1
1	1	0	1	1	+1
1	1	0	1	0	-1
1	0	0	1	0	+1
1	0	0	0	1	-1
1	0	0	1	1	+1
1	1	0	1	1	+1
1	0	0	1	0	+1
1	0	0	0	0	+1
1	0	1	1	1	+1
1	1	0	0	0	-1
1	**0**	**1**	**0**	**0**	**?**

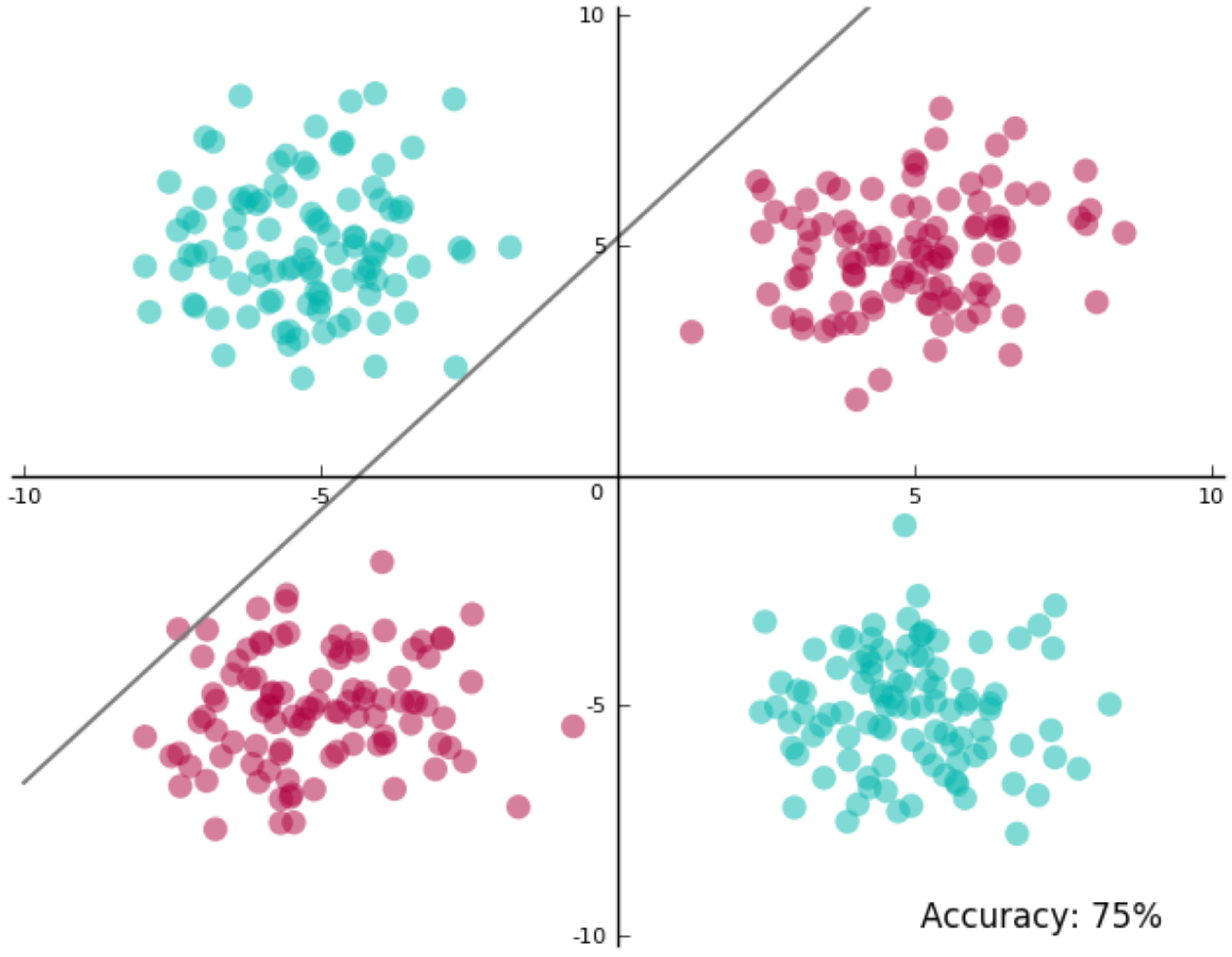
- What is the probability that new person will buy a computer?

$$\mathbf{x} = [1, 0, 1, 0, 0]$$
$$\mathbf{w} = [-.5, 1.2, 3, -2, 0.7]$$

$$\mathbf{x}^T\mathbf{w} = 2.5$$

$$BC = +1$$

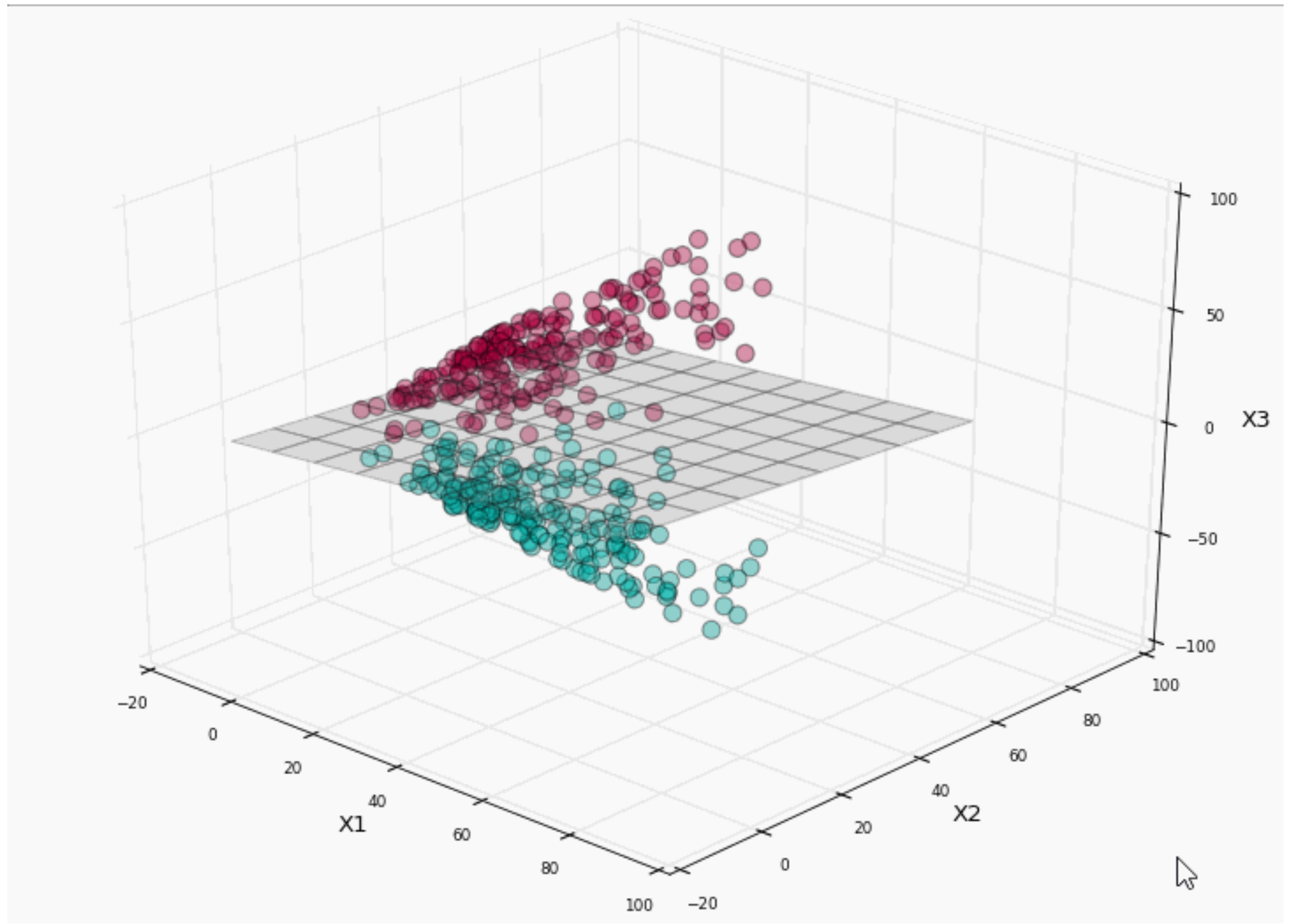
BEYOND LINEAR SVM

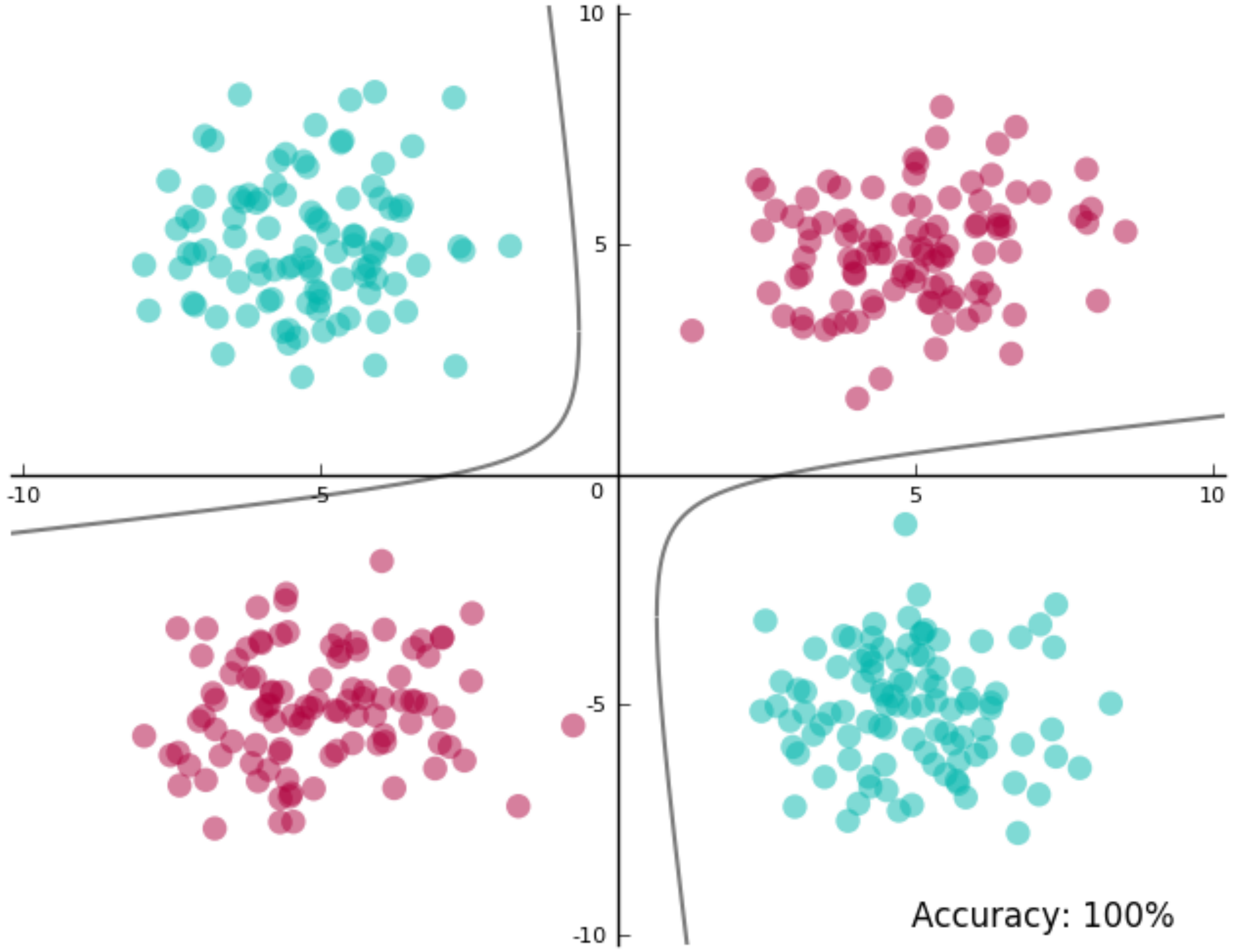


Hardly linearly-separable!

PROJECTING TO HIGHER DIMENSIONAL SPACE

- Data that is not linearly separable in lower-dimensional space is more likely to be linearly separable when projected onto higher dimensions
- $X_1 = x_1^2, X_2 = x_2^2, X_3 = \sqrt{2}x_1x_2$





EMPOWERING SVM

- Project data into a higher-dimensional space
- Find a hyperplane in the higher-dimensional space that can almost linearly separate the training examples
- Project the hyperplane back to the original lower-dimensional space to get the non-linear decision boundary!
- Which higher-dimensional space should I project the data into?

THE KERNEL TRICK

- You only need to know the dot products between data points to learn SVM and make prediction with SVM (related to primal-dual of optimization problems)
  - Given a training dataset, you only need to know $\mathbf{x}_i^T\mathbf{x}_j$ for any two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ in the training example to learn the linear SVM
  - After a linear SVM is learned, given a test data point $\mathbf{x}$, you only need to know $\mathbf{x}^T\mathbf{x}_i$ for all the data points $\mathbf{x}_i$ in the training example to make predictions
- Given a projection function $\mathbf{x} \rightarrow \phi(\mathbf{x})$
  - The linear SVM in the higher-dimensional space can be learned and used as long as we know $\phi(\mathbf{x})^T\phi(\mathbf{y})$