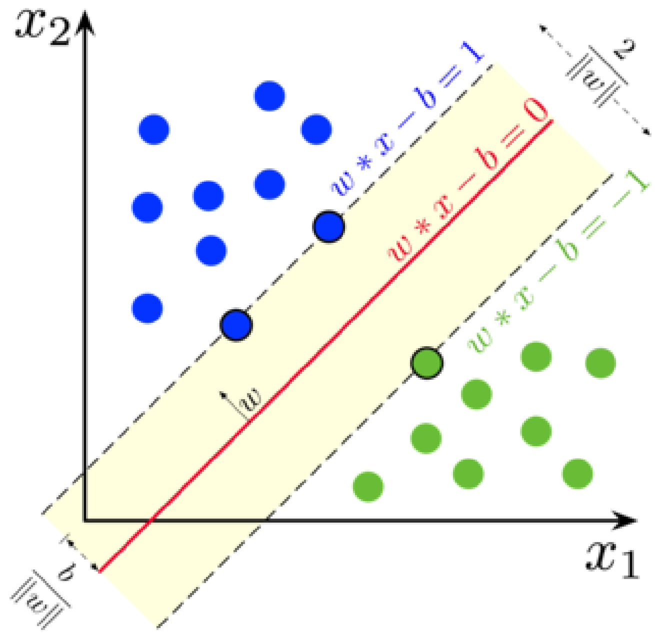
SUPERVISED MACHINE LEARNING: CLASSIFICATION

MODULE 3:

SUPPORT VECTOR MACHINES

TABLE OF CONTENTS

Overview



- **Support Vector Machine** is a supervised learning algorithm used for classification and sometimes regression.
- The main goal is to find a **hyperplane (siêu phẳng)** that best separates the classes by **maximizing the margin** (khoảng cách biên).
- Only the **support vectors** — the data points closest to the decision boundary — determine this hyperplane.

Concept and Intuition

- SVM seeks the decision boundary that **maximizes the distance** between classes, not just minimizes misclassification.
- The model avoids being overly sensitive to nearby points or outliers.
- The **margin** is defined by dotted lines equidistant from the hyperplane and nearest support vectors.

Mathematical Formulation

- **Decision boundary (in linear SVM):**

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

- **Margin distance:**

$$\text{margin} = \frac{2}{\|\mathbf{w}\|}$$

- **Prediction rule:**

$$\hat{y} = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$

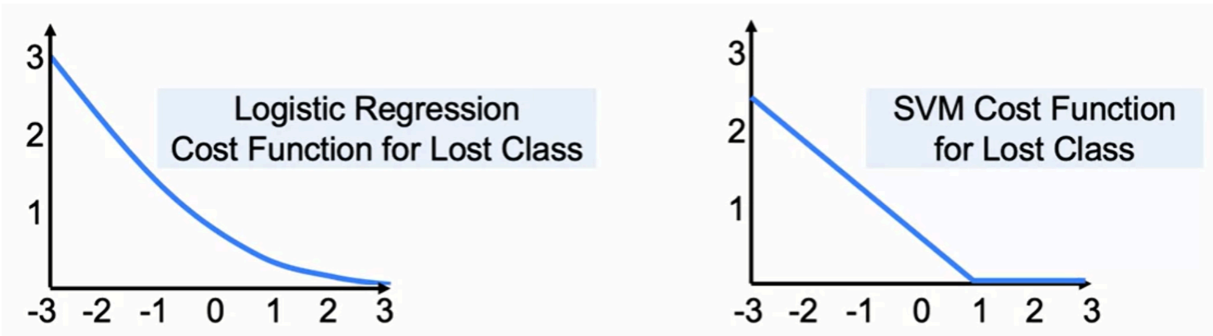
Cost Function

SVM minimizes the **hinge loss** plus a regularization term:

$$\boxed{\min_{\mathbf{w},b} \frac{1}{n} \sum_{i=1}^{n} \max(0,\ 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i + b)) + \lambda\|\mathbf{w}\|^2}$$

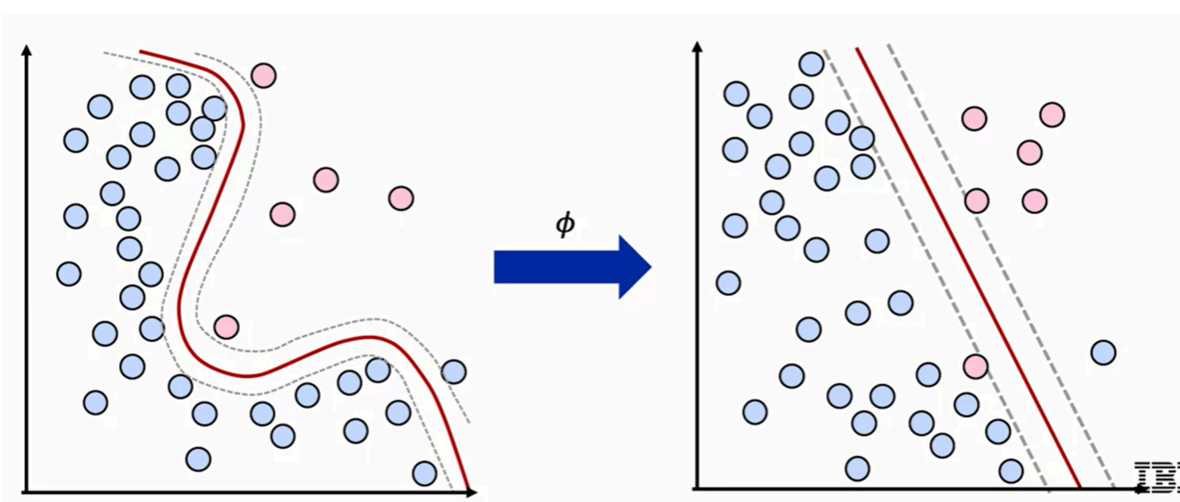
- **Hinge Loss** (Hàm mất mát bản lề) penalizes misclassified points and those inside the margin.
- **Regularization** (Chuẩn hóa / Điều chuẩn) prevents overfitting and controls complexity.
- Controlled by parameter CCC:
    - Small CCC → stronger regularization → simpler boundary.
    - Large CCC → weaker regularization → tighter fit.

Comparison: Logistic Regression vs SVM



Aspect	Logistic Regression	SVM
Cost Function	Log loss (smooth)	Hinge loss (piecewise linear)
Output	Probability	Class label
Regularization	L1/ L2	Controlled by C
Sensitivity	More to outliers	Ignores well-separated points

Kernel Trick



- **Purpose:** Extend SVM to handle **non-linear** boundaries by mapping data to higher-dimensional space.
- Achieved using **Kernel functions (hàm nhân)** that compute similarity without explicitly transforming data.
- Common kernels:
  - Linear:

$$K(x_i, x_j) = x_i^T x_j$$

  - Polynomial:

$$K(x_i, x_j) = (x_i^T x_j + c)^d$$

  - RBF/ Gaussian:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$$

- RBF kernel adds flexibility but is **computationally expensive** for large datasets.

Regularization and Parameters

- C: controls trade-off between margin width and classification error.
- γ: in RBF kernel, controls the “reach” of each support vector.
  - Low γ: smoother boundary.
  - High γ: tighter boundary, may overfit.

Implementation in Python

```
from sklearn.svm import SVC

# RBF Kernel SVM
model = SVC(kernel='rbf', C=1.0, gamma='scale')
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

- For SVM, use:

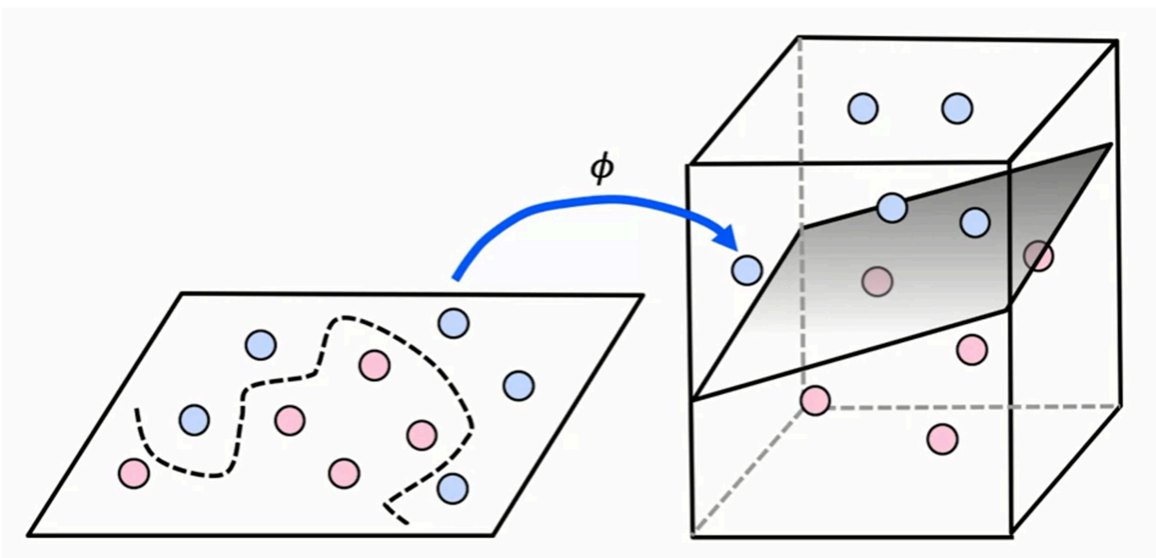
```
from sklearn.svm import LinearSVC

model = LinearSVC(C=1.0, penalty='l2')

model.fit(X_train, y_train)
```

Use `GridSearchCV` for tuning `C`, `gamma`, and `kernel`.

Kernel Approximation



- When data is large, direct kernel computation becomes slow.
- Use **approximation techniques** to map data into higher dimensions efficiently:
  - **Nystroem method**
  - **RBF Sampler**

```
from sklearn.kernel_approximation import
Nystroem
```

```
feature_map = Nystroem(kernel='rbf',
n_components=100)
X_train_trans =
feature_map.fit_transform(X_train)
X_test_trans = feature_map.transform(X_test)
```

- Then apply a linear classifier (`LinearSVC`, `LogisticRegression`) on `X_train_trans`.

Model Selection Guideline

Feature	Dataset Size	Recommended Model
Many features (>10,000)	Small dataset	Linear SVC/ Logistic Regression
Few features (<100)	Medium dataset (~10k rows)	SVC with RBF Kernel
Few features	Large dataset	Kernel Approximation + Linear Model

Key Takeaways

- SVMs find **maximum-margin hyperplanes** separating classes.
- **Hinge loss** penalizes only misclassified or marginal points.
- **Regularization (C)** balances accuracy and simplicity.
- **Kernels** enable non-linear separation by mapping to higher-dimensional space.
- **RBF kernels** improve flexibility but slow for large data.
- **Approximation methods** (Nystroem, RBF Sampler) provide scalable alternatives.
- **SVMs outperform Logistic Regression** when classes are separable but may lose probability interpretability.