Machine Learning

Chapter 1: Introduction to Machine Learning

What is Machine Learning?

At its core: Machine learning is about **predicting the future based on the past**

Examples:

- Predict how much Alice will like a movie she hasn't seen
- Recommend courses for students based on past ratings
- Detect spam emails from message content
- Forecast stock prices from historical data
- Predict properties of materials

Fundamental Goal: Making informed guesses about **unobserved properties** based on **observed properties**

What Does it Mean to Learn?

Alice's Learning Analogy:

- Alice takes a machine learning course
- Bob (teacher) wants to test if she has "learned"
- Gives her an exam to gauge understanding

What makes a good exam?

- ❌ **Bad:** History of Pottery (unrelated content)
- ❌ **Bad:** Exact lecture questions (no generalization)
- ✅ **Good:** New but related questions testing **generalization**

Key Insight: Learning = Ability to **generalize** from examples

Course Recommendation Example

Setup:

- Students rate courses from -2 (terrible) to +2 (awesome)
- Goal: Predict how Alice will rate "Algorithms"

Unfair Questions:

- How will Alice like "History of Pottery"? (No training data)
- How will Alice like "AI" she already rated +2? (Just recall)

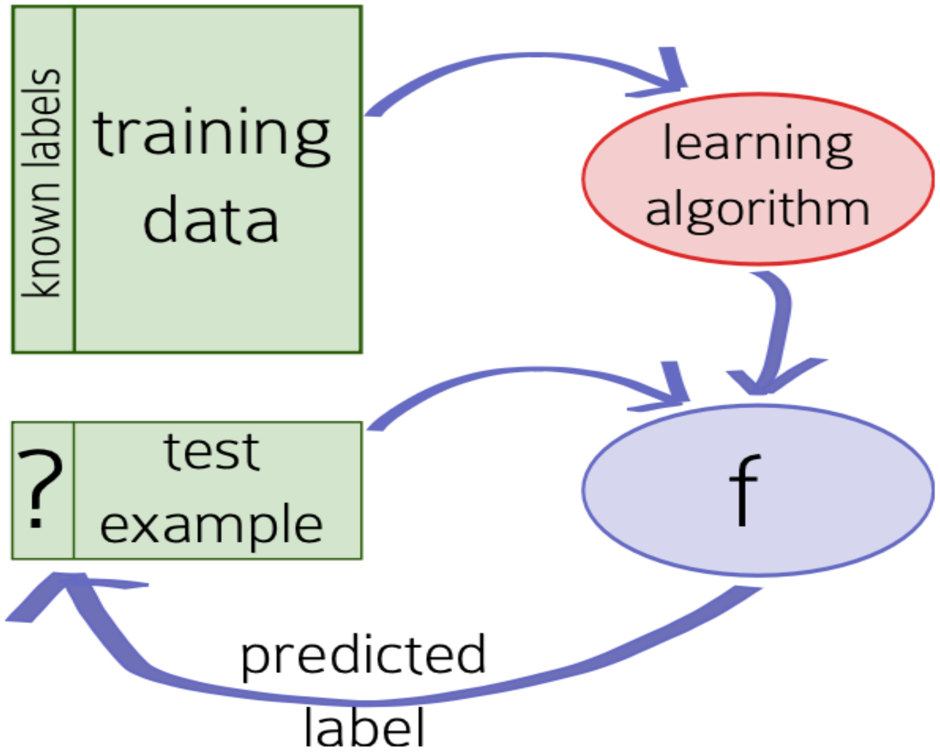
Fair Question:

- How will Alice like "Machine Learning" based on her rating of "AI"?

The Learning Framework

Training Phase:

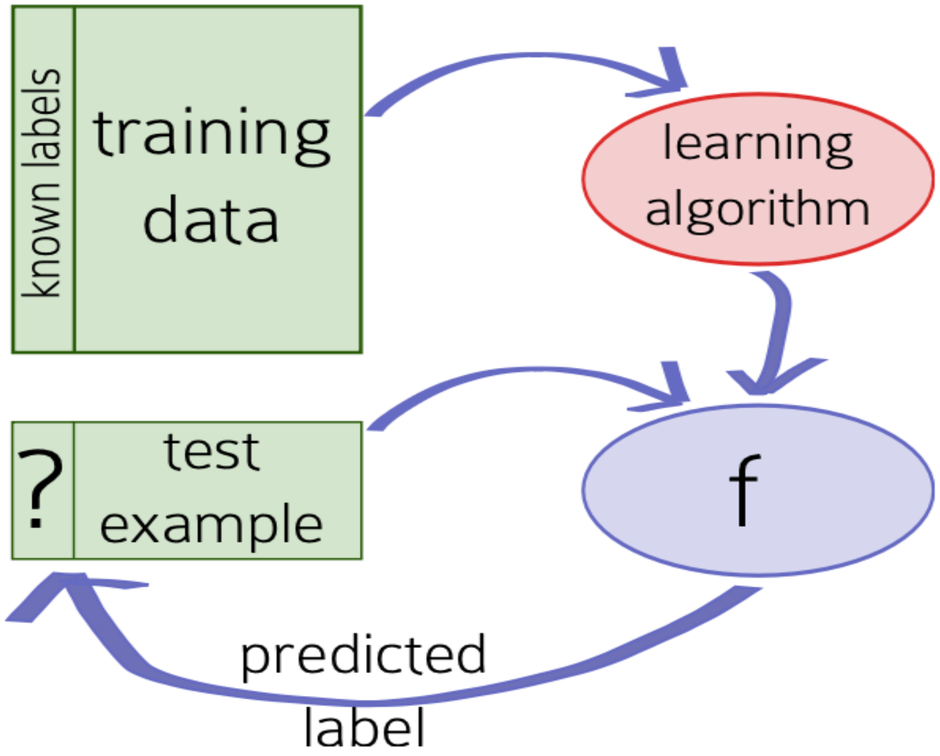
1. **Training Data:** Historical examples with known outcomes
2. **Learning Algorithm:** Processes training data
3. **Induced Function f:** Maps new examples to predictions



The Learning Framework

Testing Phase:

1. **Test Set:** New, unseen examples (closely guarded secret!)
2. **Evaluation:** How well does f perform on test data?
3. **Success:** High performance on test data



Machine Learning: A Better Solution

Machine learning is a subfield of **artificial intelligence** that focuses on:

- **Learning from data** itself
- **Recognizing patterns** in data
- Making decisions based on learned patterns
- **Without being explicitly programmed**

What is Needed to Make a Machine Learn?

1. **Dataset** (collection of examples/instances)
2. **Algorithm** (step-by-step procedure)

What the Machine Does:

- Uses the **algorithm** to create a **model** from the dataset
- Uses the model to predict and solve problems

Types of Machine Learning

Based on the availability and nature of data:

1. **Supervised Learning**
2. **Unsupervised Learning**
3. **Semi-Supervised Learning**
4. **Reinforcement Learning**

1. Supervised Learning

Dataset: Collection of labeled examples $\{(x_i, y_i)\}_{i=1}^{N}$

Where:

- x_i = feature vector

$$x_i = [x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(D)}]$$

- y_i = label (class or real number)

Goal: Produce a model that takes feature vector as input and outputs the label

2. Unsupervised Learning

Dataset: Collection of examples **without labels** $\{x_i\}_{i=1}^N$

Goal: Find hidden patterns, structure, or representations in data

Common Tasks:

- **Clustering:** Group similar examples together
- **Dimensionality Reduction:** Compress data while preserving structure
- **Density Estimation:** Model probability distribution of data
- **Outlier Detection:** Identify unusual examples

Challenge: No "correct answer" to guide learning

3. Reinforcement Learning

Setup: Agent learns through **interaction** with environment

Key Components:

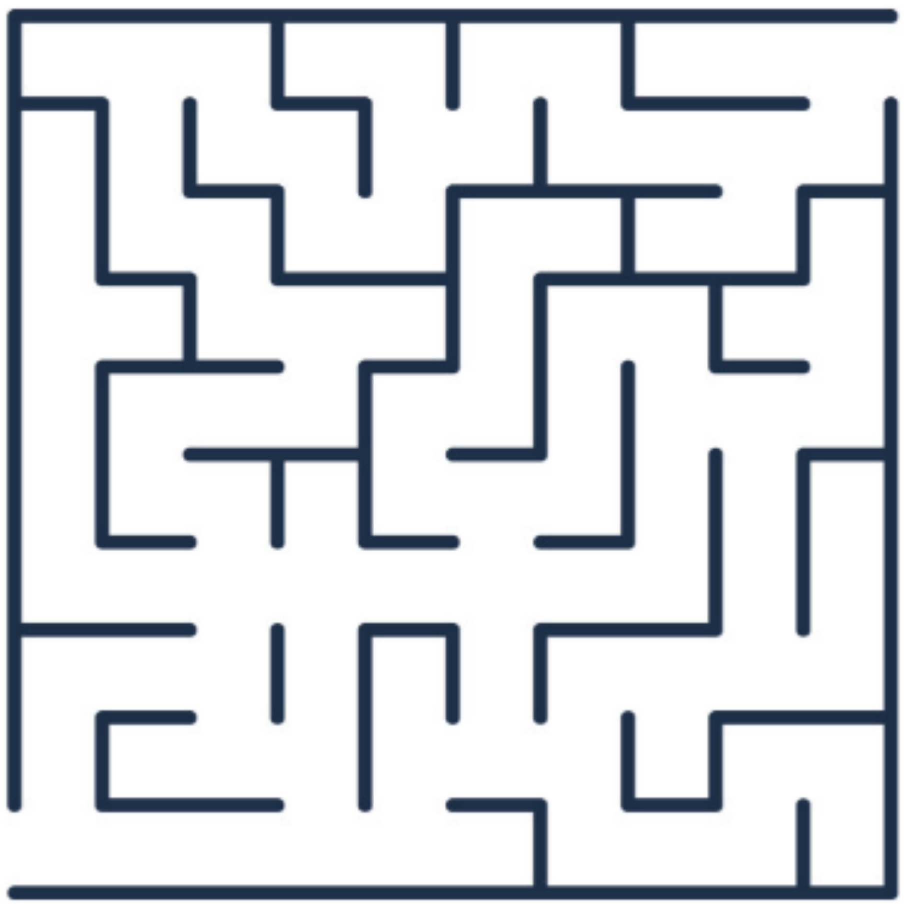
- **Agent:** The learning entity
- **Environment:** The world agent interacts with
- **Actions:** What agent can do
- **Rewards:** Feedback from environment
- **Policy:** Agent's strategy for choosing actions

Goal: Learn optimal policy to maximize cumulative reward

Examples: Game playing, robotics, autonomous driving

3. Reinforcement Learning

Eg. Solving a maze problem,



4. Semi-Supervised Learning

Dataset: Contains **both labeled and unlabeled** examples

$$\{(x_i, y_i)\}_{i=1}^{N_L} \cup \{x_j\}_{j=1}^{N_U}$$

Key Characteristic: Usually $N_U \gg N_L$ (many more unlabeled than labeled examples)

Goal: Same as supervised learning, but leverage unlabeled data to find better model

The Hope: Large amount of unlabeled data helps discover underlying structure

Why useful? Labeling is expensive, but raw data is often abundant

Semi-Supervised Learning Examples

Real-world scenarios:

- **Web page classification:** Few manually labeled pages, millions unlabeled
- **Medical diagnosis:** Few expert-labeled cases, many patient records
- **Speech recognition:** Limited transcribed audio, vast amounts of speech
- **Image classification:** Some labeled photos, millions unlabeled images

Key Insight: Unlabeled data reveals data distribution and can improve decision boundaries

Challenge: How to effectively use unlabeled data without introducing noise

Canonical Learning Problems

Different prediction types require different approaches:

1. Regression

- **Goal:** Predict real-valued numbers
- **Examples:** Stock prices, exam scores, house prices
- **Error Measurement:** Distance from true value matters

2. Binary Classification

- **Goal:** Predict yes/no, positive/negative
- **Examples:** Spam detection, medical diagnosis
- **Error Measurement:** Correct/incorrect prediction

More Canonical Problems

3. Multiclass Classification

- **Goal:** Choose one category from multiple options
- **Examples:** News categorization (sports, politics, entertainment)
- **Error Measurement:** All mistakes equally bad

4. Ranking

- **Goal:** Order objects by relevance/preference
- **Examples:** Search results, course recommendations
- **Error Measurement:** Position in ranking matters

Why categorize? Different problems need different ways to measure "goodness"

Error Measurement Examples

Regression: Stock price prediction

- Predicting \$100.05 instead of \$100.00 ≠ Predicting \$300.00 instead of \$100.00

Classification: News categorization

- Predicting "entertainment" instead of "sports" = Predicting "politics" instead of "sports"

Ranking: Search results

- Wrong result at position 1 > Wrong result at position 10

The Machine Learning Pipeline

```
Training Data → Learning Algorithm → Model f → Test Data → Evaluation
     ↓               ↓                   ↓          ↓           ↓
Past Examples    Pattern Finding   Predictions  New Examples Performance
```

Key Principle:

- Train on past data
- Test on future/unseen data
- **Never** let algorithm see test data during training (cheating!)

Success Metric: Good performance on test data = successful learning

The Problem: Detecting Spam Emails

Traditional Programming Approach:

- Write explicit rules for recognizing spam messages
- Example: If message contains "win money", "free offer" → Spam

Issues:

- Not feasible with too many/complex rules
- Easy to bypass (e.g., "freee" instead of "free")
- No learning capability
- Poor generalization

How Supervised Learning Works

Complete Picture of the Process

Let's use **supervised learning** as our example - it's the most frequently used type of machine learning in practice.

The Goal: Build a system that can automatically detect spam emails

Why This Example?

- Practical and relatable problem
- Shows all key concepts clearly
- Foundation for understanding other ML problems

Step 1: Gathering the Data

Dataset Creation:

- Collect 10,000 email messages
- Add labels manually: "spam" or "not_spam"
- Result: Collection of pairs **(input, output)**

Input: Email messages (text)
Output: Labels ("spam", "not_spam")

Key Point: Outputs can be:

- Real numbers (prices, scores)
- Labels (spam/not_spam, cat/dog)
- Vectors (bounding box coordinates)
- Sequences (part-of-speech tags)

Step 2: Feature Engineering

Problem: Convert email text → machine-readable numbers

Solution: Bag of Words Approach

- Take English dictionary (20,000 words, alphabetically sorted)
- Create feature vector where each position represents one word

Feature Vector Creation:

- Feature 1: 1 if email contains "a", otherwise 0
- Feature 2: 1 if email contains "aaron", otherwise 0
- ...
- Feature 20,000: 1 if email contains "zulu", otherwise 0

Result: Each email → 20,000-dimensional feature vector

Step 3: Label Encoding

Human-readable → Machine-readable

Original Labels: "spam", "not_spam"

Numeric Labels for SVM:

- "spam" → +1 (positive class)
- "not_spam" → -1 (negative class)

Why Transform? Different algorithms have different requirements

- Some need 0/1
- SVM specifically needs +1/-1
- Others might need probability distributions

Step 4: Learning Process

Now We Have:

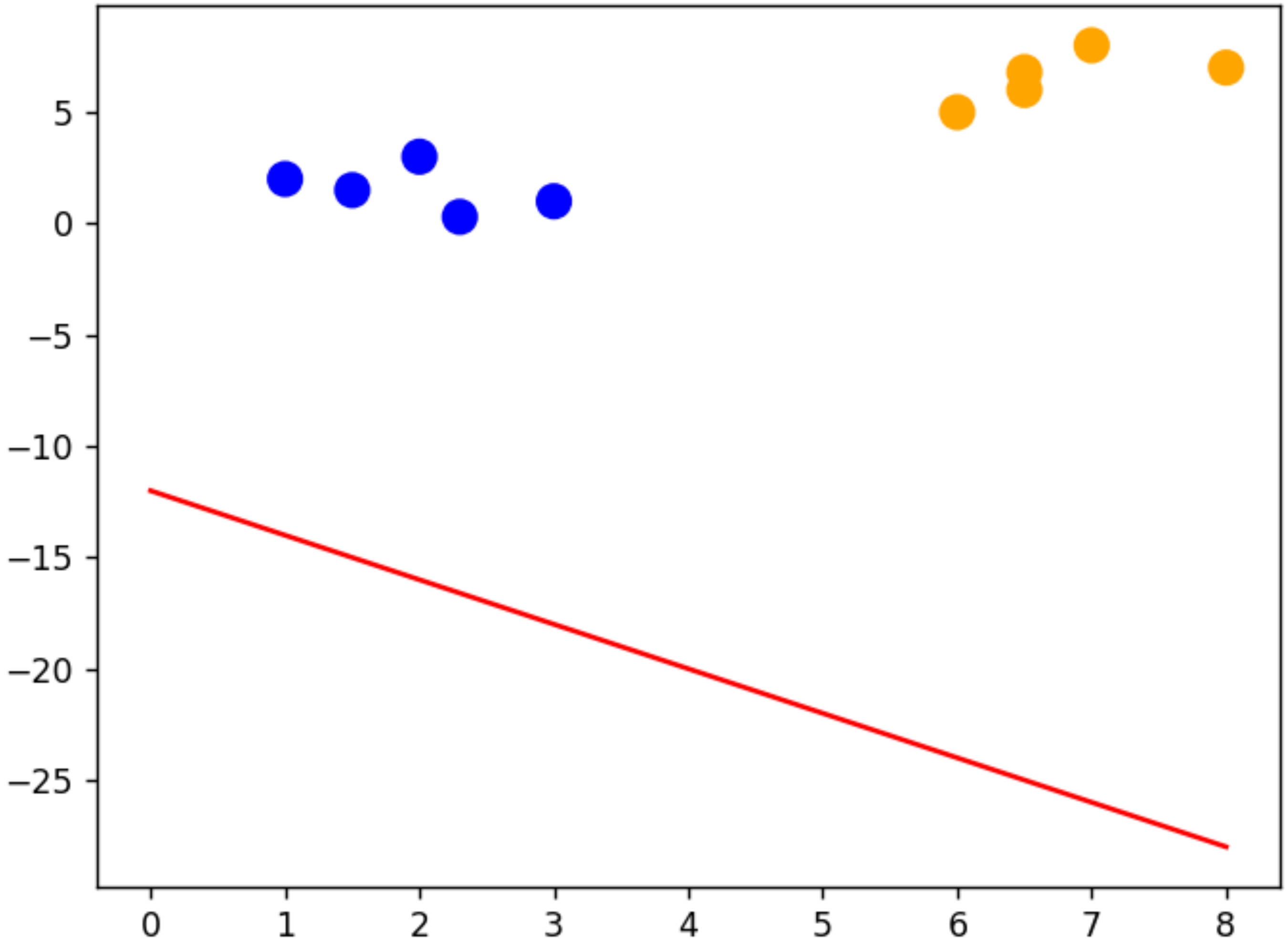
- ✅ **Dataset:** 10,000 feature vectors (20,000-dimensional)
- ✅ **Labels:** +1 (spam) or -1 (not_spam) for each email
- ✅ **Algorithm:** Support Vector Machine (SVM)

Next: Apply learning algorithm to dataset → Get the model

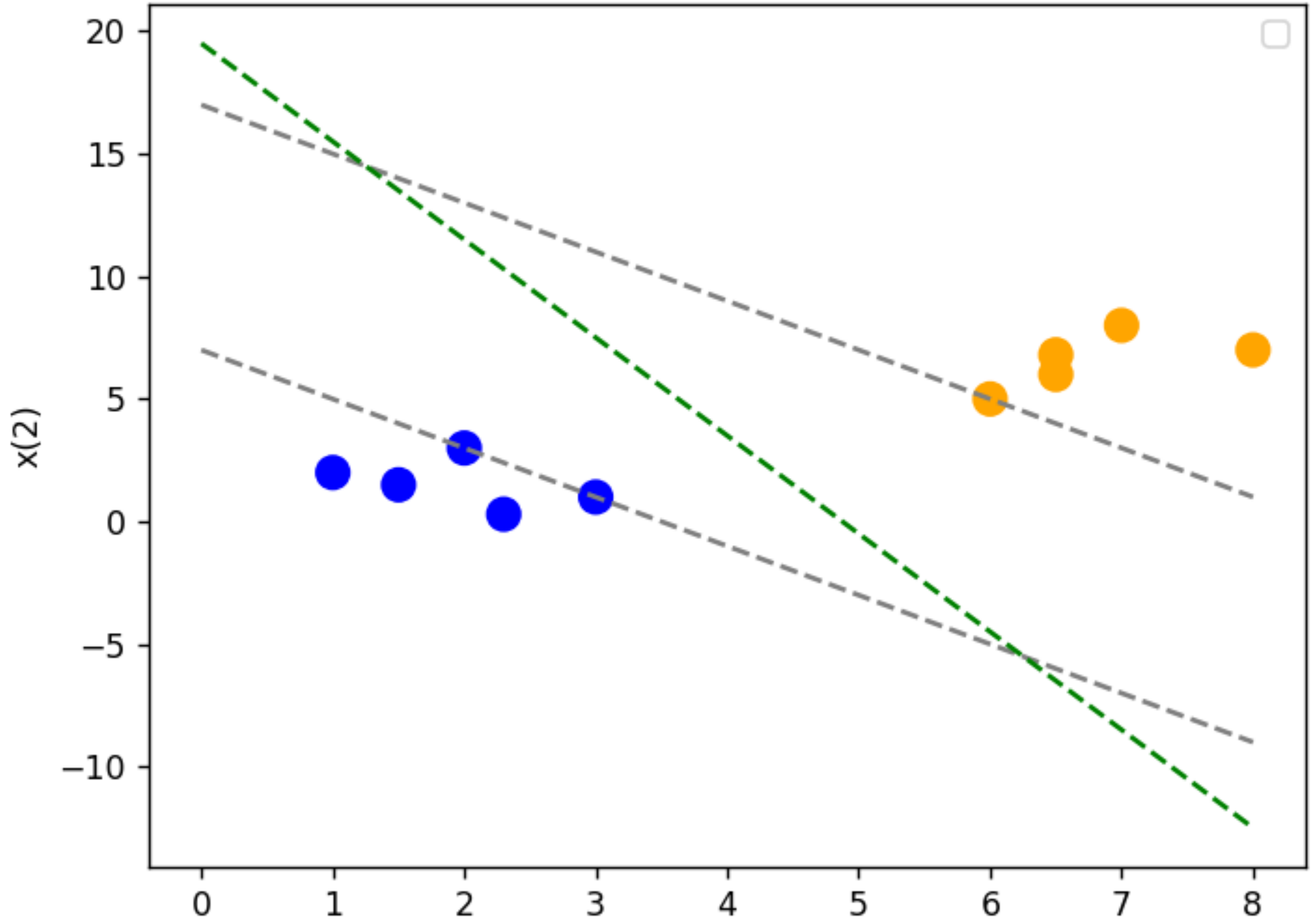
SVM's Approach:

- Sees each feature vector as point in 20,000-dimensional space
- Draws 20,000-dimensional hyperplane to separate classes
- This hyperplane becomes our **decision boundary**

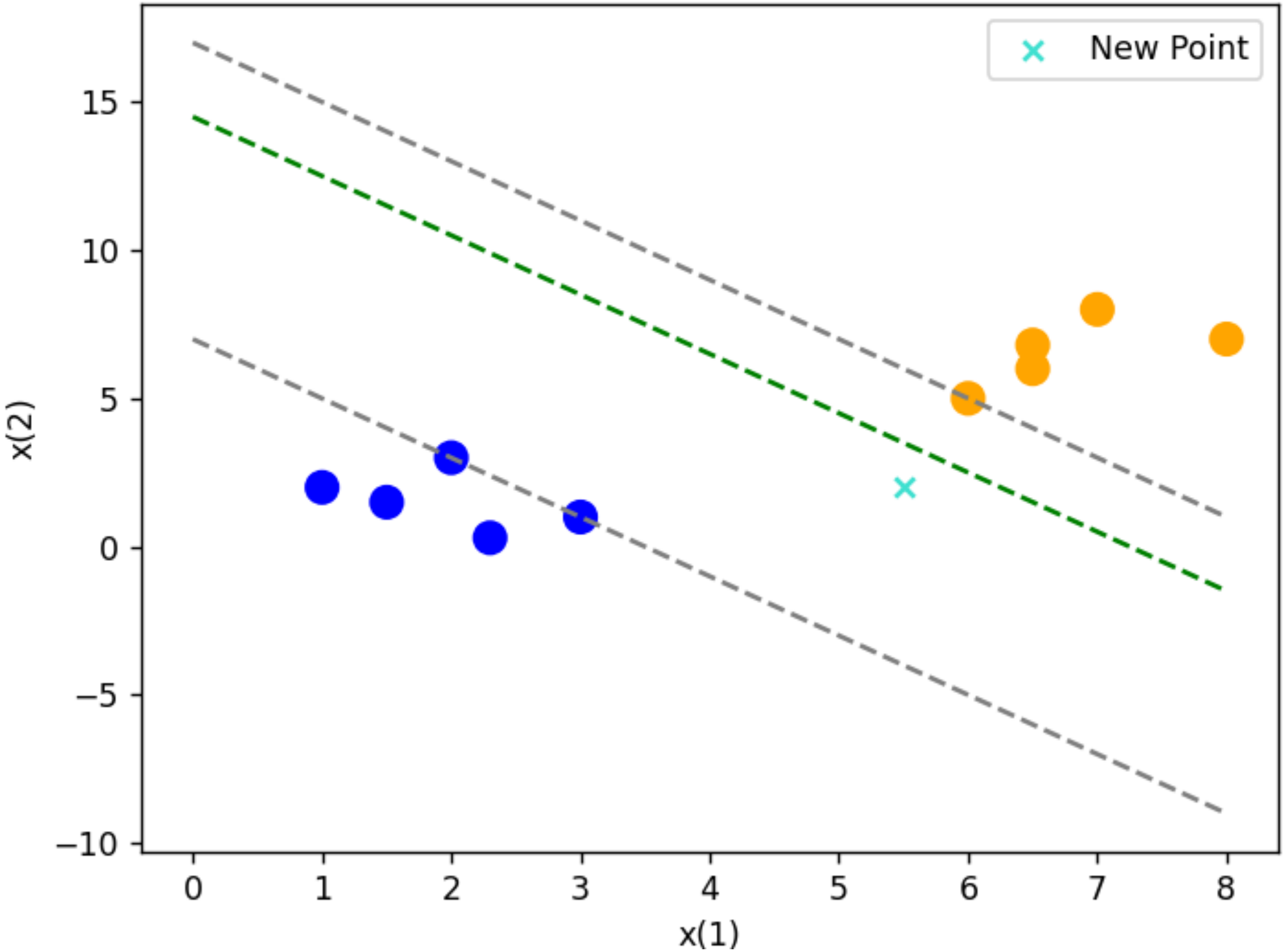
Option 1:



Option 2:



Option 3:



Support Vector Machine (SVM)

A supervised learning algorithm for classification

- **Labels:** Classes (+1 for positive, -1 for negative)
- **Creates:** Decision boundary (hyperplane) to separate data
- **Prefers:** Largest margin for better generalization

Why largest margin?

- Better generalization
- More robust to noise

SVM: Mathematical Foundation

Decision boundary:

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

Prediction:

$$y = \mathrm{sign}(\mathbf{w} \cdot \mathbf{x} - b)$$

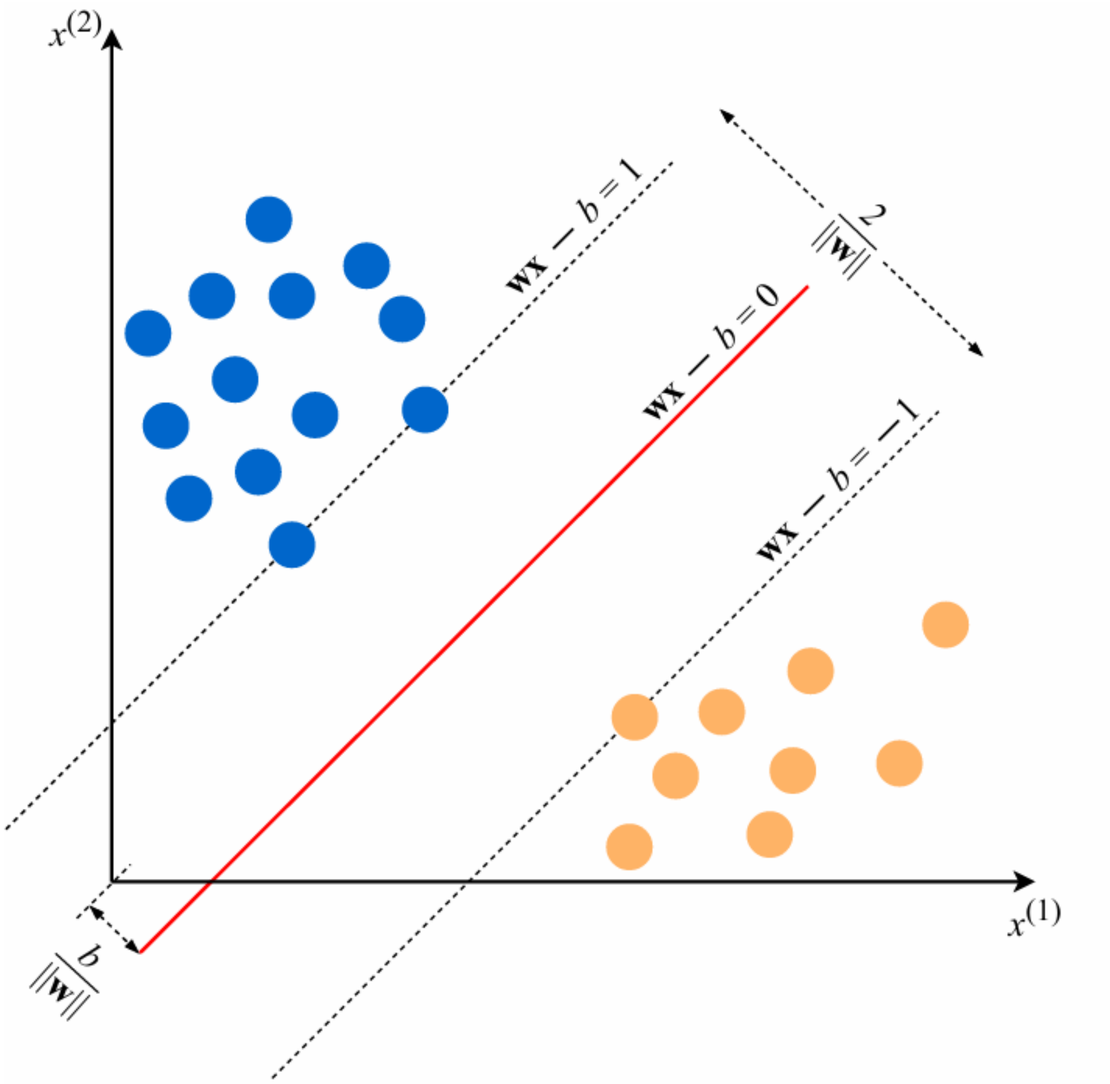
Model:

$$f(x) = \mathrm{sign}(\mathbf{w}^* \cdot \mathbf{x} - b^*)$$

Constraints: $y_i(\mathbf{w} \cdot \mathbf{x} - b) \geq 1$

Objective: Minimize $||\mathbf{w}||$ (maximize margin)

Visual Examples



Key Takeaways - Part 1

Machine Learning = Generalization

- Learn patterns from past data
- Make predictions on new, unseen data
- Success measured by test performance

Four Main Types:

1. **Supervised:** Learn from labeled examples
2. **Unsupervised:** Find patterns without labels
3. **Semi-Supervised:** Use both labeled and unlabeled data
4. **Reinforcement:** Learn through interaction

Key Takeaways - Part 2

Multiple Problem Types:

- **Regression:** Predict numbers
- **Classification:** Predict categories
- **Ranking:** Order by relevance

Golden Rule:

Never peek at test data during training!

Error Measurement Matters:

- Different problems need different error measures
- Regression: Distance from true value
- Classification: Correct/incorrect
- Ranking: Position matters

References

- The Elements of Statistical Learning
- Mathematics for Machine Learning
- Introduction to Machine Learning with Python
- A Course in Machine Learning