

Supervised Learning, Simple Linear Regression, SVM

Dr. Mohamed Mostafa

AI 511 Foundations of Artificial Intelligence

Lecture 10





Outline

- **Machine Learning Regression**
 - Simple Regression Model
 - Multiple Linear Regression
- **Support Vector Machine**
- **Underfitting and Overfitting**

Resources for this lecture

Book "Business Statistics", 7th edition, David M. Levine, et al.

.. AND ..

Andrew W. Moore, School of Computer Science, Carnegie Mellon University,

- Regression is a supervised machine learning technique which is used to predict continuous values.
- The ultimate goal of the regression algorithm is to plot a best-fit line or a curve between the data.
- Based on the number of input features and output labels, regression is classified as linear (one input and one output), multiple (many inputs and one output) and multivariate (many outputs).

- Ex. The primary objective while using linear regression is to locate the best-fit line, which implies that the error between the predicted and actual values should be kept to a minimum. There will be the least error in the best-fit line.
- The tuning of coefficient and intercept is achieved through gradient descent or a cost function — least squares method.

- **Regression analysis is used to:**

- Predict the value of a dependent variable based on the value of at least one independent variable
 - Explain the impact of changes in an independent variable on the dependent variable
- **Dependent variable:** the variable we wish to predict or explain
- **Independent variable:** the variable used to predict or explain the dependent variable

The diagram illustrates the Simple Regression Model equation: $Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$. Red arrows point from descriptive labels to the corresponding parts of the equation. A purple bracket under $\beta_0 + \beta_1 X_i$ is labeled 'Linear component', and another purple bracket under ε_i is labeled 'Random Error component'.

Population Y intercept (Constant)

Population Slope Coefficient

Independent Variable (IV)

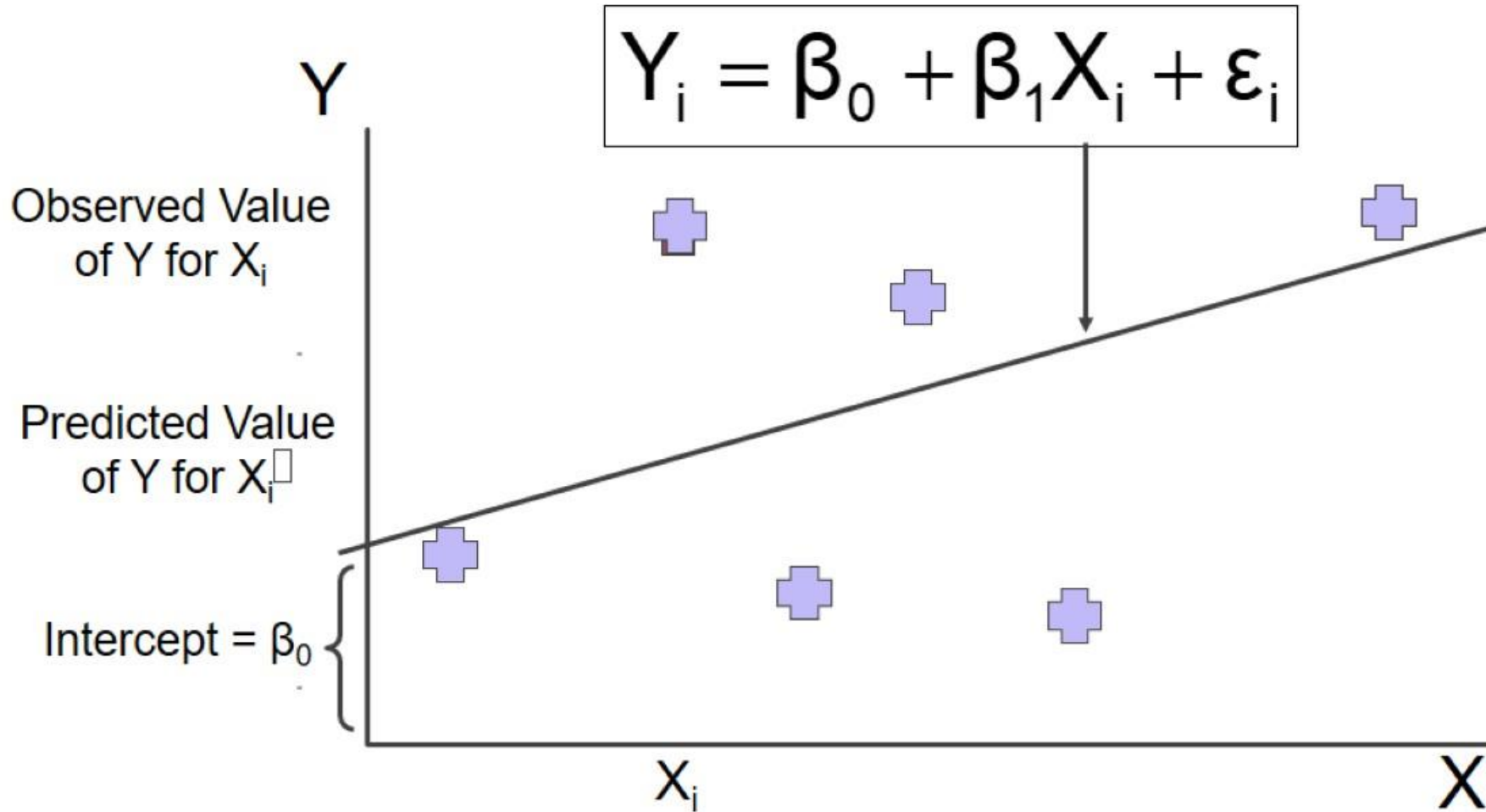
Random Error term

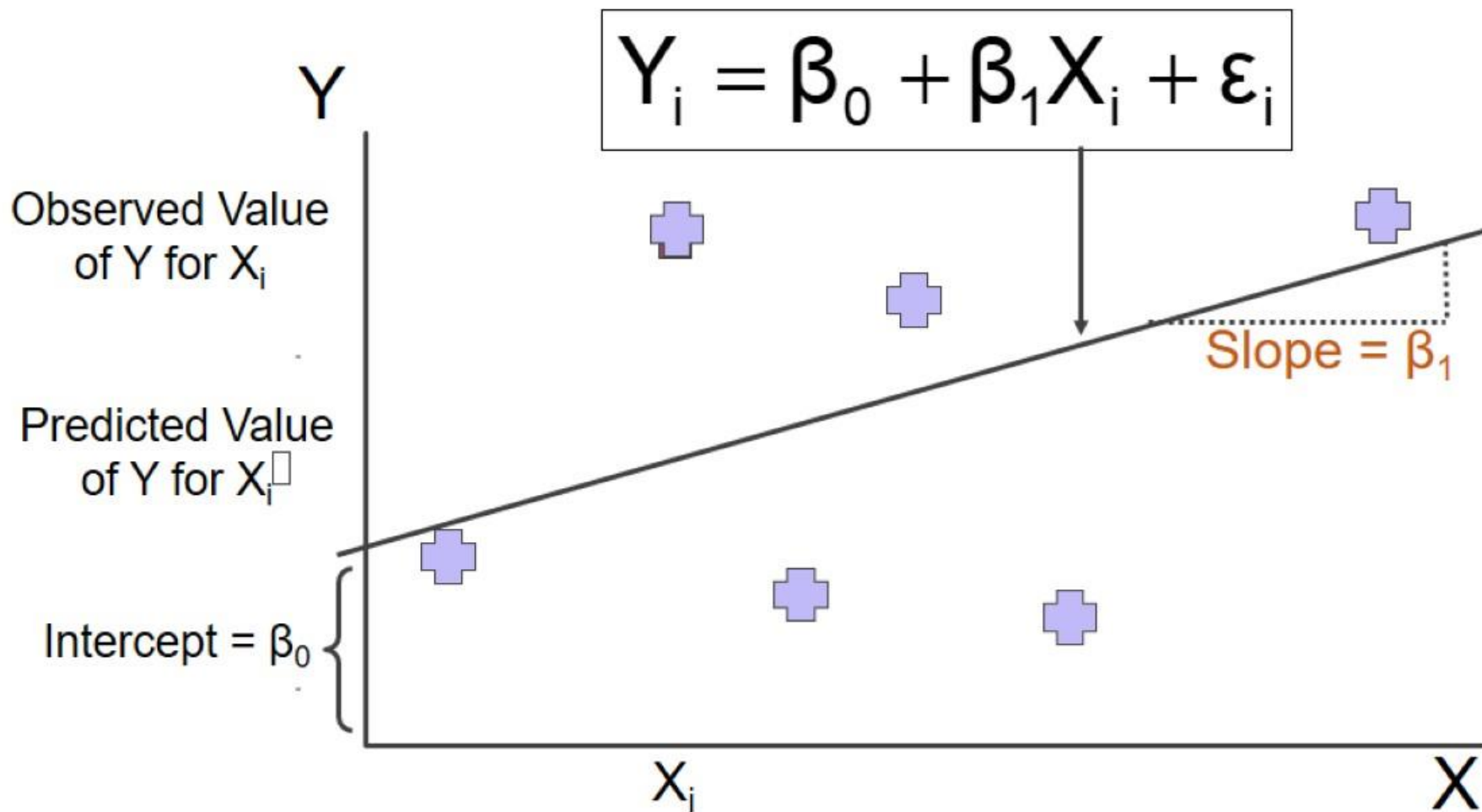
Dependent Variable (DV)

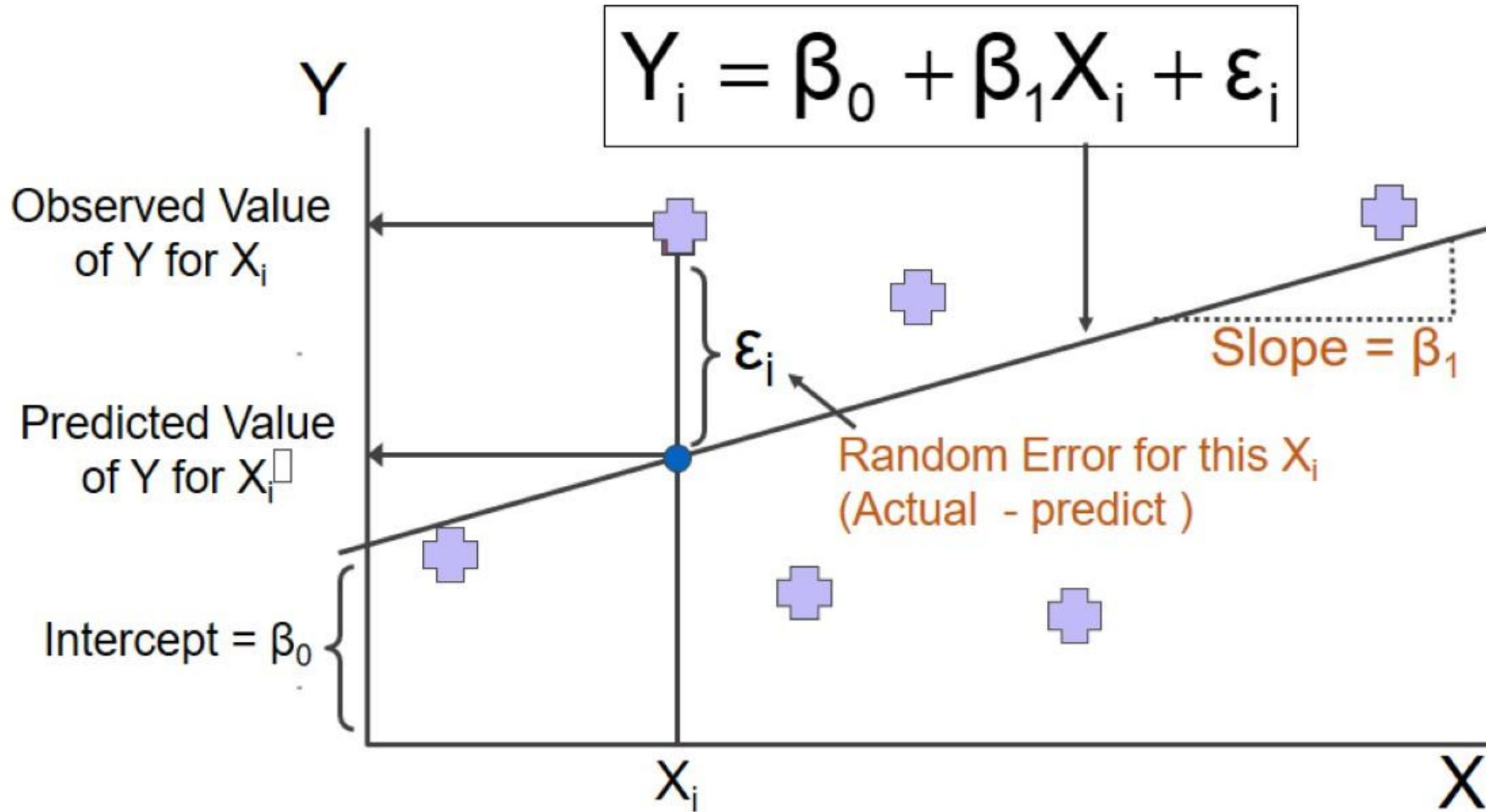
$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Linear component

Random Error component







Simple Regression Model (Prediction Line)

- The simple linear regression equation provides an estimate of the population regression line

Estimated
(or predicted)
Y value for
observation i

Estimate of
the regression
intercept

Estimate of the
regression slope

Value of X for
observation i

$$\hat{Y}_i = b_0 + b_1 X_i$$

Simple Regression Model (Prediction Line)

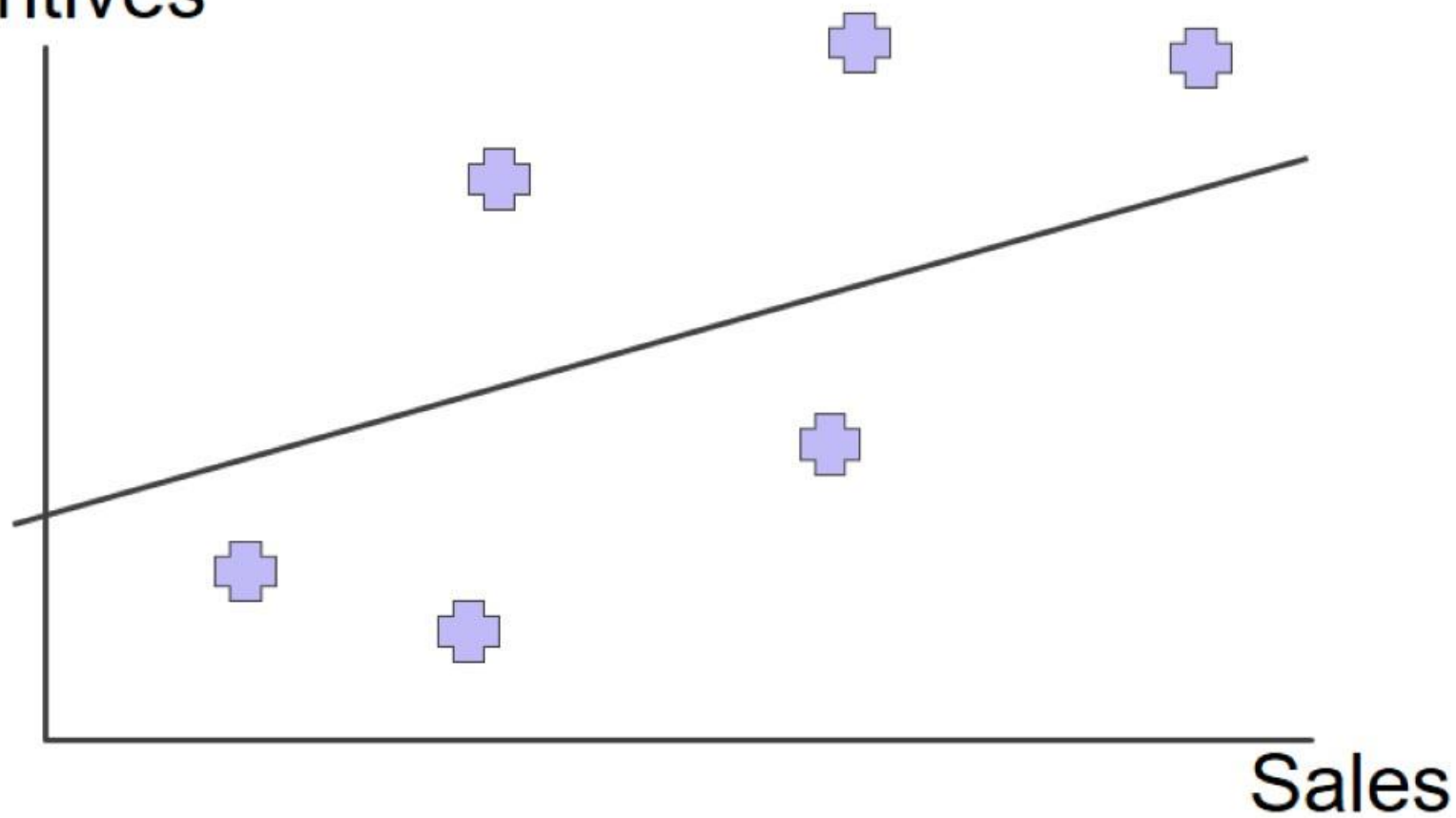
- b_0 is the estimated mean value of Y when the value of X is zero.

For example; when there is no sales the incentives will be 2K EGP.

- b_1 is the estimated change in the mean value of Y as a result of a one-unit increase in X

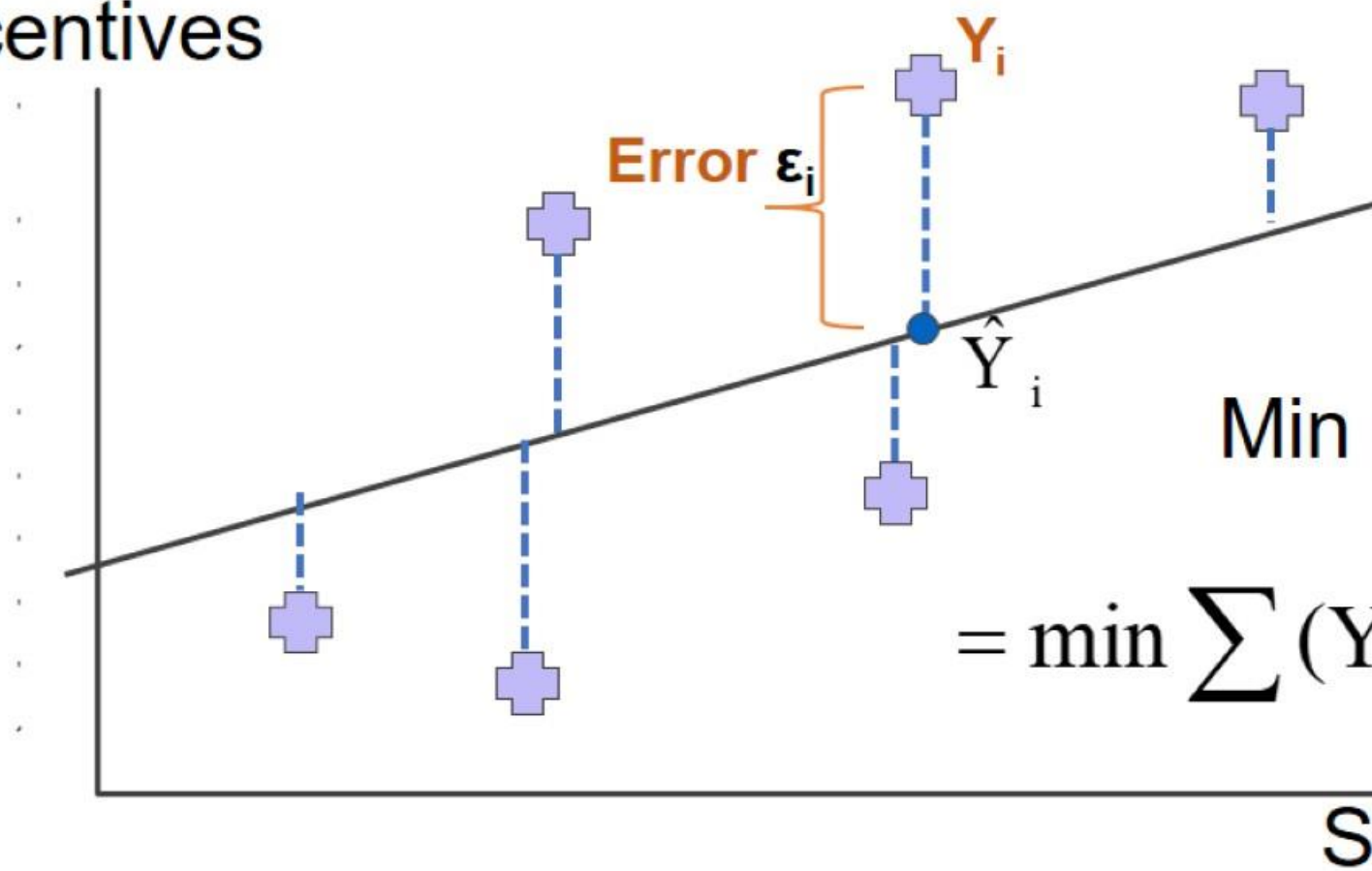
The Least Squares Method

Incentives



The Least Squares Method

Incentives



$$\text{Min } (\text{Sum}(Y - \hat{Y})^2)$$

$$= \min \sum (Y_i - (b_0 + b_1 X_i))^2$$

Simple Linear Regression Example

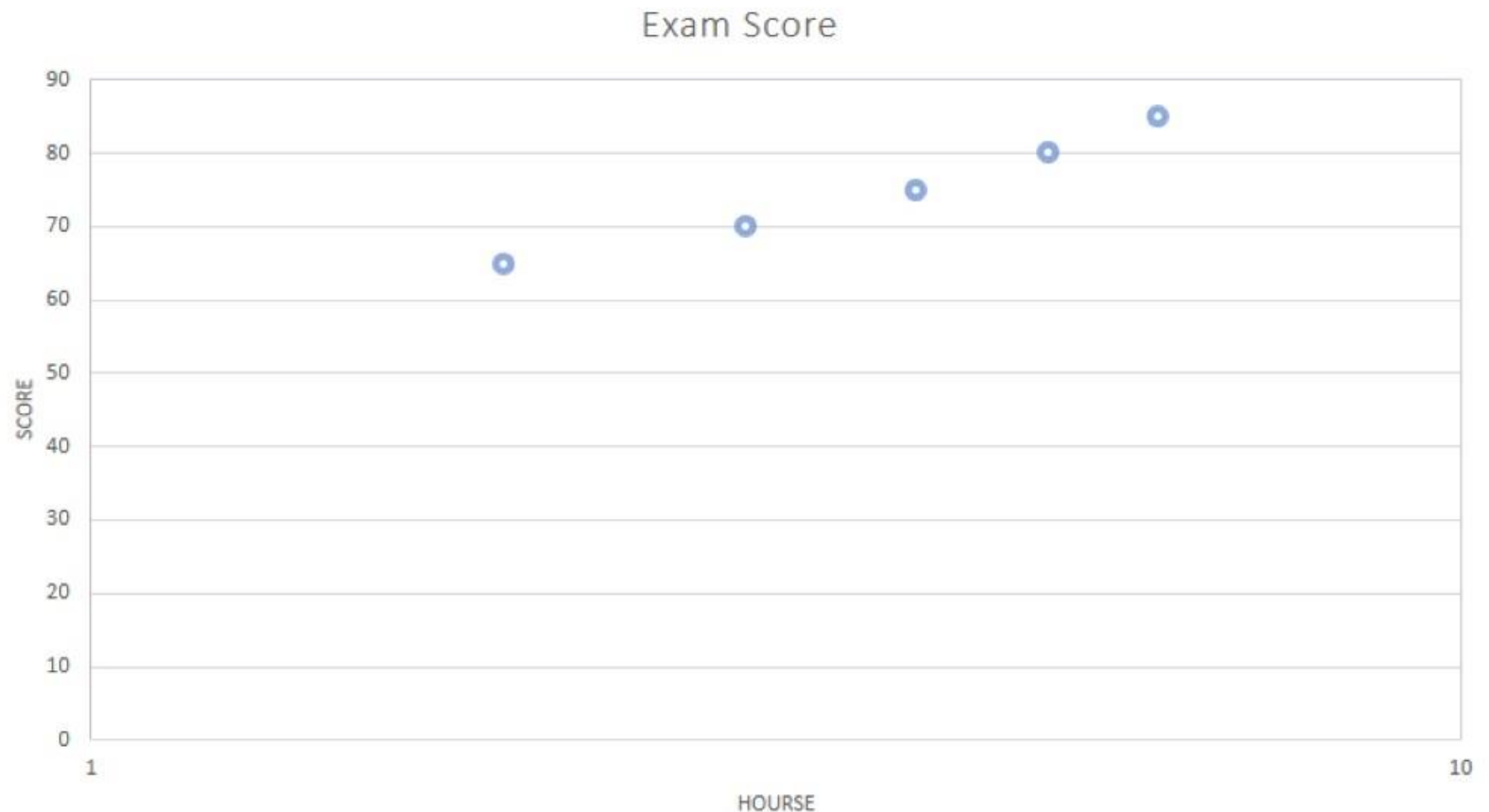
- Suppose we have a dataset containing information about the number of hours studied by students and their corresponding exam scores. Here's a hypothetical dataset:

Hours Studied	Exam Score
2	65
3	70
4	75
5	80
6	85

Simple Linear Regression Example

- Scatter Plot

Hours Studied	Exam Score
2	65
3	70
4	75
5	80
6	85



Simple Linear Regression Example

- we'll fit a line to the data that best represents the relationship between the number of hours studied and exam scores. The equation of a simple linear regression model is:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Simple Linear Regression Example

- Let's try to estimate the regression coefficients B_0 and B_1

$$\beta_1 = \frac{n(\sum XY) - (\sum X)(\sum Y)}{n(\sum X^2) - (\sum X)^2}$$

$$\beta_0 = \frac{\sum Y - \beta_1(\sum X)}{n}$$

Where:

- n is the number of data points.
- $\sum X$ and $\sum Y$ represent the sum of the hours studied and exam scores, respectively.
- $\sum XY$ represents the sum of the products of hours studied and exam scores.
- $\sum X^2$ represents the sum of the squared hours studied.

Simple Linear Regression Example

- Let's try to estimate the regression coefficients B_0 and B_1

$$\beta_1 = \frac{n(\sum XY) - (\sum X)(\sum Y)}{n(\sum X^2) - (\sum X)^2}$$

$$\beta_0 = \frac{\sum Y - \beta_1(\sum X)}{n}$$

Where:

- n is the number of data points.
- $\sum X$ and $\sum Y$ represent the sum of the hours studied and exam scores, respectively.
- $\sum XY$ represents the sum of the products of hours studied and exam scores.
- $\sum X^2$ represents the sum of the squared hours studied.

Simple Linear Regression Example

- Let's calculate these coefficients using our dataset:
- $n=5$
- $\sum X=2+3+4+5+6=20$
- $\sum Y=65+70+75+80+85=375$
- $\sum XY=(2*65)+(3*70)+(4*75)+(5*80)+(6*85)=2005$
- $\sum X^2=(2^2)+(3^2)+(4^2)+(5^2)+(6^2)=90$

Simple Linear Regression Example

- Let's calculate the coefficients

$$\beta_1 = \frac{(5*2005)-(20*375)}{(5*90)-(20)^2}$$

$$\beta_1 \approx 5$$

$$\beta_0 = \frac{375 - \beta_1(20)}{5}$$

$$\beta_0 \approx 55$$

Simple Linear Regression Example

- Then the estimated regression equation is:

$$\text{Exam Exam Score} = 55 + 5 \times \text{Hours Studied}$$

we use the previous equation to predict exam scores for new values of hours studied. For example, if a student studies for 7 hours:

$$\text{Exam Score} = 55 + 5 \times 7 = 90$$

Multiple Linear Regression

- This involves more than one independent variable and one dependent variable.

$$y = \beta_0 + \beta_1 X + \beta_2 X + \dots \dots \dots \beta_n X$$

where:

- Y is the dependent variable
- $X_1, X_2, \dots, X_p$ are the independent variables
- β_0 is the intercept
- $\beta_1, \beta_2, \dots, \beta_n$ are the slopes

Practice



Predicting House Prices Based on Square Footage

Suppose you're a real estate agent trying to help clients determine the selling price of their homes. One of the factors that significantly influences the price of a house is its square footage. You decide to **use linear regression to build a model that predicts house prices based on square footage.**

Square Footage (sqft)	Selling Price (\$)
1500	200,000
1800	220,000
2000	250,000
2200	270,000
2500	300,000
2700	320,000
3000	350,000
3200	370,000
3500	400,000
3800	420,000

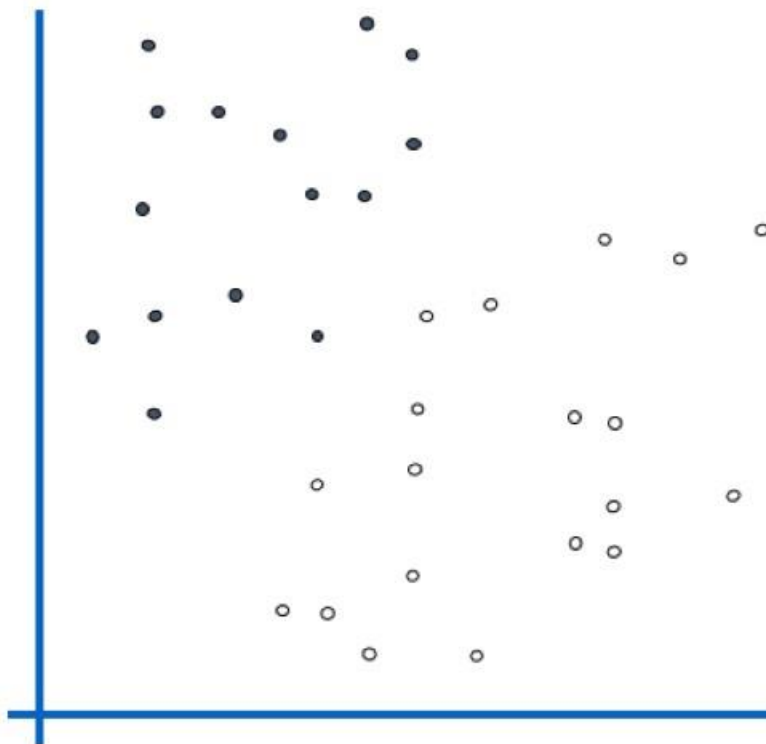
Support Vector Machine Algorithm

Support Vector Machine (SVM)

- Support Vector Machine (SVM) is a powerful machine learning algorithm used for linear or nonlinear classification, regression, and even outlier detection tasks.
- SVMs can be used for a variety of tasks, such as text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection.
- SVMs are adaptable and efficient in a variety of applications because they can manage high-dimensional data and nonlinear relationships.
- SVM algorithms are very effective as we try to find the maximum separating hyperplane between the different classes available in the target feature.

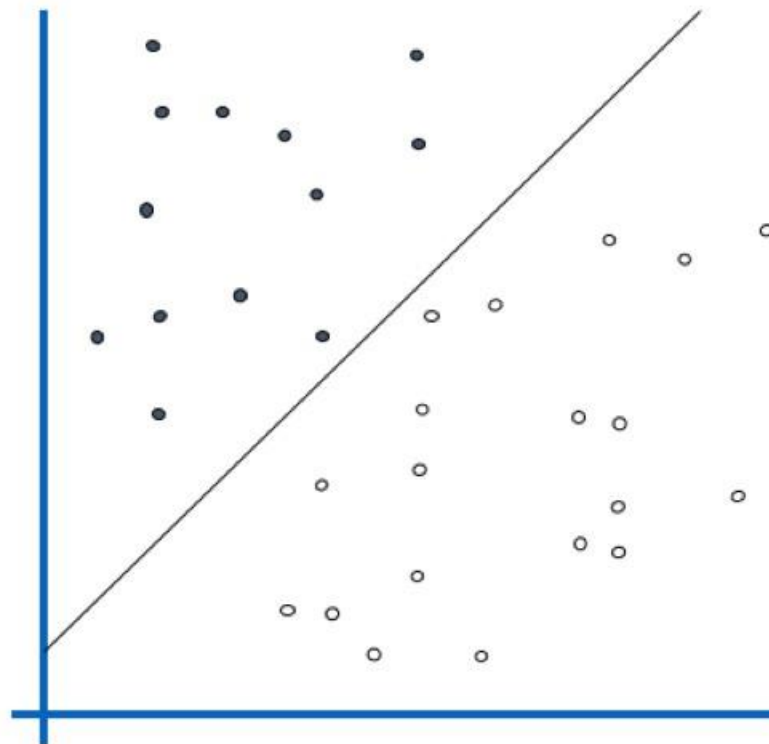
How would you classify this data?

- denotes +1
- denotes -1

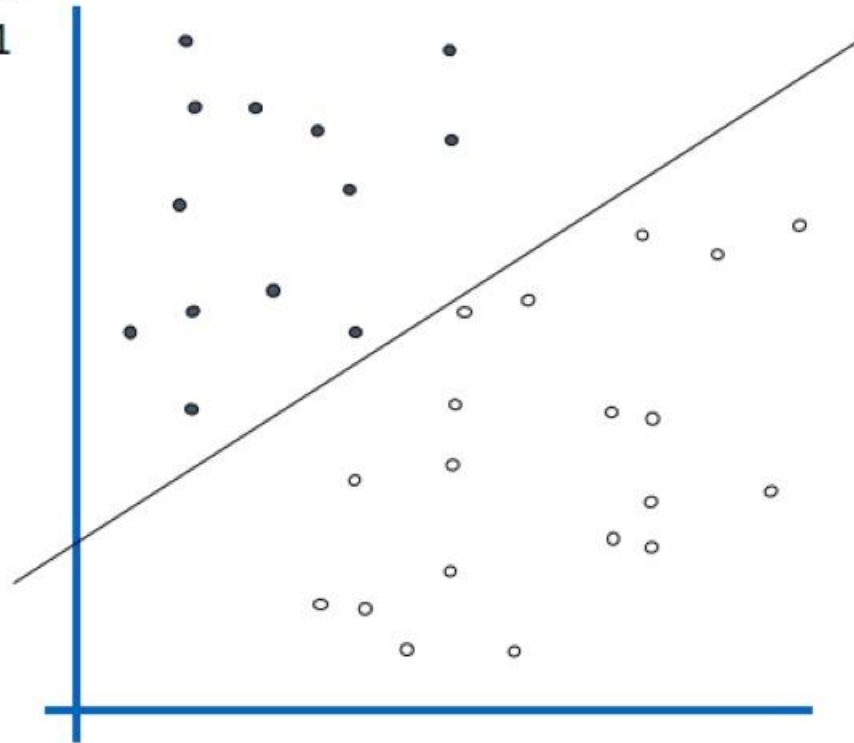


Linear Classifiers

- denotes +1
- denotes -1

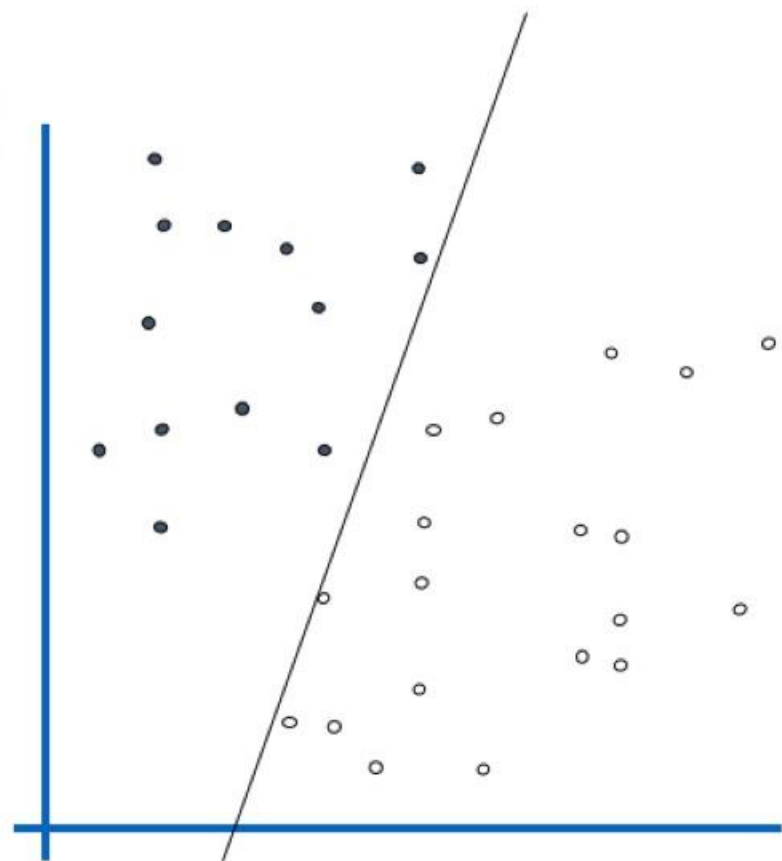


- denotes +1
- denotes -1



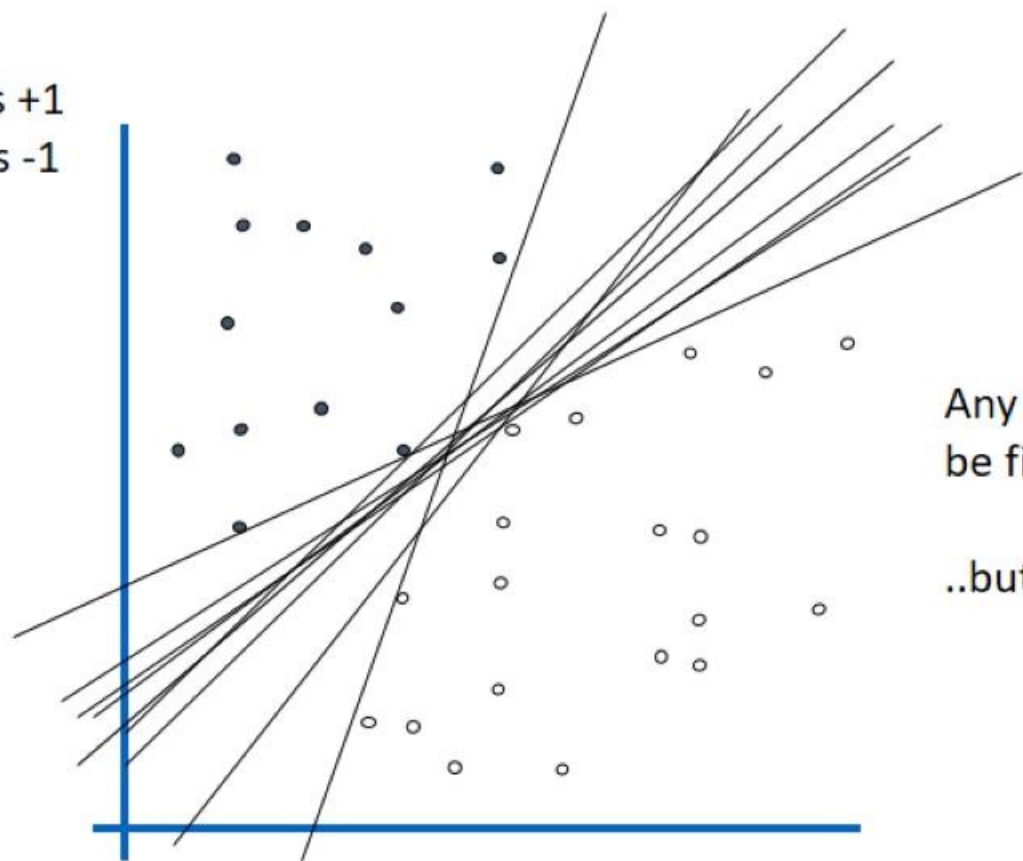
Linear Classifiers

- denotes +1
- denotes -1



Linear Classifiers

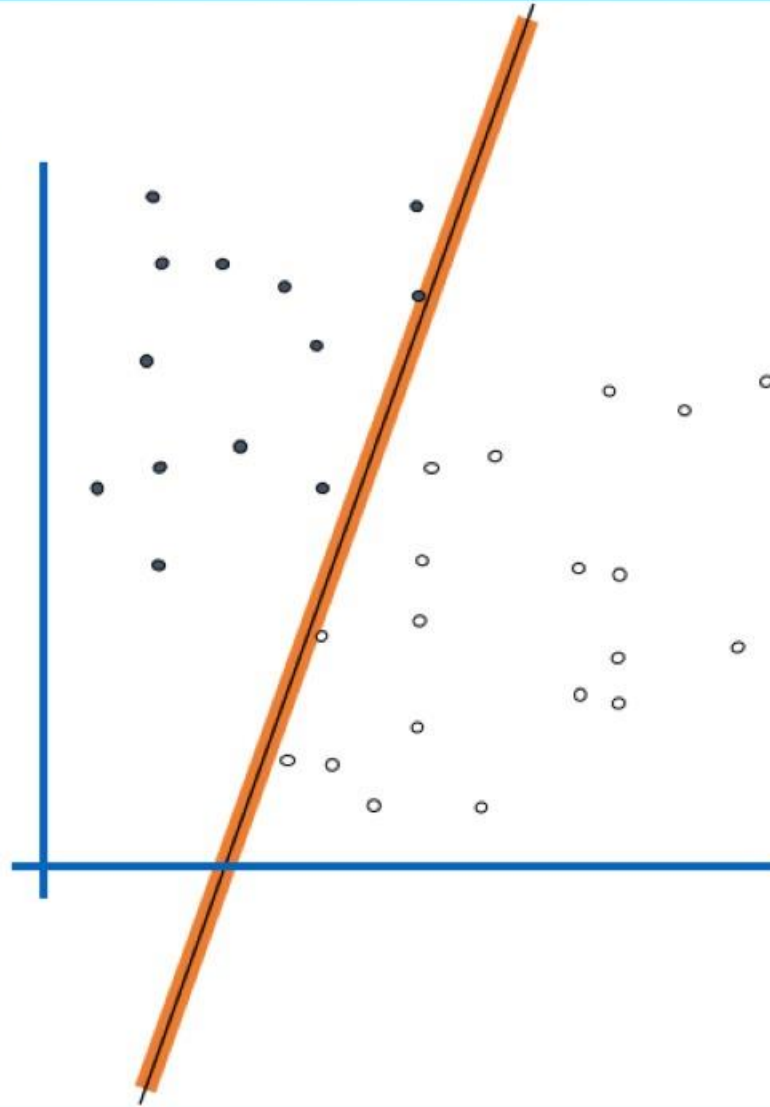
- denotes +1
- denotes -1



Any of these would
be fine..

..but which is best?

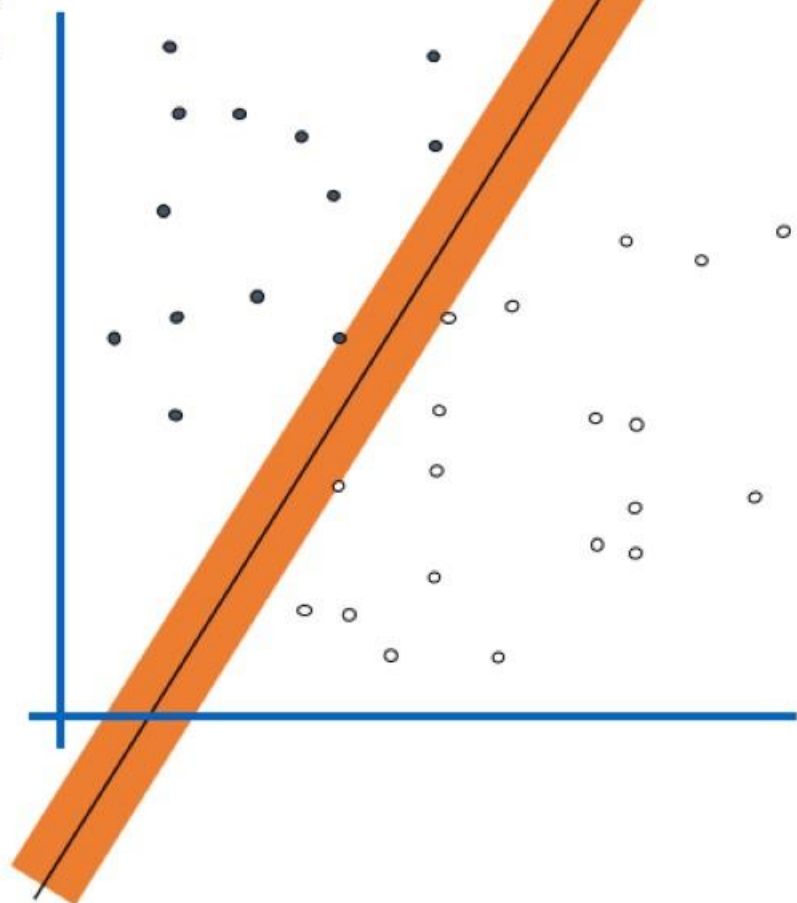
- denotes +1
- denotes -1



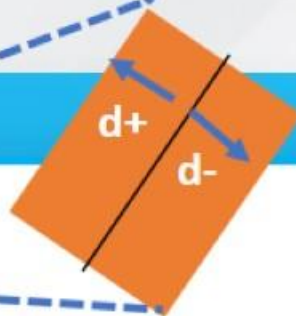
Define the **margin** of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

Maximum Margin

- denotes +1
- denotes -1

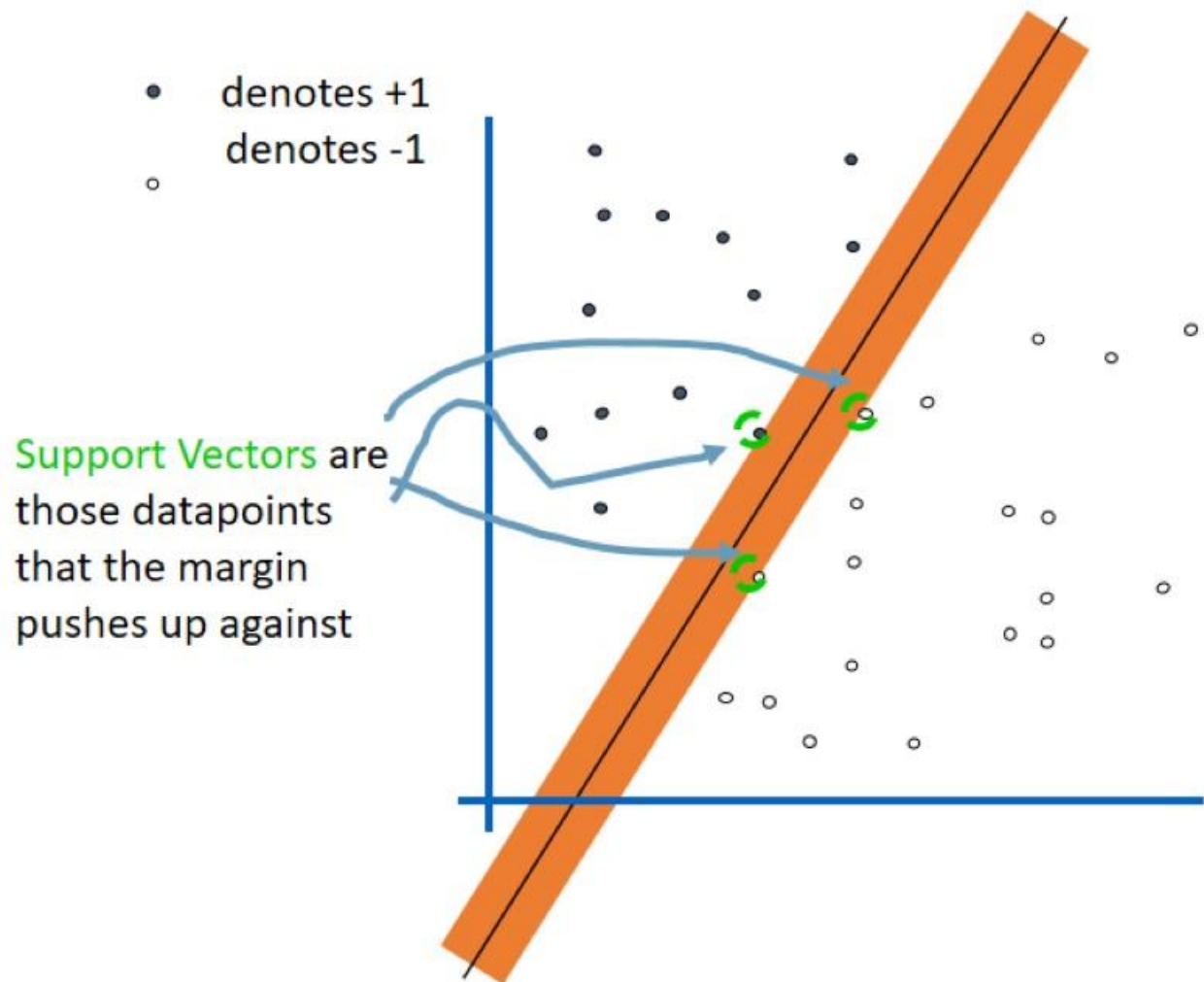


The maximum margin linear classifier is the linear classifier with the, um, maximum margin.
This is the simplest kind of SVM (Called an Linear SVM)



Marginal Distance

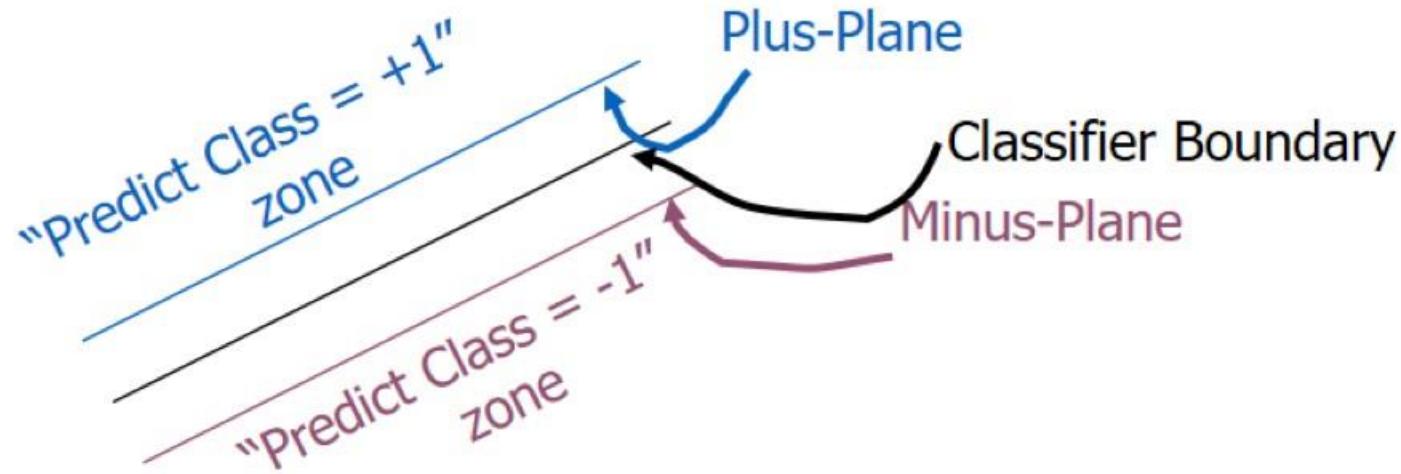
Maximum Margin



The maximum margin linear classifier is the linear classifier with the, um, maximum margin.

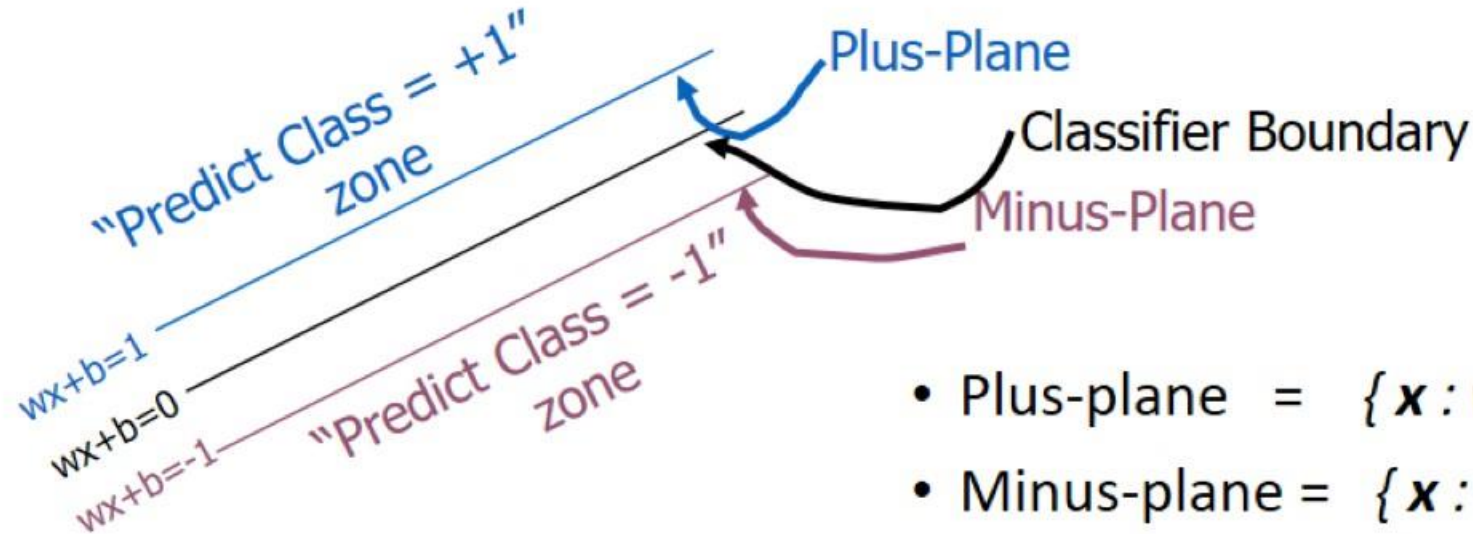
This is the simplest kind of SVM (Called an Linear SVM)

Specifying a line and margin



- How do we represent this mathematically?
- ...in m input dimensions?

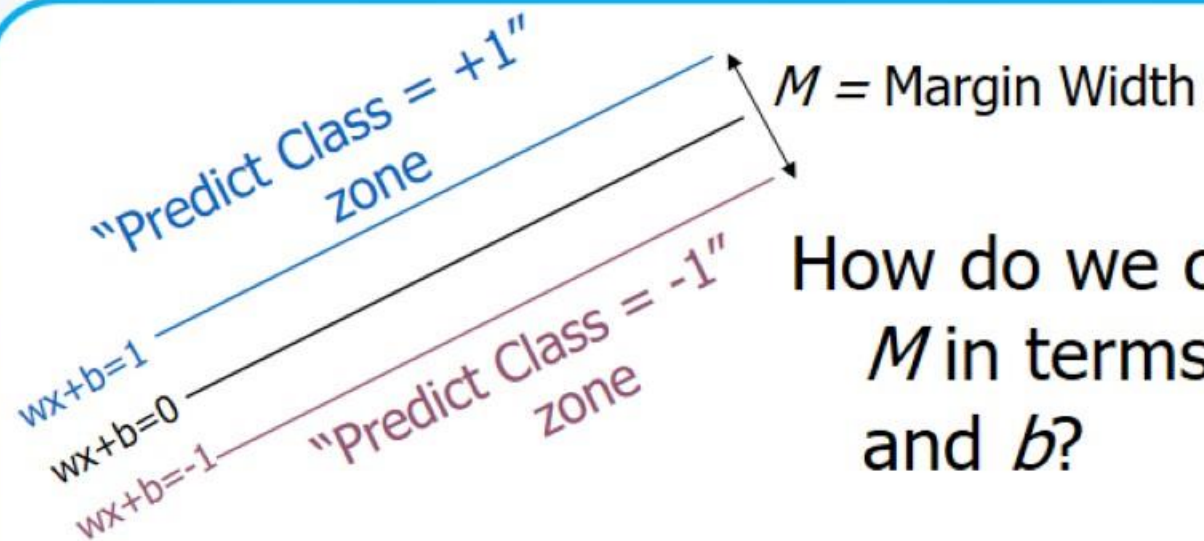
Specifying a line and margin



- Plus-plane = $\{\mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = +1\}$
- Minus-plane = $\{\mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = -1\}$

Classify as.. $+1$ if $\mathbf{w} \cdot \mathbf{x} + b \geq 1$
 -1 if $\mathbf{w} \cdot \mathbf{x} + b \leq -1$

Specifying a line and margin

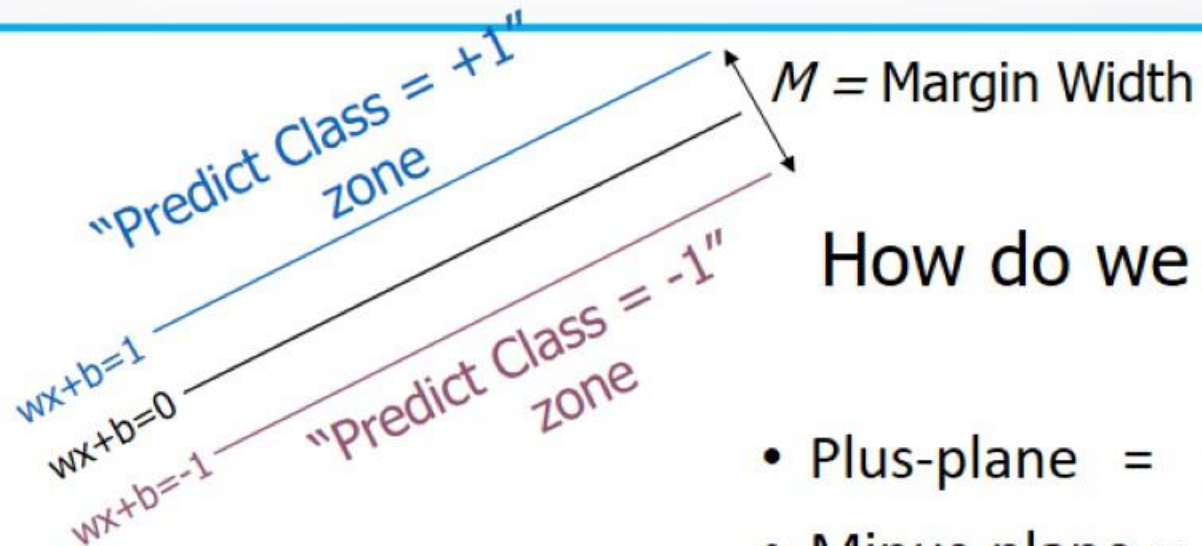


How do we compute M in terms of $\mathbf{w}$ and b ?

- Plus-plane = $\{\mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = +1\}$
- Minus-plane = $\{\mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = -1\}$

Claim: The vector $\mathbf{w}$ is perpendicular to the Plus Plane. **Why?**

Specifying a line and margin



How do we compute M in terms of $\mathbf{w}$ and b ?

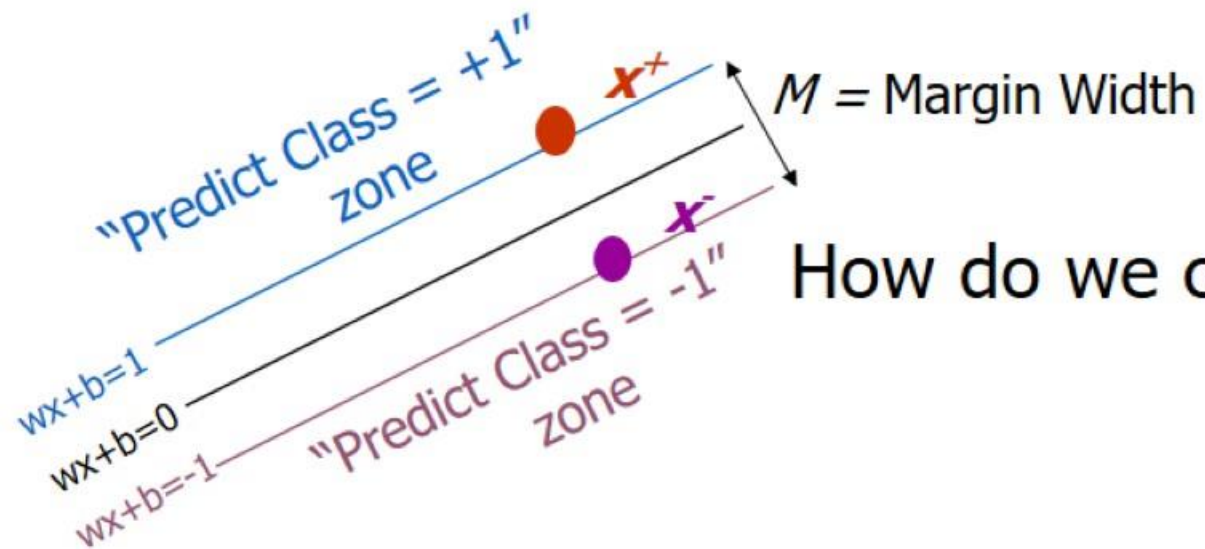
- Plus-plane = $\{ \mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = +1 \}$
- Minus-plane = $\{ \mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = -1 \}$

Claim: The vector $\mathbf{w}$ is perpendicular to the Plus Plane. **Why?**

Let $\mathbf{u}$ and $\mathbf{v}$ be two vectors on the Plus Plane. What is $\mathbf{w} \cdot (\mathbf{u} - \mathbf{v})$?

And so of course the vector $\mathbf{w}$ is also perpendicular to the Minus Plane

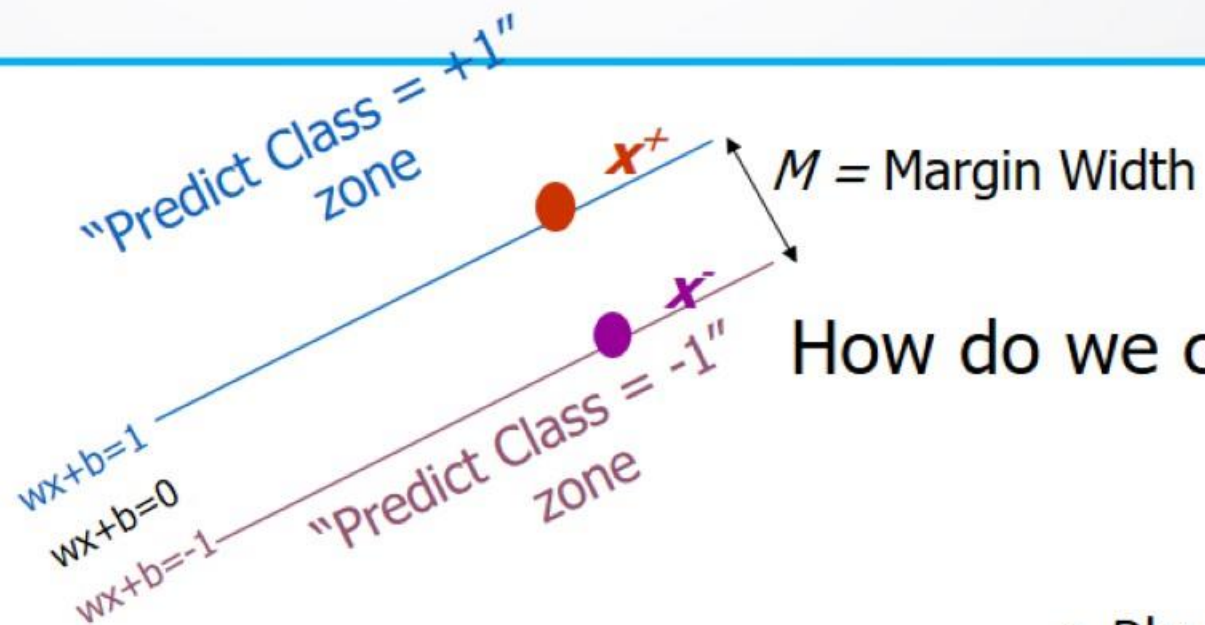
Specifying a line and margin



How do we compute M in terms of $\mathbf{w}$ and b ?

- Plus-plane = $\{\mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = +1\}$
- Minus-plane = $\{\mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = -1\}$
- The vector $\mathbf{w}$ is perpendicular to the Plus Plane
- Let $\mathbf{x}^-$ be any point on the minus plane
- Let $\mathbf{x}^+$ be the closest plus-plane-point to $\mathbf{x}^-$.

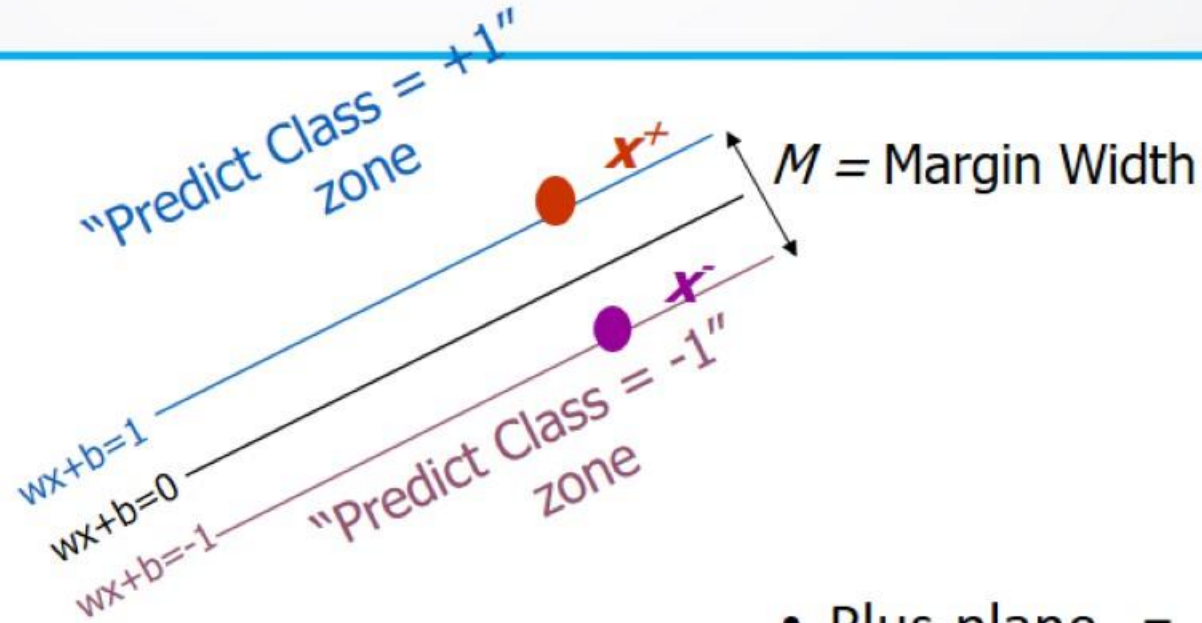
Specifying a line and margin



How do we compute M in terms of $\mathbf{w}$ and b ?

- Plus-plane = $\{\mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = +1\}$
- Minus-plane = $\{\mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = -1\}$
- The vector $\mathbf{w}$ is perpendicular to the Plus Plane
- Let $\mathbf{x}^-$ be any point on the minus plane
- Let $\mathbf{x}^+$ be the closest plus-plane-point to $\mathbf{x}^-$.
- **Claim:** $\mathbf{x}^+ = \mathbf{x}^- + \lambda \mathbf{w}$ for some value of λ . **Why?**

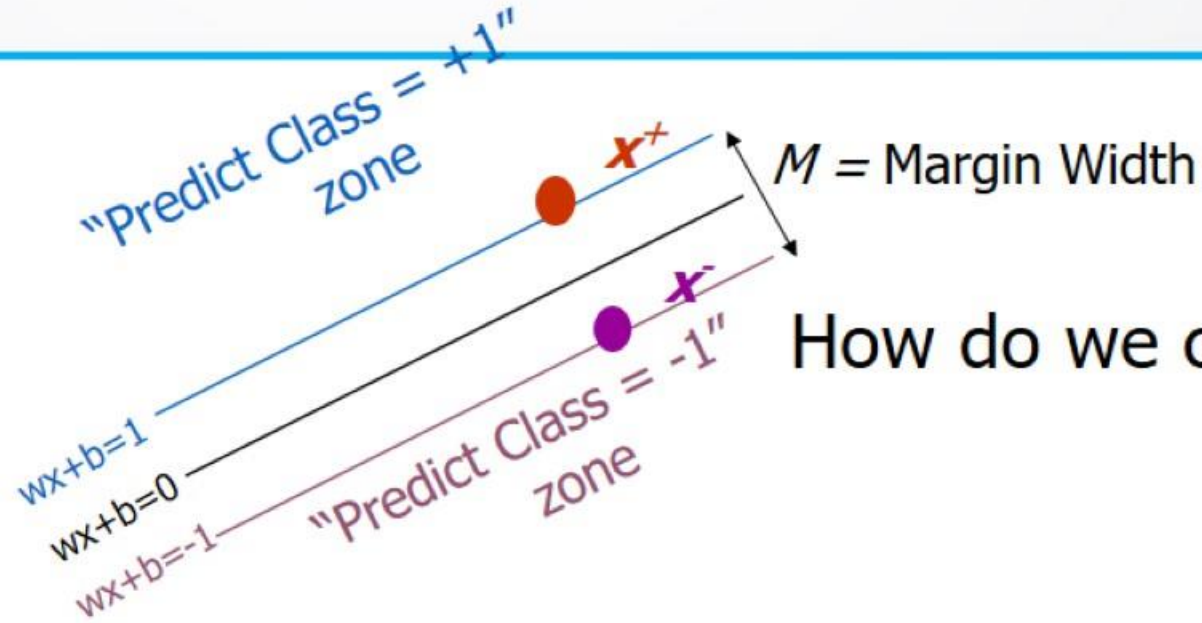
Specifying a line and margin



The line from $\mathbf{x}$ to $\mathbf{x}^+$ is perpendicular to the planes. So to get from $\mathbf{x}$ to $\mathbf{x}^+$ travel some distance in direction $\mathbf{w}$.

- Plus-plane = $\{\mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = +1\}$
- Minus-plane = $\{\mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = -1\}$
- The vector $\mathbf{w}$ is perpendicular to the Plus Plane
- Let $\mathbf{x}$ be any point on the minus plane
- Let $\mathbf{x}^+$ be the closest plus-plane-point to $\mathbf{x}$.
- **Claim:** $\mathbf{x}^+ = \mathbf{x} + \lambda \mathbf{w}$ for some value of λ . **Why?**

Specifying a line and margin



How do we compute M in terms of $\mathbf{w}$ and b ?

What we know:

- $\mathbf{w} \cdot \mathbf{x}^+ + b = +1$
- $\mathbf{w} \cdot \mathbf{x}^- + b = -1$
- $\mathbf{x}^+ = \mathbf{x}^- + \lambda \mathbf{w}$
- $\underline{|\mathbf{x}^+ - \mathbf{x}^-|} = M$

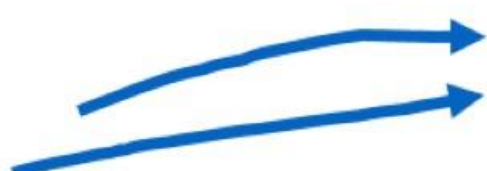
It's now easy to get M in terms of $\mathbf{w}$ and b

Specifying a line and margin

What we know:

- $\mathbf{w} \cdot \mathbf{x}^+ + b = +1$
- $\mathbf{w} \cdot \mathbf{x}^- + b = -1$
- $\mathbf{x}^+ = \mathbf{x}^- + \lambda \mathbf{w}$
- $|\mathbf{x}^+ - \mathbf{x}^-| = M$

It's now easy to get M in terms of $\mathbf{w}$ and b



$$\mathbf{w} \cdot (\mathbf{x}^- + \lambda \mathbf{w}) + b = 1$$

$\Rightarrow$

$$\mathbf{w} \cdot \mathbf{x}^- + b + \lambda \mathbf{w} \cdot \mathbf{w} = 1$$

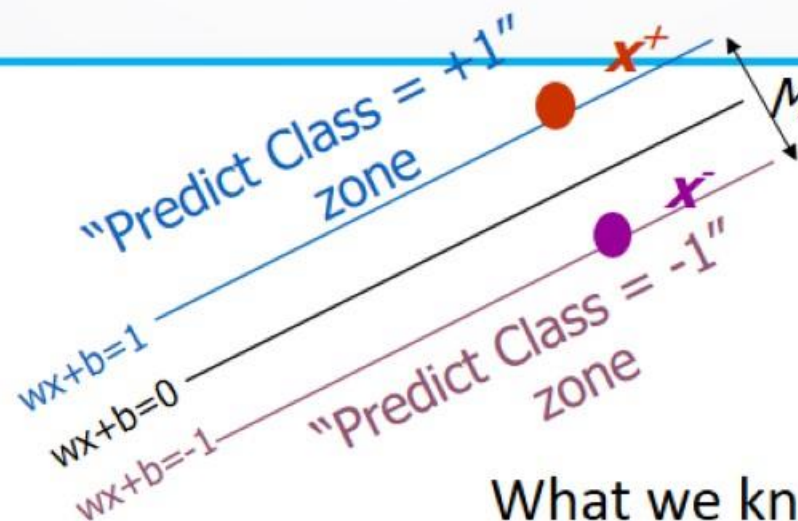
$\Rightarrow$

$$-1 + \lambda \mathbf{w} \cdot \mathbf{w} = 1$$

$\Rightarrow$

$$\lambda = \frac{2}{\mathbf{w} \cdot \mathbf{w}}$$

Specifying a line and margin



What we know:

- $w \cdot x^+ + b = +1$
- $w \cdot x^- + b = -1$
- $x^+ = x^- + \lambda w$
- $|x^+ - x^-| = M$
- $\lambda = \frac{2}{w \cdot w}$

$$M = \text{Margin Width} = \frac{\text{Max } 2}{\sqrt{w \cdot w}} = \frac{2}{\|w\|_2}$$

$$\begin{aligned} M &= |x^+ - x^-| = |\lambda w| = \\ &= \lambda |w| = \lambda \sqrt{w \cdot w} \\ &= \frac{2\sqrt{w \cdot w}}{w \cdot w} = \frac{2}{\sqrt{w \cdot w}} \end{aligned}$$

$$\begin{aligned} xw &= \begin{bmatrix} 3 \\ 4 \end{bmatrix} \\ &= \sqrt{(3)^2 + (4)^2} \\ &= \sqrt{9 + 16} \\ &= \sqrt{25} \\ &= 5 \end{aligned}$$

Specifying a line and margin

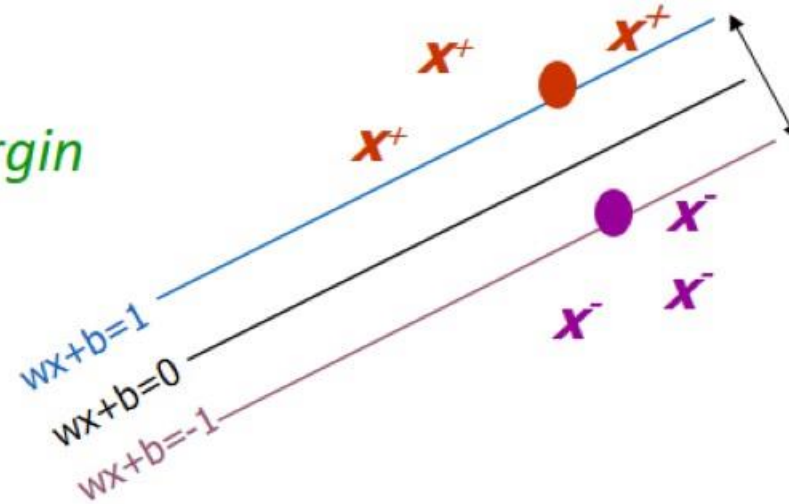
Given a guess of $\mathbf{w}$ and b we can

- Compute whether all data points in the correct half-planes
- Compute the width of the margin

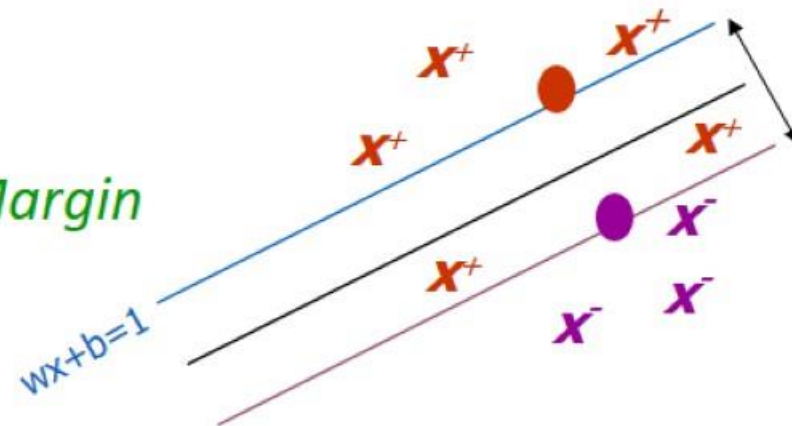
So now we just need to write a program to search the space of $\mathbf{w}$'s and b 's to find the widest margin that matches all the datapoints.

Margin Classifier

Hard Margin

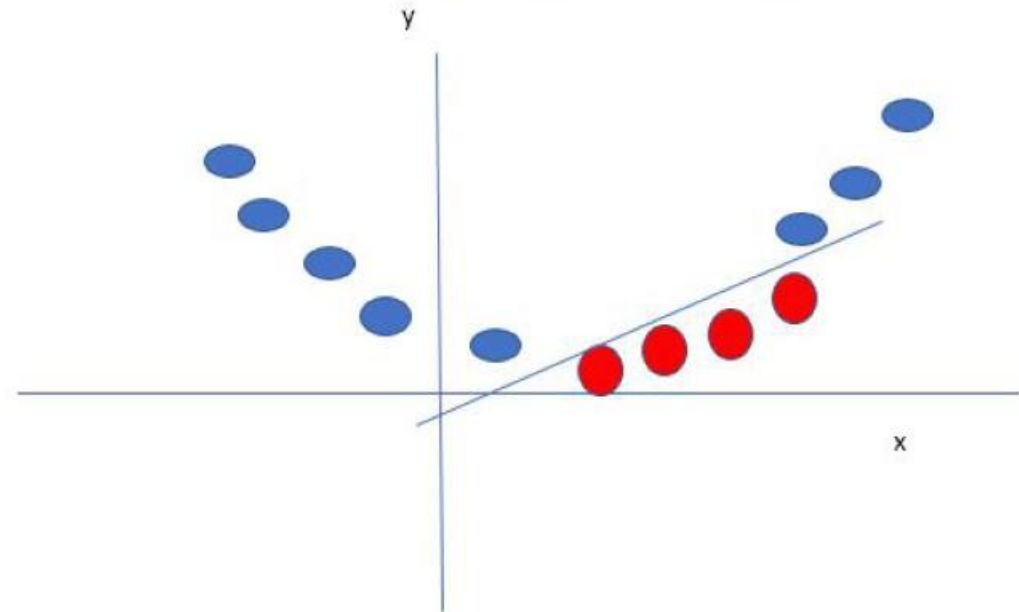
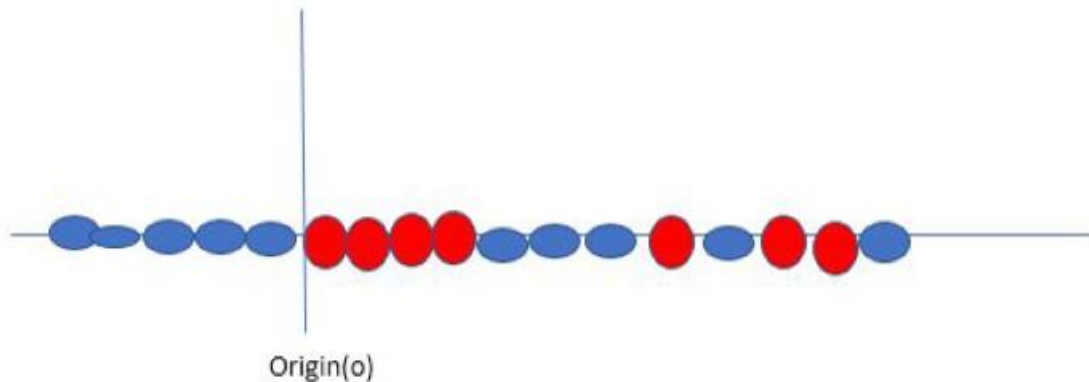


Soft Margin

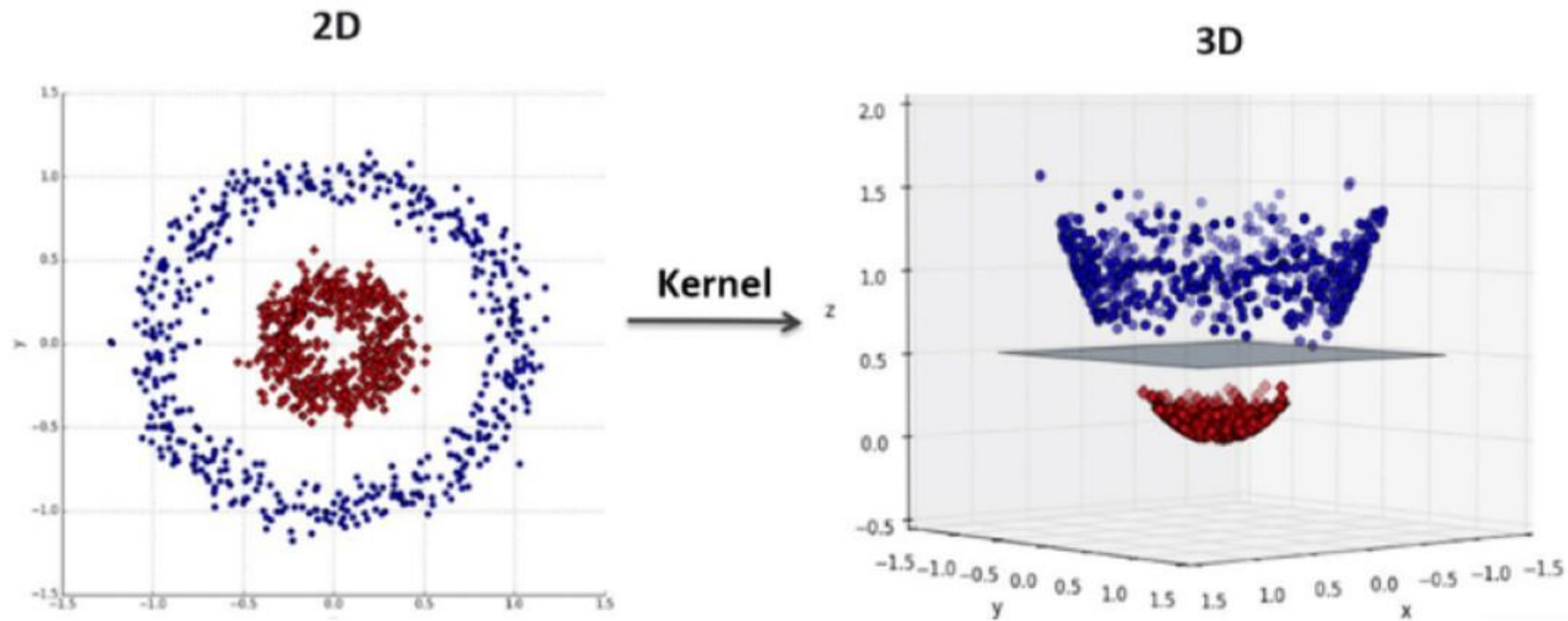


Support Vector Machine Kernels

- **Kernel:** Kernel is the mathematical function, which is used in SVM to transfer (map) the original input data points into high-dimensional feature spaces, so, that the hyperplane can be easily found out even if the data points are not linearly separable in the original input space. Some of the common kernel functions are linear, polynomial, radial basis function(RBF), and sigmoid.

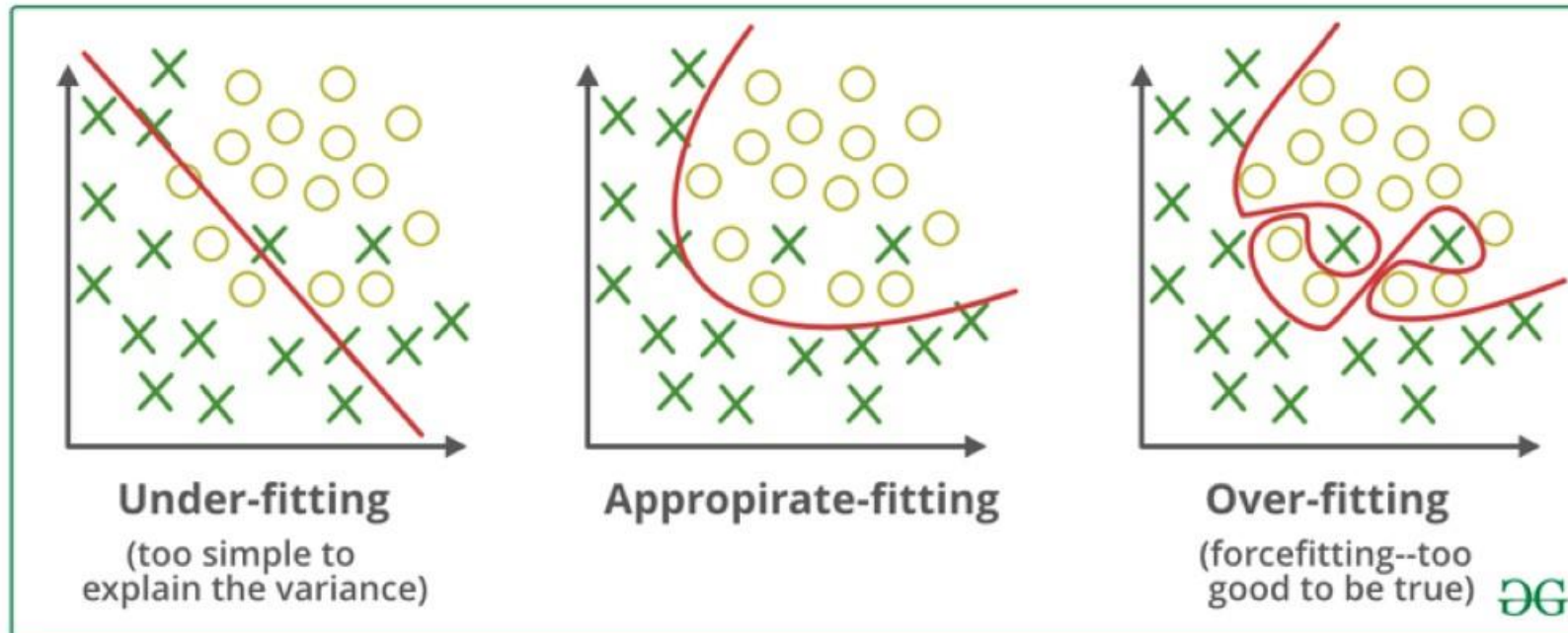


Polynomial kernel SVM Example



Underfitting and Overfitting

- **Bias** and **variance** are two fundamental concepts in machine learning that help us understand the performance of models and the underlying factors contributing to their errors.



Underfitting and Overfitting

- **Bias:**
- Bias refers to the error introduced by the simplifying assumptions made by a model when trying to capture the underlying relationships between features and the target variable.
- A high bias model tends to oversimplify the data and may not capture the true patterns, leading to underfitting.

- **Variance:**
- Variance refers to the amount by which the model's predictions vary for different training datasets.
- A high variance model is overly sensitive to the training data and captures random noise instead of true patterns, leading to overfitting.
- Models with high variance tend to perform well on the training data but poorly on unseen data because they memorize the training examples instead of learning generalizable patterns.

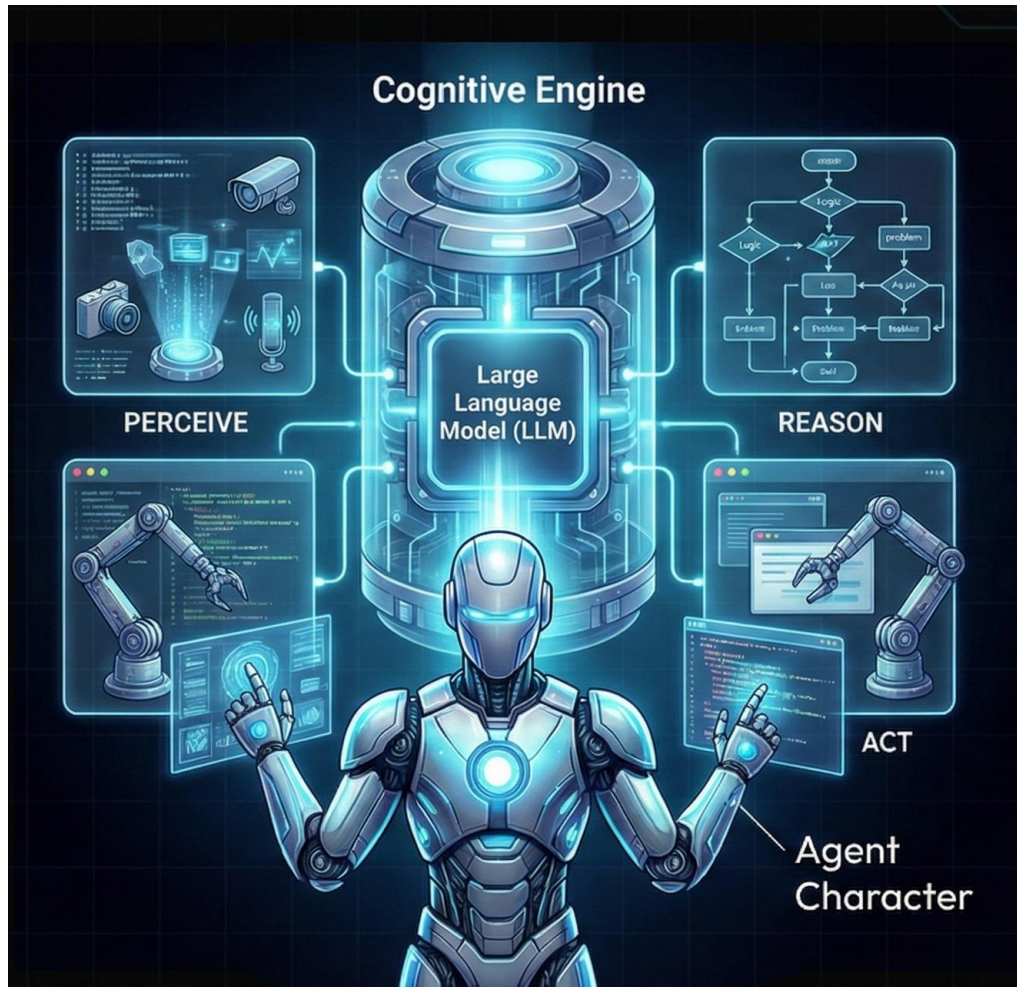
Additional Resources

- **"Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow"**
Book by Aurélien Géron
- **Research paper "Support-Vector Networks" by Vladimir Vapnik**
 - <https://link.springer.com/article/10.1007/BF00994018>

Recent AI Approaches

Agentic AI

Defining Agentic AI



Agentic AI represents a shift from "chatting" to "doing". These architectures utilize Large Language Models (LLMs) as a cognitive engine to:

- **Perceive** their digital or physical environment.
- **Reason** about the steps needed to solve a problem.
- **Act** using external tools (APIs, browsers, code interpreters).

Key Capabilities

- **Self-Directed Planning**

Agents can break down a high-level goal (e.g., "Research competitor X") into a sequence of actionable steps without human intervention.

- **Contextual Memory**

Unlike stateless sessions, agents maintain short-term and long-term memory, allowing them to learn from past interactions and maintain continuity.

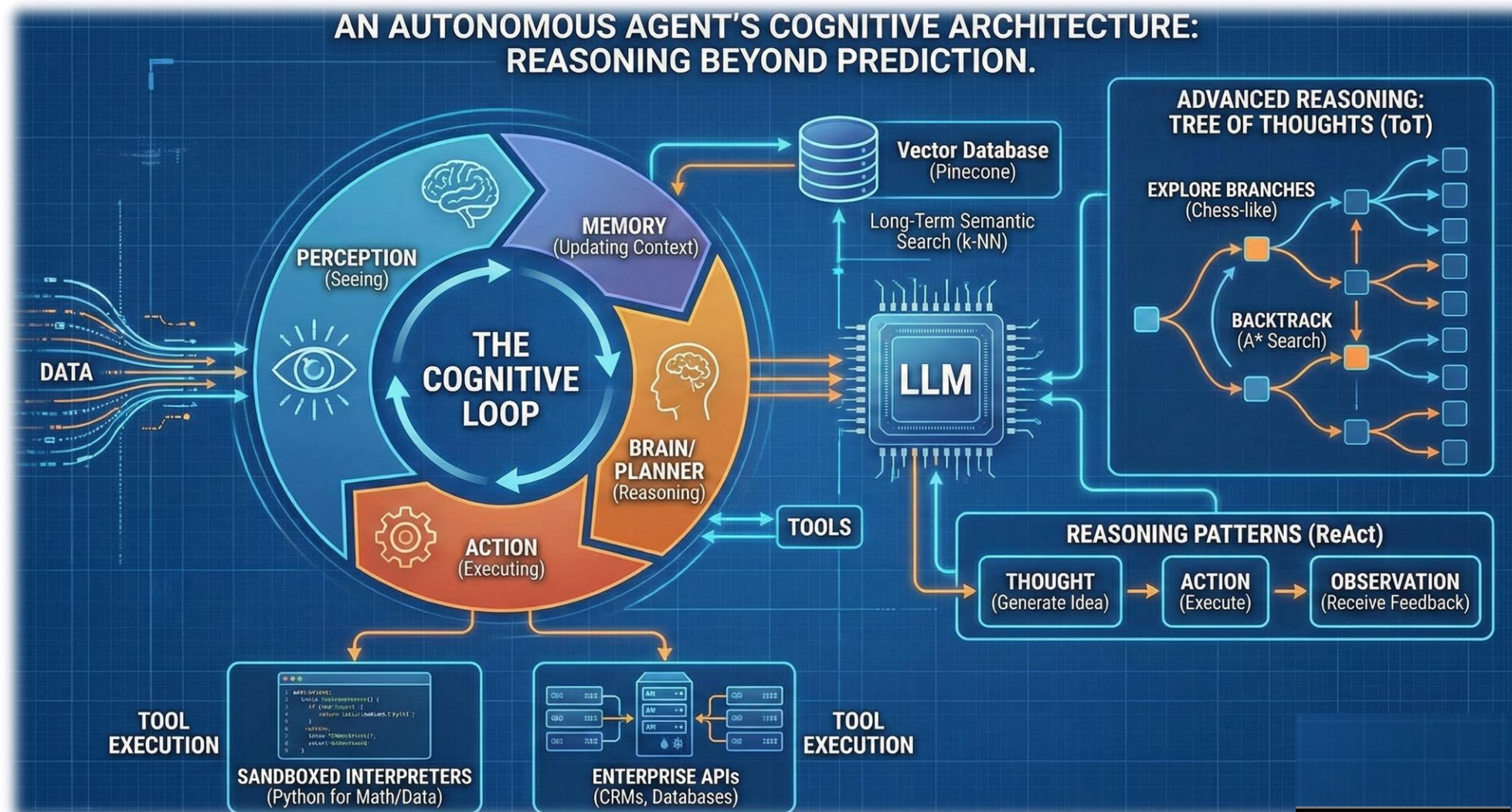
- **Tool Use & Execution**

The ability to call functions, query databases, browse the web, and write/execute code to validate their own outputs.

- **Reflection & Iteration**

Agents can critique their own work, catch errors, and refine their approach in a "thought loop" before presenting a final result.

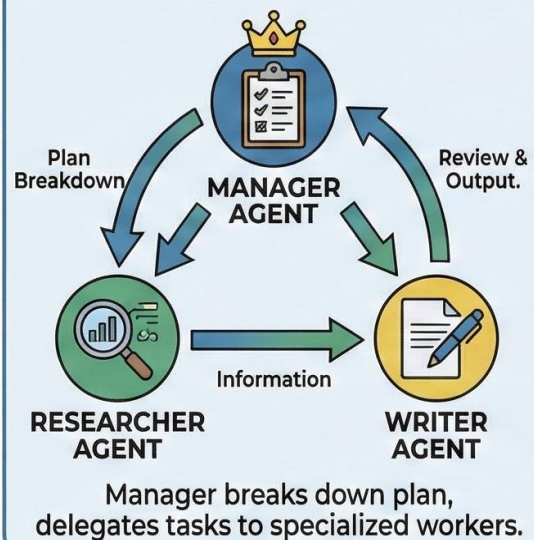
Agentic AI Frameworks (The "Brain")



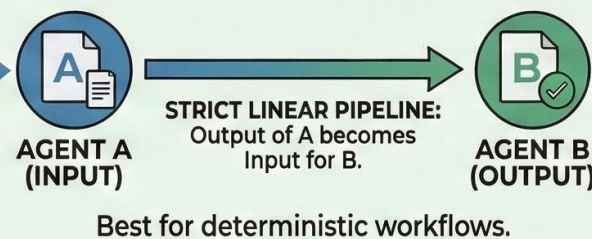
Multi-Agent Systems (The "Team")

ORCHESTRATION PATTERNS

HIERARCHICAL (DELEGATION)



SEQUENTIAL HANDOFF (PIPELINE)



JOINT CHAT (SWARM)

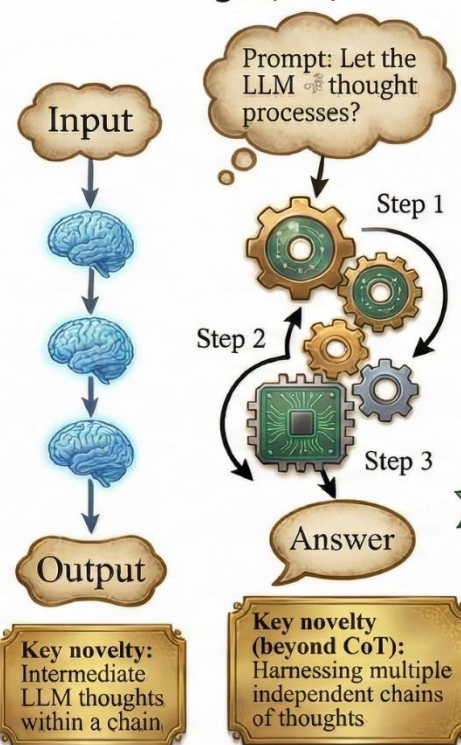


Evolution of Reasoning

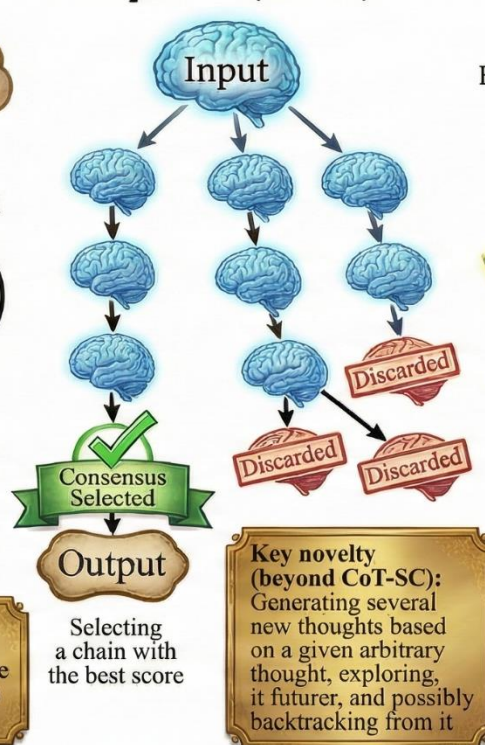
Basic Input-Output (IO)



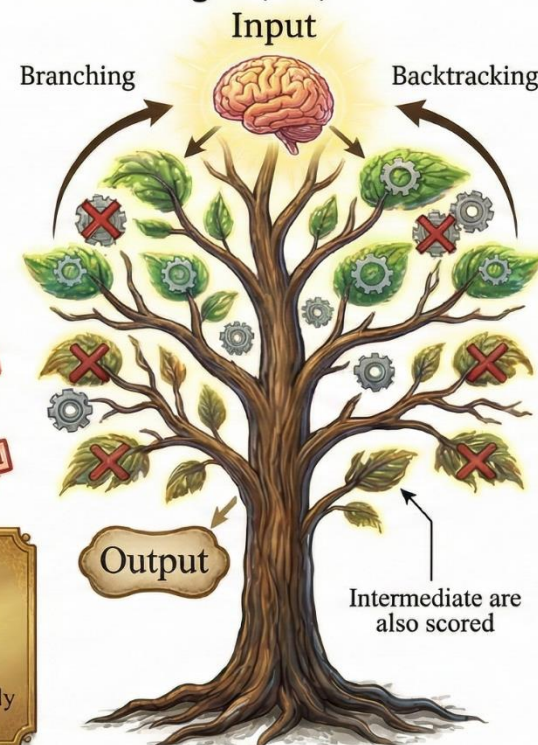
Chain-of-Thought (CoT)



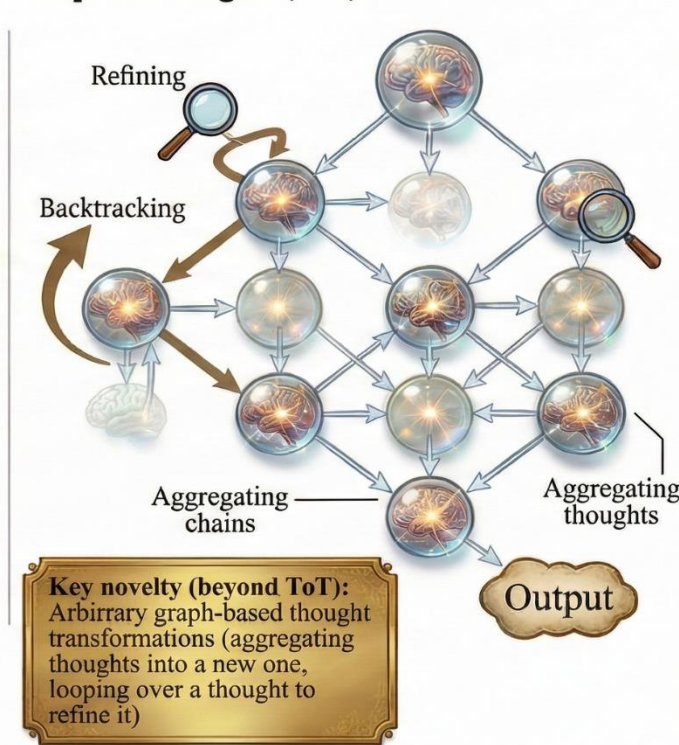
Multiple CoTs (CoT-SC)



Tree of Thoughts (ToT)



Graph of Thoughts (GoT)



Evolution of Reasoning

Basic Input-Output (IO)



1. Basic Input–Output (IO)

Description

This is the most basic reasoning mode.

The model receives an input.

- It produces an output **directly**, without showing or generating intermediate reasoning steps.

- It has **no deliberate thought process**.

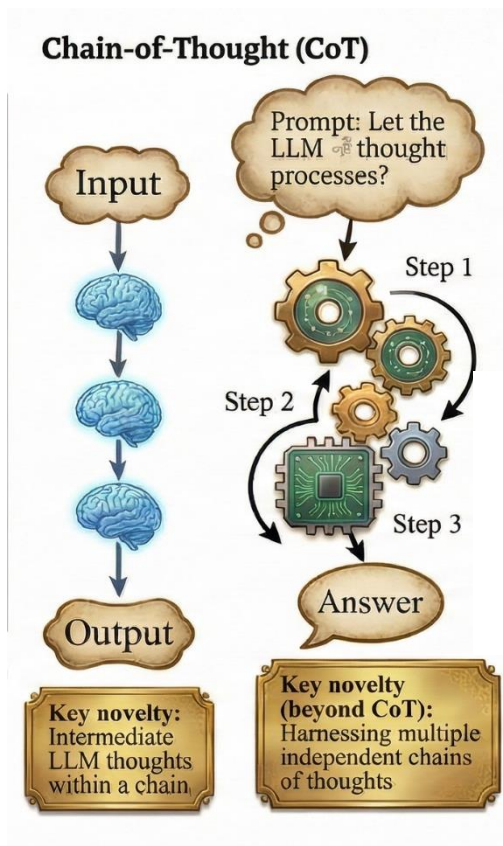
Example

- **User:** “What is 12×8 ?”

Model: “96.”

No steps are shown; the answer appears immediately.

Evolution of Reasoning



2. Chain-of-Thought (CoT)

Description

Chain-of-thought makes the model **think step-by-step** in a *single linear chain*. The LLM generates a sequence of intermediate reasoning steps. This helps solve math problems, logic puzzles, explanations, etc. Only **one chain** of reasoning is followed.

Example

User: "A train travels 300 km in 3 hours. What is its speed?"

Model (CoT):

Speed = Distance / Time

Distance = 300 km

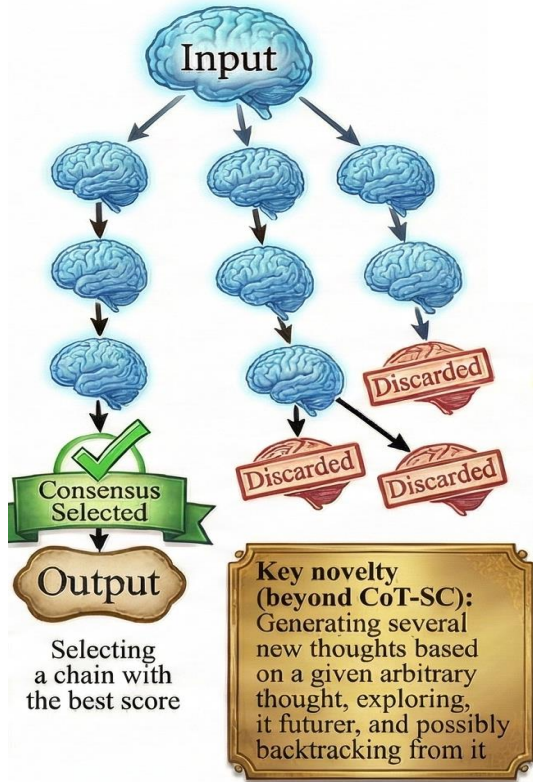
Time = 3 h

Speed = $300 / 3 = 100$ km/h

Final Answer: 100 km/h

Evolution of Reasoning

Multiple CoTs (CoT-SC)



3. Multiple CoTs (Self-Consistency, CoT-SC)

Description

Instead of using **one chain**, the model generates **many independent reasoning chains**, then selects the best or most consistent answer.

Reduces errors from a single incorrect reasoning path.

Similar to brainstorming multiple solutions, then choosing the best.

Example

User: "What is the next number: 2, 4, 8, 16, ...?"

The model generates **multiple CoTs**:

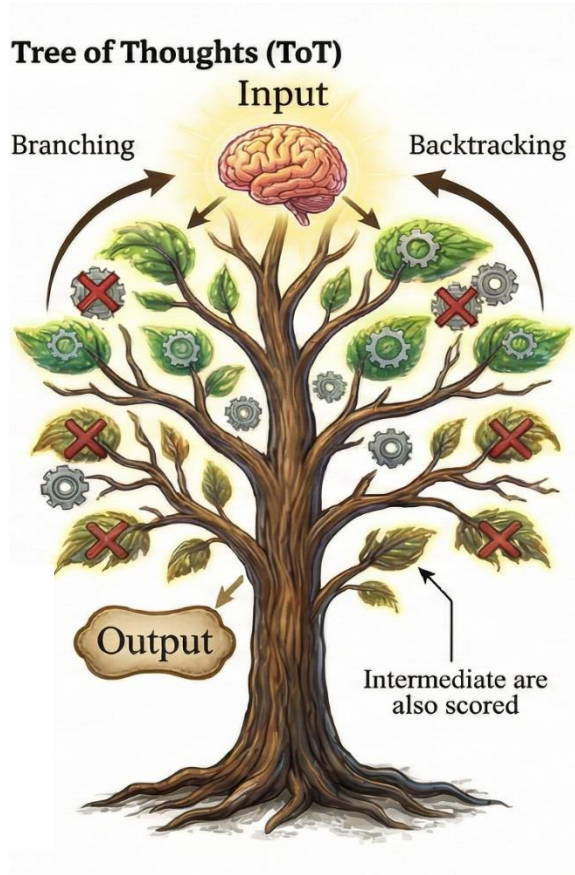
Chain 1: "This doubles each time → next is 32."

Chain 2: "Pattern increases by $\times 2$ → next is 32."

Chain 3: "Geometric progression → next is 32."

All agree → **consensus = 32.**

Evolution of Reasoning



4. Tree of Thoughts (ToT)

Description

Instead of a single chain, ToT allows **branching**:

At each step, the model can generate **multiple possible next thoughts**.

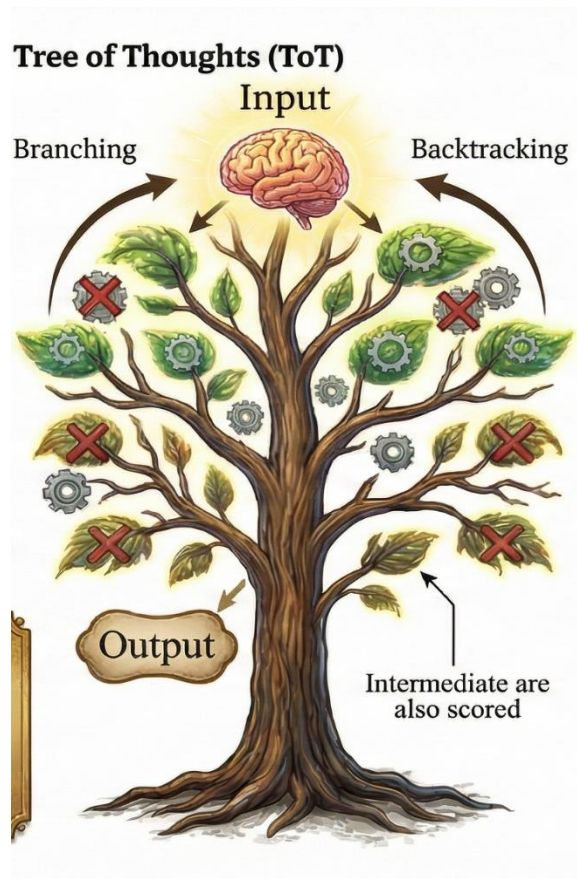
The model can **score** these thoughts, **explore**, and **backtrack** if needed.

This helps with: planning, puzzles, long-term strategy problems, and creative writing.

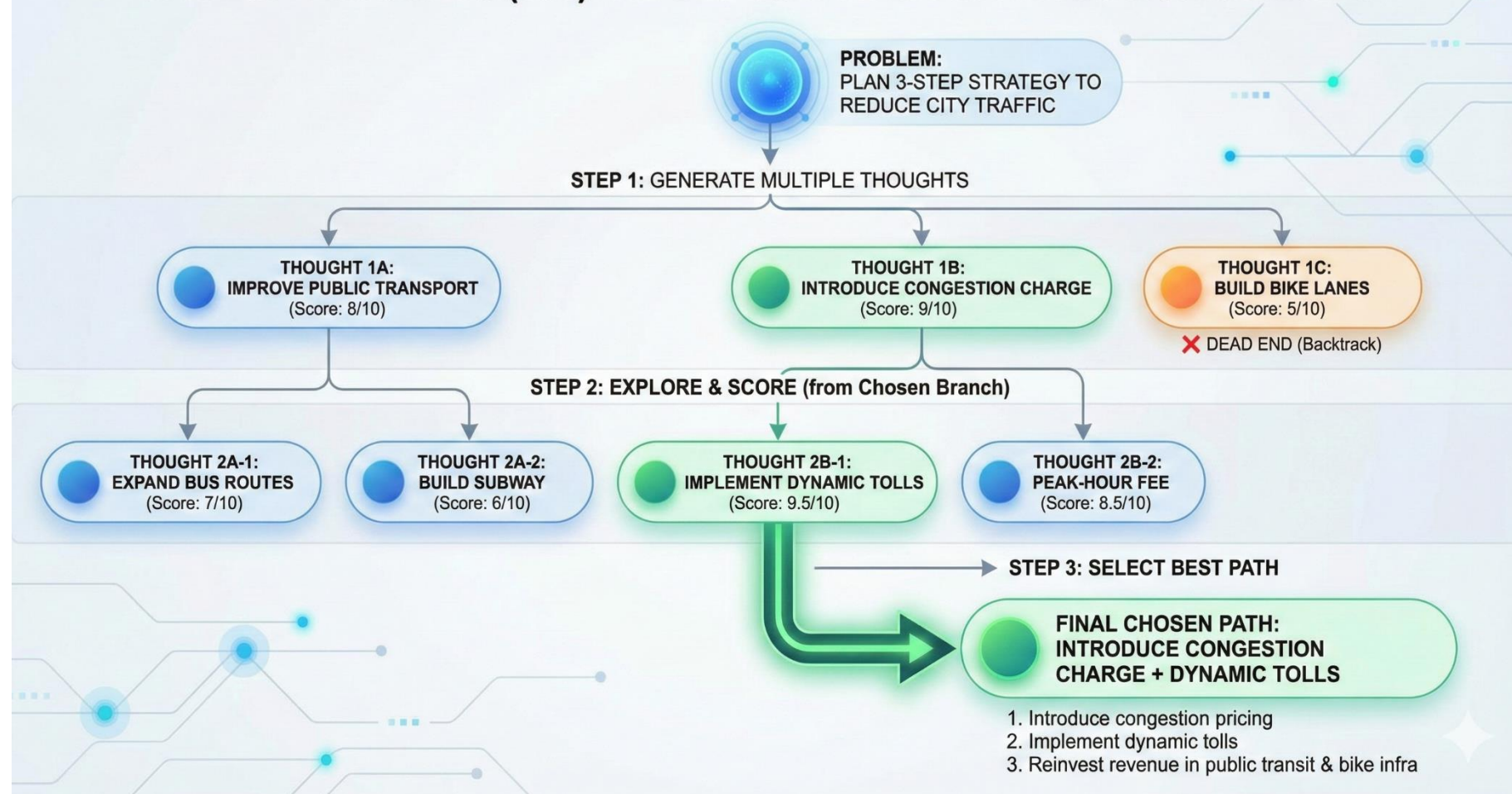
Example

Problem: *“Plan a 3-step strategy to reduce traffic in a city.”*

Evolution of Reasoning

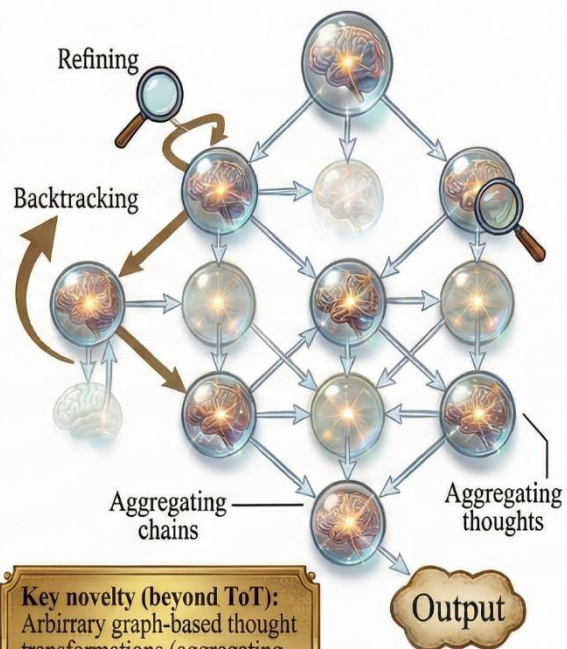


TREE OF THOUGHTS (ToT): PLANNING A TRAFFIC REDUCTION STRATEGY



Evolution of Reasoning

Graph of Thoughts (GoT)



5. Graph of Thoughts (GoT)

Description

This is the most advanced method.

Thoughts are not linear or tree-shaped — they form a **graph**.

The model can **aggregate multiple thoughts into a new one**, or **loop back and refine** a thought.

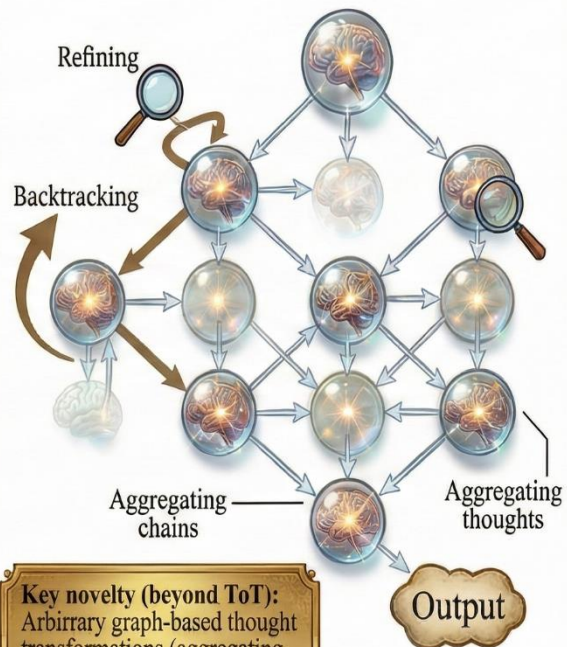
GoT supports: Complex reasoning, Deep planning, Multi-constraint optimization, and Collaborative thought merging

Example

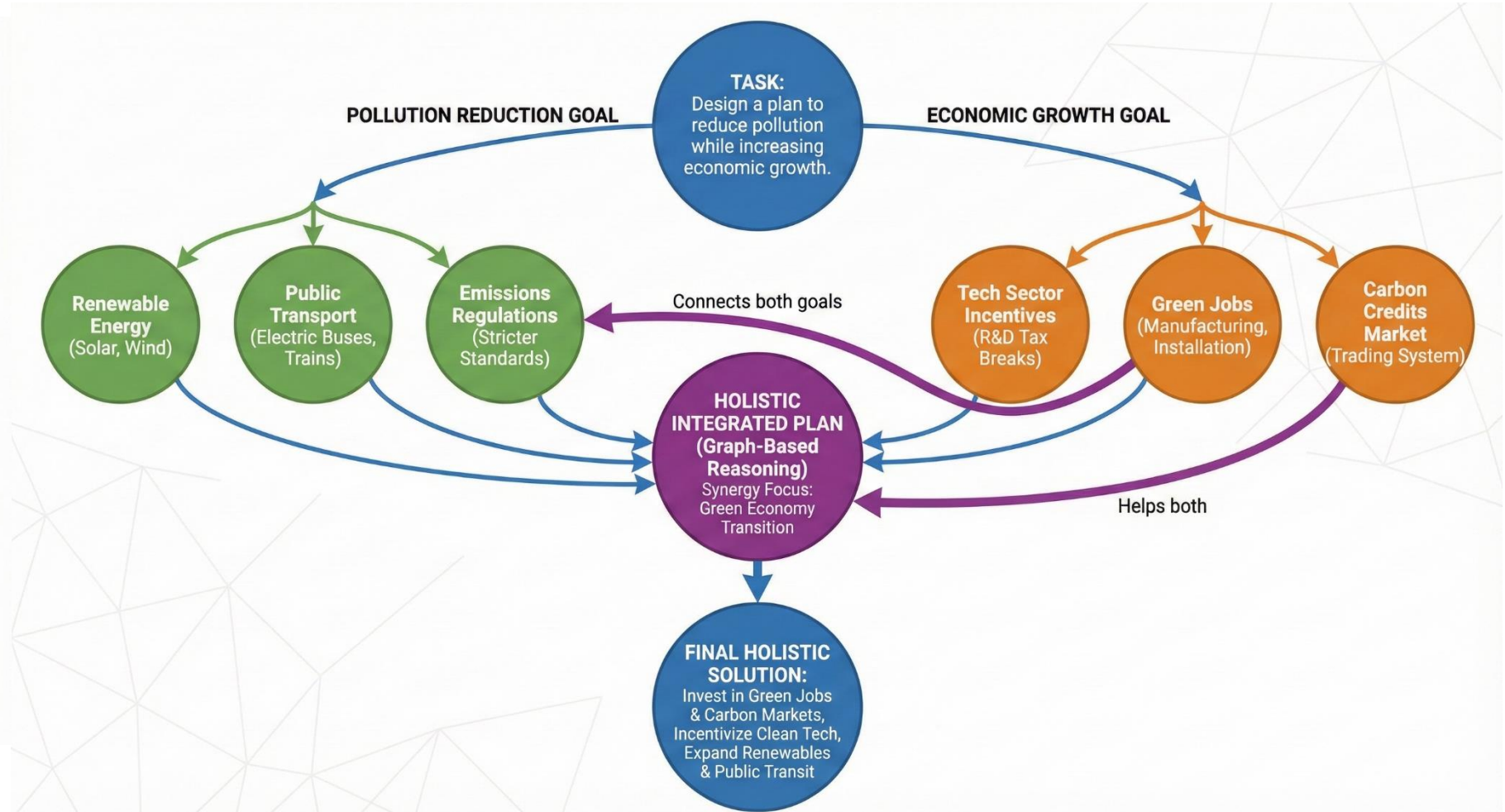
Task: *“Design a plan to reduce pollution while increasing economic growth.”*

Evolution of Reasoning

Graph of Thoughts (GoT)



Key novelty (beyond ToT):
Arbitrary graph-based thought transformations (aggregating thoughts into a new one, looping over a thought to refine it)



Orchestration Frameworks

LangGraph

The Controller

Moving beyond linear chains (DAGs) to **cyclic graphs**.

Essential for agents that need to loop, retry, and maintain state over long periods.

AutoGen

The Conversation

Microsoft's framework where agents "talk" to solve problems.

Excellent for multi-agent simulation and code execution.

CrewAI

The Role-Player

Focuses on "Role-Playing" agents (Researcher, Writer).

High-level abstraction that makes building teams intuitive.

AI Agent Protocols

MCP (Model Context Protocol) – Anthropic

- Enables secure two-way communication between an LLM and external tools.
- Think of it as the USB-C for AI agents connecting to real-time data.

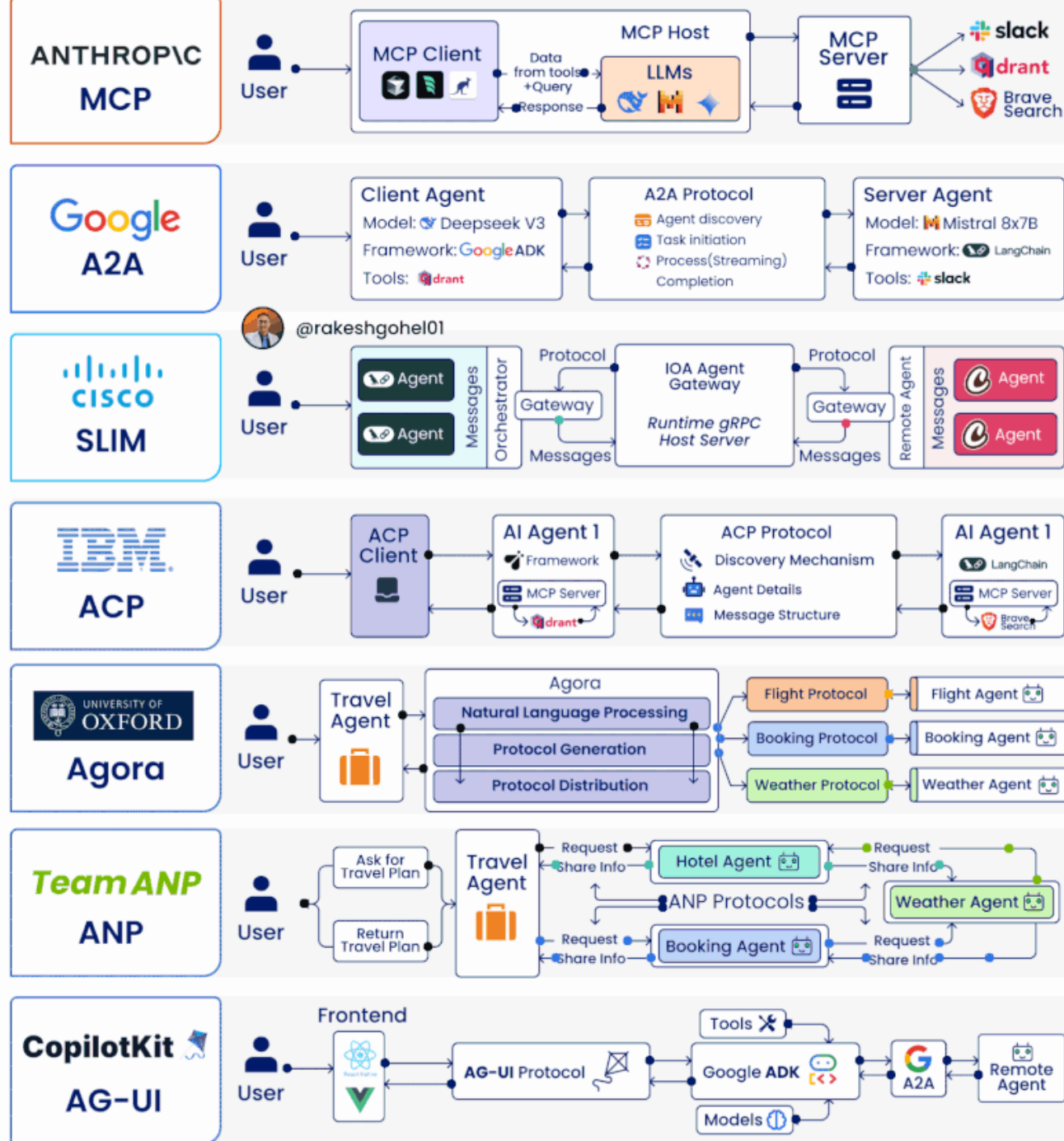
A2A (Agent 2 Agent Protocol) – Google

- Built for agent-to-agent communication, regardless of vendor or framework.
- Enables agents to discover, collaborate, and execute multi-step tasks.

ACP (Agent Communication Protocol) – IBM

- A RESTful protocol enabling persistent, secure agent conversations.
- Focused on asynchronous, scalable, and observable execution.

AI Agent Protocols



Scientific Challenges

Performance & Latency

Real-time control loops (50-200 Hz) require inference speeds that current large models struggle to meet. Edge computing and model distillation are critical research areas.

Scalability & Coordination

How do we coordinate thousands of autonomous agents without creating a communication bottleneck? We need new "Agent-to-Agent" protocols and decentralized consensus mechanisms.



Thank You