Lecture 7: Introduction to Contextual Bandits

ID 6002W: Online and Reinforcement Learning

Web-enabled M.Tech. program, IIT Madras

Suryanarayana Sankagiri

2nd June, 2026

The Story So Far

Multi-armed Bandit | Best-arm Identification | Regret Minimisation | UCB | Thompson Sampling

estimate → quantify uncertainty → act using optimism or sampling

<u>Assumptions so far:</u> fixed, featureless arms

one user type · fixed arm set · fixed means $\mu_1, \ldots, \mu_K$

How realistic are these assumptions?

Yahoo! Front Page, 2009

A real personalisation problem

- Setting: Four editor-curated articles shown at a time on yahoo.com (F1–F4, refreshed hourly)
- Problem statement: choose one article of these four as main story, based on individual interests
- Reward signal: whether the user clicks the highlighted story

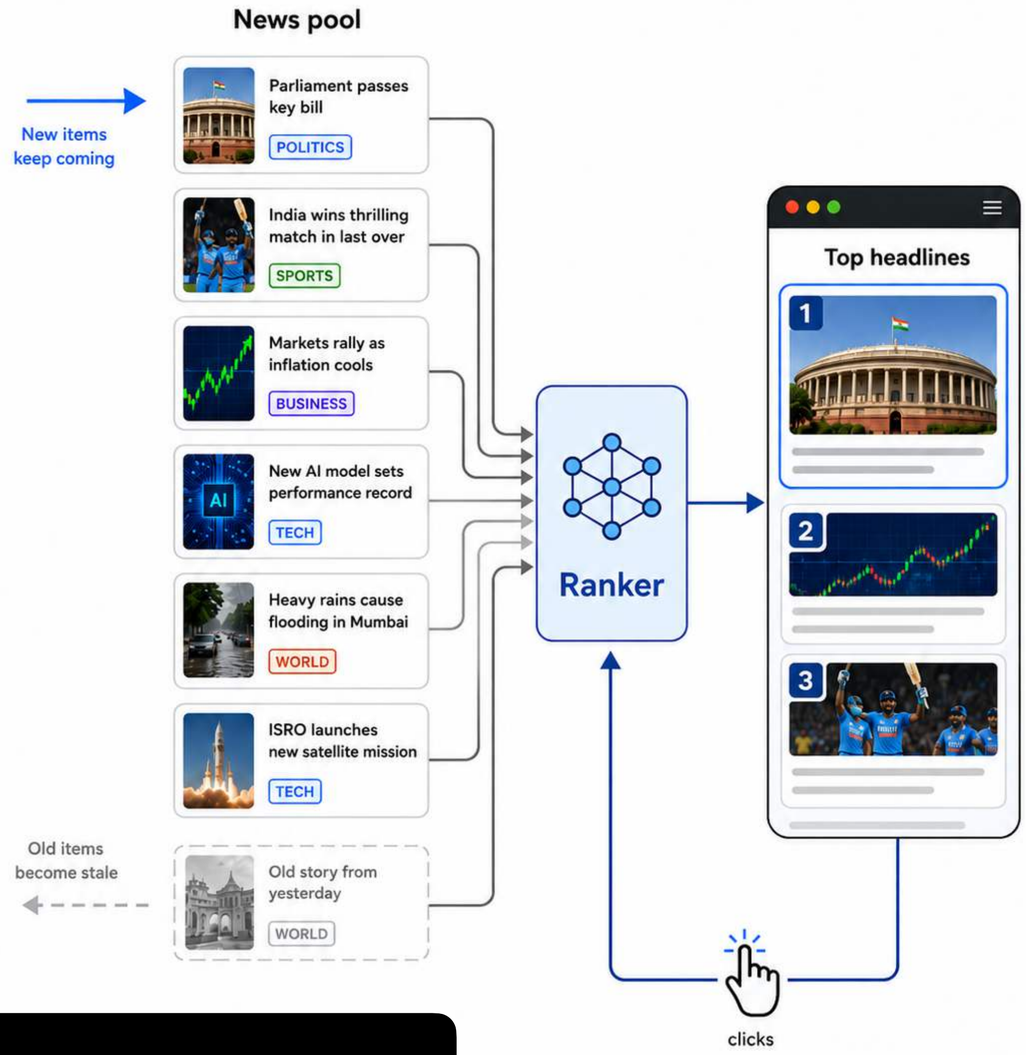
Figure 1: A snapshot of the "Featured" tab in the Today Module on Yahoo! Front Page. By default, the article at F1 position is highlighted at the story position.

Influential paper: A Contextual-Bandit Approach to Personalised News Article Recommendation

Where The Assumptions Break

In real-world systems

- **Arms change**
  - *New items arrive, old items go away*
- **Users differ**
  - *We need personalisation; no single best item for all*
- **Many arms**
  - *Can’t explore all options one by one*



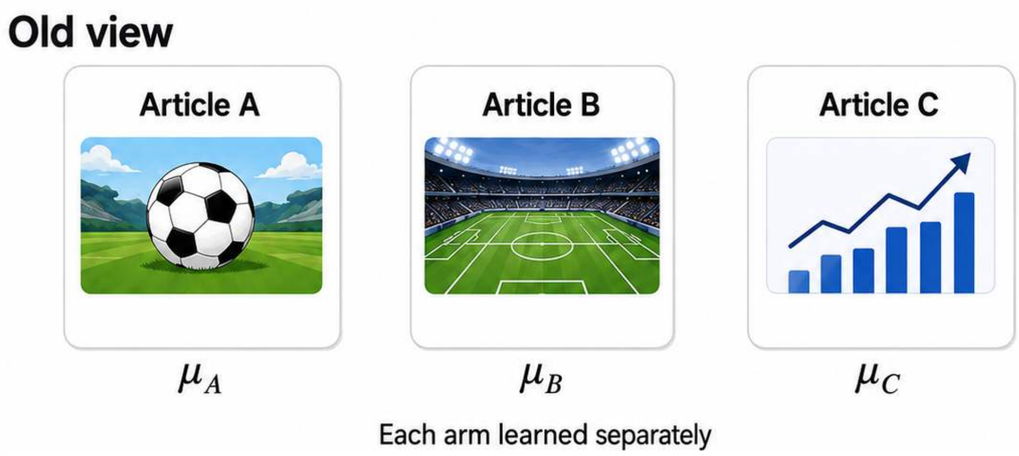
Learning about one user/item should help with similar users/items

Describing Items By Features

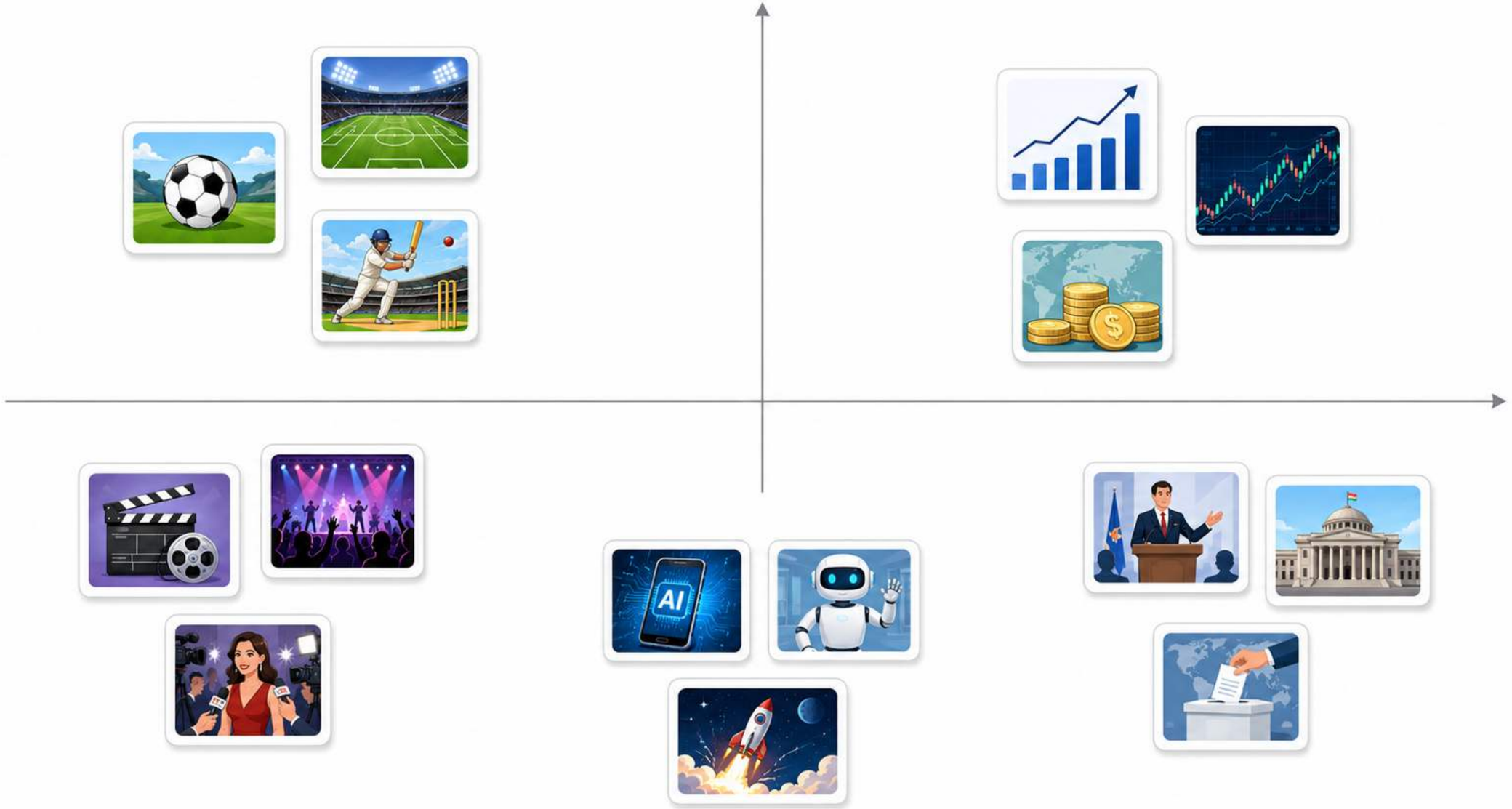
Why features buy you transfer

- In vanilla bandits, an arm is just an index i
- Now, each arm a comes with feature vector x_a
  - Example: a news article may have features like topic, length, recency, source
- Similar features ⇒ similar predictions
  - Learning about one article gives some idea about other similar articles — even a new one

A new article may be unseen, but its features are not!



Features Have Semantic Meaning



The Contextual Bandit Protocol

- At each round $t = 1,2,\ldots$
  - Observe user u_t and an arm set $\mathcal{A}_t$, with a feature vector $x_{t,a}$ for each arm a
  - Choose an arm $a_t \in \mathcal{A}_t$
  - Observe reward r_{t,a_t} — only for the chosen arm
  - Update

NEW: a feature vector per arm, per round

SAME: bandit feedback — we only get pulled arm's reward

What goes into the feature $x_{t,a}$?

What is the quantity to be learned?

From Counting to Predicting

- Until now, learning meant estimating an arm reward by simple averaging:

pull arm i many times, average the rewards → $\hat{\mu}_i$

- With features, estimating an arm becomes a prediction problem: $x_a \xrightarrow{f} \hat{\mu}_a$

Does this remind you of some supervised learning problem?

- When rewards are binary → classification
- When rewards are real → regression

Contextual Bandits with Item Features

Basic Modelling Assumptions

We need to learn a function from features to rewards

- The simplest possible function is a *linear function*
- We model the expected reward as a dot product:

$$\mu_a = \mathbb{E}[X_t \,|\, A_t = x_a] = \langle \theta, x_a \rangle$$

Exactly like linear regression

- One *unknown* weight vector θ, shared across all arms
- All arm features $x_a \in \mathbb{R}^d \Rightarrow \theta \in \mathbb{R}^d$

θ is the parameter to be learned (estimated) from data

Contextual Bandits with Item Features

Basic Modelling Assumptions

We need to learn a function from features to rewards

- Linear functions model real-valued rewards. What if the reward is binary?
- We pass the linear model through a non-linear link function

$$\mu_a = \sigma\left(\langle \theta, x_a \rangle\right)$$

Exactly like logistic regression

$\sigma(t) = 1/(1 + \exp(-t))$

- Once again, we have one *unknown* weight vector θ, shared across all arms
- Similar to linear model. We will focus on linear model in this course

Contextual Bandits with Item Features

What we gain, what we miss

- **The gain:**
  - a brand-new article (published five minutes ago) has *known* features x
  - From past data, we have an *estimate* $\hat{\theta}$
  - $\rightarrow \langle \hat{\theta}, x \rangle$ gives an immediate estimate of article quality, before a single click
- **The miss:**
  - $\langle \hat{\theta}, x \rangle$ depends only on the article, not on who is asking
  - A sports fan and a finance professional get identical recommendations

Arm churn: ✓

Personalisation: ✗

Can incorporating user features give us personalisation?

Personalisation

First attempt

Can we learn a separate contextual bandit for each user?

Replace $\theta \rightarrow \theta_u$ to get $\mu_{u,a} = \langle \theta_u, x_a \rangle$

Does not work well: too little data per user!

What if we split user into groups?

Sports fans $\rightarrow \theta_{\text{sports}}$

Investors $\rightarrow \theta_{\text{finance}}$

Cinema buffs $\rightarrow \theta_{\text{cinema}}$

Better, but brittle. How many groups? How to sort a new user?

Best of Both Worlds

Combining user and item features

$$x_{t,a} = \phi(\text{user at time } t, \text{article } a)$$

Learn one shared θ across all users and items:

$$\mu_{t,a} = \mathbb{E}[X_{t,a} \mid x_{t,a}] = \langle \theta, x_{t,a} \rangle$$

✓ Personalised: $x_{t,a}$ depends on the user

✓ Knowledge transfer: θ is learned across users and items

✓ Flexible: incorporates known aspects of new users and items

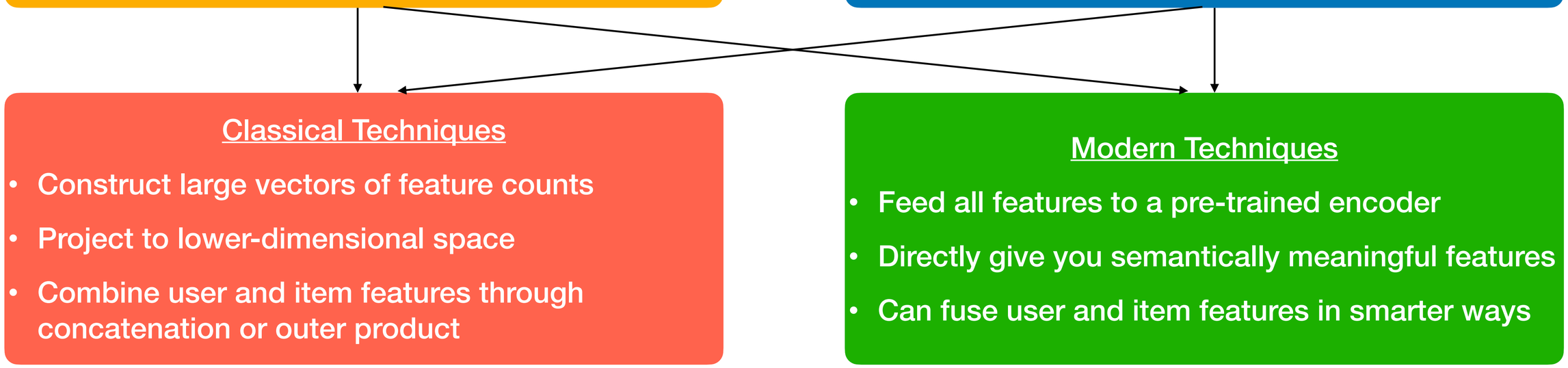
Goal: choose arms with high reward → estimate θ

How Do We Get Features?

An engineering challenge

User features: demographics, device, location, past consumption history

Item features: length, time-of-origin, keywords, bag-of-words (BM25)



Classical Techniques

- Construct large vectors of feature counts
- Project to lower-dimensional space
- Combine user and item features through concatenation or outer product

Modern Techniques

- Feed all features to a pre-trained encoder
- Directly give you semantically meaningful features
- Can fuse user and item features in smarter ways

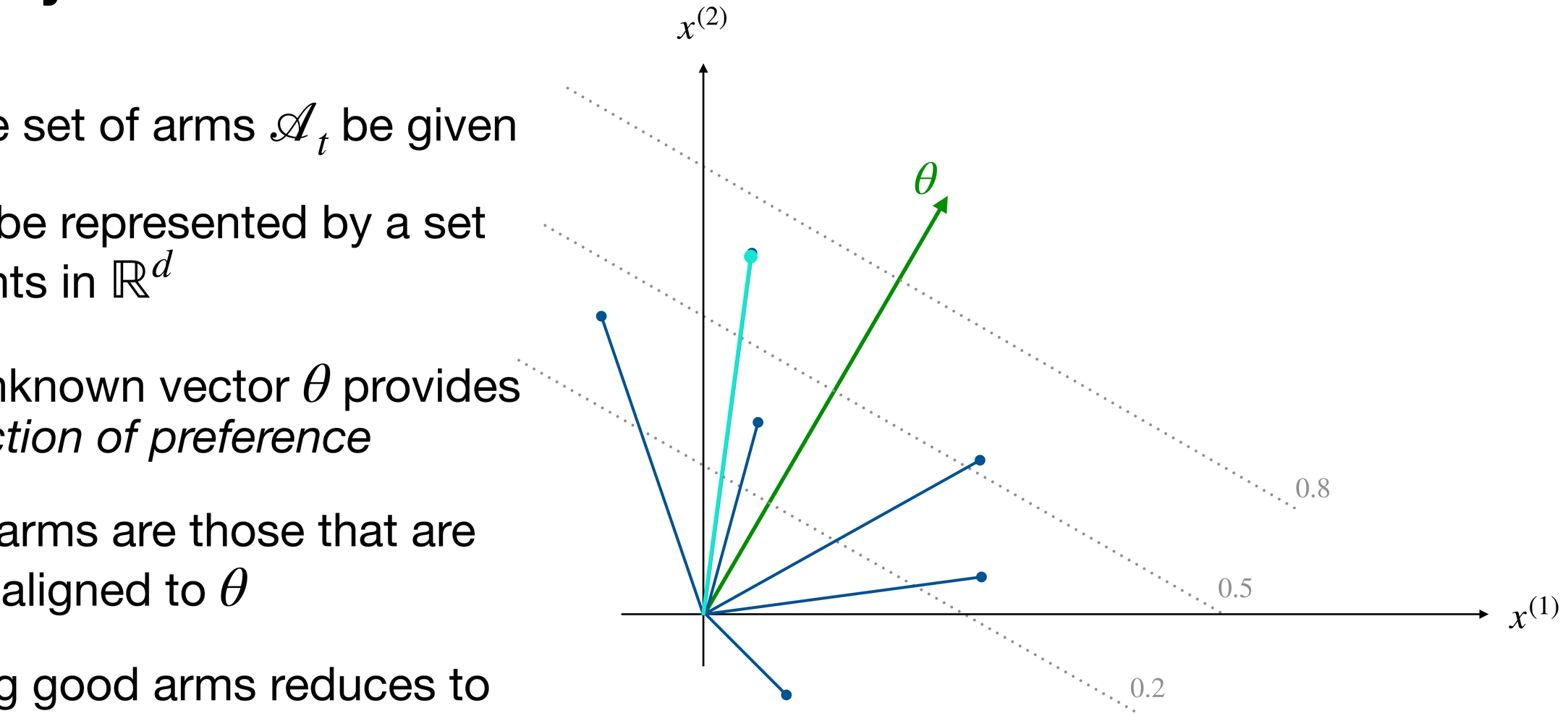
We will cover more practical aspects in weeks 6/7

A Deeper Dive Into Linear Bandits

Arms in Feature Space

Geometry of the model

- Let the set of arms $\mathcal{A}_t$ be given
- It can be represented by a set of points in $\mathbb{R}^d$
- The unknown vector θ provides a *direction of preference*
- Good arms are those that are better aligned to θ
- Finding good arms reduces to estimating θ



How Do We Find θ?

Least squares estimation

After n rounds, we have observed $(x_1, r_1), (x_2, r_2), \ldots, (x_n, r_n)$

Where $x_s = x_{s,a_s}$ is the feature vector of the arm actually chosen

Estimate θ by solving the least-squares problem

$$\hat{\theta} = \arg\min_{\theta} \sum_{s=1}^{n} \left(r_s - \langle \theta, x_s \rangle \right)^2$$

Difference from linear regression: you can *control* the features x_s that is used in the estimate

Importance of Regulariser

Ridge regression

In bandits, early data can be scarce ⇒ least squares problem does not have a solution

Solution: add a regulariser to the least-squares formulation

$$\hat{\theta} = \arg\min_{\theta} \sum_{s=1}^{n} \left(r_s - \langle \theta, x_s \rangle \right)^2 + \lambda \|\theta\|^2$$

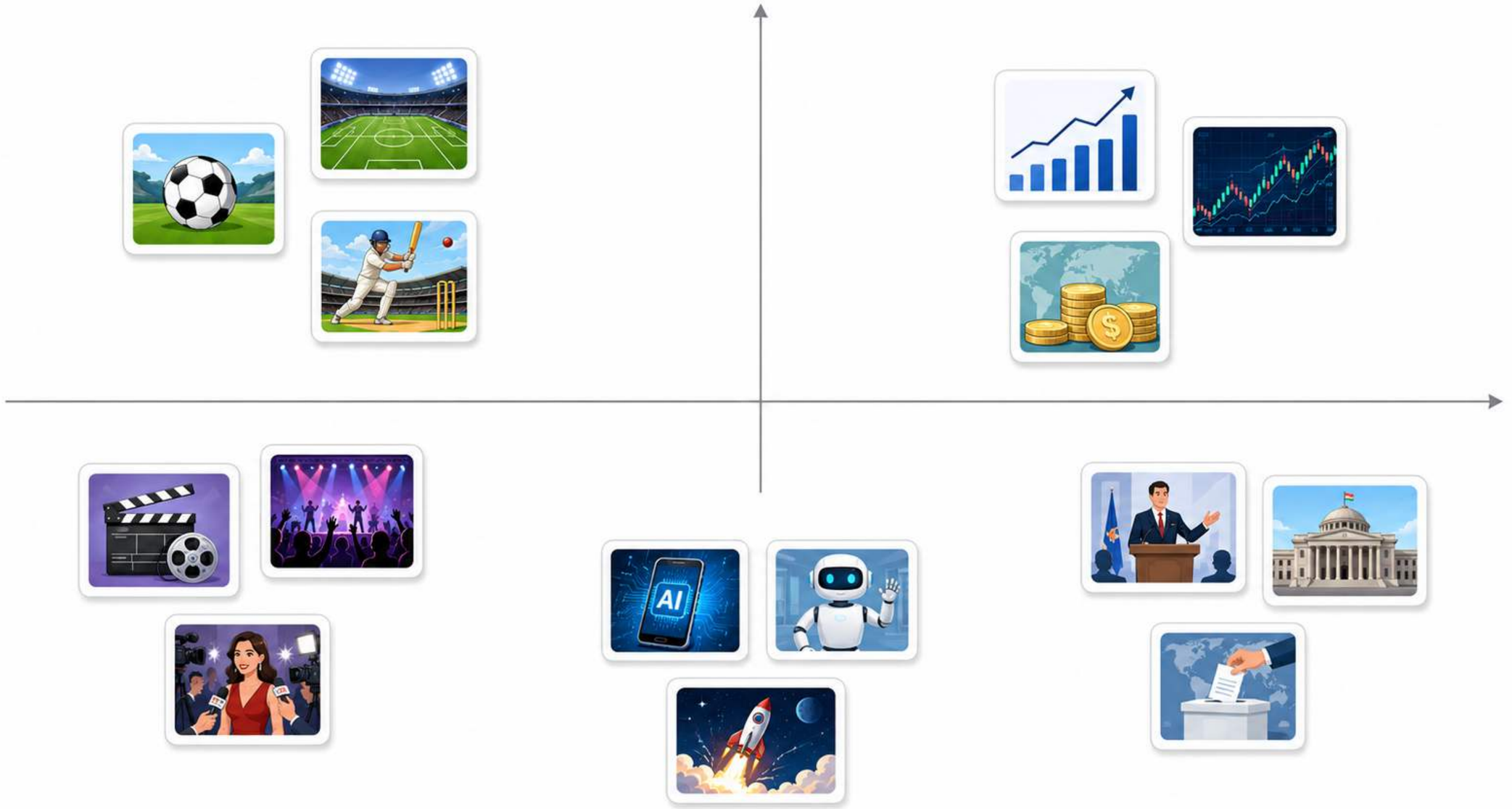
Equivalent closed form:

$$\hat{\theta} = V_n^{-1} b_n$$

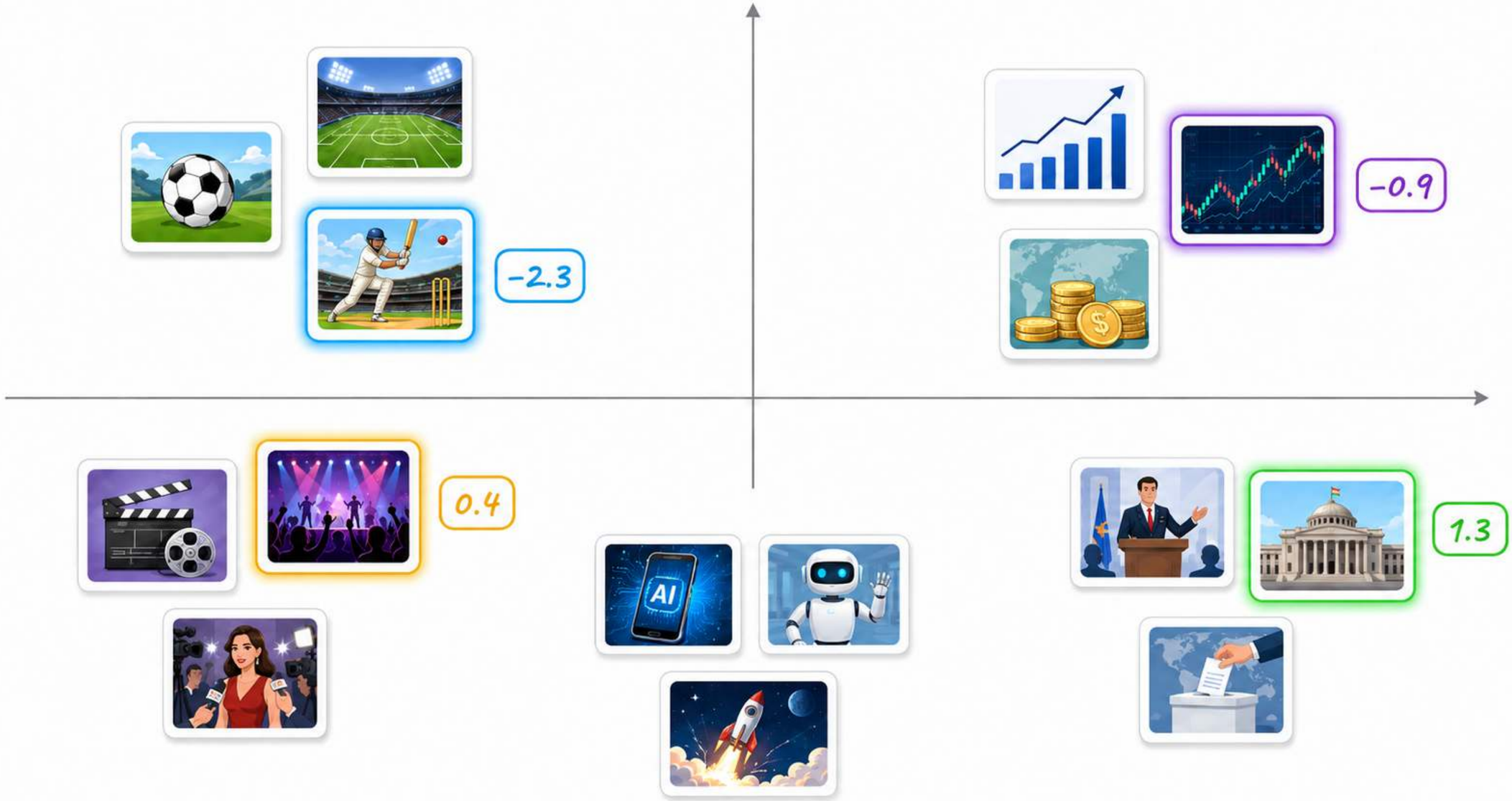
This matrix will be very important going forward

$$V_n = \lambda I + \sum_{s=1}^{n} x_s x_s^\top \qquad b_n = \sum_{s=1}^{n} x_s r_s$$

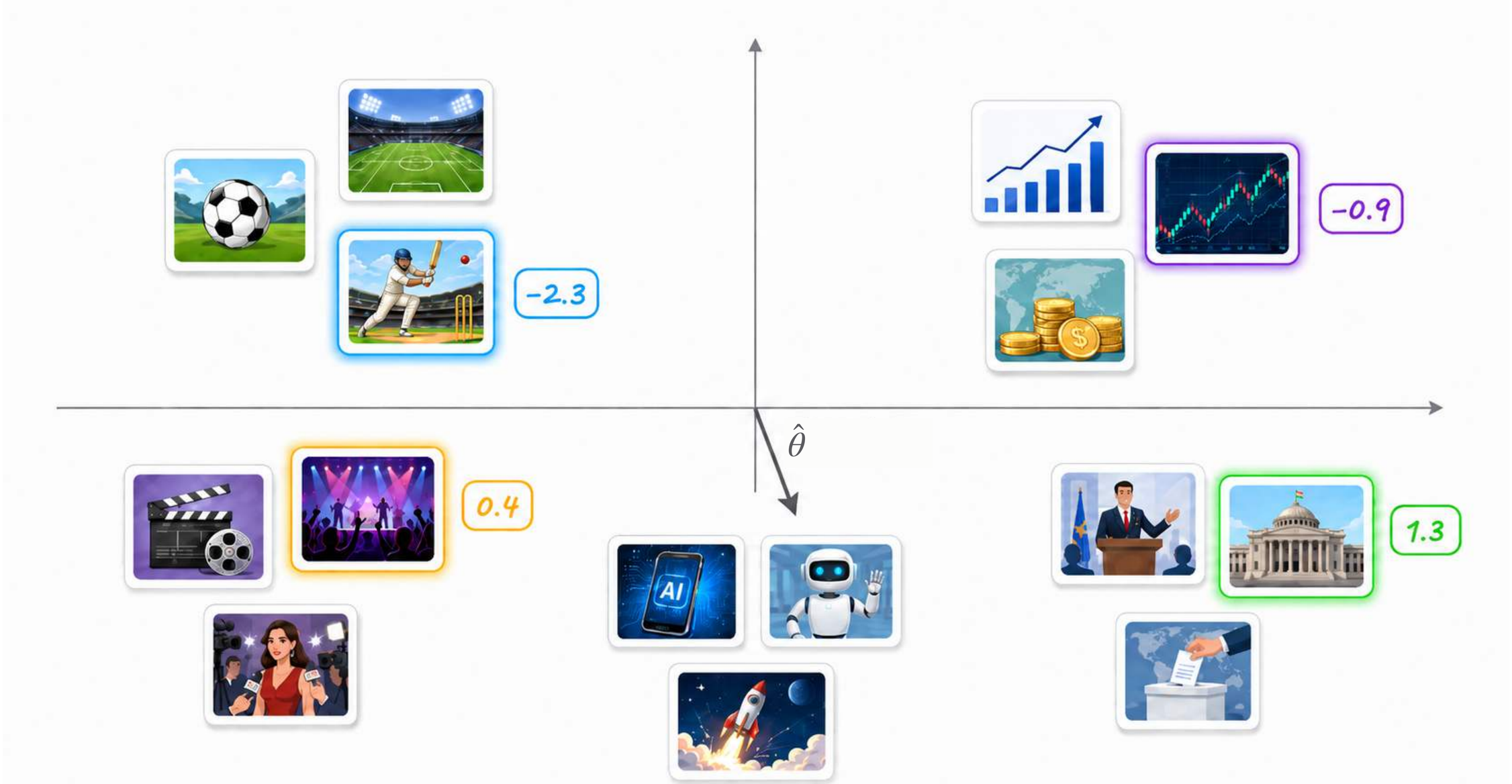
A Worked Example



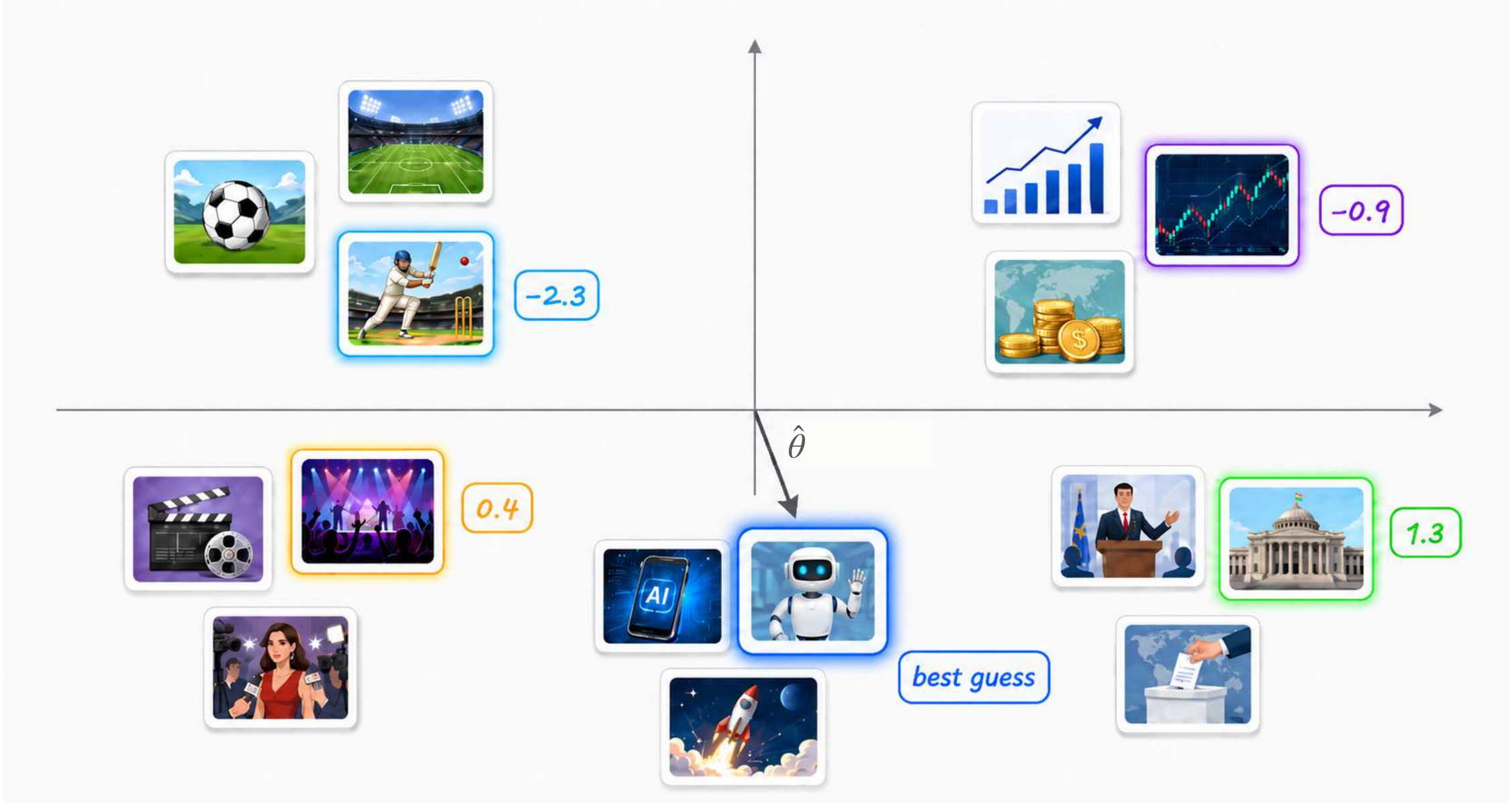
A Worked Example



A Worked Example



A Worked Example



Towards A Bandit Algorithm

Is the best guess enough?

Given the least squares estimate $\hat{\theta}_n$ after n rounds,
what arm do we recommend at time $n+1$?

- The natural rule: pick the arm with the highest predicted reward

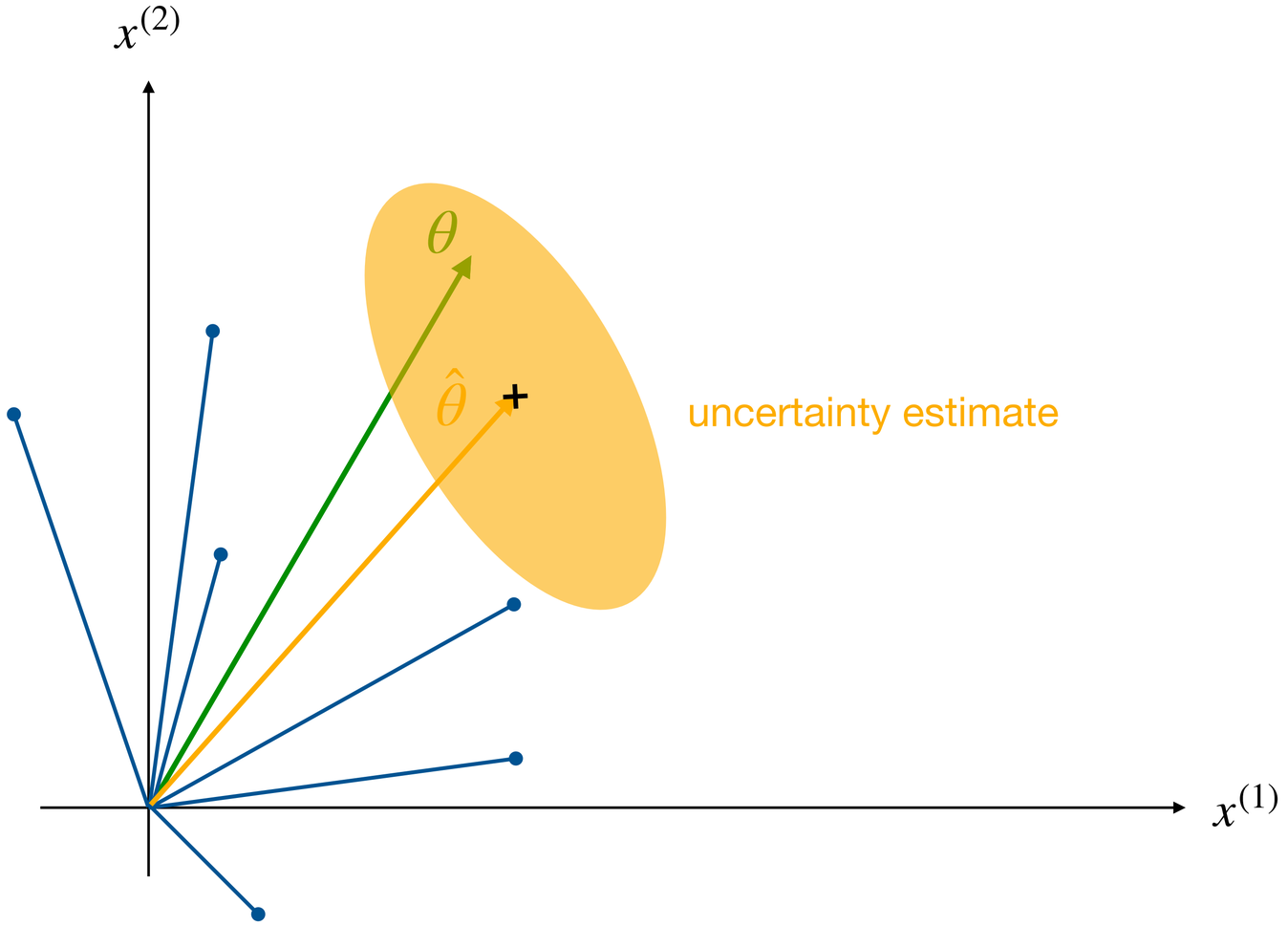
$$a_{n+1} = \arg\max_a \langle \hat{\theta}, x_{t,a} \rangle$$

- Easy to implement—but we know from basic bandits that this won't work!
  - All estimates are imperfect ⇒ greedy can get stuck on the wrong arm
  - We can do ε-greedy. But surely there are better approaches?

Intelligent Exploration

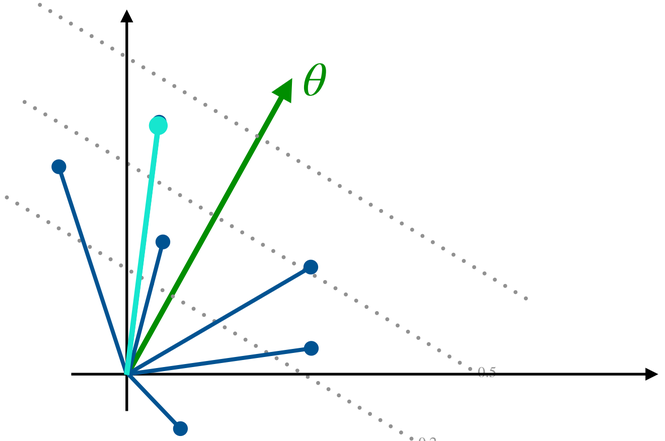
What would it look like?

- To explore intelligently, we revisit concepts we have already seen for bandits: optimism (UCB) or sampling (Thompson).
- Both need one ingredient: *a measure of how uncertain our estimate is*



Conclusion

- We saw the contextual bandit model, where each arm has features
- Features allow us to generalise across users and items—useful in practical systems
- Both user and item features can be combined to handle personalisation and arm churn
- Modelling rewards as a linear function of features gives us linear bandits,
  which has linear regression (least-squares estimation) at its core
- Next time: capturing uncertainty about $\hat{\theta}$ and designing LinUCB



Homework Logistics

- Homework 1 solution has been posted online
- Homework 2, based on UCB and TS (Lectures 4, 5, 6) will be released on Thursday
  - Due date: Sunday, 14 June, 11:59 PM
  - Similar structure as HW1
- Homework 3, based on linear bandits (Lectures 7, 8, 9): release date TBD
- TAs: Sukanya: da25d900@smail.iitm.ac.in
  Roshan: bt21d401@smail.iitm.ac.in

Questions?