

Lecture 10: Linear Bandits in Practice

ID 6002W: Online and Reinforcement Learning

Web-enabled M.Tech. program, IIT Madras

The Linear Bandits Model

An extension to multi-armed bandits

The Linear Bandits Model

An extension to multi-armed bandits

- The basic multi-armed bandit setting had K arms
 - ◉ each with fixed but unknown rewards μ_i
 - No relation between $\mu_1, \dots, \mu_K$ is assumed

The Linear Bandits Model

An extension to multi-armed bandits

- The basic multi-armed bandit setting had K arms
 - ◉ each with fixed but unknown rewards μ_i
 - No relation between $\mu_1, \dots, \mu_K$ is assumed
- The linear bandit model adds some structure to $\mu_1, \dots, \mu_K$
 - ◉ $\mu_i = \langle \theta, x_i \rangle$: $x_i \in \mathbb{R}^d$ known, $\theta \in \mathbb{R}^d$ unknown
 - Allows us to use data from one arm to learn about other arms

The Linear Bandits Model

An extension to multi-armed bandits

The Linear Bandits Model

An extension to multi-armed bandits

- In basic multi-armed bandits, goal is to **minimise cumulative regret**
- In linear bandits too, the goal is to **minimise cumulative regret**

The Linear Bandits Model

An extension to multi-armed bandits

- In basic multi-armed bandits, goal is to **minimise cumulative regret**
- In linear bandits too, the goal is to **minimise cumulative regret**
- ⦿ We estimate each μ_i separately (simple averaging) and maintain confidence intervals for each μ_i
- ⦿ We estimate θ from each data point (least-squares) and maintain a confidence ellipsoid around θ

The Linear Bandits Model

An extension to multi-armed bandits

- In basic multi-armed bandits, goal is to **minimise cumulative regret**
- In linear bandits too, the goal is to **minimise cumulative regret**
- ◉ We estimate each μ_i separately (simple averaging) and maintain confidence intervals for each μ_i
- ◉ We estimate θ from each data point (least-squares) and maintain a confidence ellipsoid around θ
- ▶ We apply the **optimism principle**: for each arm, assume its mean is the best it can reasonably be, then play the best arm accordingly
- ▶ We apply the **optimism principle**: for each arm, assume its mean is the best it can reasonably be, then play the best arm accordingly

The Linear Bandits Model

An extension to multi-armed bandits

- In basic multi-armed bandits, goal is to **minimise cumulative regret**
 - ◉ We estimate each μ_i separately (simple averaging) and maintain confidence intervals for each μ_i
 - We apply the **optimism principle**: for each arm, assume its mean is the best it can reasonably be, then play the best arm accordingly
 - ✓ Regret scales linearly with K
- In linear bandits too, the goal is to **minimise cumulative regret**
 - ◉ We estimate θ from each data point (least-squares) and maintain a confidence ellipsoid around θ
 - We apply the **optimism principle**: for each arm, assume its mean is the best it can reasonably be, then play the best arm accordingly
 - ✓ Regret scales linearly with d

The Linear Bandits Model

An extension to multi-armed bandits

[illegible]

The Linear Bandits Model

An extension to multi-armed bandits

[illegible]

The Linear Bandits Model

An extension to multi-armed bandits

Basic Multi-Armed Bandit	Linear Bandit
K unrelated arms	Arms have features $x_i \in \mathbb{R}^d$
Unknown means: $(\mu_1, \dots, \mu_K)$	Shared known parameter: $\theta \in \mathbb{R}^d$
No relation between μ_i	$\mu_i = \langle \theta, x_i \rangle$

The Linear Bandits Model

An extension to multi-armed bandits

Basic Multi-Armed Bandit	Linear Bandit
K unrelated arms	Arms have features $x_i \in \mathbb{R}^d$
Unknown means: $(\mu_1, \dots, \mu_K)$	Shared known parameter: $\theta \in \mathbb{R}^d$
No relation between μ_i	$\mu_i = \langle \theta, x_i \rangle$
Estimate each μ_i separately	Estimate θ from all data

The Linear Bandits Model

An extension to multi-armed bandits

Basic Multi-Armed Bandit	Linear Bandit
K unrelated arms	Arms have features $x_i \in \mathbb{R}^d$
Unknown means: $(\mu_1, \dots, \mu_K)$	Shared known parameter: $\theta \in \mathbb{R}^d$
No relation between μ_i	$\mu_i = \langle \theta, x_i \rangle$
Estimate each μ_i separately	Estimate θ from all data
Confidence interval per arm	Confidence ellipsoid for θ

The Linear Bandits Model

An extension to multi-armed bandits

Basic Multi-Armed Bandit	Linear Bandit
K unrelated arms	Arms have features $x_i \in \mathbb{R}^d$
Unknown means: $(\mu_1, \dots, \mu_K)$	Shared known parameter: $\theta \in \mathbb{R}^d$
No relation between μ_i	$\mu_i = \langle \theta, x_i \rangle$
Estimate each μ_i separately	Estimate θ from all data
Confidence interval per arm	Confidence ellipsoid for θ
Optimism: largest plausible μ_i	Optimism: largest plausible $\langle \theta, x_i \rangle$

The Linear Bandits Model

An extension to multi-armed bandits

Basic Multi-Armed Bandit	Linear Bandit
K unrelated arms	Arms have features $x_i \in \mathbb{R}^d$
Unknown means: $(\mu_1, \dots, \mu_K)$	Shared known parameter: $\theta \in \mathbb{R}^d$
No relation between μ_i	$\mu_i = \langle \theta, x_i \rangle$
Estimate each μ_i separately	Estimate θ from all data
Confidence interval per arm	Confidence ellipsoid for θ
Optimism: largest plausible μ_i	Optimism: largest plausible $\langle \theta, x_i \rangle$
Regret depends on K	Dependence shifts to d

Today's Lecture

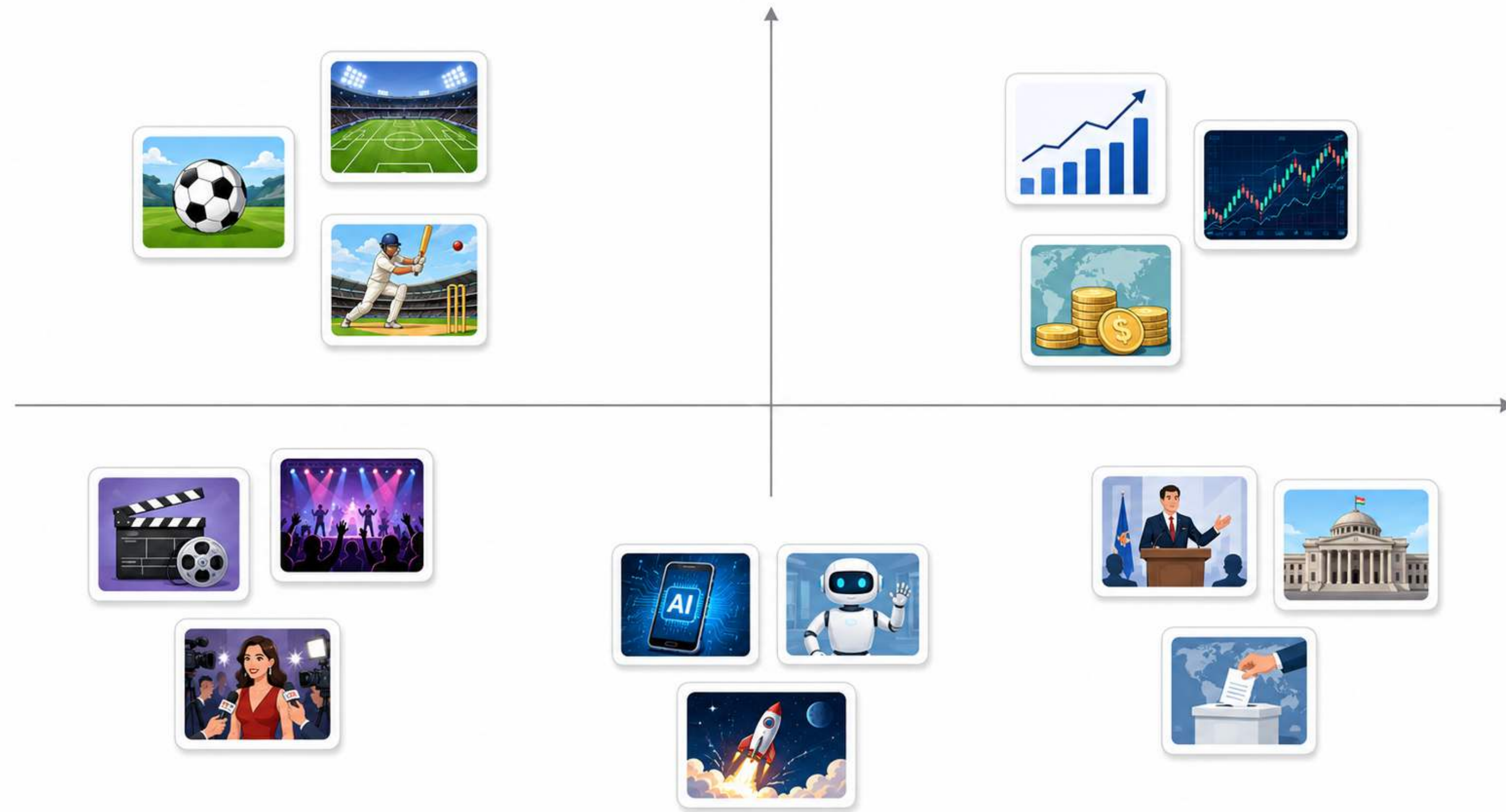
Outline

- What is the interpretation of the context vector x and unknown vector θ ?
- How do we obtain the context vector x in practice?
- Can the context set $\mathcal{X} = \{x_1, \dots, x_K\}$ change with time? How does that change the algorithm?
- Can the model be generalised to make personalised decisions?

This lecture is about the practical aspects of linear bandits

Interpreting Context Vectors

- We know that $\mu_i = \langle \theta, x_i \rangle$
- This means, arms i, j with $x_i \approx x_j$ will have $\mu_i \approx \mu_j$ (for any θ)
- Intuitively, arms that are ‘similar’ should have similar rewards; closer the arms, closer the rewards
- So context vector x_i should capture some semantic meaning about arm i



Representative image of context vectors of different news articles.
Articles of similar topics should have context vectors that are close together

Why Linear Functions?

- The model says $\mu_i = \langle \theta, x_i \rangle$
- In other words, $y(t) = \langle \theta, x(t) \rangle + \eta_t$
- This model gives us a closed-form solution for $\hat{\theta}$
 - The solution can be easily iterated at each time t
- More importantly, it gives us a confidence set with theoretical guarantees:
 - With probability at least $1 - \delta$, $\|\theta - \hat{\theta}\|_{V_t} \leq \beta$

Is the reward truly a linear function of the arm features?

Is Linearity Too Restrictive?

- The model says $\mu_i = \langle \theta, x_i \rangle$
 - **But x_i need not be the raw item features**
- Raw input (article text, image, user profile) is passed through a feature map φ

$$x_i = \varphi(\text{raw item } i)$$

- φ must be **fixed before** we run the bandit algorithm
 - The bandit learns θ , not φ
- Richer $\varphi \Rightarrow$ larger d . More expressive, but slower learning (larger regret)

The real question is, how to choose φ ?

Where Do Context Vectors Come From?

Where Do Context Vectors Come From?

Let us understand with the example of news article recommendation

Hand-Crafted Features

Coarse Features

- Each news article has a topic/category:
 - Encode category as one-hot or multi-hot vector
 - d categories $\Rightarrow d$ -dimensional context vectors
- ✓ Low-dimensional, interpretable features
- Misses subtler differences within a category

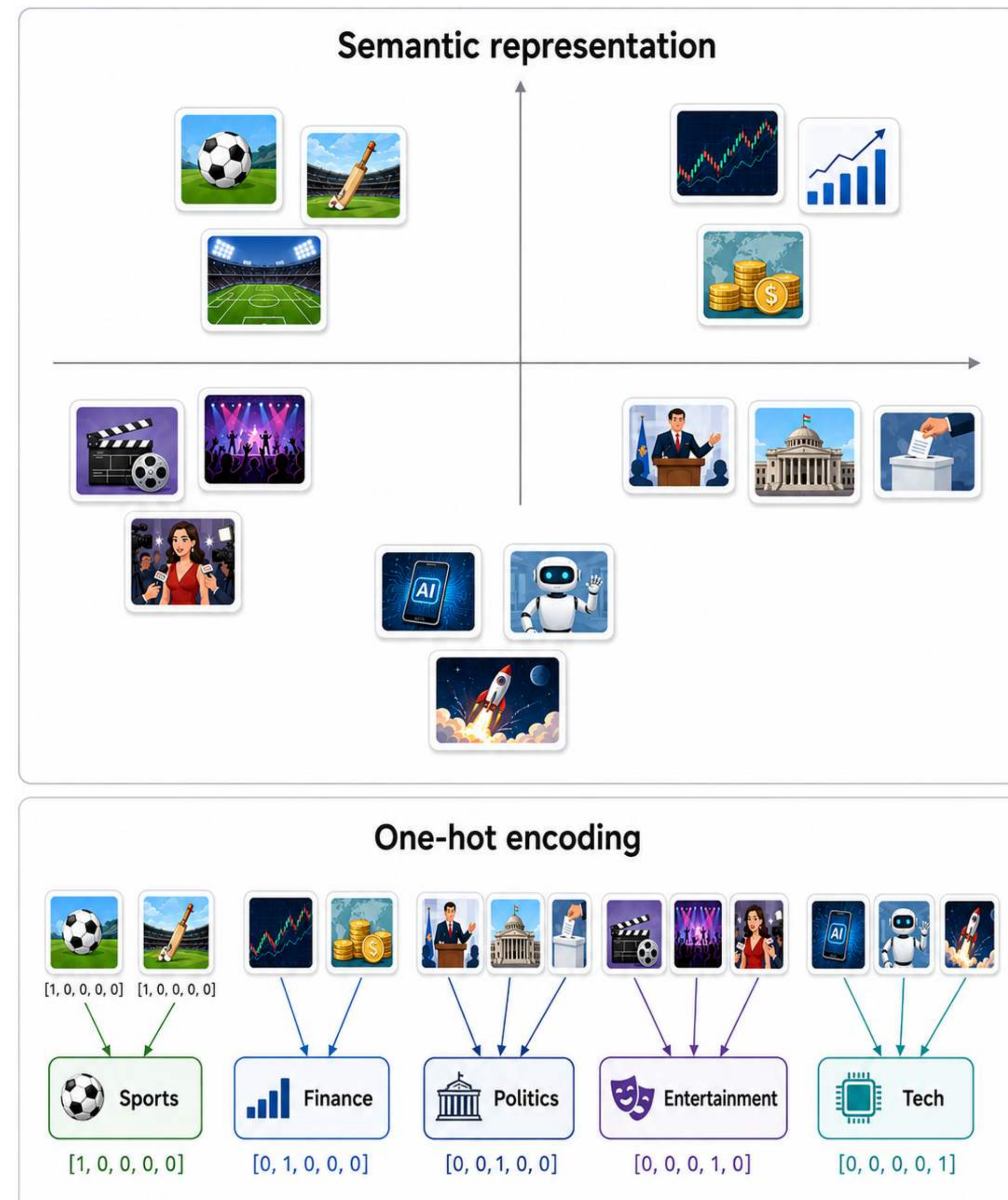
Hand-Crafted Features

Coarse Features

- Each news article has a topic/category:
 - Encode category as one-hot or multi-hot vector
 - d categories $\Rightarrow d$ -dimensional context vectors

✓ Low-dimensional, interpretable features

- Misses subtler differences within a category



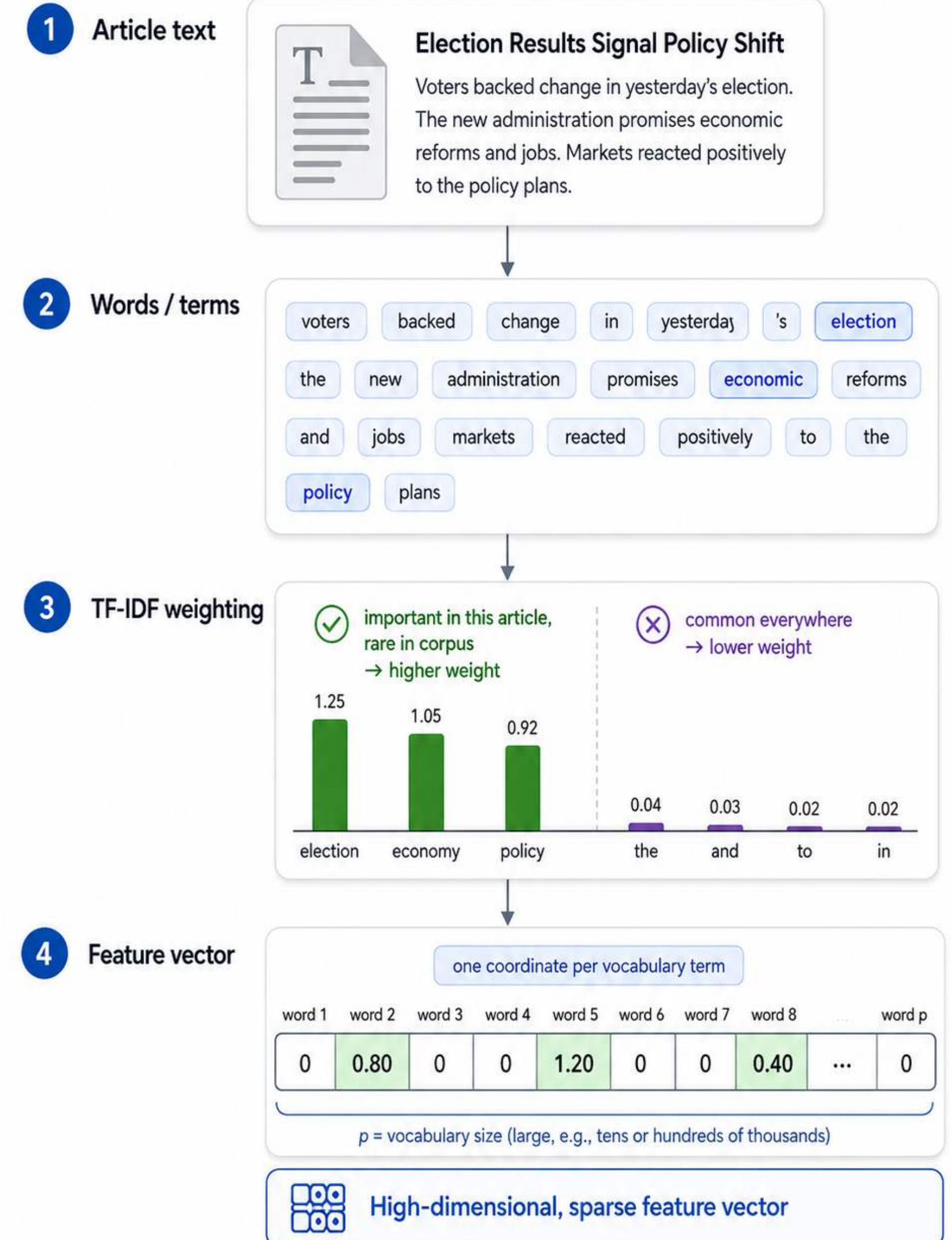
No notion that politics is closer to finance than entertainment

Hand-Crafted Features

Fine Features

- Each news article is a piece of text
 - Use calculate text-based features: bag-of-words, TF-IDF, BM25, etc.
 - Each word/term becomes a possible feature
- ✓ Captures richer item information
- Very high-dimensional $\Rightarrow$ poor learning

TF-IDF features

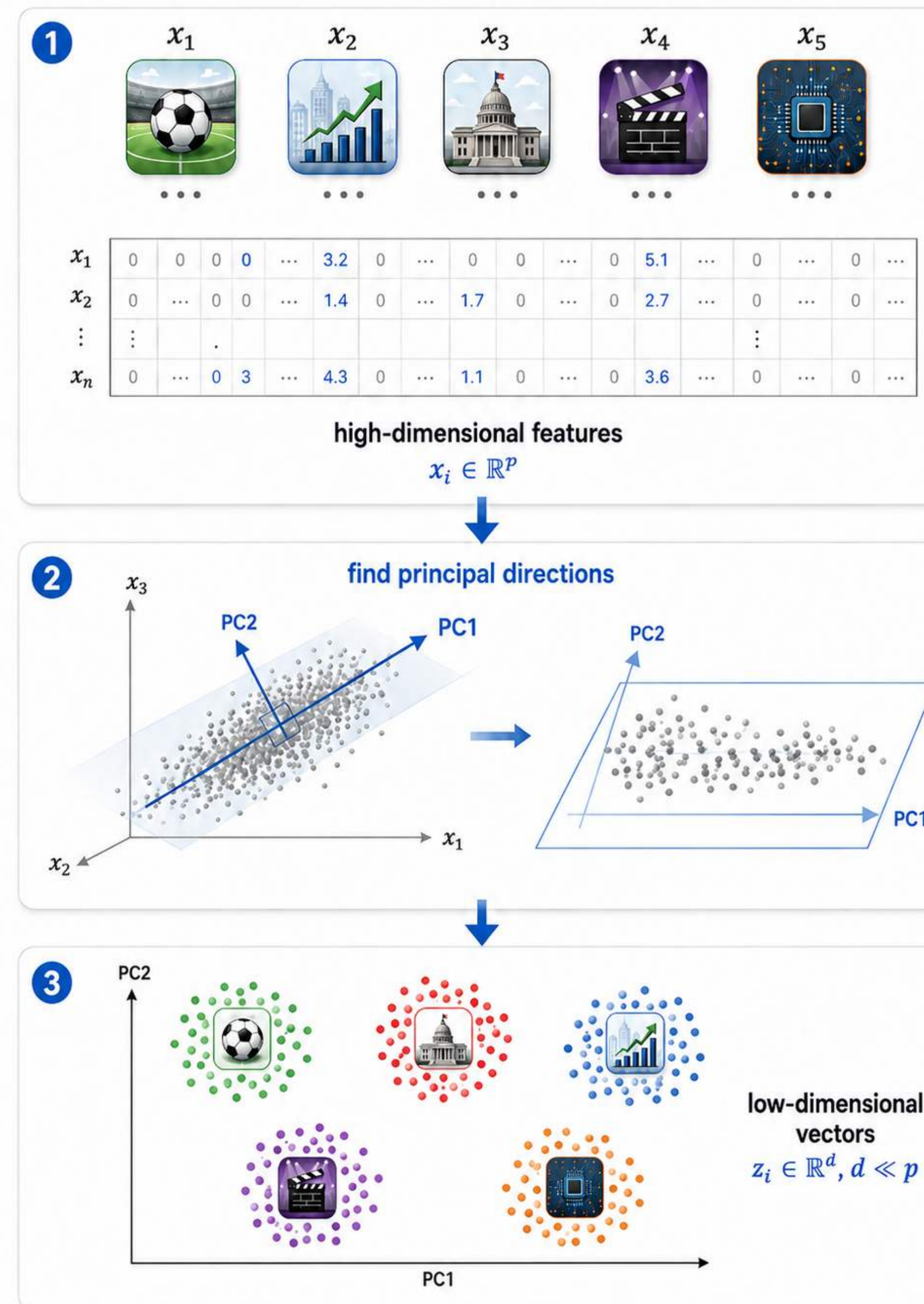


Hand-Crafted Features

Dimensionality Reduction

- Start with high-dimensional item vectors x_i
- We can do dimensionality reduction, e.g. via Principle Component Analysis (PCA)
- It takes projections of each x_i onto low-dimensional space, preserving distinctive qualities
- ✓ Keeps richer signals in fewer dimensions
 - Loses direct interpretability of x_i

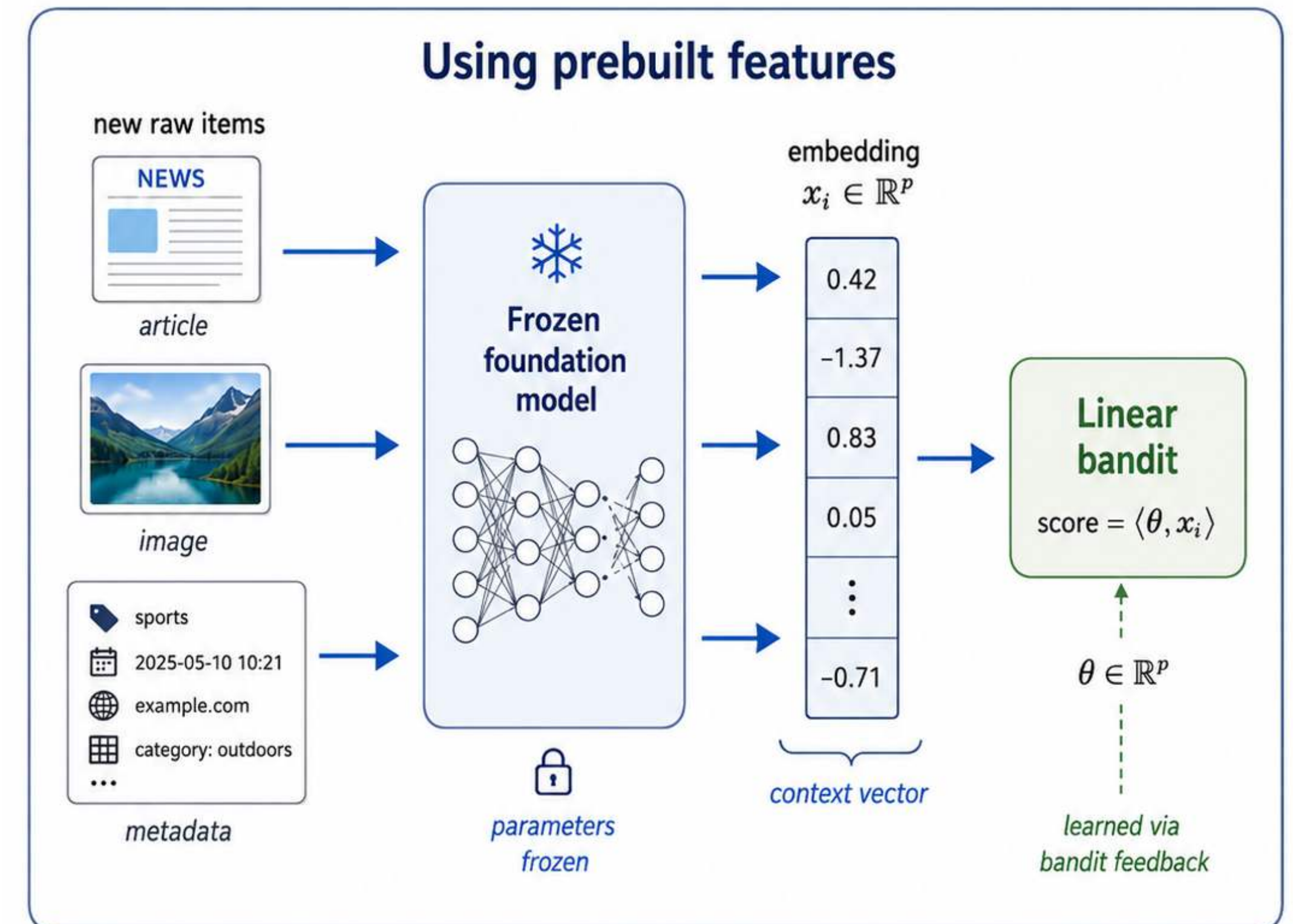
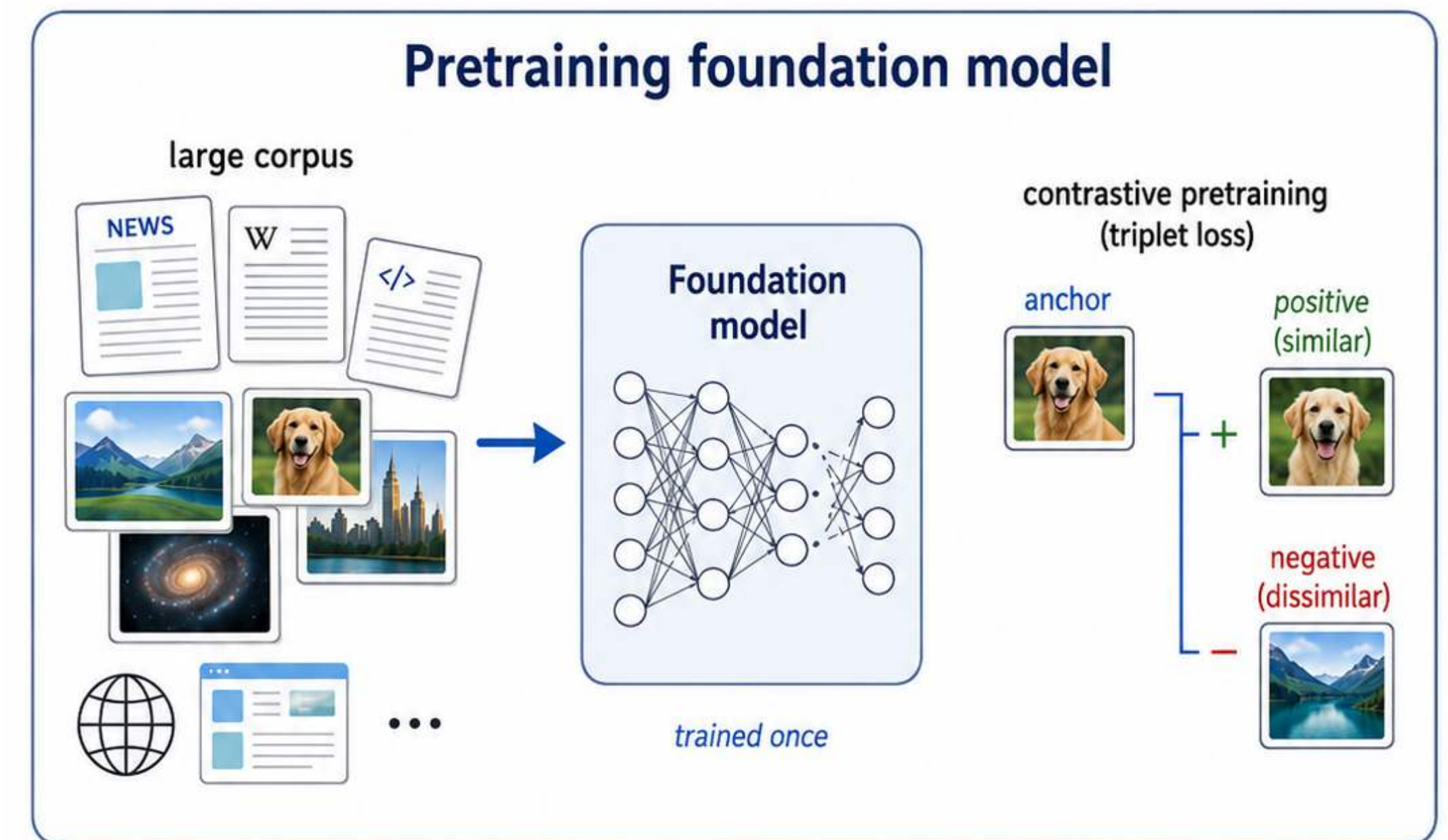
PCA for dimensionality reduction



Neural Network Features

Off-the-shelf foundation models

- Foundation models are gigantic neural networks trained on a huge corpus of data
- They learn embeddings that reflect semantic similarity
 - CLIP-style embeddings for images
 - BERT-style embeddings for text
- ✓ Captures rich semantics from complex data types
- High dimensional features ($d = 768, 1024$), potentially *not aligned* with task at hand



Neural Network Features

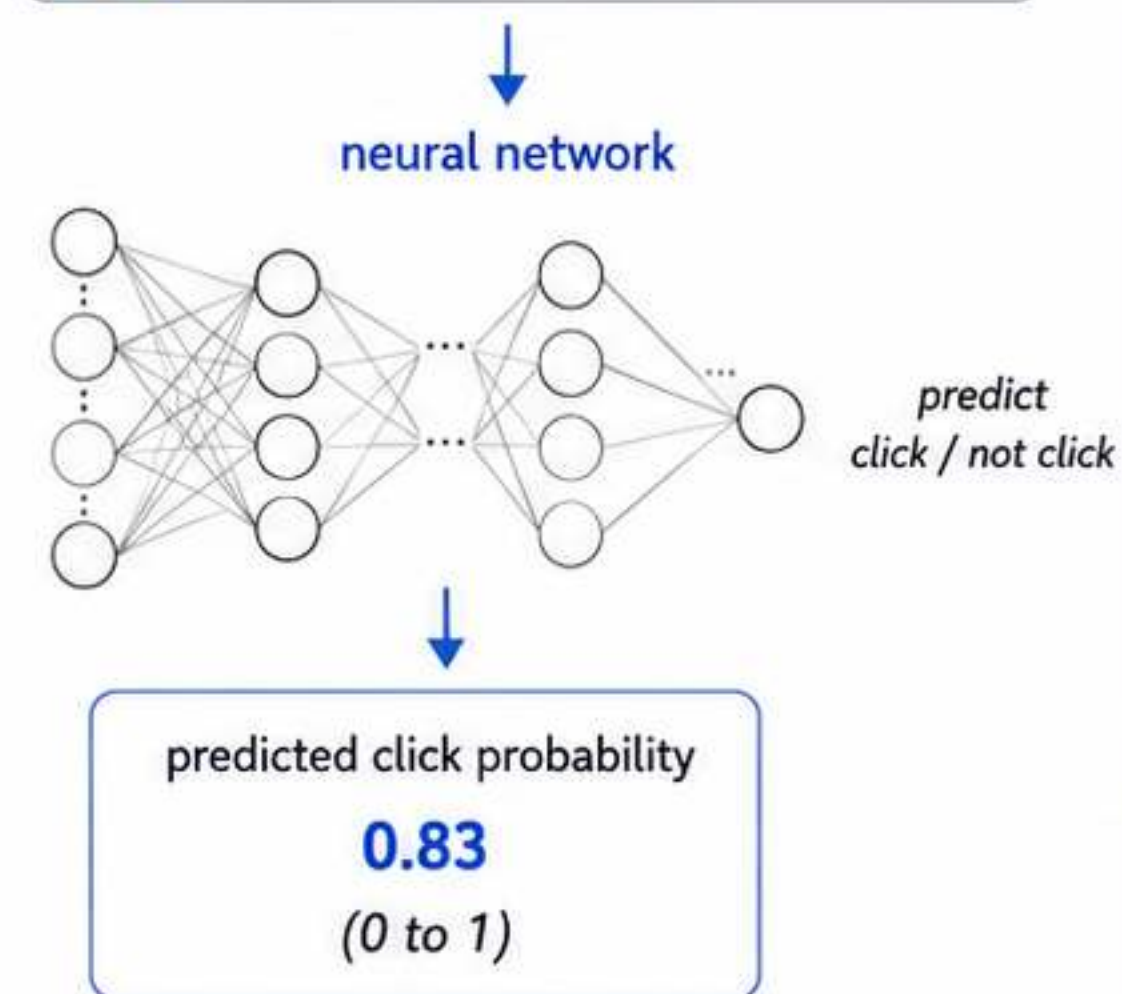
Training a neural network ourselves

- We have logged data: which articles were shown, clicked/not clicked
- Train a neural network to predict clicks
- Remove the last layer and use the rest as φ
- ✓ Extracts rich features relevant to the task at hand
- Needs lots of past data to be effective

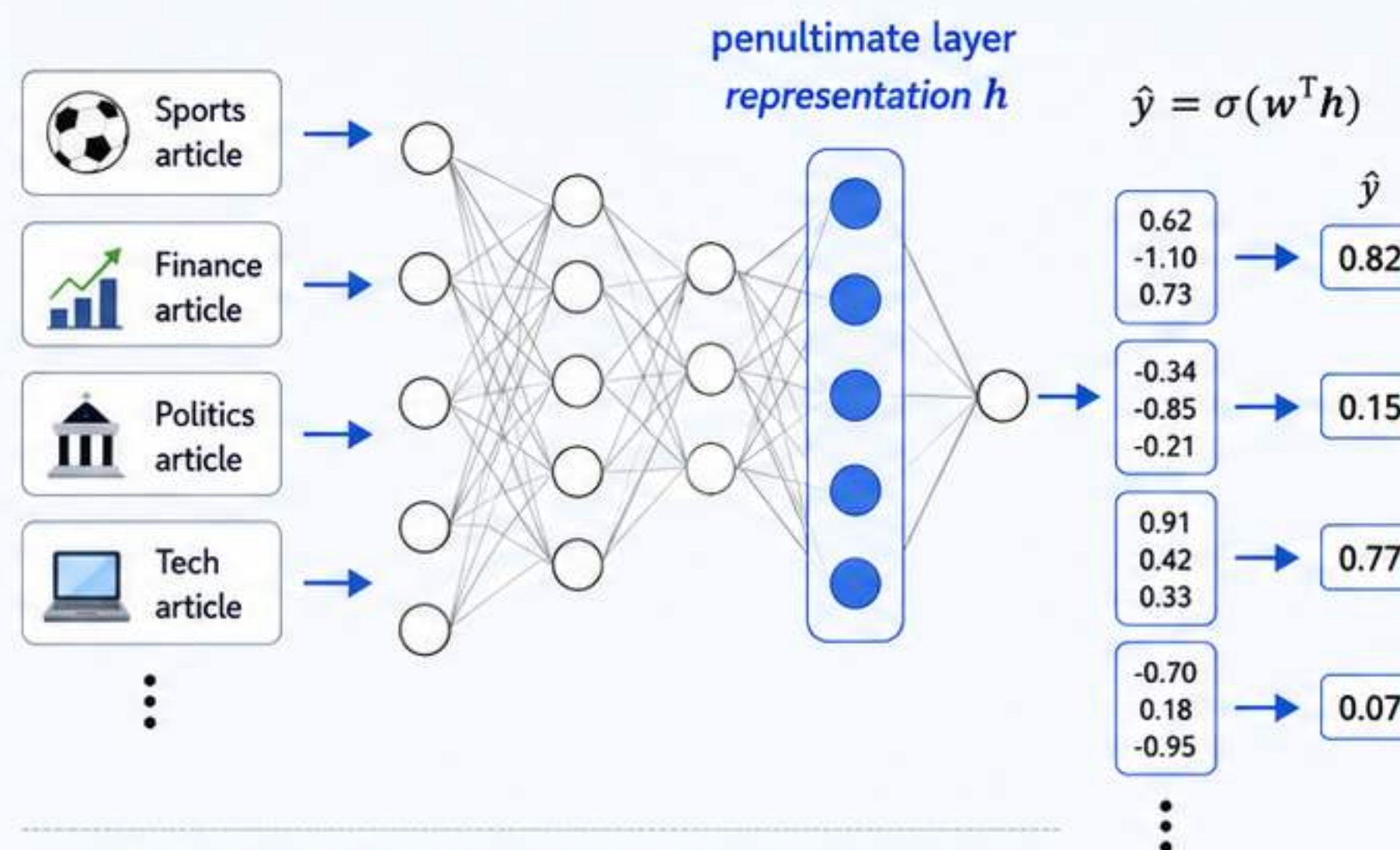
Neural Network Features

Training a neural network ourselves

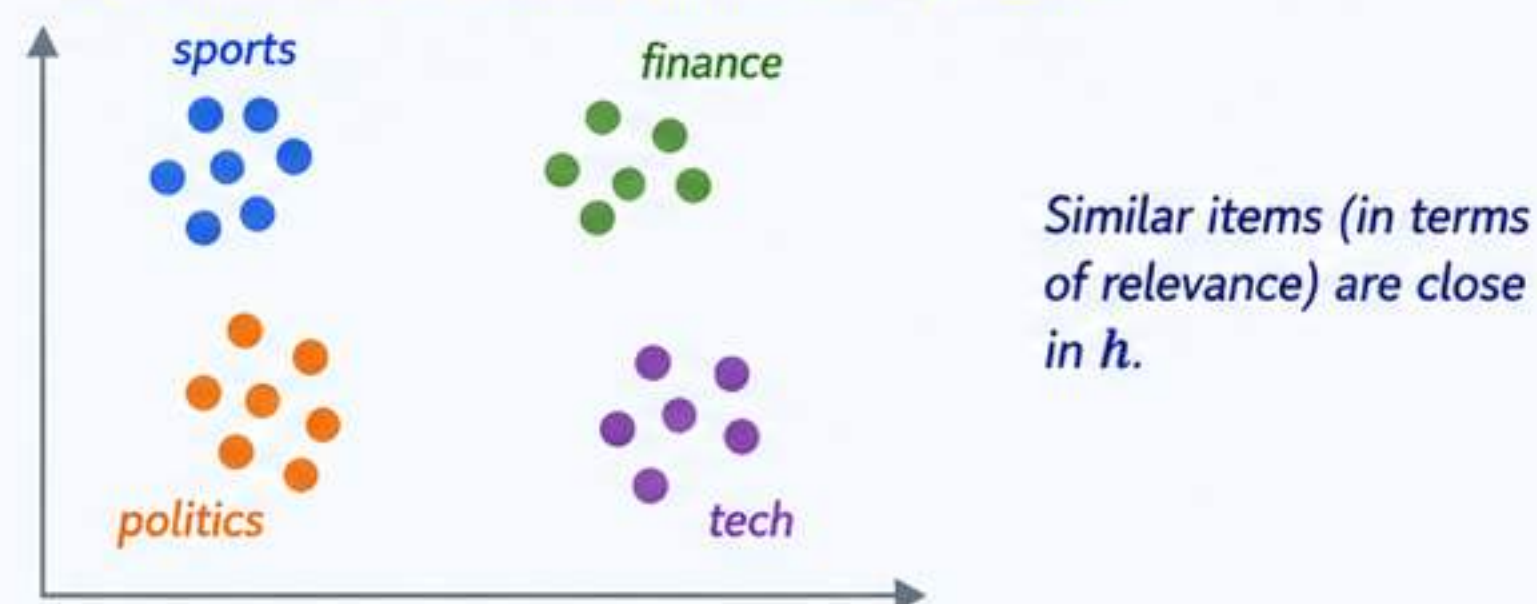
1 Train on historical data



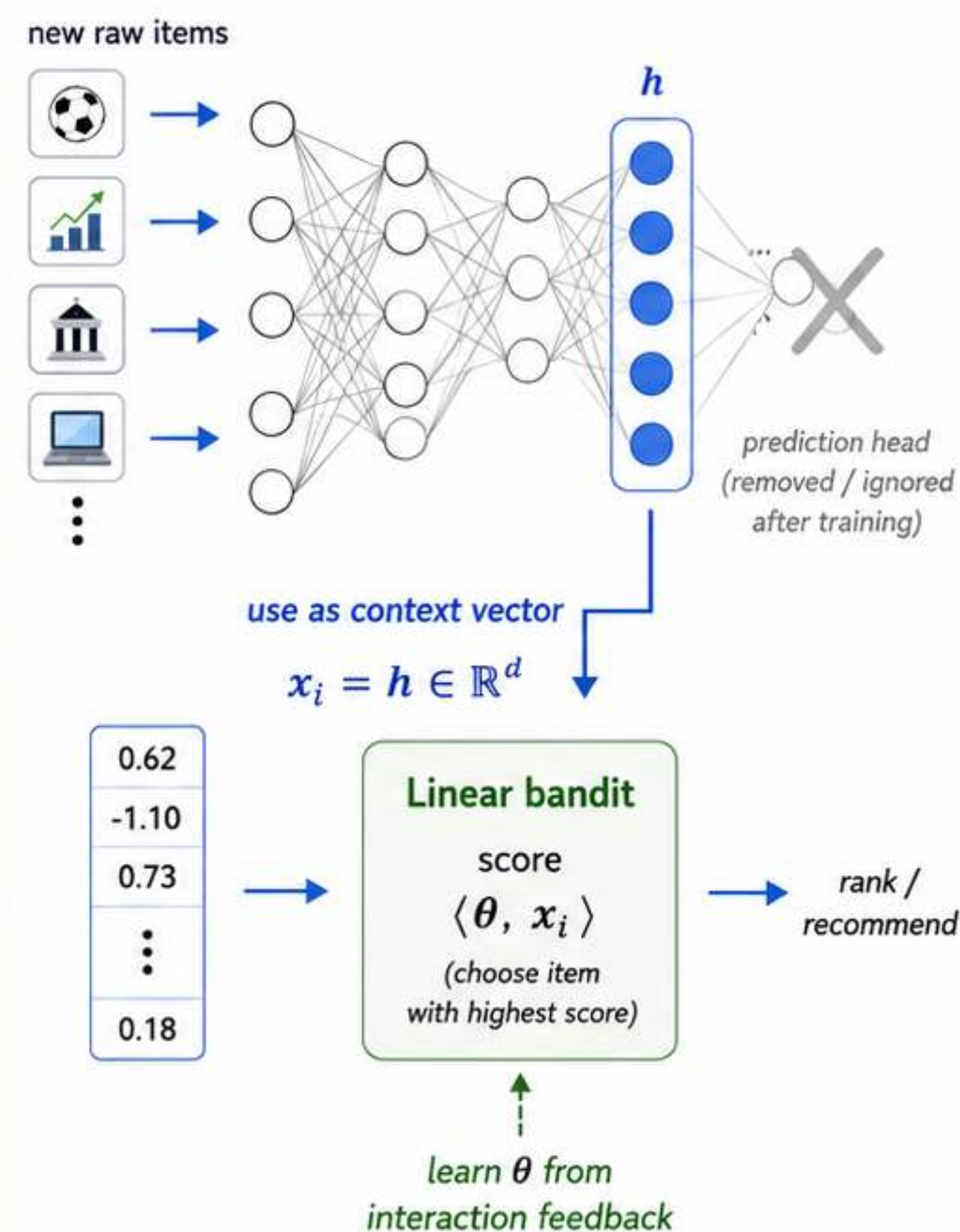
2 Why the penultimate layer matters



To predict clicks well, h must capture useful patterns.



3 Use h as the context vector

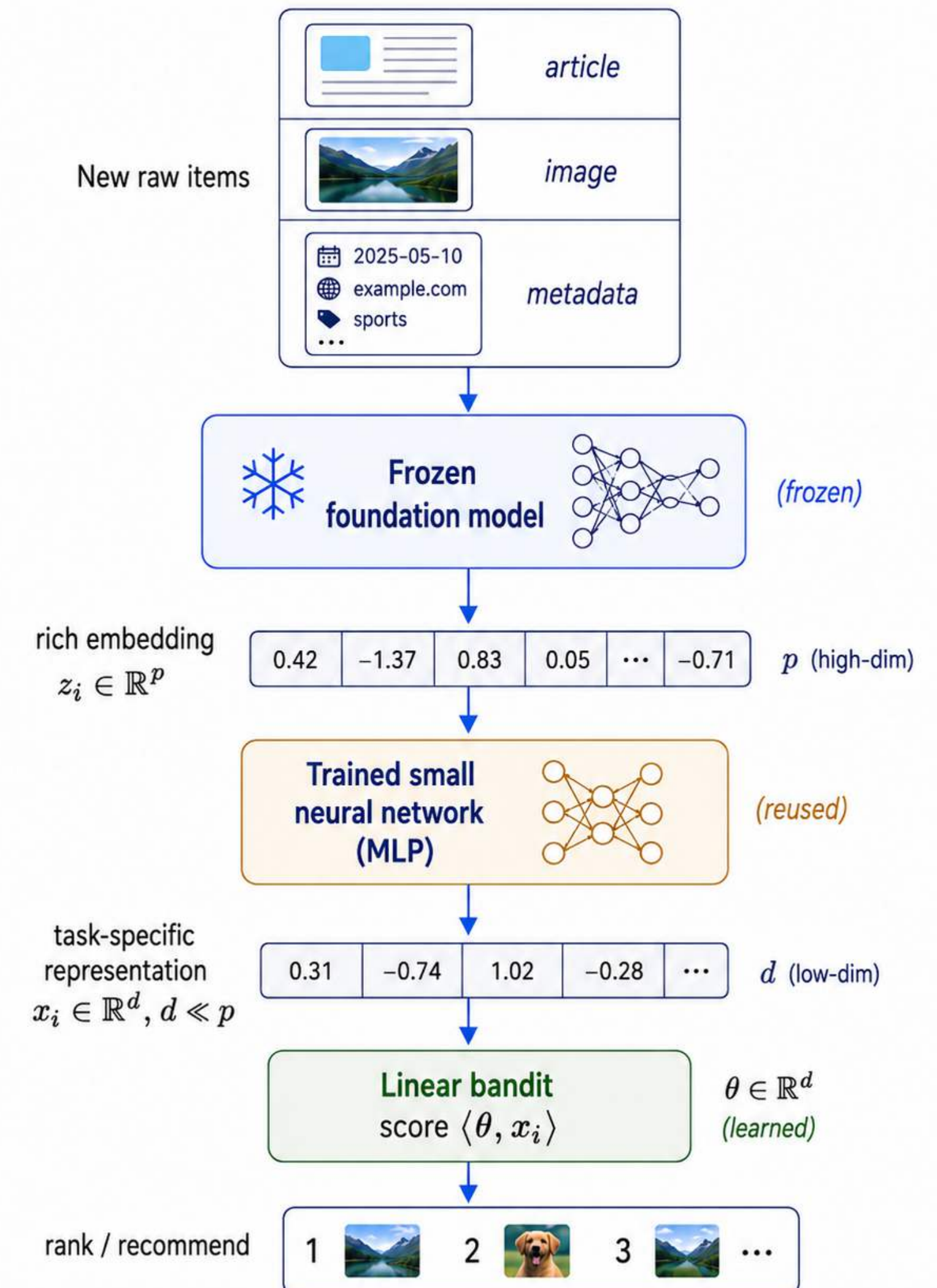


Neural Network Features

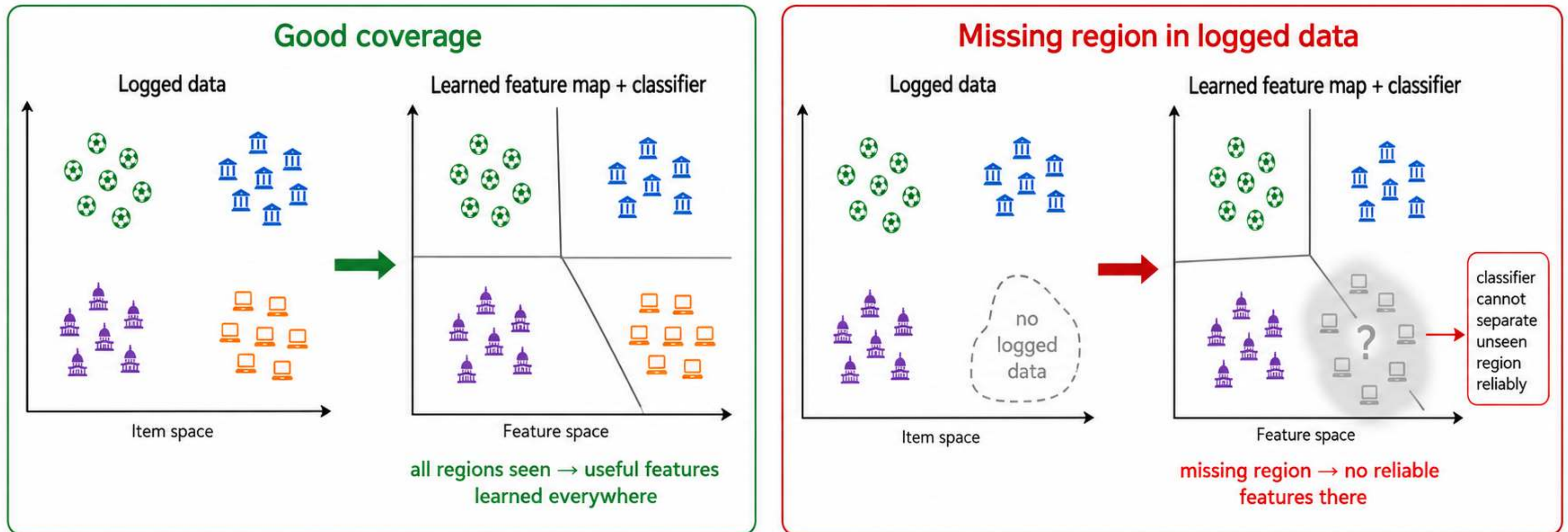
Modern Paradigm

- Use foundation models and custom network, back-to-back
- Foundation model gives rich embedding
- Small custom network maps it to task-specific feature
- Bandit algorithm works with this low-dimensional representation
- ✓ Rich features, less data hungry
- Still depends on good exploratory logged data

Using the learned representation in the bandit



Why Exploratory Logging Matters



If a whole region is never explored in the logged data, the supervised learner cannot learn a good representation for that region.

Final Words on Feature Engineering

Bandit-Specific Caveats

Expressiveness

Dimension d

Interpretability

- Feature scaling matters: if some items have large features and others have very small, then the large feature items will automatically win!
-

Extending The Linear Bandit Model

Time-Varying Action Sets

- The set of arms available may not remain fixed throughout
 - News articles: new ones published, old ones retired
 - Products: new ones listed, old ones out of stock
 - Ads on Google: ad pool changes as a function of search

But there are interesting consequences on the learning dynamics

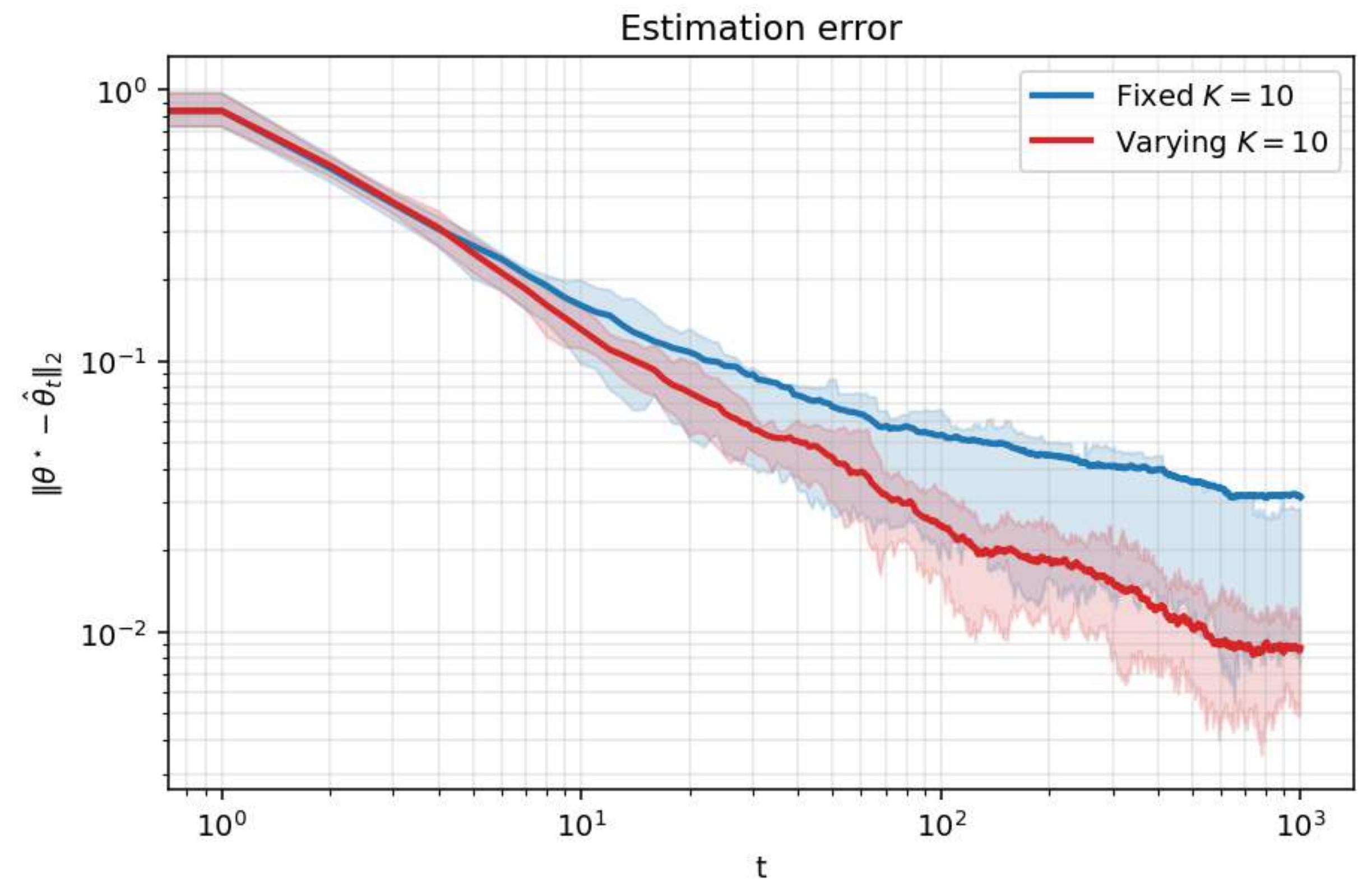
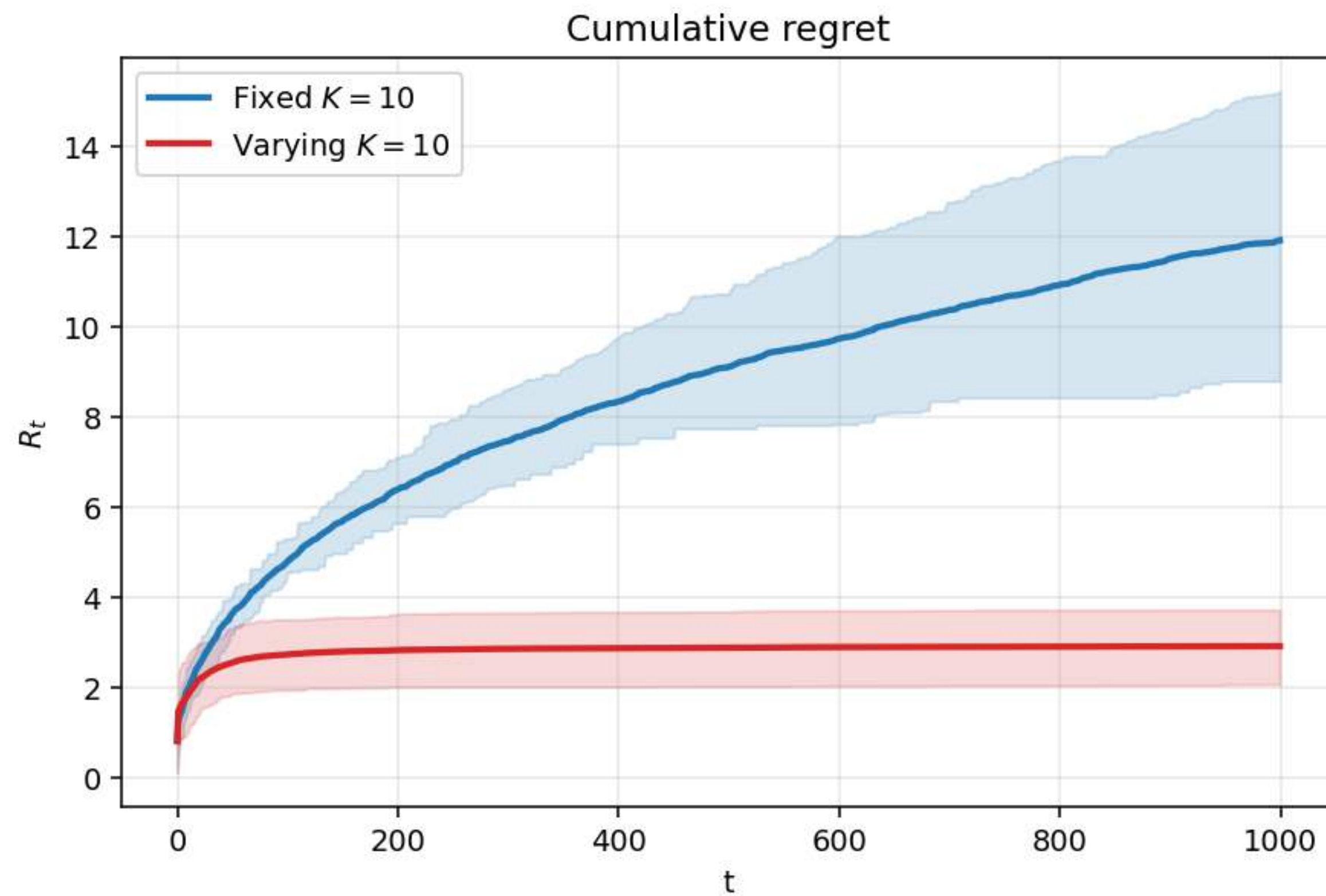
Time-Varying Action Sets

- The set of arms available may not remain fixed throughout
 - News articles: new ones published, old ones retired
 - Products: new ones listed, old ones out of stock
 - Ads on Google: ad pool changes as a function of search
- The linear bandit algorithms we have studied can handle this case!
- Simply compute $\arg \max_{x \in \mathcal{X}_t} \langle \hat{\theta}_t, x \rangle + \sqrt{\beta} \|x\|_{V_t^{-1}}: \mathcal{X}_t = \{x_{1,t}, x_{2,t}, \dots, x_{K,t}\}$

But there are interesting consequences on the learning dynamics

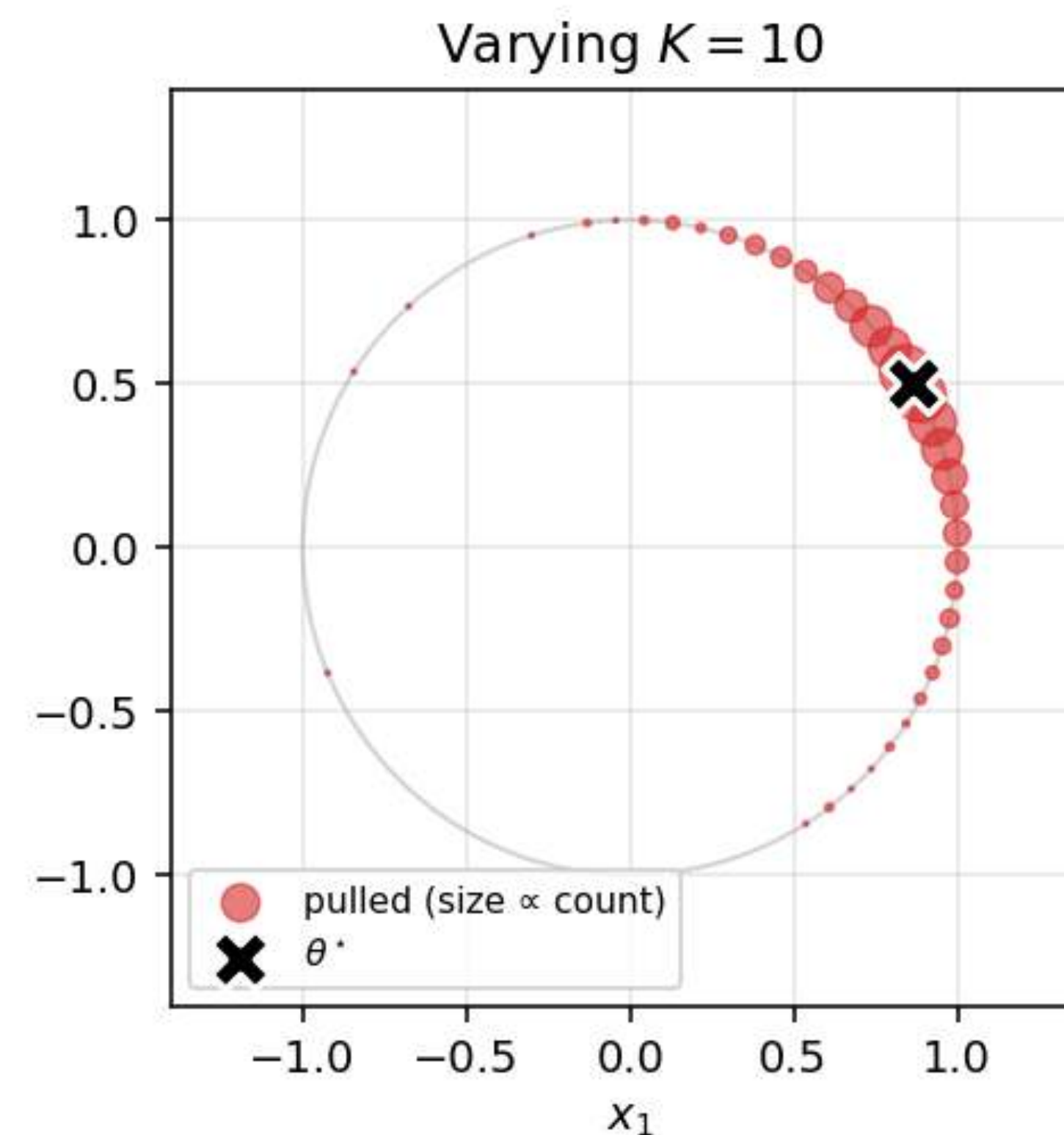
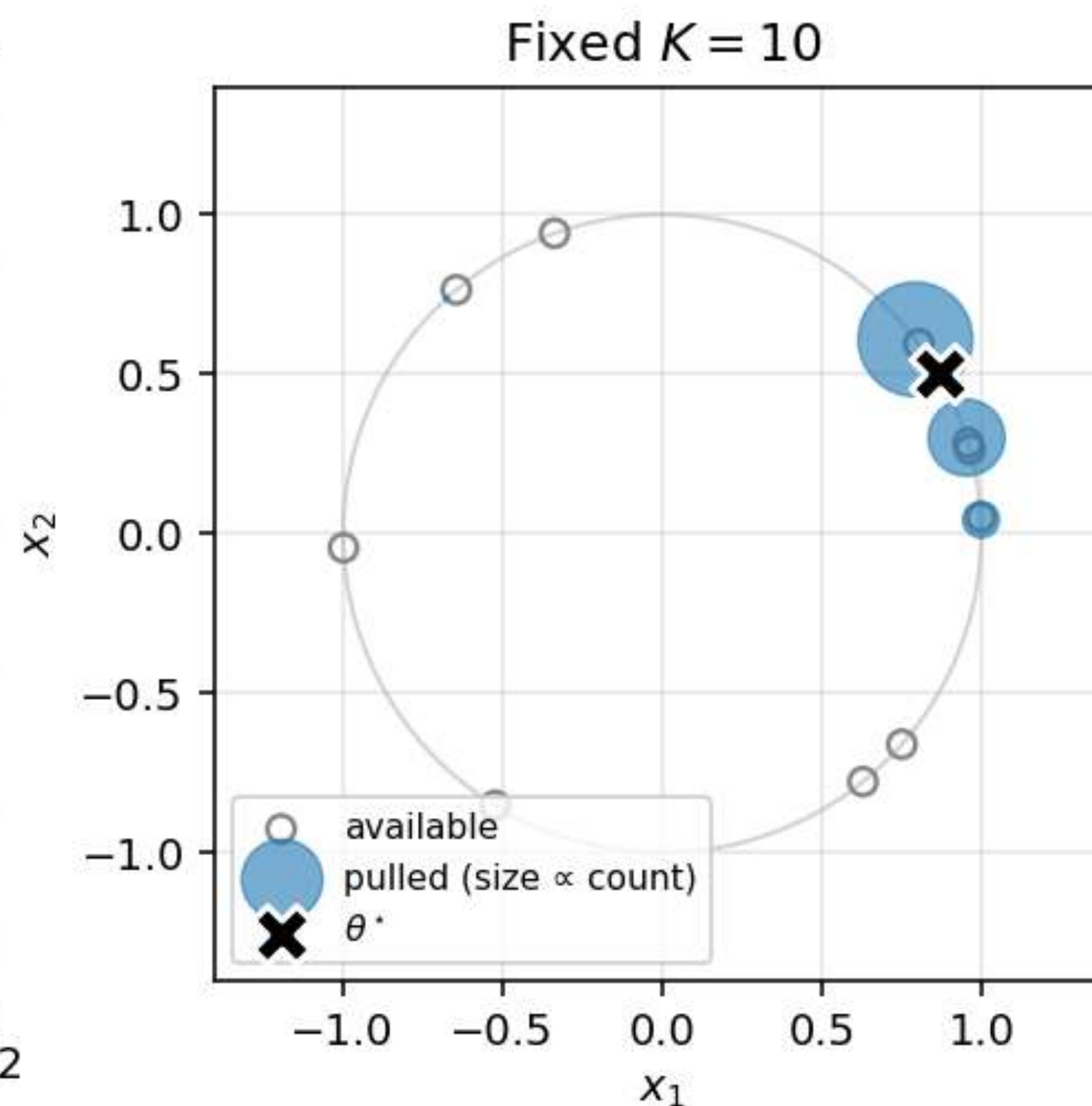
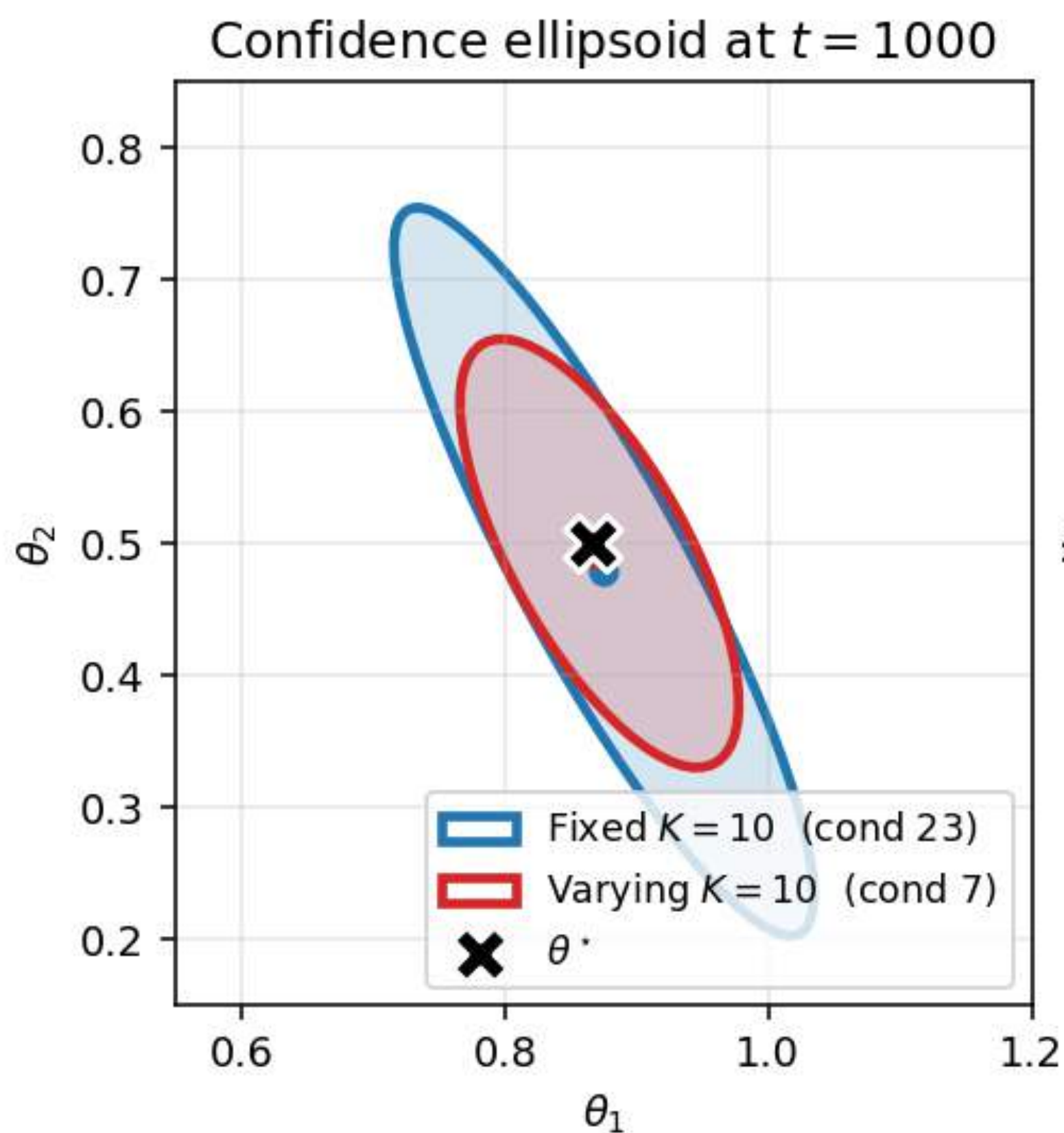
Regret and Estimation Error

Fixed context versus varying context: $K = 10$



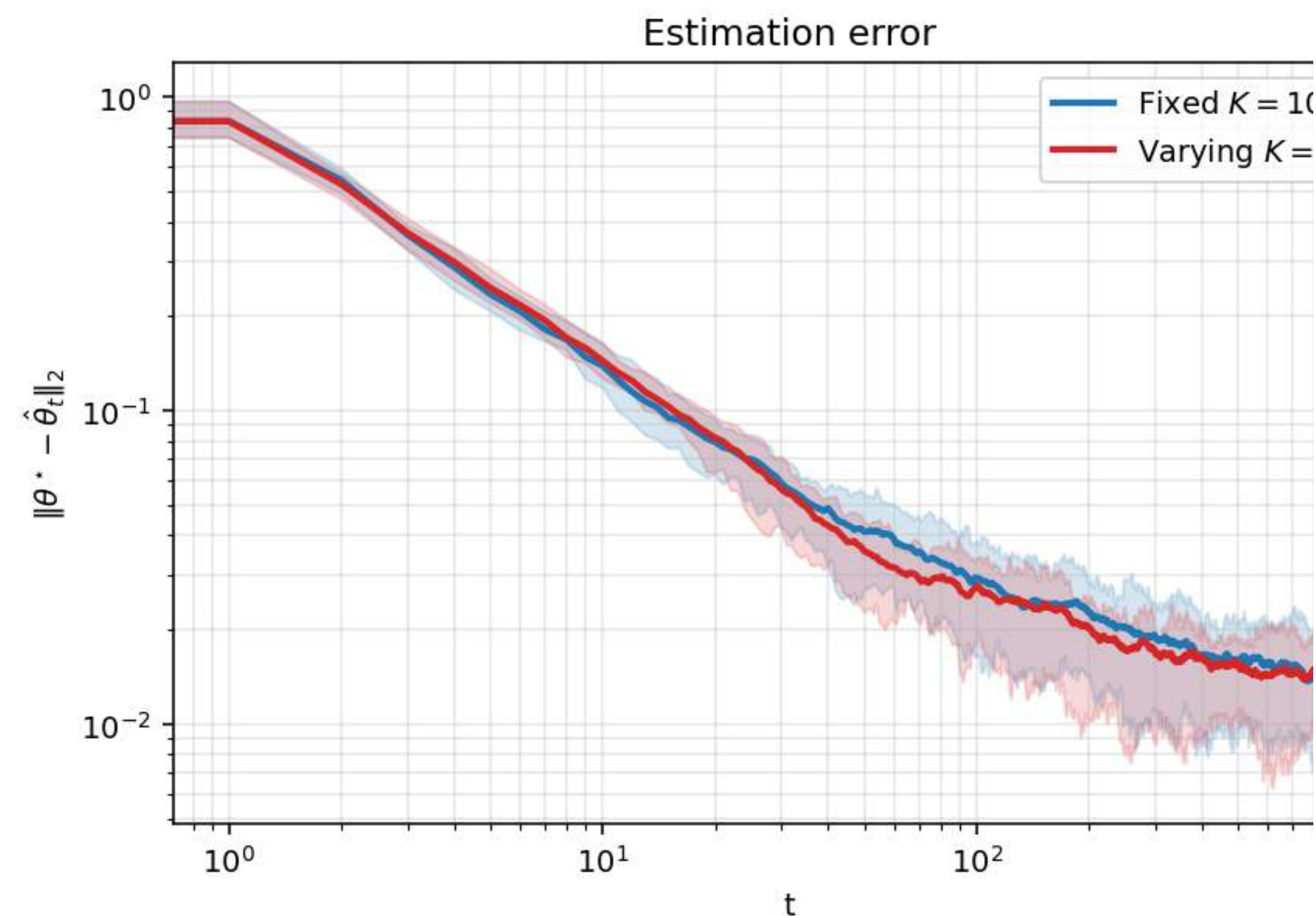
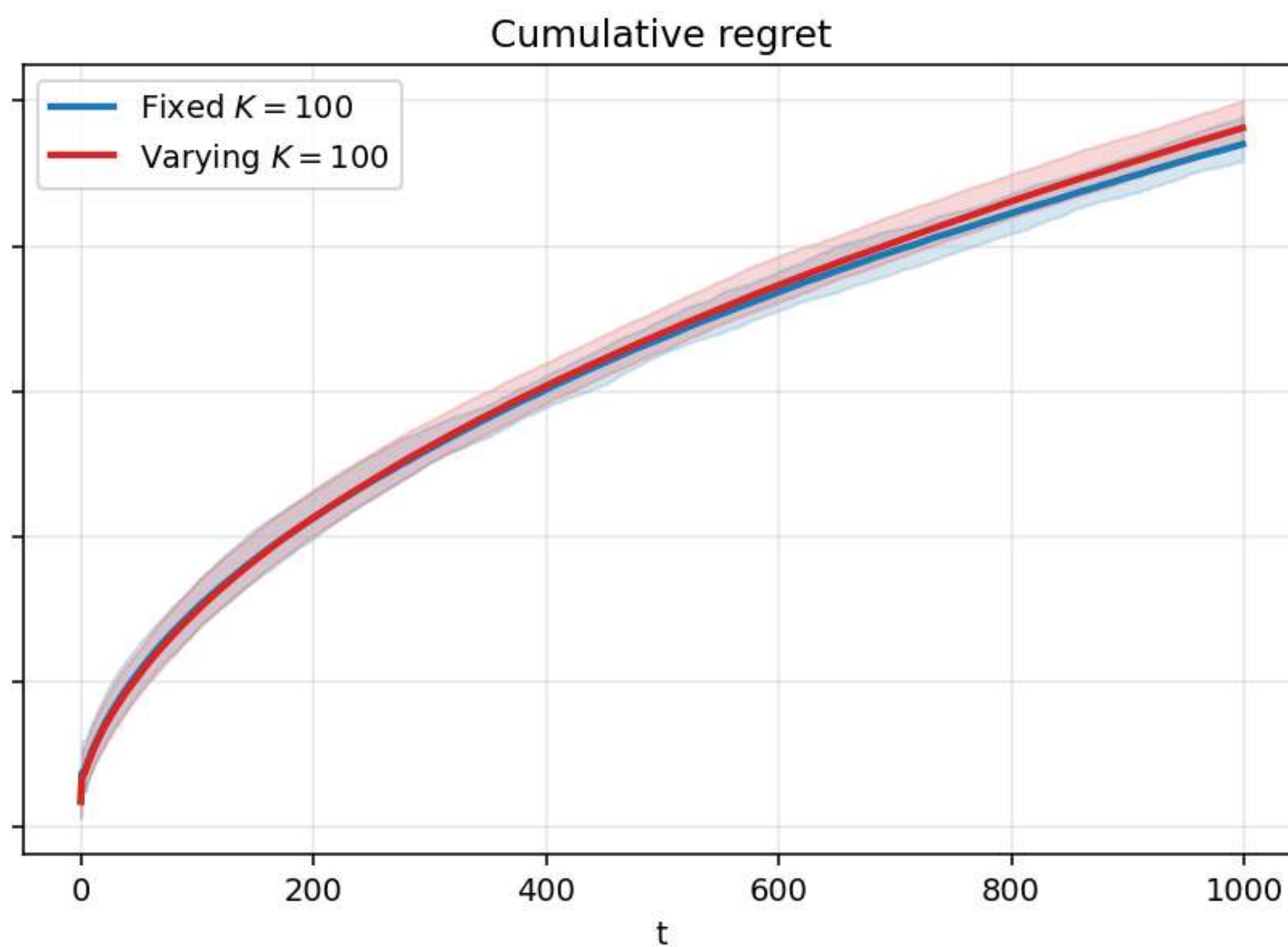
Confidence Ellipsoids and Arm Scatter Plots

Fixed context versus varying context: $K = 10$



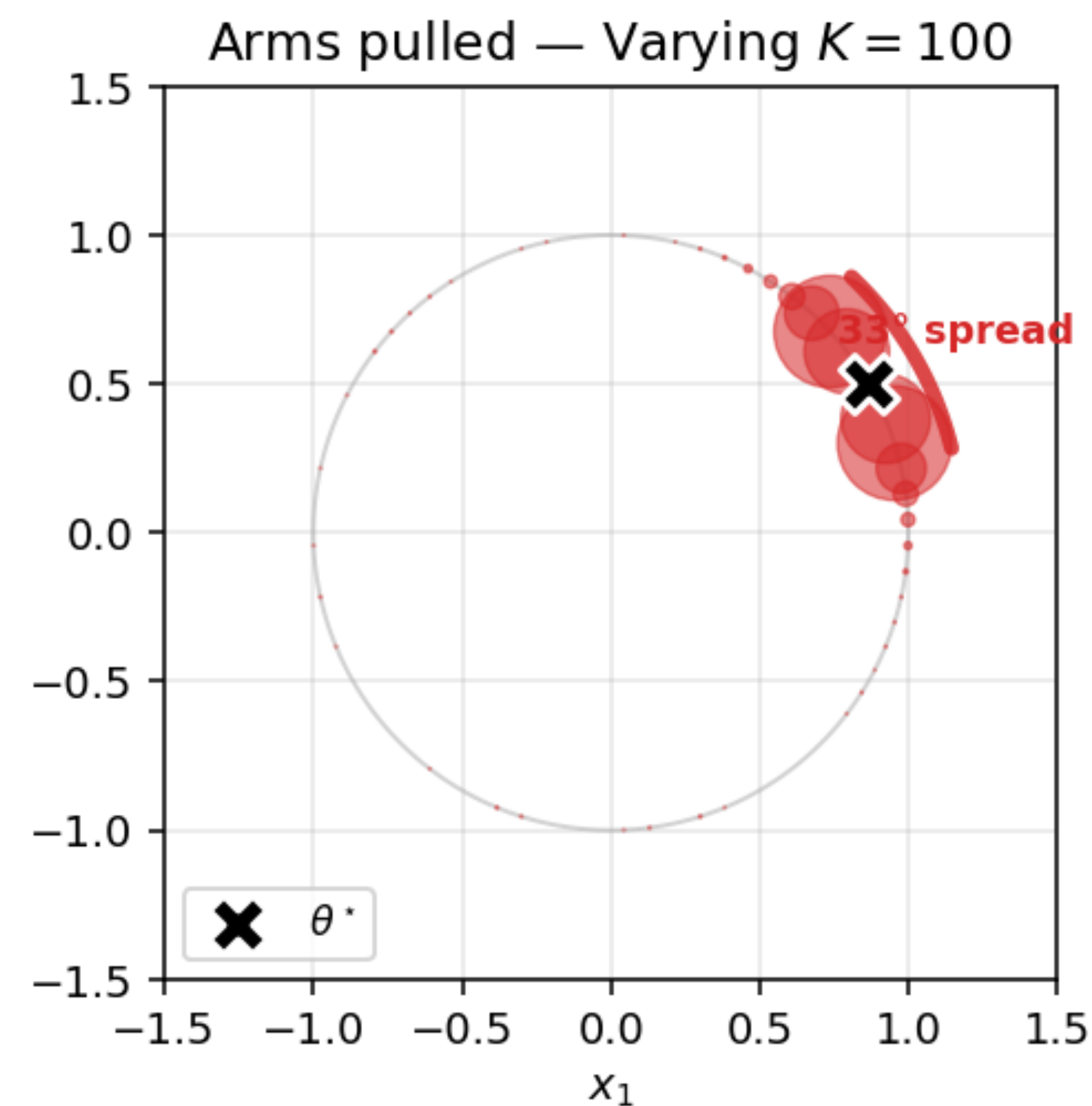
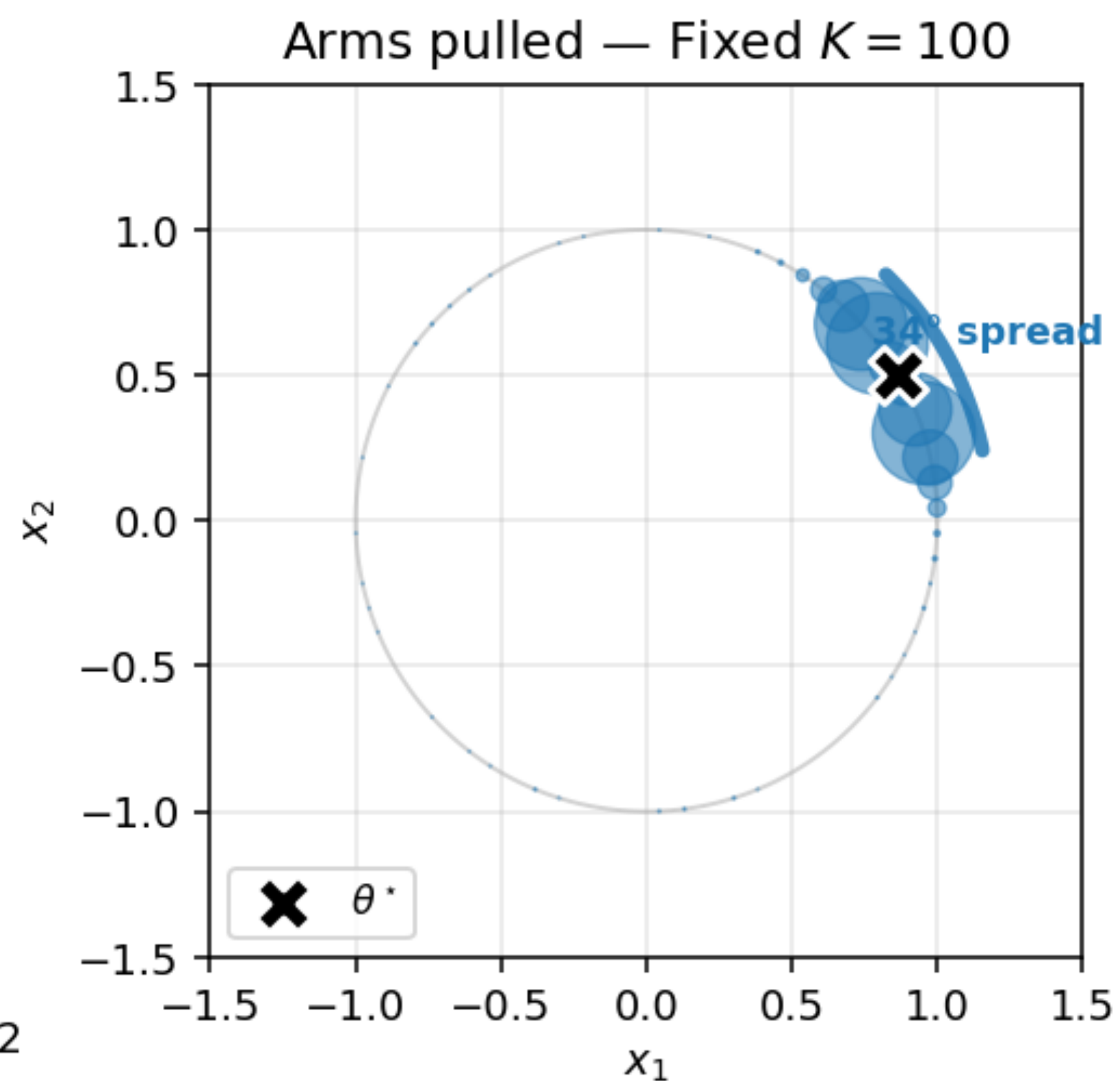
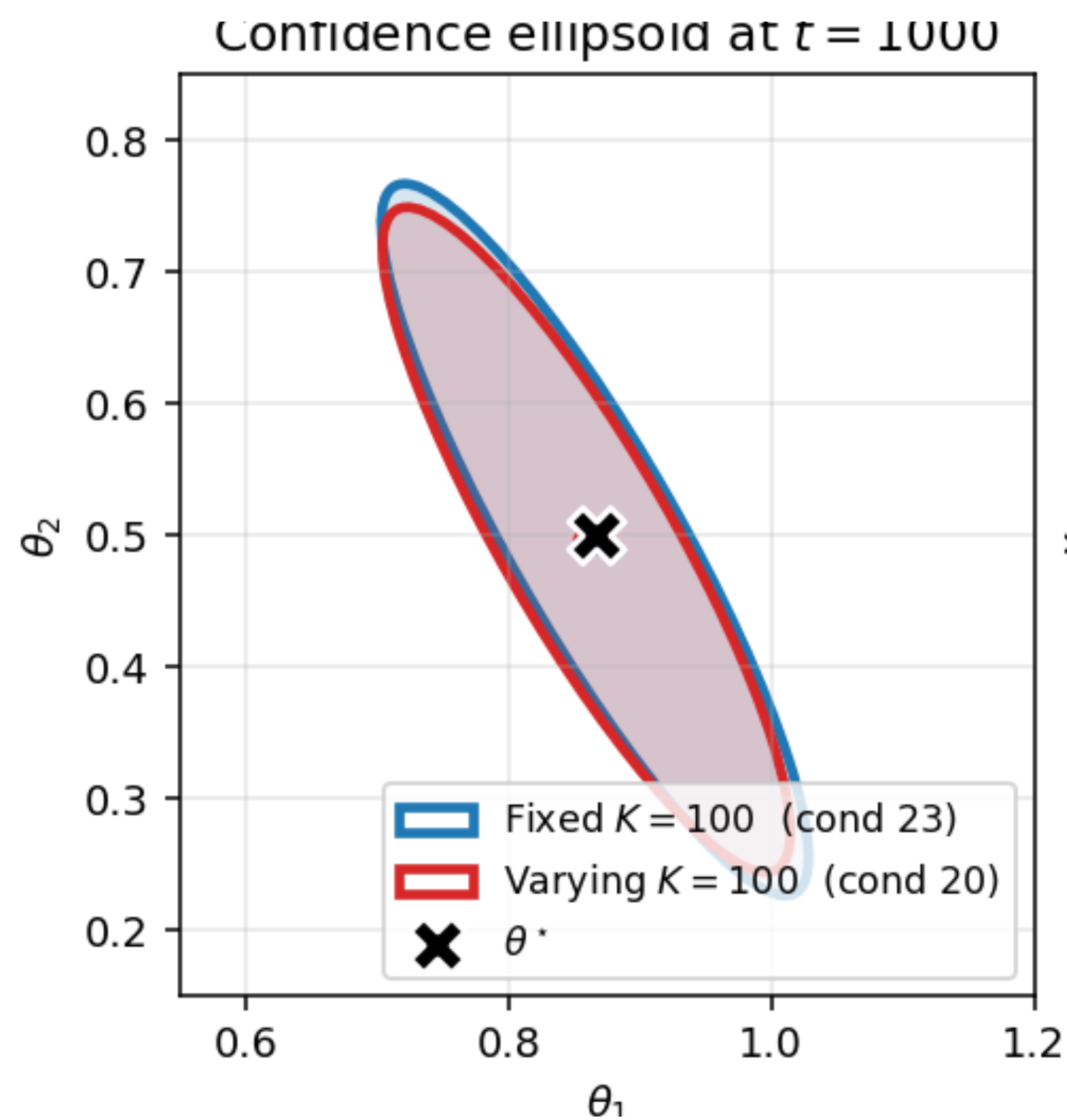
Regret and Estimation Error

Fixed context versus varying context: $K = 100$



Confidence Ellipsoids and Arm Scatter Plots

Fixed context versus varying context: $K = 100$



Can Linear Bandits Give Personalised Recommendations?

Benefits of Personalisation

- In the current model of news article recommendation, x captures news item features
- $\mu_i = \langle \theta, x_i \rangle$: this model implies that there is an overall best ‘type’ of news article, characterised by θ

Benefits of Personalisation

- In the current model of news article recommendation, x captures news item features
- $\mu_i = \langle \theta, x_i \rangle$: this model implies that there is an overall best ‘type’ of news article, characterised by θ
- In reality, user preferences vary greatly
 - Some may prefer sports, others may love politics

Benefits of Personalisation

- In the current model of news article recommendation, x captures news item features
- $\mu_i = \langle \theta, x_i \rangle$: this model implies that there is an overall best ‘type’ of news article, characterised by θ
- In reality, user preferences vary greatly
 - Some may prefer sports, others may love politics
- How can we bring in user identity into the bandit problem?

Three Options for Personalisation

Three Options for Personalisation

- Option 1: Have a separate bandit per user
 - θ_u for user u , interpreted as the user's feature vector
 - Known facts about the user can be incorporated as a prior

Three Options for Personalisation

- Option 1: Have a separate bandit per user
 - θ_u for user u , interpreted as the user's feature vector
 - Known facts about the user can be incorporated as a prior
- Option 2: Cluster users according to type, one bandit per cluster
 - Use past data about users to cluster
 - Need some time to put a new user into a cluster

Three Options for Personalisation

- Option 1: Have a separate bandit per user
 - θ_u for user u , interpreted as the user's feature vector
 - Known facts about the user can be incorporated as a prior
- Option 2: Cluster users according to type, one bandit per cluster
 - Use past data about users to cluster
 - Need some time to put a new user into a cluster
- Option 3: Incorporate user features in x
 - $x_{t,i} = \varphi(\text{user at time } t, \text{item } i)$
 - One shared θ , interpreted as weights given to each feature

What Works In Practice?

What Works In Practice?

- Per user θ_u : Data from one user cannot be used for another user.
 - Does not scale—too little data per user

What Works In Practice?

- Per user θ_u : Data from one user cannot be used for another user.
 - Does not scale—too little data per user
- Clustered users: Used by Deezer to recommend playlists (Bendada et al, 2020)
 - Found that using past data to cluster users was better than using joint user-item feature vectors

What Works In Practice?

- Per user θ_u : Data from one user cannot be used for another user.
 - Does not scale—too little data per user
- Clustered users: Used by Deezer to recommend playlists (Bendada et al, 2020)
 - Found that using past data to cluster users was better than using joint user-item feature vectors
- Joint features: Used by the Yahoo! News article recommendation paper (Li et al, 2009)
 - Today, we use foundation models for extracting user profiles as well

Conclusion

- Linear bandits is an amazing, super-flexible model
- Continues to be used today across many industries, especially any instance of a recommendation system

Conclusion

- Linear bandits is an amazing, super-flexible model
- Continues to be used today across many industries, especially any instance of a recommendation system
- For it to work in practice, we need to do careful feature engineering

Conclusion

- Linear bandits is an amazing, super-flexible model
- Continues to be used today across many industries, especially any instance of a recommendation system
- For it to work in practice, we need to do careful feature engineering
- Modern day data science makes smart use of neural networks!

Conclusion

- Linear bandits is an amazing, super-flexible model
- Continues to be used today across many industries, especially any instance of a recommendation system
- For it to work in practice, we need to do careful feature engineering
- Modern day data science makes smart use of neural networks!
- We will explore a few more practical aspects about bandits in upcoming lectures