Keynote Talk at 2021 KDD Workshop on Multi-Armed Bandits and Reinforcement Learning

A Map of Bandits for E-commerce

Yi Liu, Lihong Li

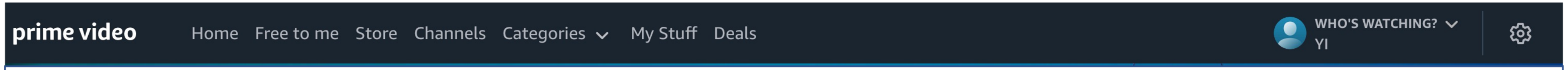
Aug 15th, 2021

KDD2021 MarbleKDD 2021 amazon

Summary

- The rich Bandit literature offers a diverse toolbox of algorithms
- Hard for practitioners to find the right solution for problem at hand
  - Typical textbooks focus on designing and analyzing algorithms
  - Typical surveys present a list of individual applications

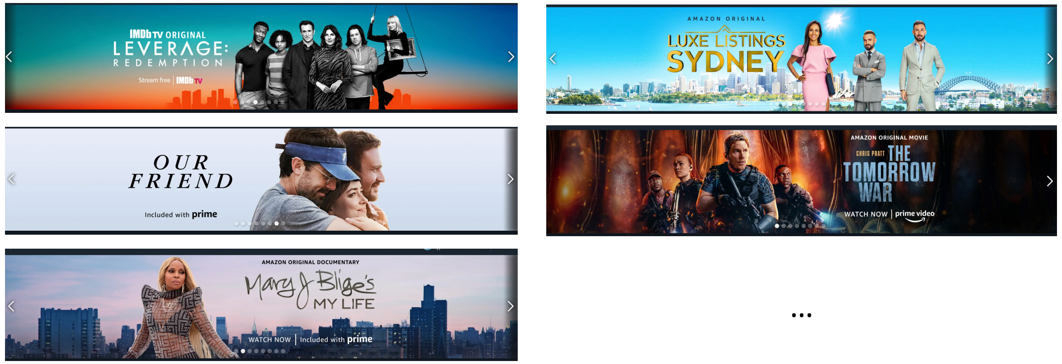
- This talk: a "map" towards closing the gap in mapping applications to appropriate Bandit algorithms.
  - Focus on a small number of key decision points related to reward/actions
  - Focus on E-commerce examples, but applicable to other applications



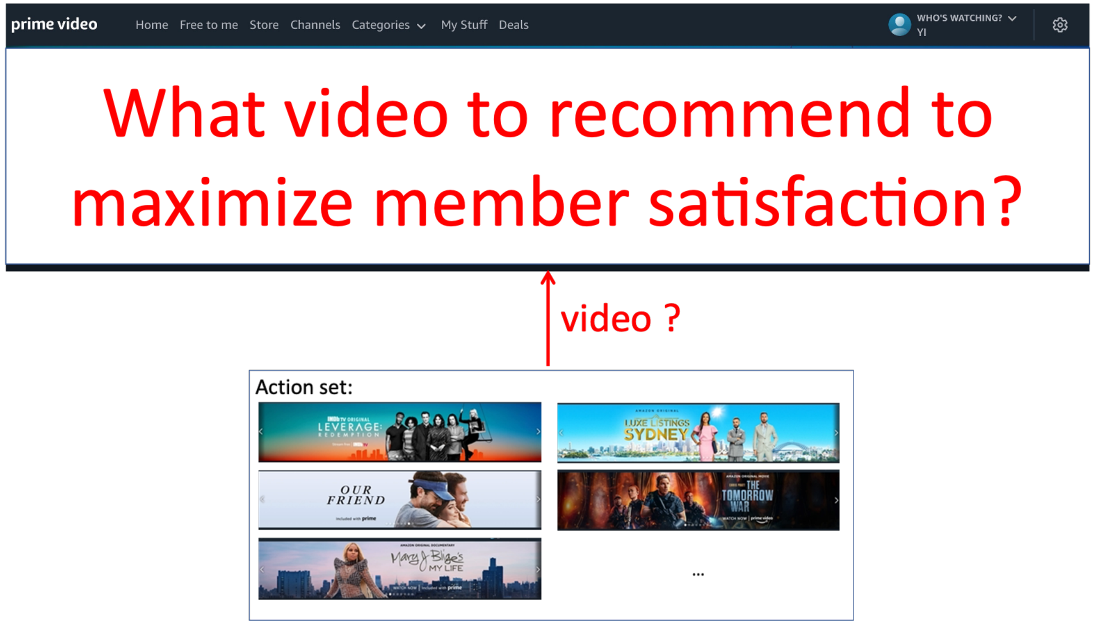
What video to recommend to maximize member satisfaction?

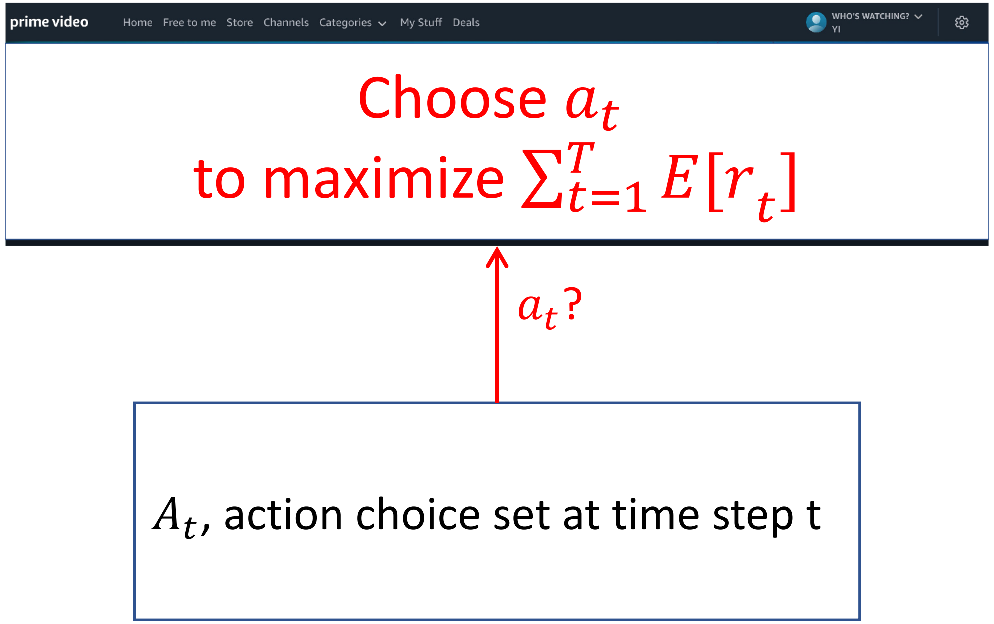
video ?

Action set:

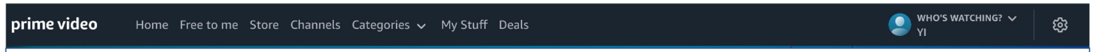


...


prime video
Home
Free to me
Store
Channels
Categories
My Stuff
Deals
WHO'S WATCHING?
YI
What video to recommend to maximize member satisfaction?
video ?
Action set:
LEVERAGE REDEMPTION
LUXE LISTINGS SYDNEY
OUR FRIEND
THE TOMORROW WAR
Mary J. Blige's MY LIFE
...


prime video
Home
Free to me
Store
Channels
Categories
My Stuff
Deals
WHO'S WATCHING?
YI
Choose a_t
to maximize $\sum_{t=1}^{T} E[r_t]$
a_t?
A_t, action choice set at time step t

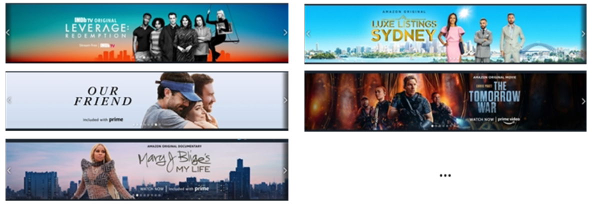
If satisfaction is defined when member streams.



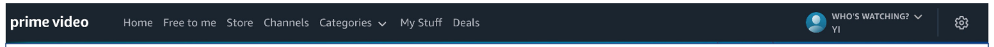
What video to recommend to maximize member satisfaction?

video ?

Action set:



...



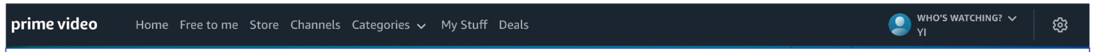
Choose a_t

to maximize $\sum_{t=1}^{T} E[r_t]$

a_t?

A_t, action choice set at time step t

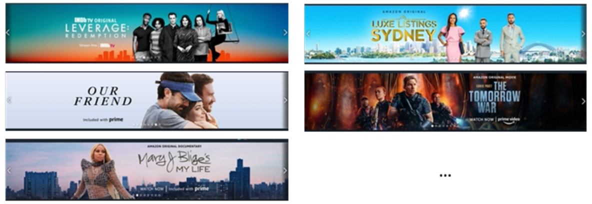
If satisfaction is defined when member streams.



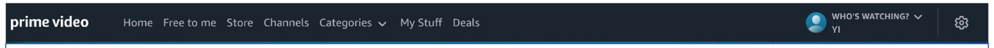
What video to recommend to maximize member satisfaction?

video ?

Action set:



...



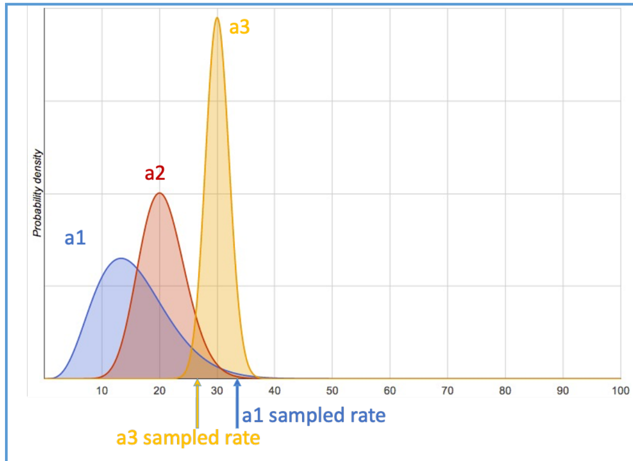
Choose a_t to maximize $\sum_{t=1}^{T} E[r_t]$

a_t?

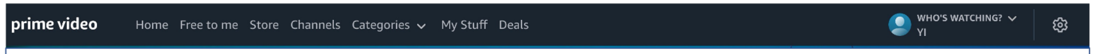
A_t, action choice set at time step t

Bandit algorithm: Bayesian Linear Probit Regression (BLIP) for reward modeling + Thomson Sampling for exploration

- r_t = 1 if streaming, and 0 otherwise
- $E[r_t] = \Phi(\boldsymbol{w} \cdot \boldsymbol{\phi}(a_t))$
- Assume w's follow Gaussian distribution and enforce the assumption when updating.



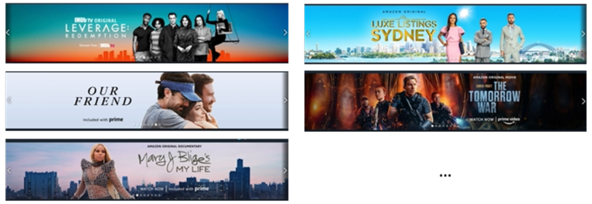
~~If satisfaction is defined when member streams.~~ If we measure satisfaction by how long they spend on watching videos.



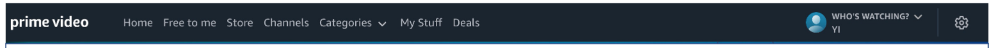
What video to recommend to maximize member satisfaction?

video ?

Action set:



...



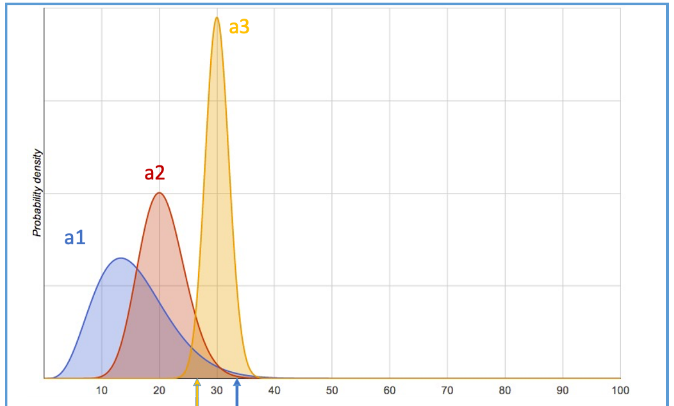
Choose a_t
to maximize $\sum_{t=1}^{T} E[r_t]$

a_t?

A_t, action choice set at time step t

Bandit algorithm: Bayesian Linear ~~Probit~~ Regression ~~(BLIP)~~ for reward modeling + Thomson Sampling for exploration

- ✗ ~~$r_t = 1$ if streaming, and 0 otherwise~~
- ✗ ~~$E[r_t] = \Phi(\boldsymbol{w} \cdot \boldsymbol{\phi}(a_t))$~~
- Assume w's follow Gaussian distribution and enforce the assumption when updating.



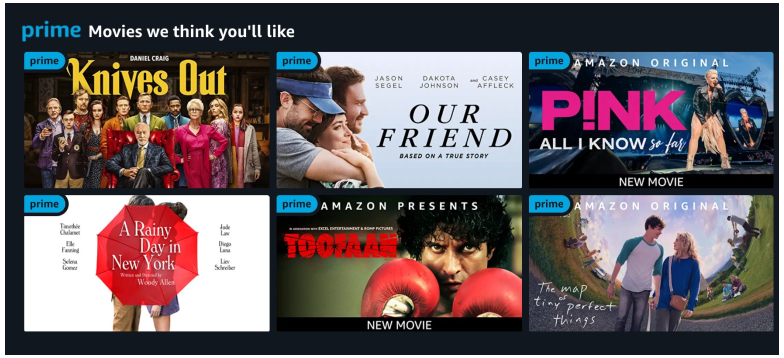
~~If satisfaction is defined when member streams.~~ If we measure satisfaction by how long they spend on watching videos.

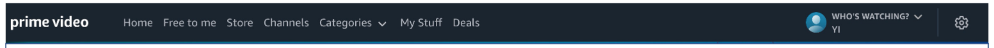
How to recommend

a subset of videos

to maximize member satisfaction?

6 videos





Choose a_t

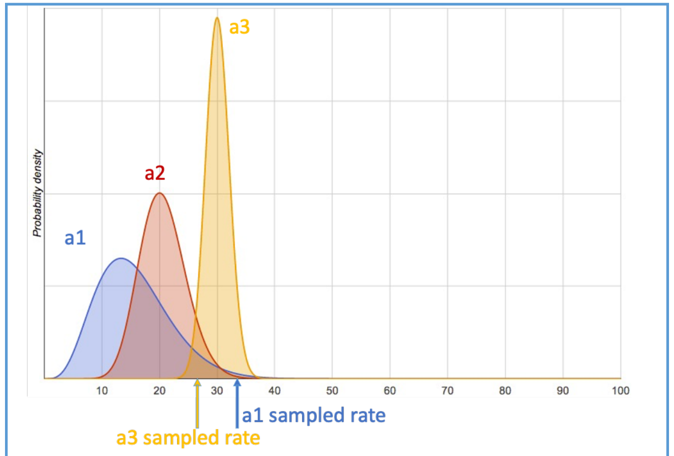
to maximize $\sum_{t=1}^{T} E[r_t]$

~~a_t?~~ ✗

A_t, action choice set at time step t

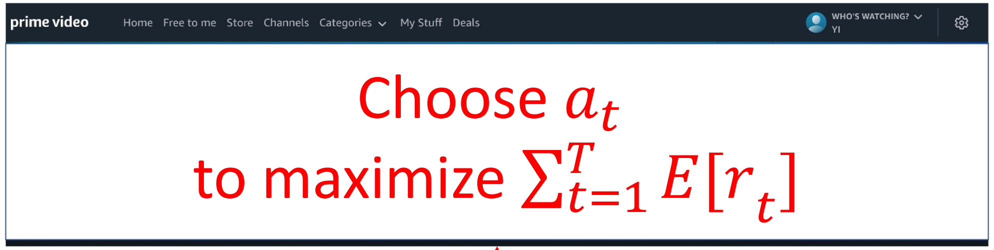
Bandit algorithm: Bayesian Linear ~~Probit~~ Regression ~~(BLIP)~~ for reward modeling + Thomson Sampling for exploration

- ✗ ~~r_t = 1 if streaming, and 0 otherwise~~
- ✗ ~~$E[r_t] = \Phi(\boldsymbol{w} \cdot \boldsymbol{\phi}(a_t))$~~
- Assume w's follow Gaussian distribution and enforce the assumption when updating.



Action is a combinatorial object.

What marketing content to recommend to maximize offer signup?



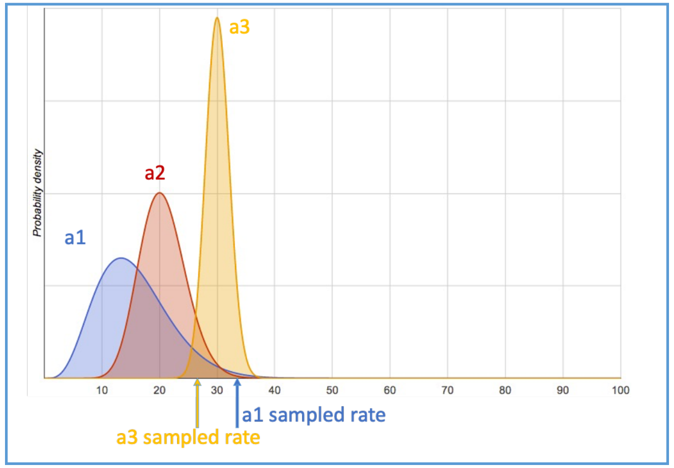
Choose a_t to maximize $\sum_{t=1}^{T} E[r_t]$

a_t?

A_t, action choice set at time step t

Bandit algorithm: Bayesian Linear Probit Regression (BLIP) for reward modeling + Thomson Sampling for exploration

- r_t = 1 if streaming, and 0 otherwise
- $E[r_t] = \Phi(\boldsymbol{w} \cdot \underline{\boldsymbol{\phi}(a_t)})$?
- Assume w's follow Gaussian distribution and enforce the assumption when updating.



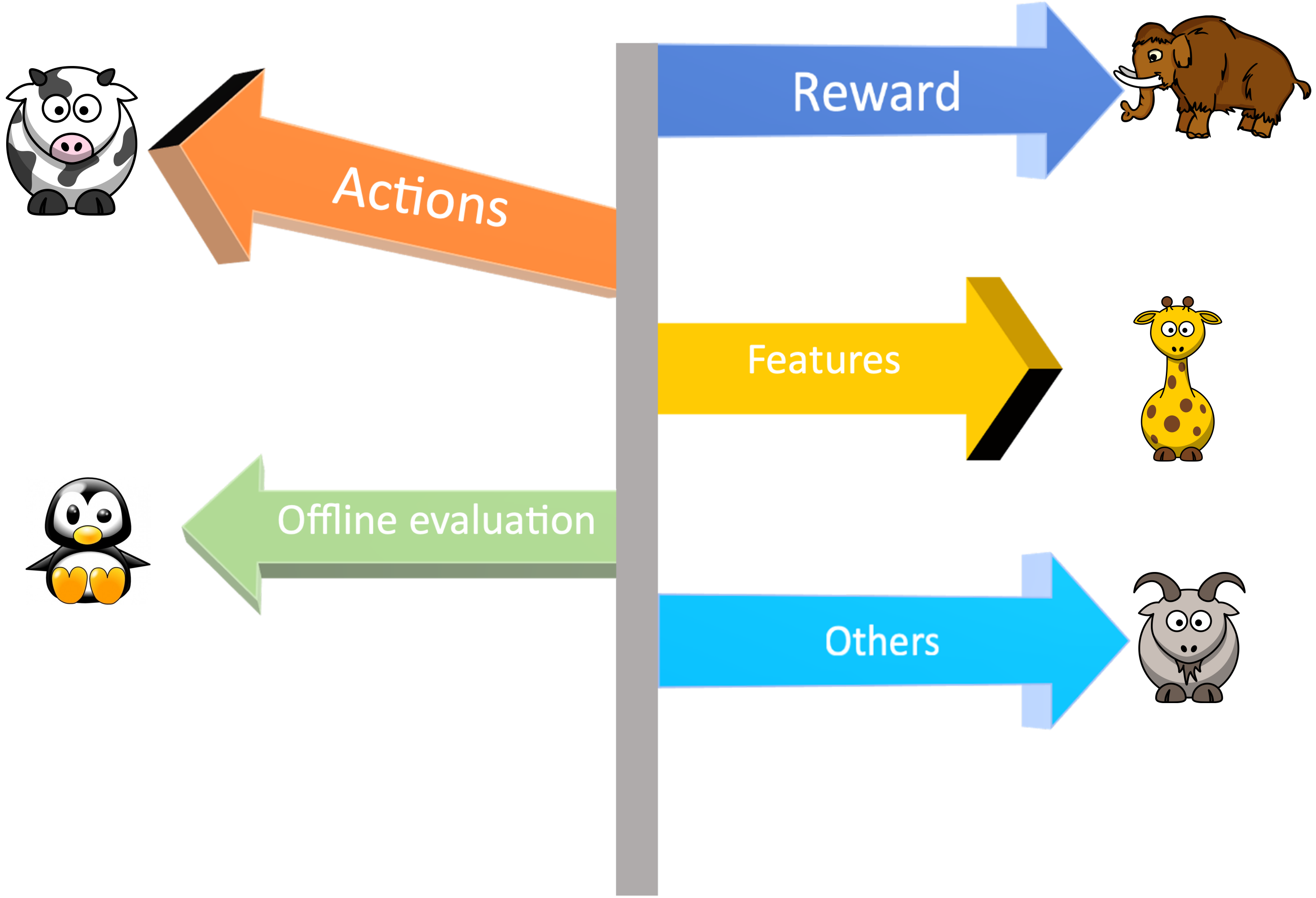
Which Bandit algorithms are for your problem?

Business problems with different characteristics $\xrightarrow{?}$ A zoo of Bandit algorithms

What marketing content to recommend to maximize offer signup?

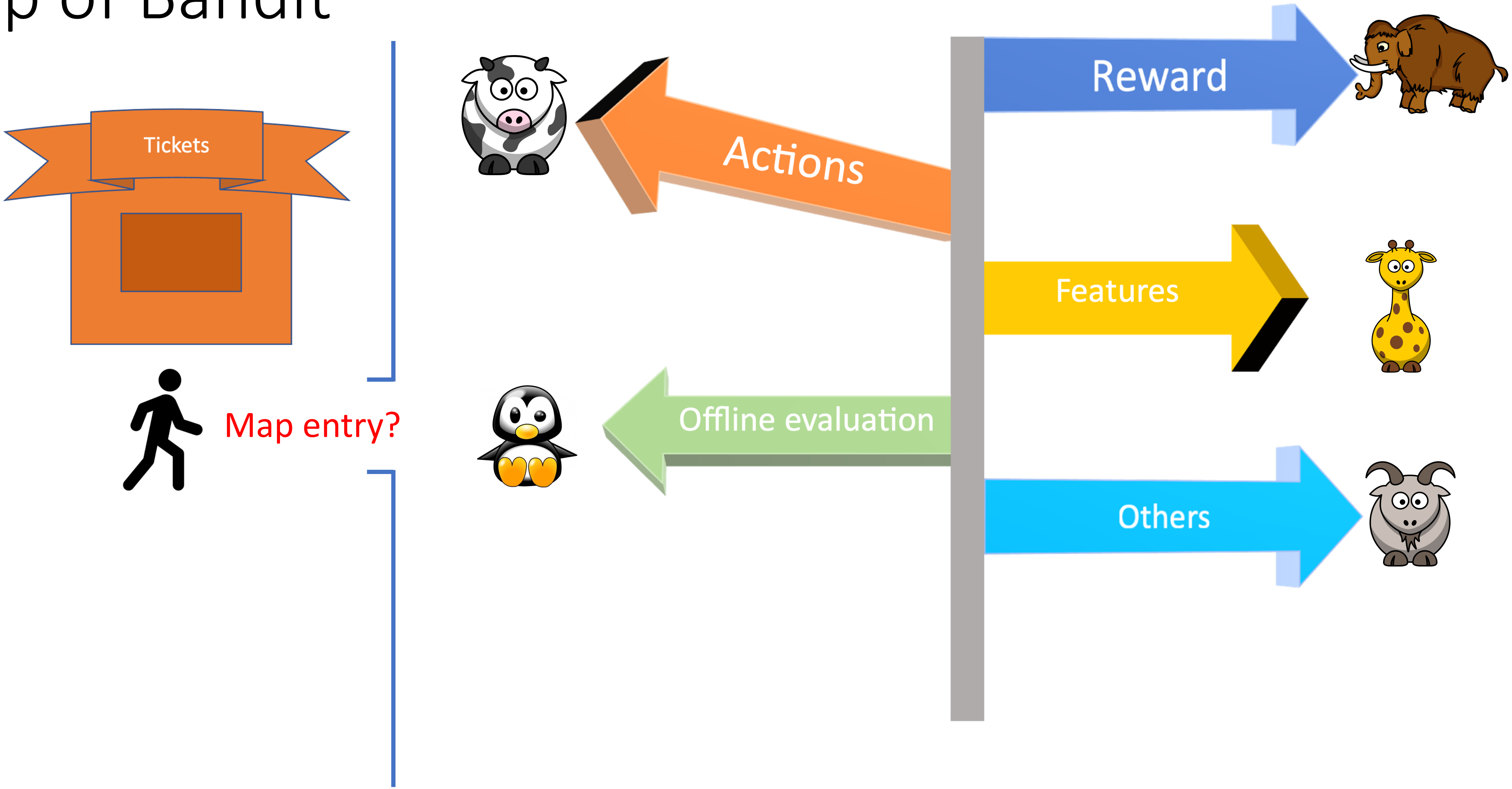
Beat-the-Mean
BLIP
LinTSPBMRank
OTFLinUCB
LinGreedy
NeuralUCB
C^2UCB
CombLinTS
Multivariate bandits

A Map of Bandit



http://clipart-library.com/cartoon-pictures-of-animals.html

A Map of Bandit



http://clipart-library.com/cartoon-pictures-of-animals.html

Map Entry?

Upsell the $9.99 Premium membership plan

Action choice set:

Download music.

Listen anywhere.

No ad interruptions.

Enjoy nonstop music.

Play any song.

Even on mobile.

Unlimited skips.

Just hit next.

Map Entry?

Upsell the $9.99 Premium membership plan

✓ X ✓ X

Action choice set:

Download music.

Listen anywhere.

No ad interruptions.

Enjoy nonstop music.

Play any song.

Even on mobile.

Unlimited skips.

Just hit next.

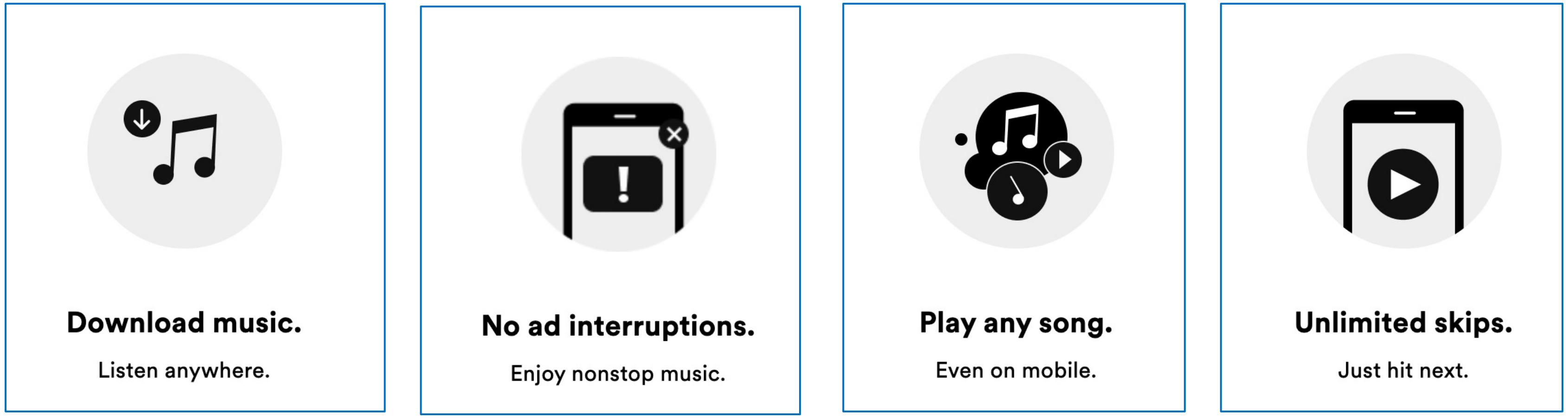
Map Entry? No!

- Rewards for *all* actions are observed.
- It is a full-information setting.
- Supervised learning should be considered.

Upsell the $9.99 Premium membership plan

✓ X ✓ X

Action choice set:



Map Entry?

Upsell the $9.99 Premium membership plan

?

Action choice set:

Download music.

Listen anywhere.

No ad interruptions.

Enjoy nonstop music.

Play any song.

Even on mobile.

Unlimited skips.

Just hit next.

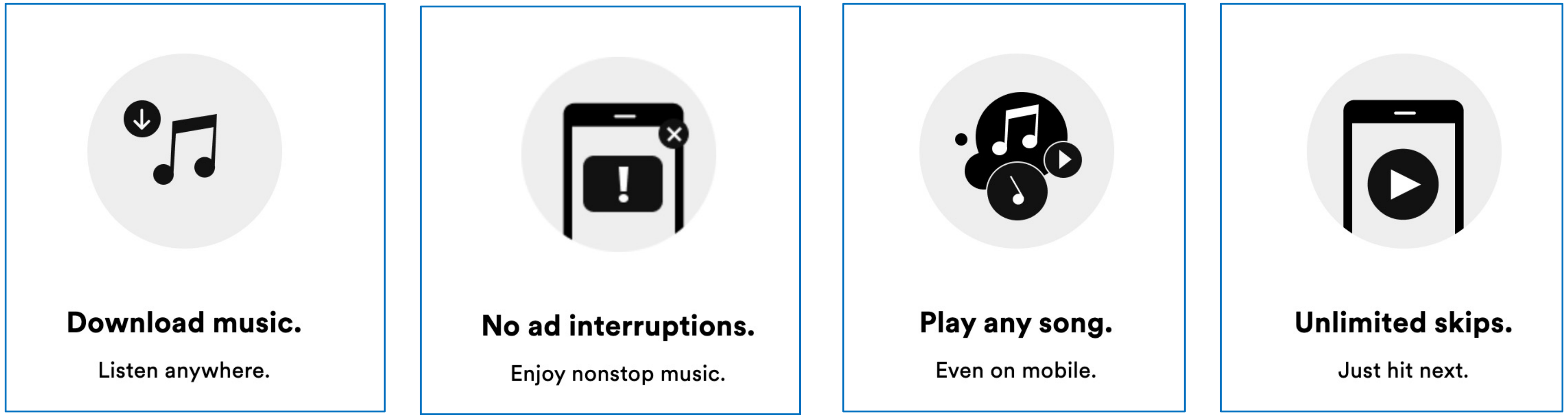
Map Entry? Yes!

Upsell the $9.99 Premium membership plan

Only the reward of
the selected action is returned.

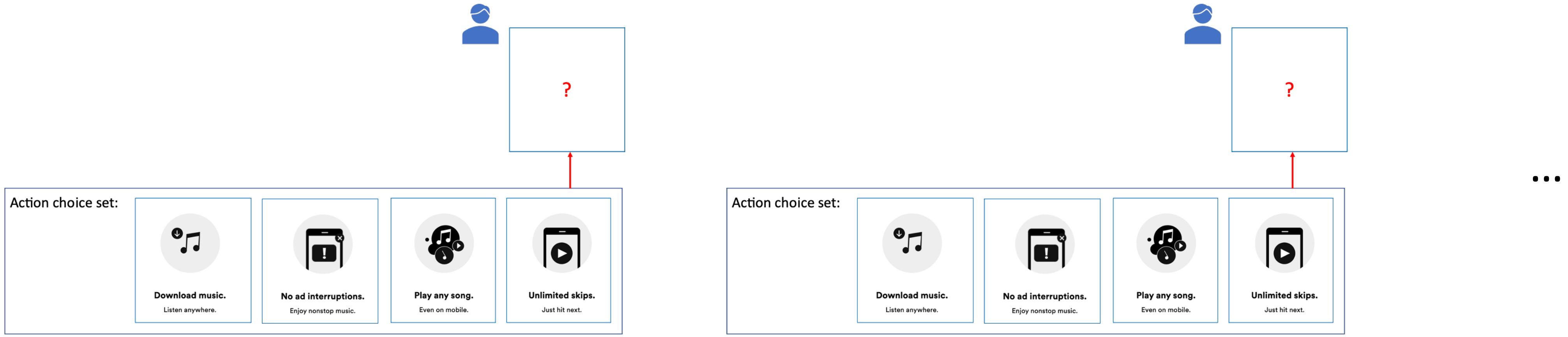
?

Action choice set:



Map Entry?

Upsell the $9.99 Premium membership plan

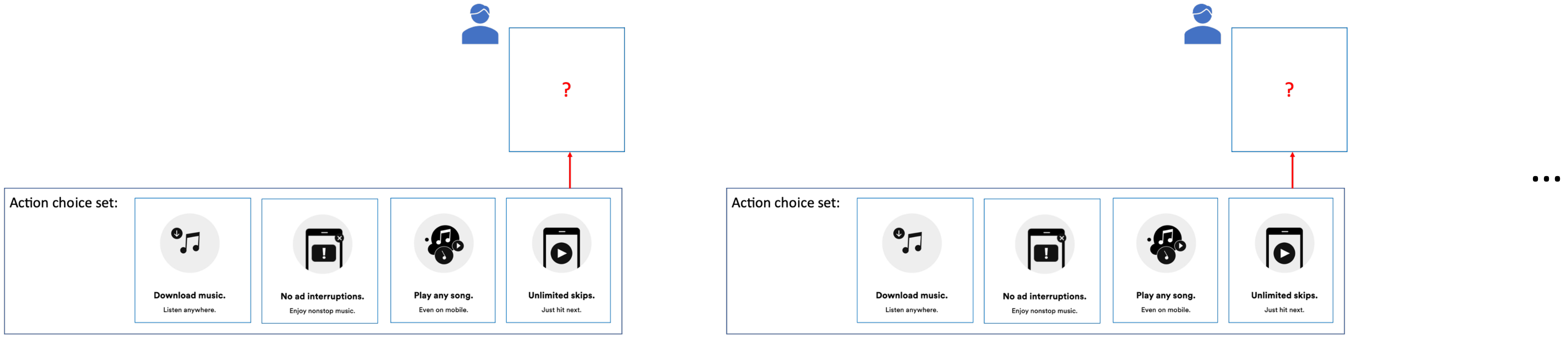


t = 0 t = 1

Map Entry? Yes!

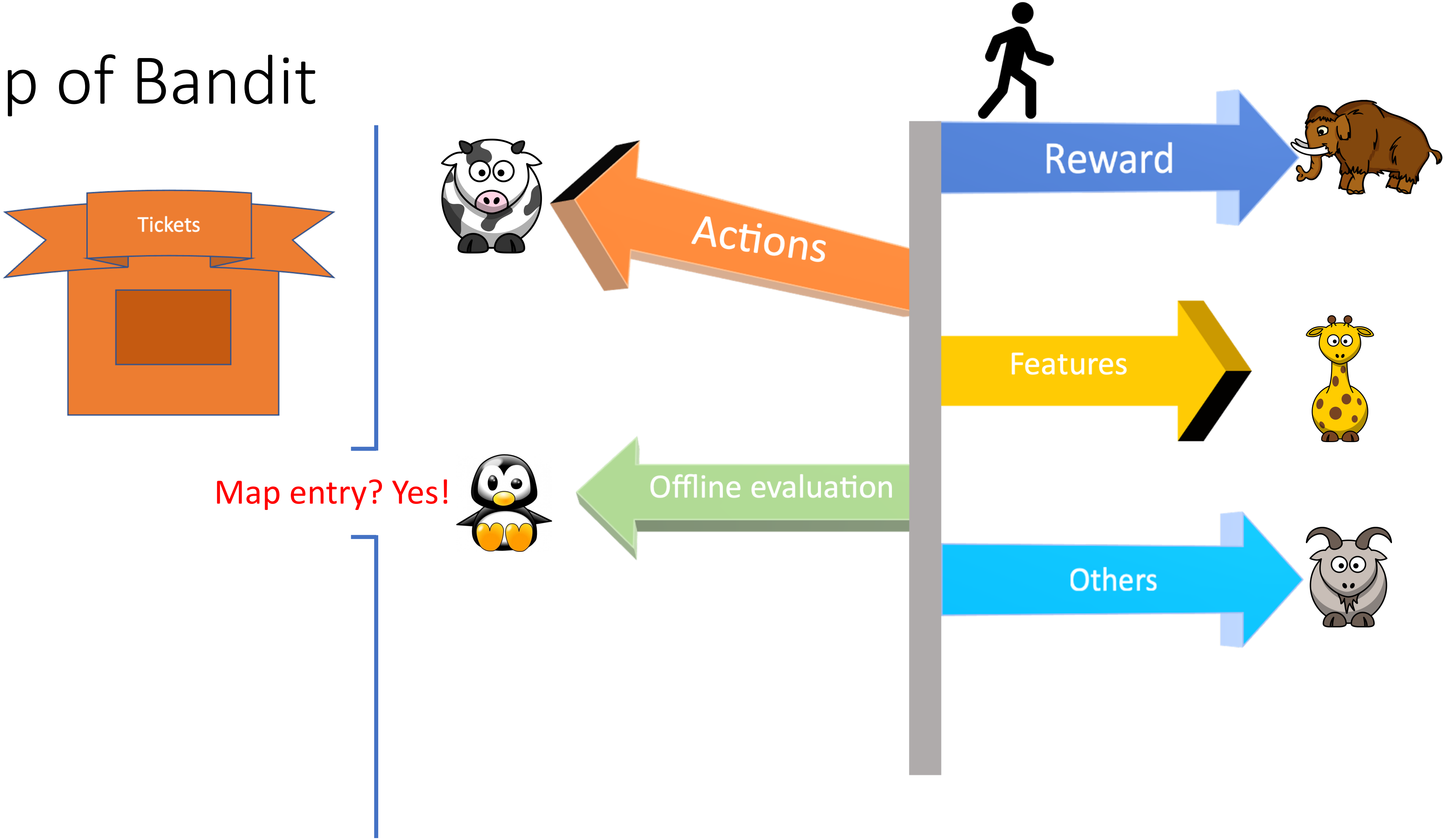
- Consider the dependency between actions.
- Reason with long-term rewards.
- Bandit is a good baseline for more general reinforcement learning setting.

Upsell the $9.99 Premium membership plan

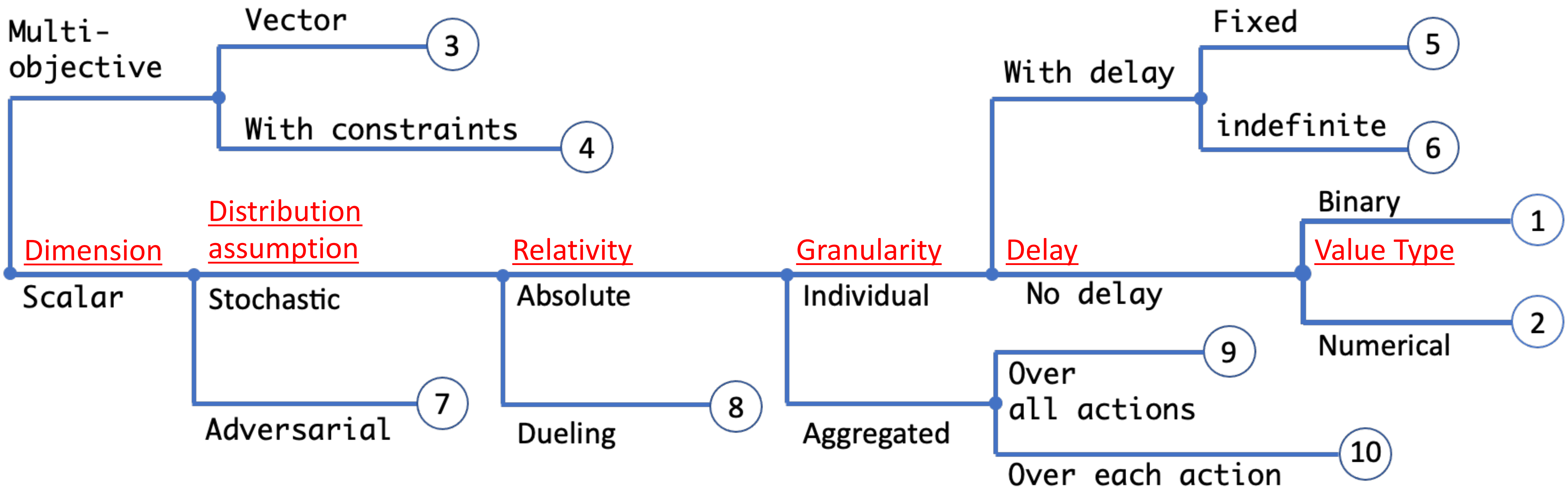


t = 0 t = 1

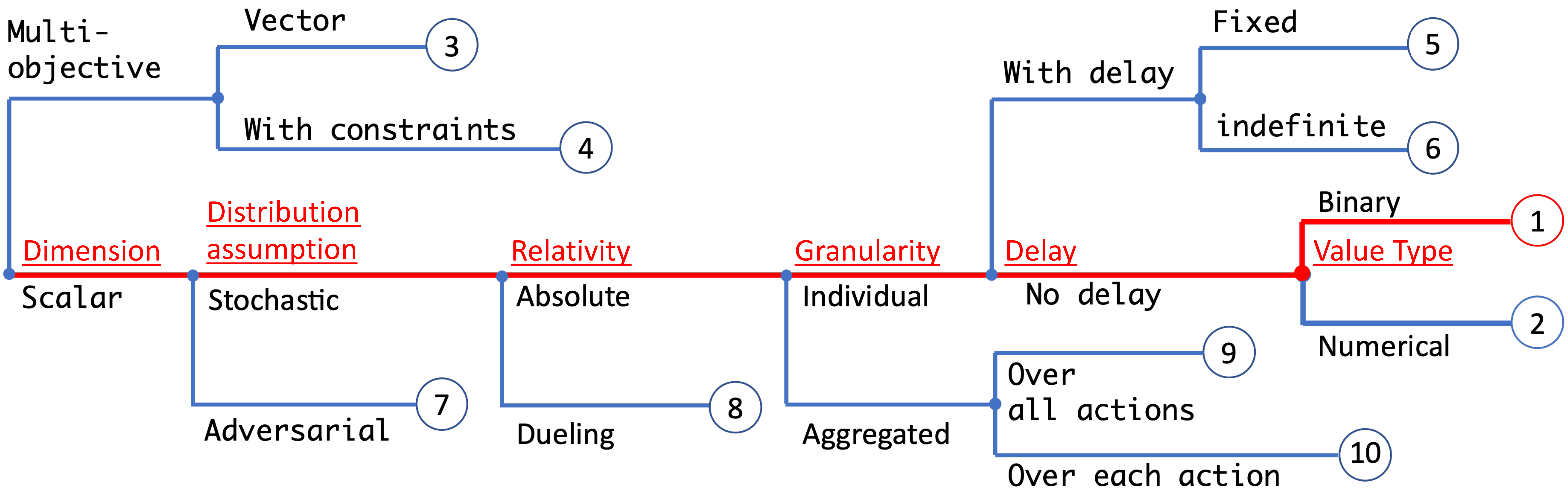
A Map of Bandit



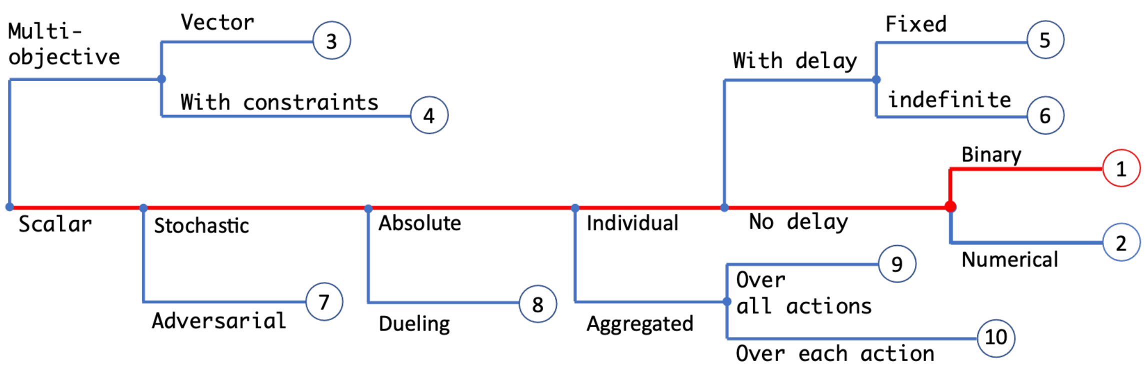
Bandit Problems by Reward Properties

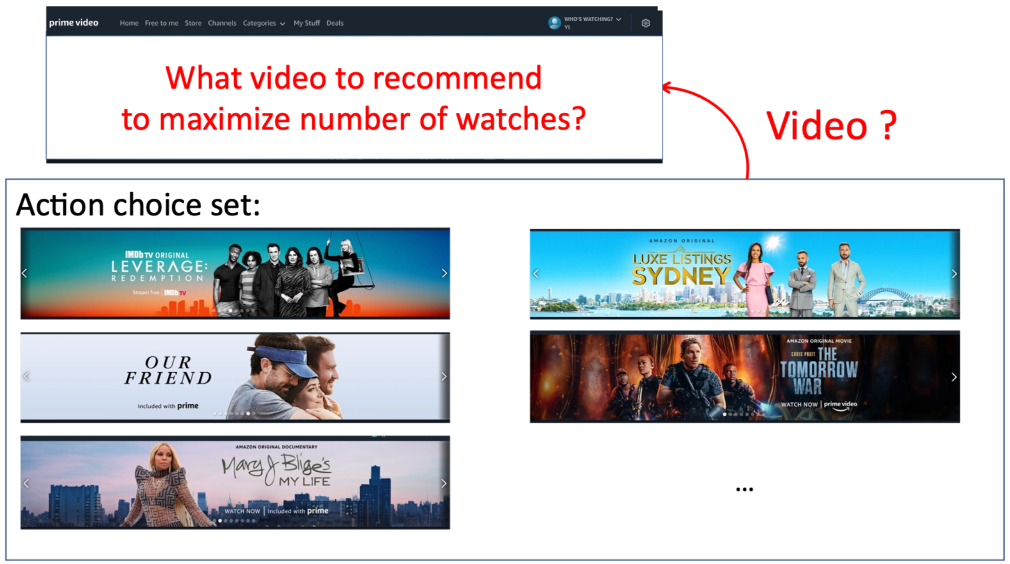


Bandit Problems by Reward Properties

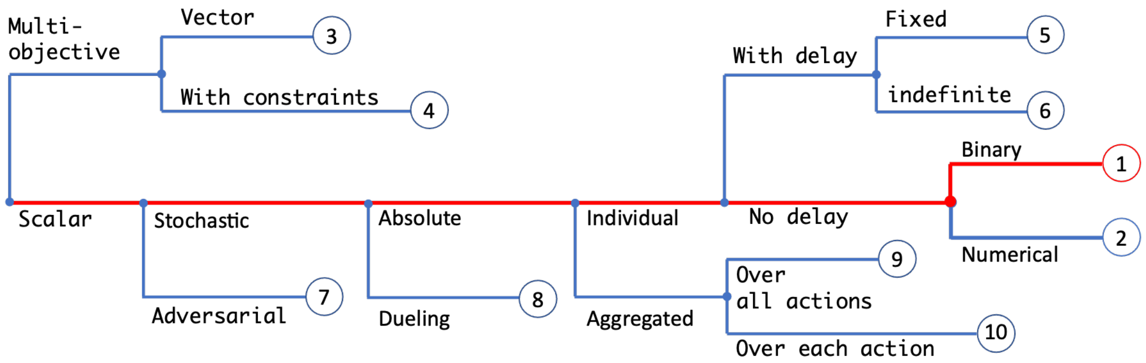


Node 1: Binary reward





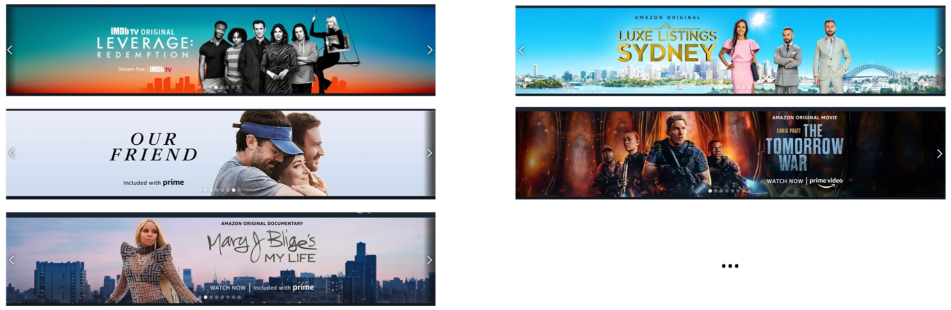
Node 1: Binary reward



What video to recommend
to maximize number of watches?

Video ?

Action choice set:



...

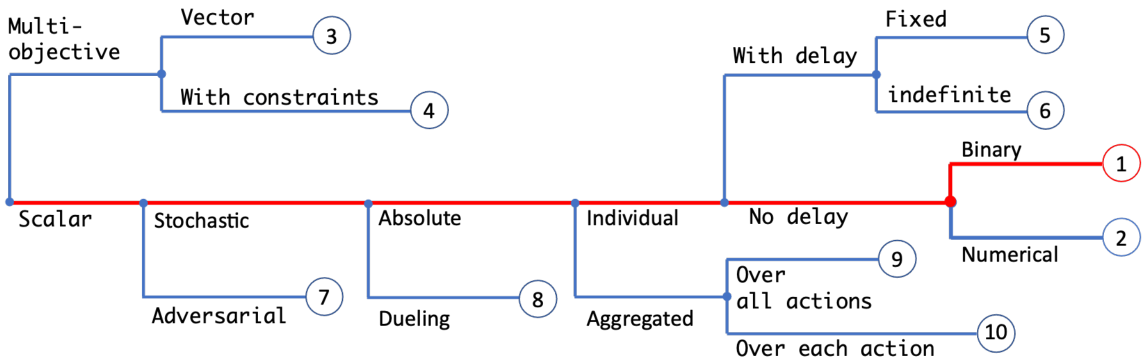
Algorithm 3 Regularized logistic regression with batch updates

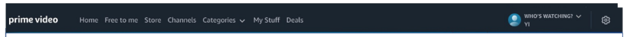
Require: Regularization parameter $\lambda > 0$.
$m_i = 0,\ q_i = \lambda$. {Each weight w_i has an independent prior $\mathcal{N}(m_i, q_i^{-1})$}
for $t = 1, \ldots, T$ **do**
Get a new batch of training data $(\mathbf{x}_j, y_j),\ \ j = 1, \ldots, n$.
Find $\mathbf{w}$ as the minimizer of: $\frac{1}{2}\sum_{i=1}^{d} q_i(w_i - m_i)^2 + \sum_{j=1}^{n} \log(1 + \exp(-y_j \mathbf{w}^\top \mathbf{x}_j))$.
$m_i = w_i$
$q_i = q_i + \sum_{j=1}^{n} x_{ij}^2 p_j(1 - p_j),\ \ p_j = (1 + \exp(-\mathbf{w}^\top \mathbf{x}_j))^{-1}$ {Laplace approximation}
end for

Reward = 1 if customer streams; 0 otherwise.

Ref:
O. Chapelle and L. Li, "An empirical evaluation of Thompson sampling," in NIPS, 2011.

Node 1: Binary reward

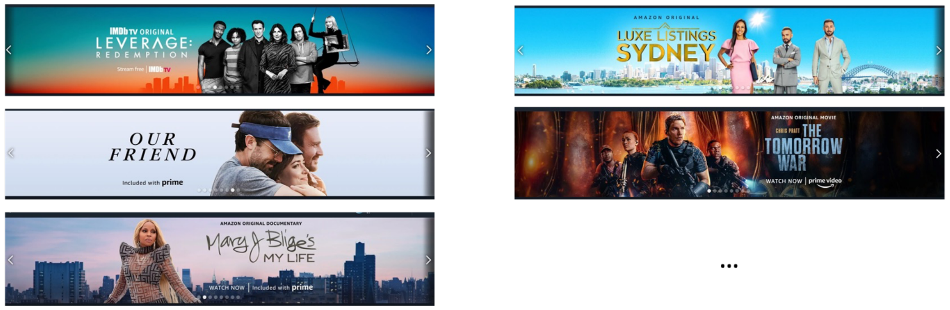




What video to recommend
to maximize number of watches?

Video ?

Action choice set:



...

Algorithm 3 Regularized logistic regression with batch updates

Require: Regularization parameter $\lambda > 0$.

$m_i = 0,\ q_i = \lambda$. {Each weight w_i has an independent prior $\mathcal{N}(m_i, q_i^{-1})$}

for $t = 1, \ldots, T$ **do**

Get a new batch of training data $(\mathbf{x}_j, y_j),\ \ j = 1, \ldots, n$.

Find $\mathbf{w}$ as the minimizer of: $\displaystyle \frac{1}{2}\sum_{i=1}^{d} q_i(w_i - m_i)^2 + \sum_{j=1}^{n} \log(1 + \exp(-y_j \mathbf{w}^\top \mathbf{x}_j))$.

$m_i = w_i$

$q_i = q_i + \displaystyle\sum_{j=1}^{n} x_{ij}^2 p_j(1 - p_j),\ \ p_j = (1 + \exp(-\mathbf{w}^\top \mathbf{x}_j))^{-1}$ {Laplace approximation}

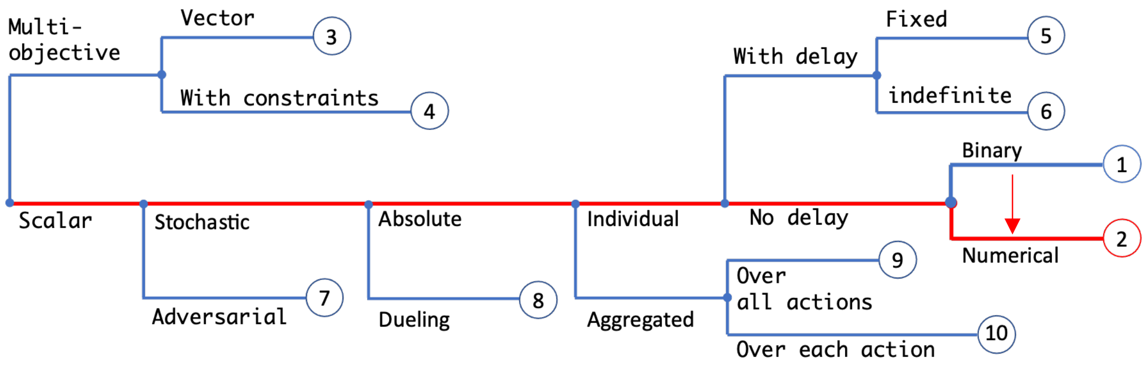
end for

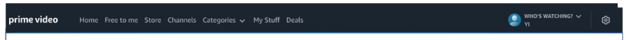
Reward = 1 if customer streams; 0 otherwise.

Ref:

O. Chapelle and L. Li, "An empirical evaluation of Thompson sampling," in NIPS, 2011.

Node 2: Numerical reward

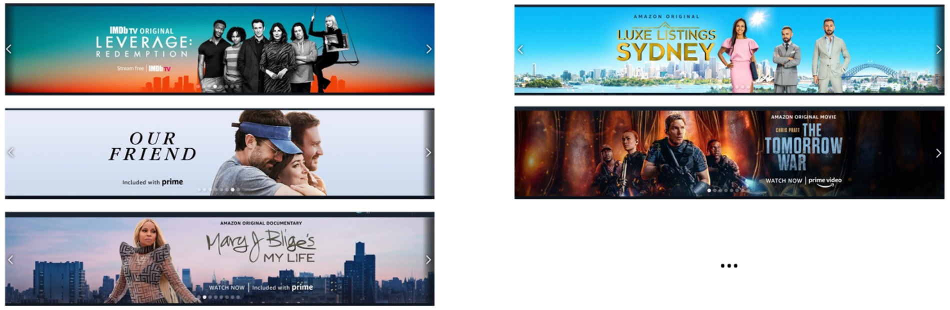




What video to recommend
to maximize ~~number of watches?~~
watching time

Video ?

Action choice set:



...

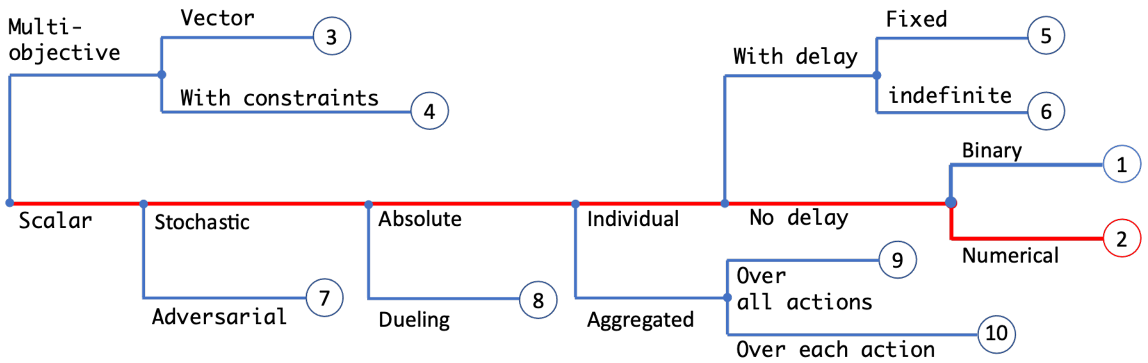
Algorithm 3 Regularized logistic regression with batch updates

Require: Regularization parameter $\lambda > 0$.
  $m_i = 0$, $q_i = \lambda$. {Each weight w_i has an independent prior $\mathcal{N}(m_i, q_i^{-1})$}
  for $t = 1, \ldots, T$ **do**
    Get a new batch of training data $(\mathbf{x}_j, y_j)$, $j = 1, \ldots, n$.
    Find $\mathbf{w}$ as the minimizer of: $\frac{1}{2}\sum_{i=1}^{d} q_i (w_i - m_i)^2 + \sum_{j=1}^{n} \log(1 + \exp(-y_j \mathbf{w}^\top \mathbf{x}_j))$.
    $m_i = w_i$
    $q_i = q_i + \sum_{j=1}^{n} x_{ij}^2 p_j (1 - p_j)$, $p_j = (1 + \exp(-\mathbf{w}^\top \mathbf{x}_j))^{-1}$ {Laplace approximation}
  end for

X

Reward is numerical.

Node 2: Numerical reward



Algorithm 1 Thompson Sampling for Contextual bandits

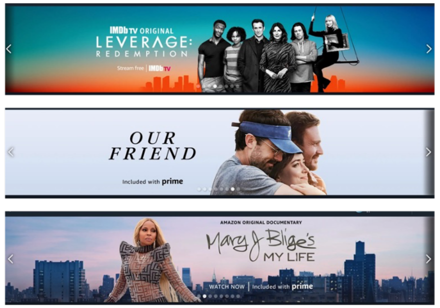
for all $t = 1, 2, \ldots,$ **do**
  Sample $\tilde{\mu}(t)$ from distribution $\mathcal{N}(\hat{\mu}(t), v^2 B(t)^{-1})$.
  Play arm $a(t) := \arg\max_i \underline{b_i(t)^T \tilde{\mu}(t)}$, and observe reward $r_{a(t)}(t)$.
end for

prime video Home Free to me Store Channels Categories My Stuff Deals

What video to recommend
to maximize ~~number of watches?~~
watching time

Video ?

Action choice set:

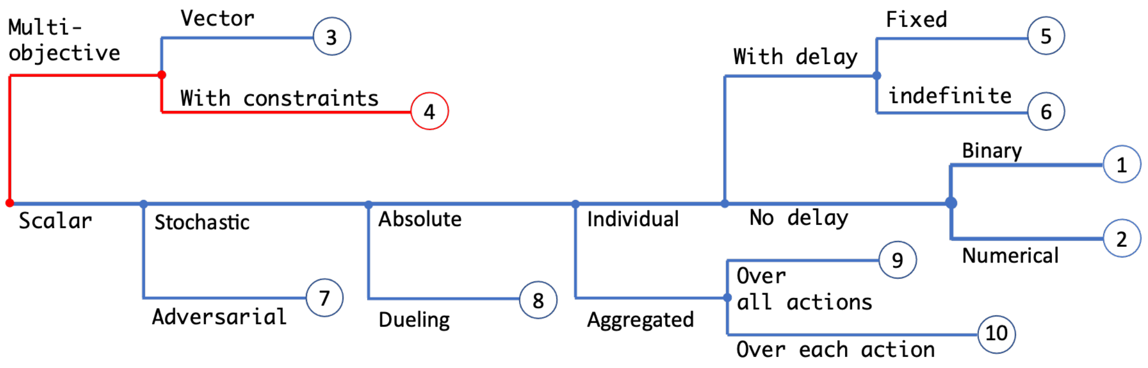


...

Reward is numerical.

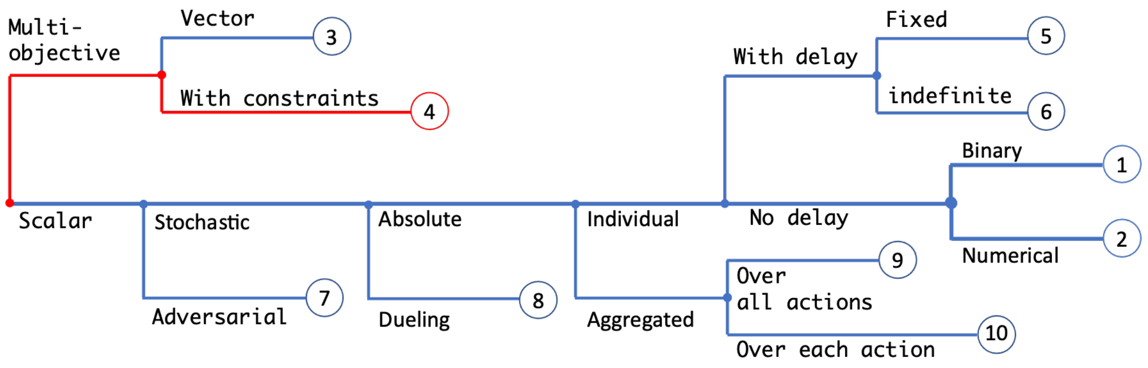
Ref:
S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs," in ICML, 2013.

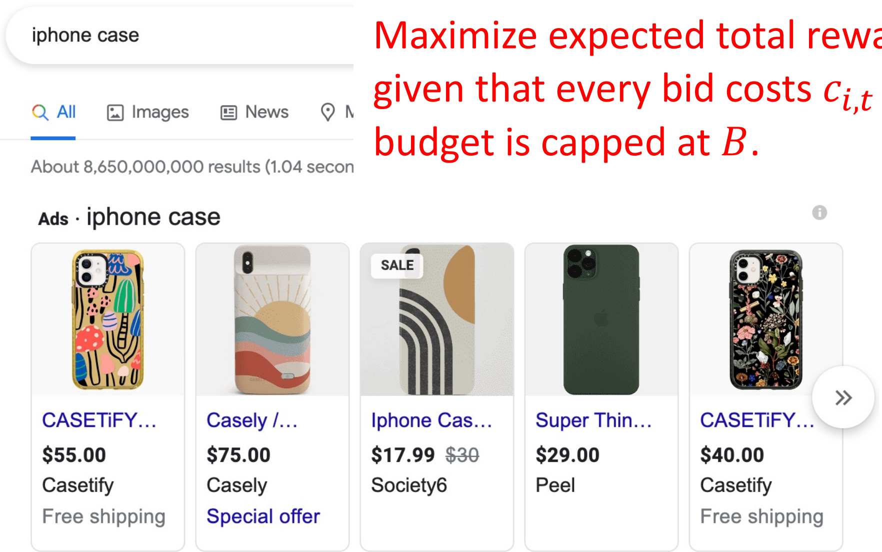
Node 4: Optimization with constraints



What if it is not cost free to take an action?
What if every reward is received at a cost?

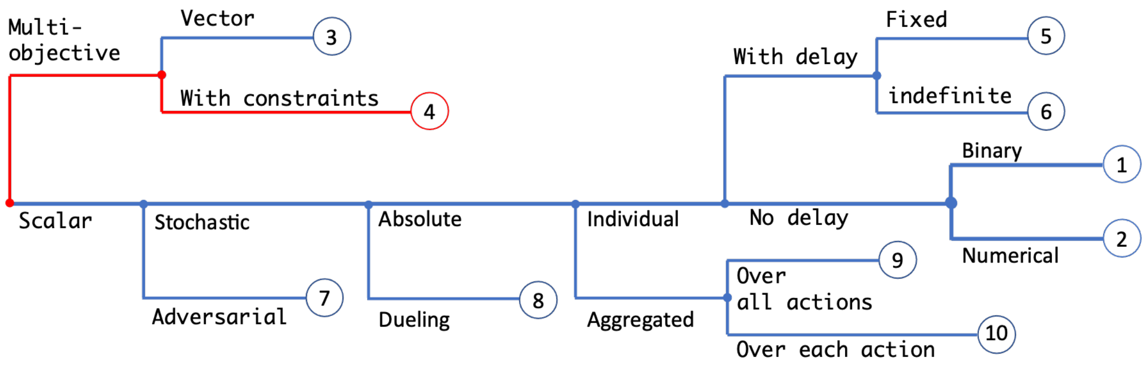
Node 4: Optimization with constraints

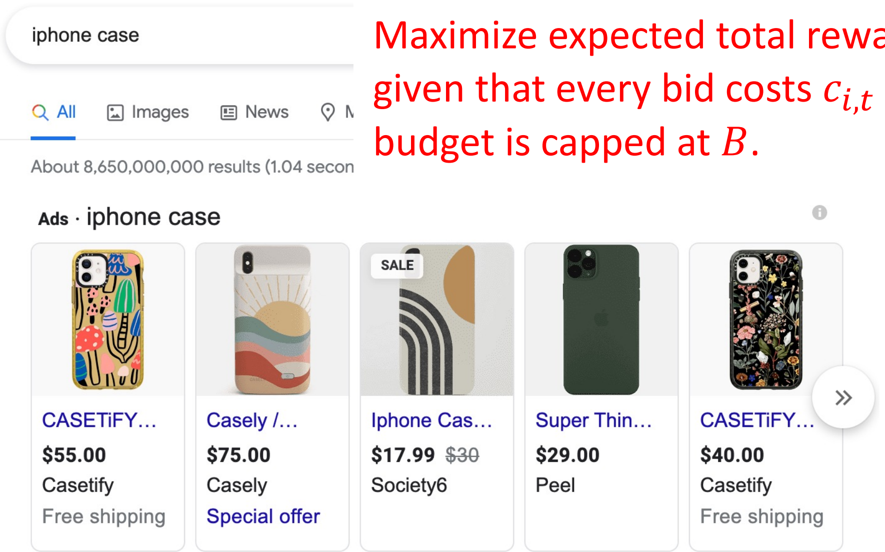




Maximize expected total reward $E\left[\sum R_{i,t}\right]$ given that every bid costs $c_{i,t}$ and the budget is capped at B.

Node 4: Optimization with constraints





Maximize expected total reward $E\left[\sum R_{i,t}\right]$ given that every bid costs $c_{i,t}$ and the budget is capped at B.

Algorithm 1 UCB-BV1/UCB-BV2

Initialization: Pull each arm i once in the first K steps, set $t = K$.

while $\sum_{s=1}^{t} c_{a_s,s} \leq B$ **do**

Set $t = t + 1$.

Calculate the index $D_{i,t}$ of each arm i as follows.

UCB-BV1

Exploitation Exploration

$$D_{i,t} = \frac{\bar{r}_{i,t}}{\bar{c}_{i,t}} + \frac{(1+\frac{1}{\lambda})\sqrt{\frac{\ln(t-1)}{n_{i,t}}}}{\lambda - \sqrt{\frac{\ln(t-1)}{n_{i,t}}}}$$

UCB-BV2

$$D_{i,t} = \frac{\bar{r}_{i,t}}{\bar{c}_{i,t}} + \frac{1}{\lambda_t}\left(1 + \frac{1}{\lambda_t - \sqrt{\frac{\ln(t-1)}{n_{i,t}}}}\right)\sqrt{\frac{\ln(t-1)}{n_{i,t}}}$$

Pull the arm a_t with the largest index: $a_t = \arg\max_i D_{i,t}$.

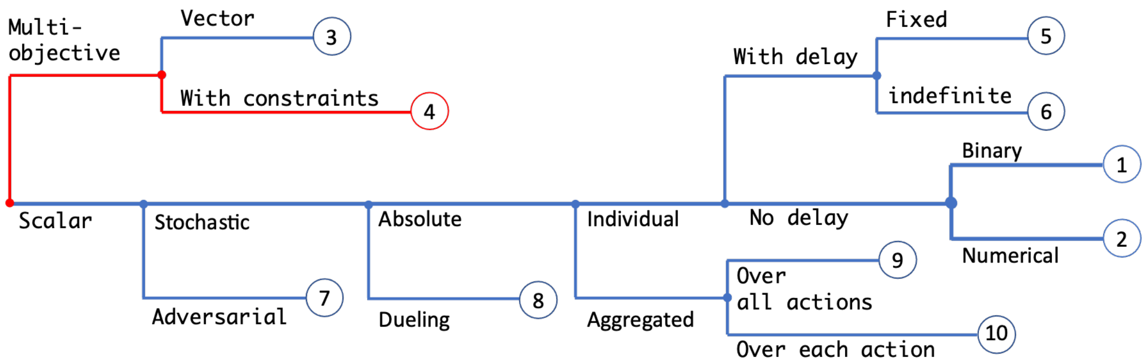
end while

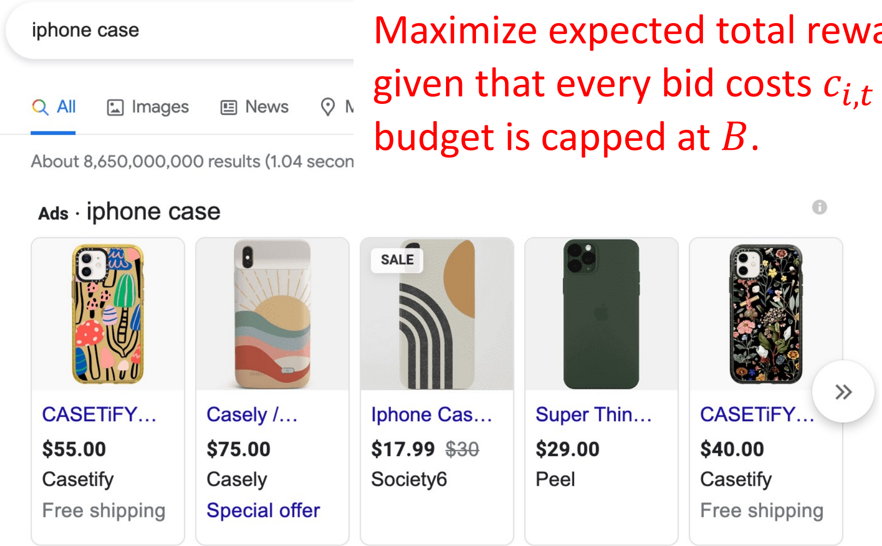
Return: $t_B = t - 1$.

Ref:

W. Ding, T. Qin, X. Zhang, and T. Liu, "Multi-armed bandit with budget constraint and variable costs," in AAAI, 2013.

Node 4: Optimization with constraints





Maximize expected total reward $E[\sum R_{i,t}]$ given that every bid costs $c_{i,t}$ and the budget is capped at B.

Algorithm 1 UCB-BV1/UCB-BV2

Initialization: Pull each arm i once in the first K steps, set $t = K$.

while $\sum_{s=1}^{t} c_{a_s,s} \le B$ **do**

Set $t = t + 1$.

Calculate the index $D_{i,t}$ of each arm i as follows.

UCB-BV1

Exploitation

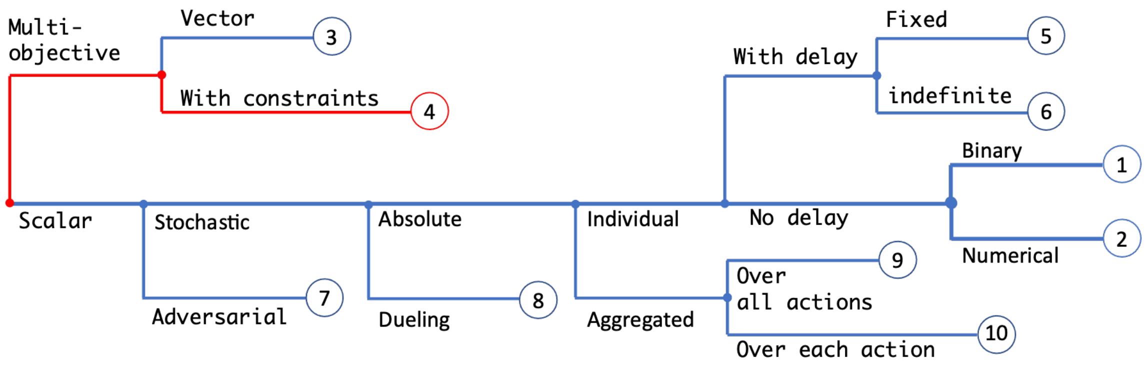
Exploration

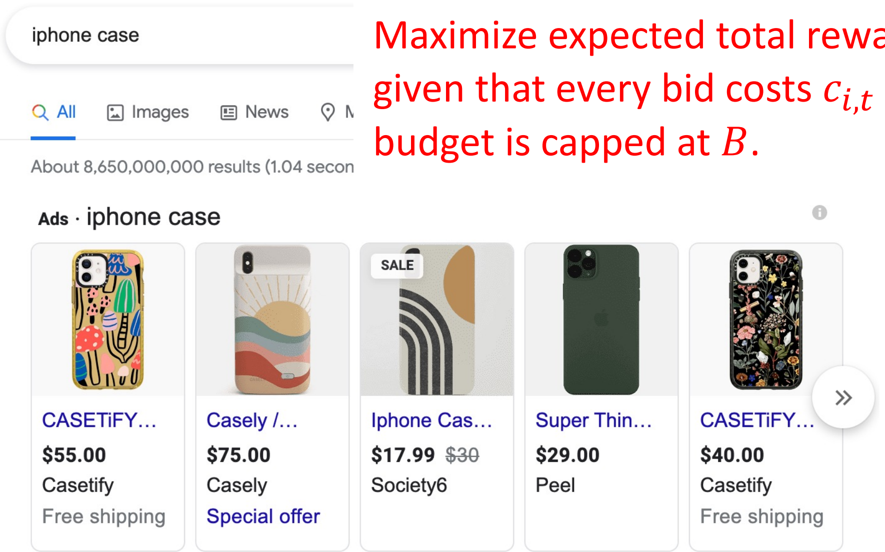
$$D_{i,t} = \frac{\bar{r}_{i,t}}{\bar{c}_{i,t}} + \frac{(1+\frac{1}{\lambda})\sqrt{\frac{\ln(t-1)}{n_{i,t}}}}{\lambda - \sqrt{\frac{\ln(t-1)}{n_{i,t}}}}$$

- The add of the term guarantees a regret bound of $O(\ln(B))$.
- The proof utilizes Chernoff-Hoeffding inequality as in most UCB algorithms while recognizing costs.
- λ is lower bound of the expected costs across arms.

Ref:
W. Ding, T. Qin, X. Zhang, and T. Liu, "Multi-armed bandit with budget constraint and variable costs," in AAAI, 2013.

Node 4: Optimization with constraints





Maximize expected total reward $E\left[\sum R_{i,t}\right]$ given that every bid costs $c_{i,t}$ and the budget is capped at B.

Algorithm 1 UCB-BV1/UCB-BV2

Initialization: Pull each arm i once in the first K steps, set $t = K$.

while $\sum_{s=1}^{t} c_{a_s,s} \le B$ **do**

Set $t = t + 1$.

Calculate the index $D_{i,t}$ of each arm i as follows.

UCB-BV1

$$D_{i,t} = \frac{\bar{r}_{i,t}}{\bar{c}_{i,t}} + \frac{(1+\frac{1}{\lambda})\sqrt{\frac{\ln(t-1)}{n_{i,t}}}}{\lambda - \sqrt{\frac{\ln(t-1)}{n_{i,t}}}}$$

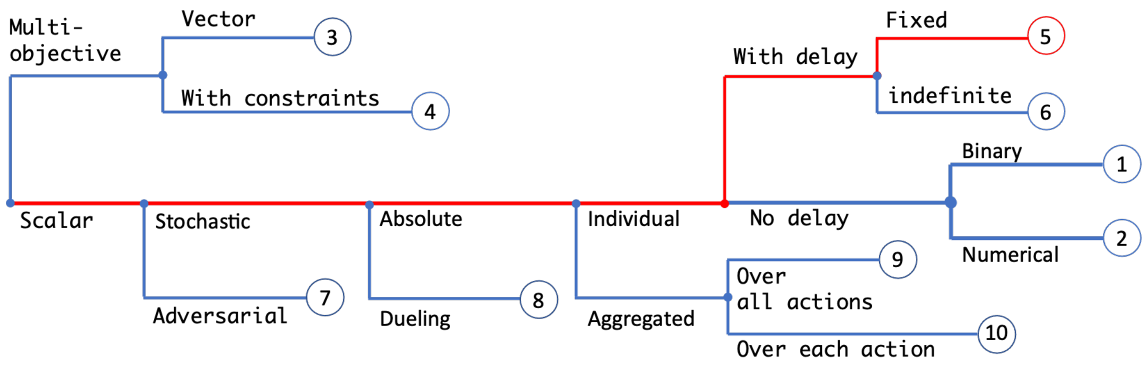
UCB-BV2

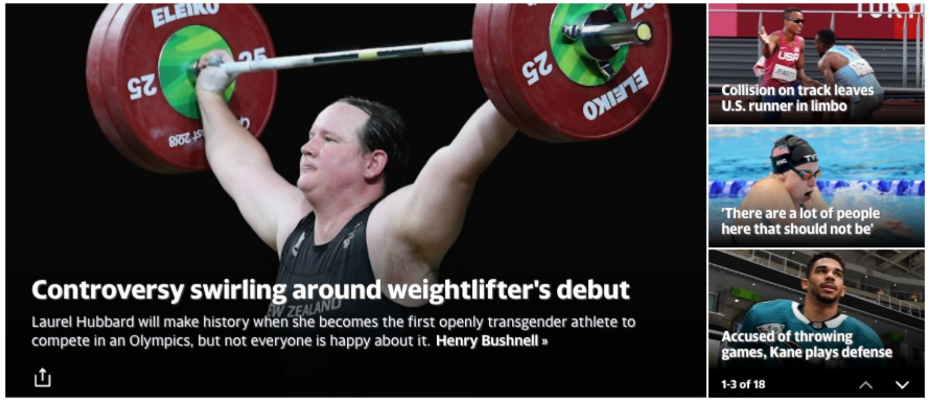
$$D_{i,t} = \frac{\bar{r}_{i,t}}{\bar{c}_{i,t}} + \frac{1}{\lambda_t}\left(1 + \frac{1}{\lambda_t - \sqrt{\frac{\ln(t-1)}{n_{i,t}}}}\right)\sqrt{\frac{\ln(t-1)}{n_{i,t}}}$$

- λ_t is estimated as the estimated minimum of the expected costs by using their empirical observations.

Ref:
W. Ding, T. Qin, X. Zhang, and T. Liu, "Multi-armed bandit with budget constraint and variable costs," in AAAI, 2013.

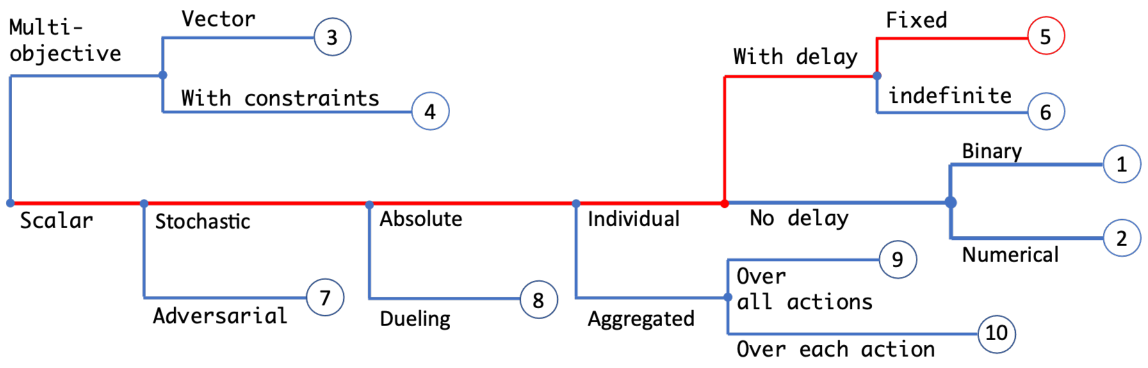
Node 5: Fixed (bounded) reward delays

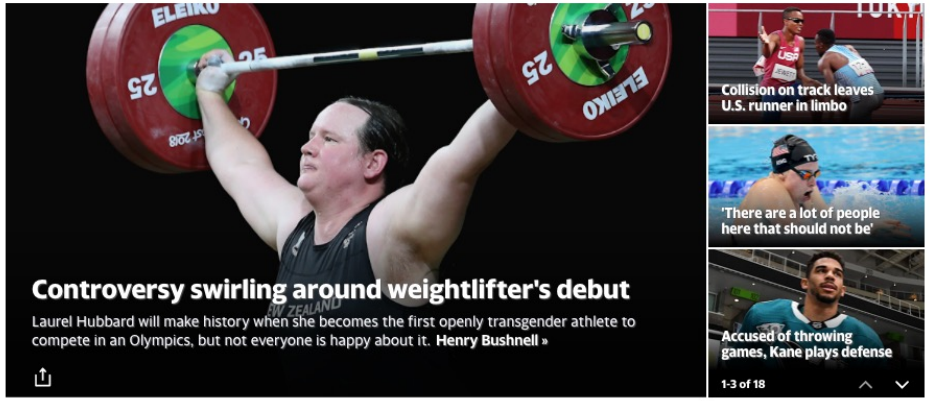




For news or social media, feedback is typically not able to come back immediately because of various runtime constraints.
Instead it usually arrives in batches over a certain period of time.

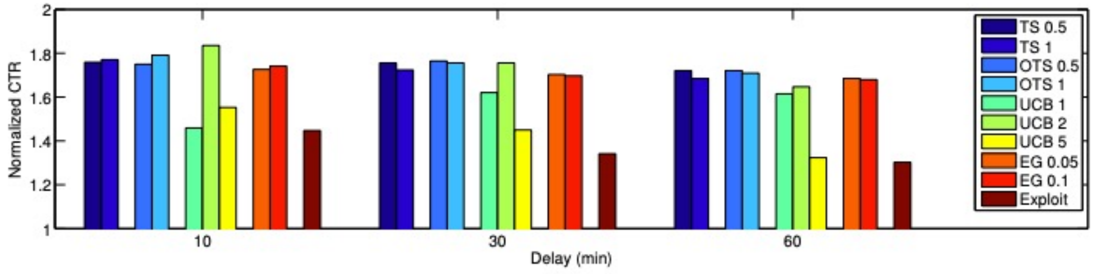
Node 5: Fixed (bounded) reward delays





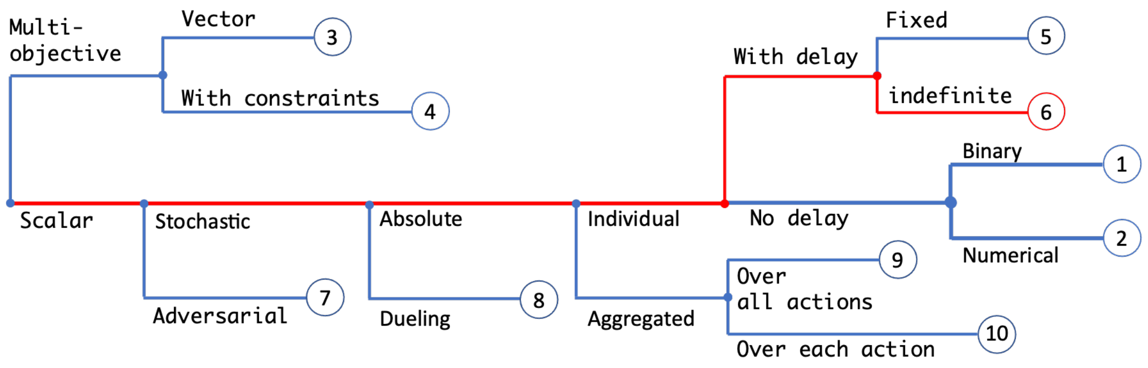
For news or social media, feedback is typically not able to come back immediately because of various runtime constraints.
Instead it usually arrives in batches over a certain period of time.

Thompson sampling is robust to delay in reward.



Ref:
O. Chapelle and L. Li, “An empirical evaluation of Thompson sampling,” in NIPS, 2011.

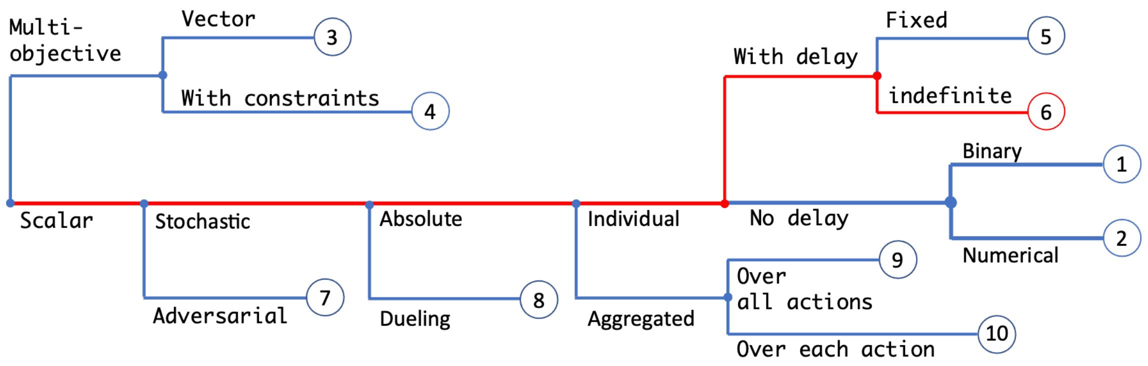
Node 6: Indefinite reward delays



What if the delay is not fixed/bounded but indefinite?

- Have you watched a movie on a weekend because of a recommendation during the week?
- Have you bought a product a month after your saw its advertisement?

Node 6: Indefinite reward delays

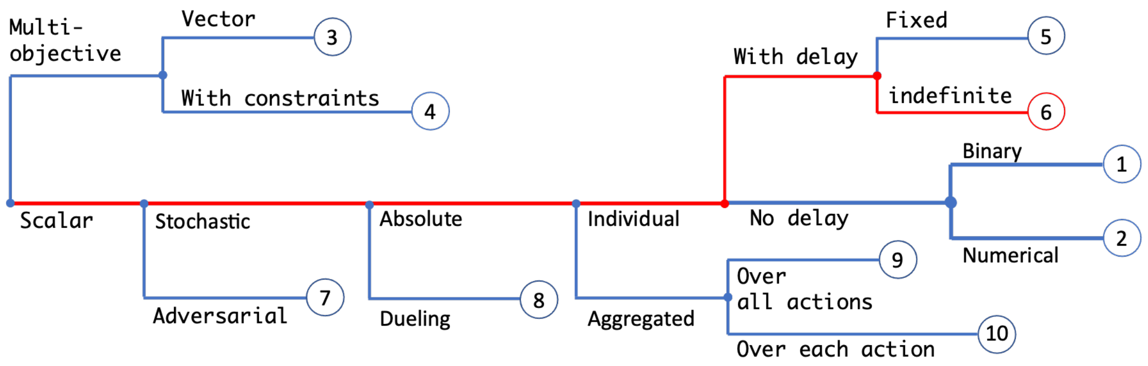


Maximize expected total reward $E[\sum R_{i,t}]$ given there is indefinite delay in receiving the reward signal.

Or state as:

Maximize expected total reward $E[\sum R_{i,t}]$ when the learner only observes delayed positive events.

Node 6: Indefinite reward delays

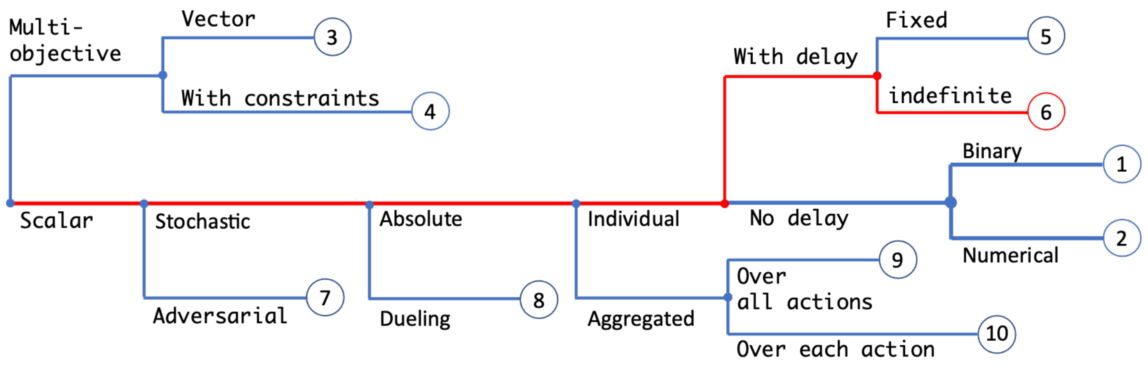


Maximize expected total reward $E[\sum R_{i,t}]$ given there is indefinite delay in receiving the reward signal.

Or state as:
Maximize expected total reward $E[\sum R_{i,t}]$ when the learner only observes delayed positive events.

Using surrogate metrics,
same-day buy instead of waiting for days/weeks,
is a pragmatic way to deal with delay.

Node 6: Indefinite reward delays



Maximize expected total reward $E[\sum R_{i,t}]$ given there is indefinite delay in receiving the reward signal.

Or state as:

Maximize expected total reward $E[\sum R_{i,t}]$ when the learner only observes delayed positive events.

If a reward has not converted within m rounds, the algorithm assumes it will never convert.

Algorithm 1 `OTFLinUCB`

Input: Window parameter $m > 0$, confidence level $\delta > 0$ and $\lambda > 0$.

for $t = 2, \ldots, T$ **do**

Receive action set $\mathcal{A}_t$

Compute width of confidence interval:

$$\alpha_{t,\delta} = 2f_{t,\delta} + \sum_{s=t-m}^{t-1} \|A_s\|_{V_t(\lambda)^{-1}}$$

Compute the least squares estimate $\hat{\theta}_t^{\mathrm{W}}$ using (2)

Compute the optimistic action:

Exploration

$$A_t = \arg\max_{a \in \mathcal{A}_t} \langle a, \hat{\theta}_t^{\mathrm{W}} \rangle + \alpha_{t,\delta} \|a\|_{V_t(\lambda)^{-1}}$$

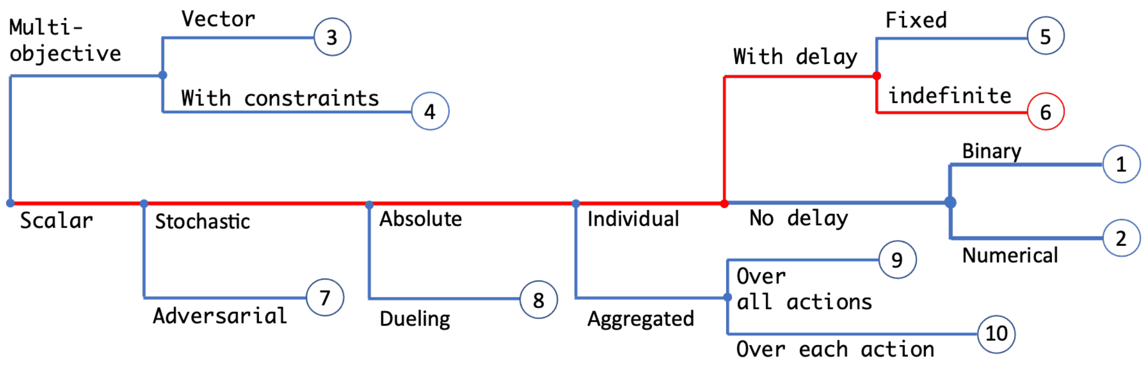
Exploitation

Play A_t and receive observations

end for

Ref:
C. Vernade, A. Carpentier, T. Lattimore, G. Zappella, B. Ermis, and M. Brueckner, "Linear bandits with stochastic delayed feedback," in ICML, 2020.

Node 6: Indefinite reward delays



Maximize expected total reward $E\left[\sum R_{i,t}\right]$ given there is indefinite delay in receiving the reward signal.

Or state as:

Maximize expected total reward $E\left[\sum R_{i,t}\right]$ when the learner only observes delayed positive events.

If a reward has not converted within m rounds, the algorithm assumes it will never convert.

Algorithm 1 OTFLinUCB

Input: Window parameter $m > 0$, confidence level $\delta > 0$ and $\lambda > 0$.

for $t = 2, \ldots, T$ **do**

Receive action set $\mathcal{A}_t$

Compute width of confidence interval:

$$\alpha_{t,\delta} = 2f_{t,\delta} + \sum_{s=t-m}^{t-1} \|A_s\|_{V_t(\lambda)^{-1}}$$

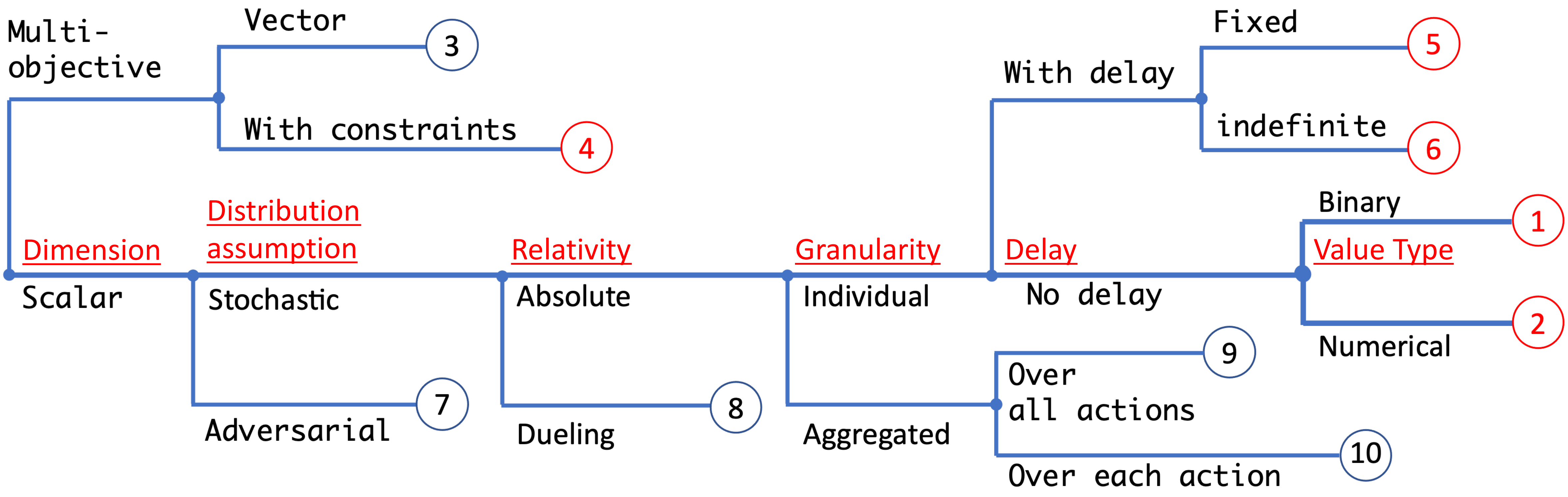
Compute the least squares estimate $\hat{\theta}_t^W$ using (2)

L_2-Regularized least square estimation where rewards that convert after more than m rounds are ignored.

Ref:

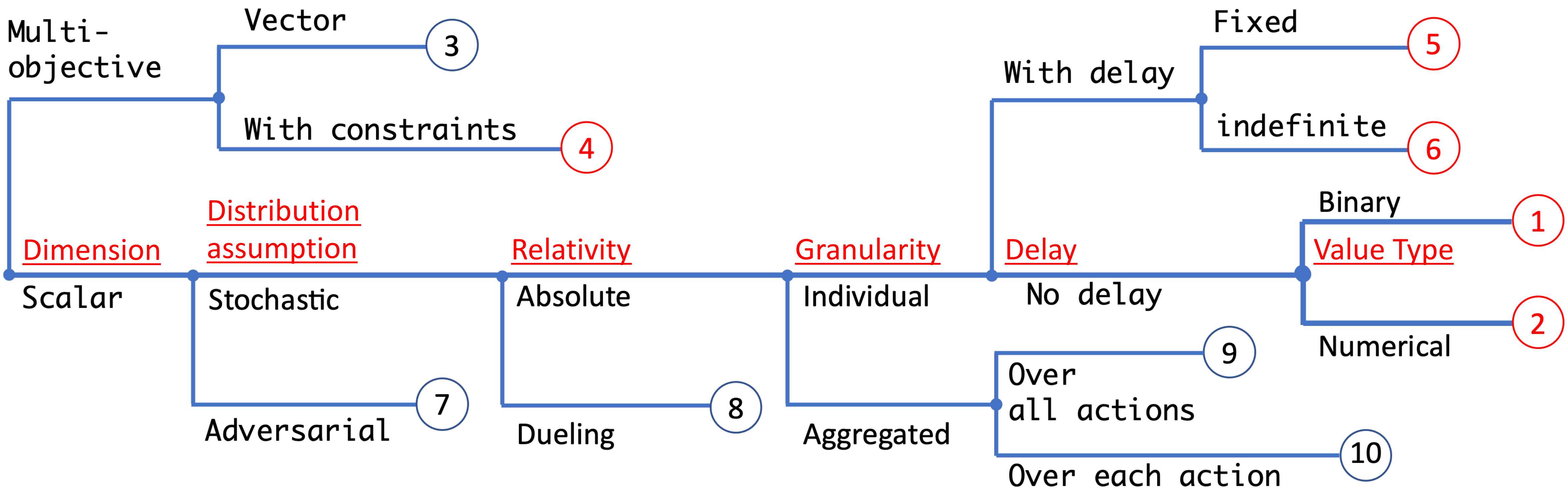
C. Vernade, A. Carpentier, T. Lattimore, G. Zappella, B. Ermis, and M. Brueckner, “Linear bandits with stochastic delayed feedback,” in ICML, 2020.

Bandit Problems by Reward Properties



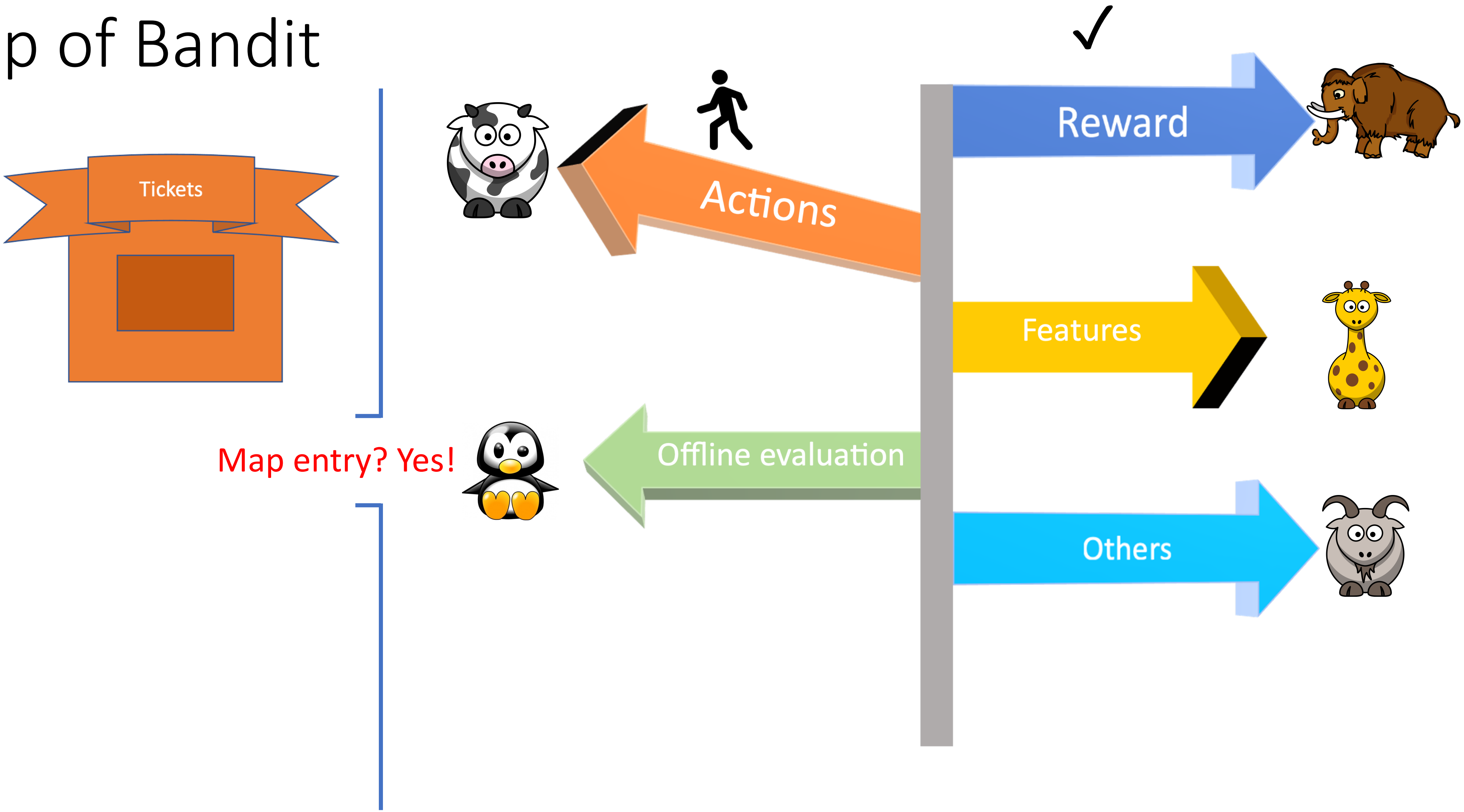
- No distribution assumption -> Adversarial 7
- Action preference instead of absolute reward -> Dueling 8
- Reward depends on multiple actions -> aggregated 9 10

Bandit Problems by Reward Properties

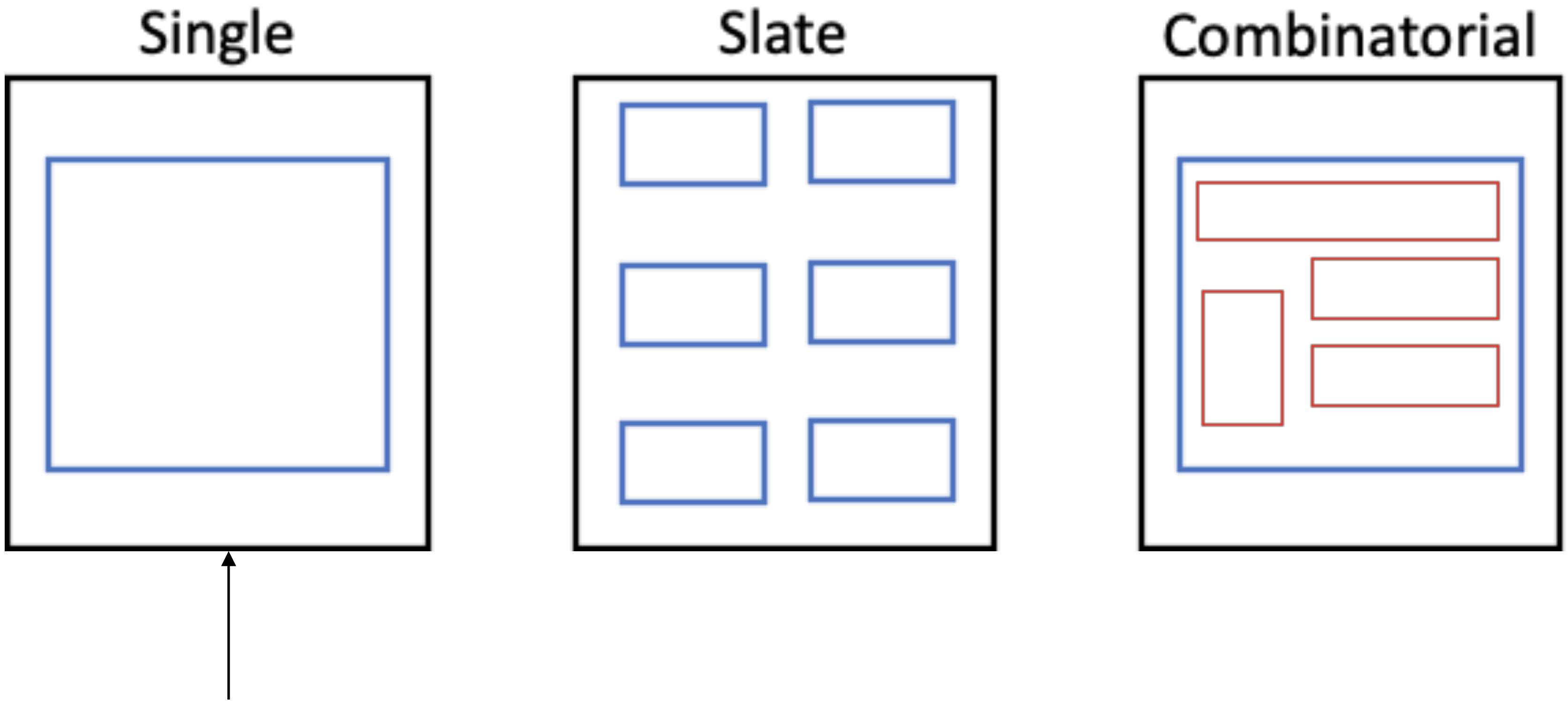


Nodes 3–10 are not exhaustive as the splits are not mutually exclusive.
For instance, an adversarial Bandit can also be a dueling one and there can be delay in reward.
In practice, however, such combinations appear uncommon.

A Map of Bandit



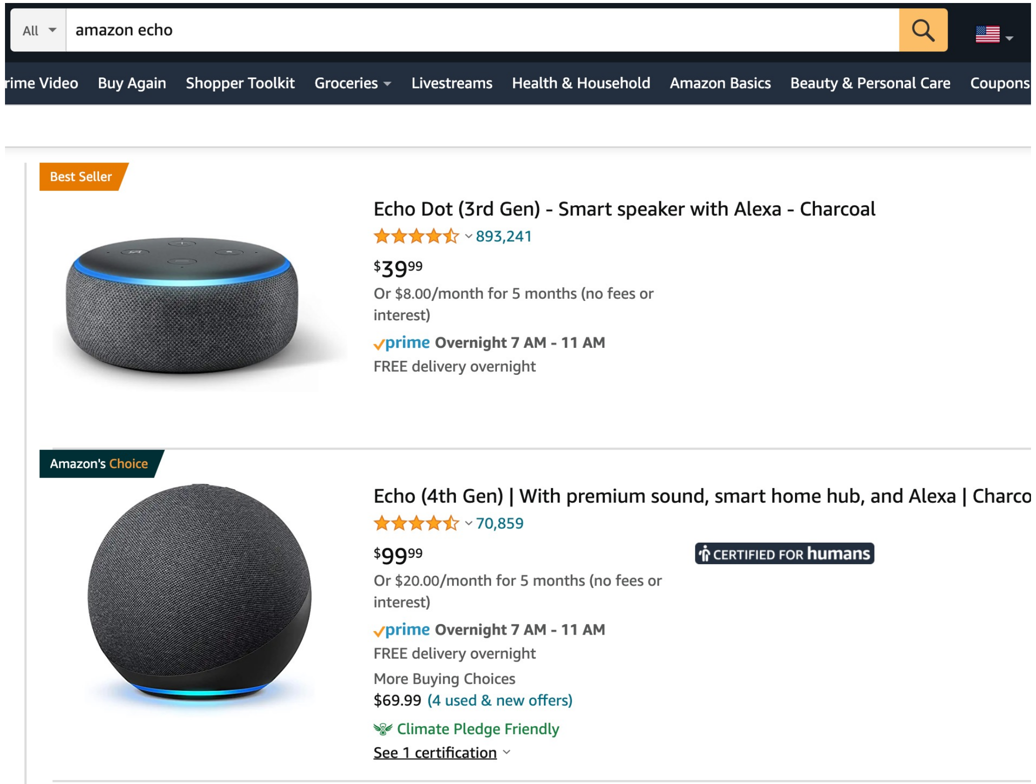
Common Action Types



The baseline case:
pull an arm and observe a reward afterwards

Slate Actions

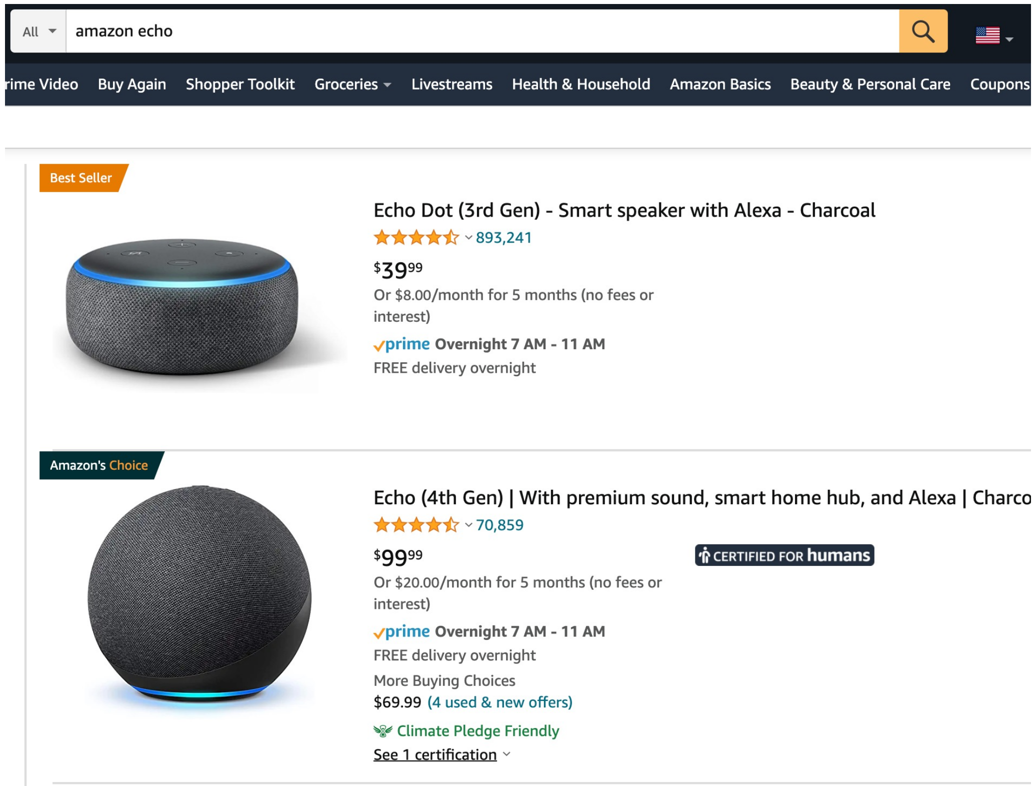
Return a ranked result list for user's search query



The goal is to maximize the total revenue per search result return, while you can track the revenue for each shown product.

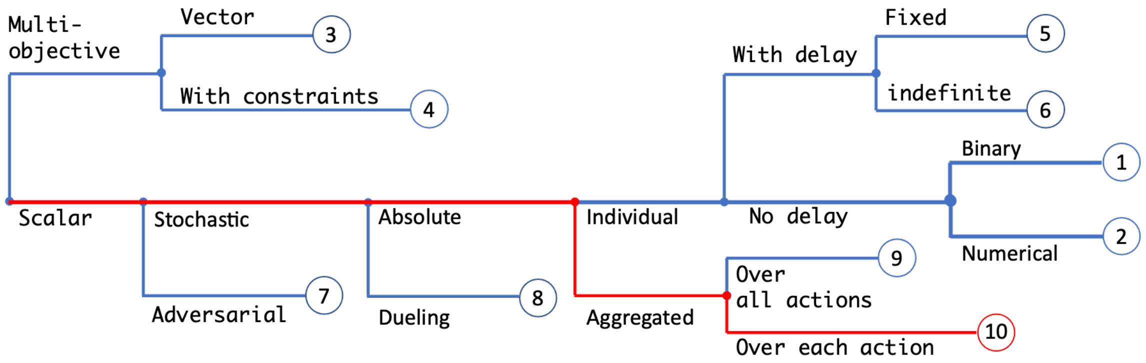
Slate Actions

Return a ranked result list for user's search query



The goal is to maximize the total revenue per search result return, while you can track the revenue for each shown product.

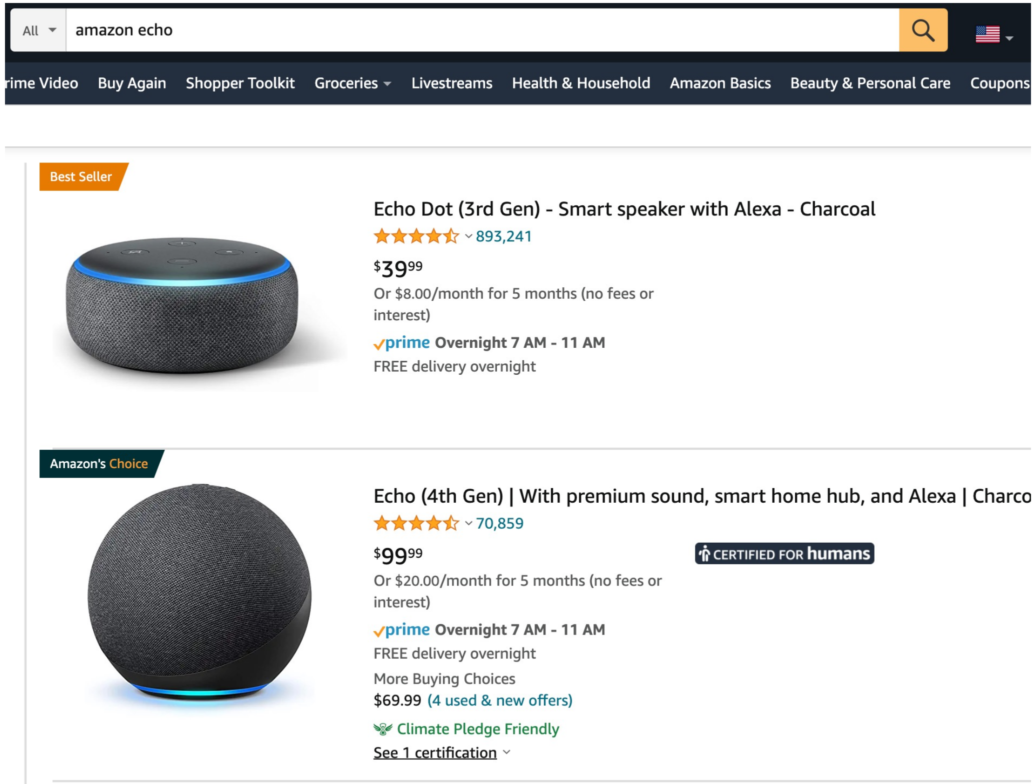
⇔ Node 10;
Reward Granularity:
aggregated over each action



Semi-bandit is defined.

Slate Actions

Return a ranked result list for user's search query



The goal is to maximize the total revenue per search result return, while you can track the revenue for each shown product.

Algorithm 2 Combinatorial Linear Thompson Sampling

Input: Combinatorial structure $(E, \mathcal{A})$, generalization matrix $\Phi \in \mathbb{R}^{L \times d}$, algorithm parameters $\lambda, \sigma > 0$, oracle $\mathtt{ORACLE}$

Initialize $\Sigma_1 \leftarrow \lambda^2 I \in \mathbb{R}^{d \times d}$ and $\bar{\theta}_1 = 0 \in \mathbb{R}^d$

for all $t = 1, 2, \ldots, n$ **do**

  Sample $\theta_t \sim N\left(\bar{\theta}_t, \Sigma_t\right)$

  Compute $A^t \leftarrow \mathtt{ORACLE}(E, \mathcal{A}, \Phi\theta_t)$

  Choose set A^t, and observe $\mathbf{w}_t(e), \forall e \in A^t$

  Compute $\bar{\theta}_{t+1}$ and Σ_{t+1} based on Algorithm 1

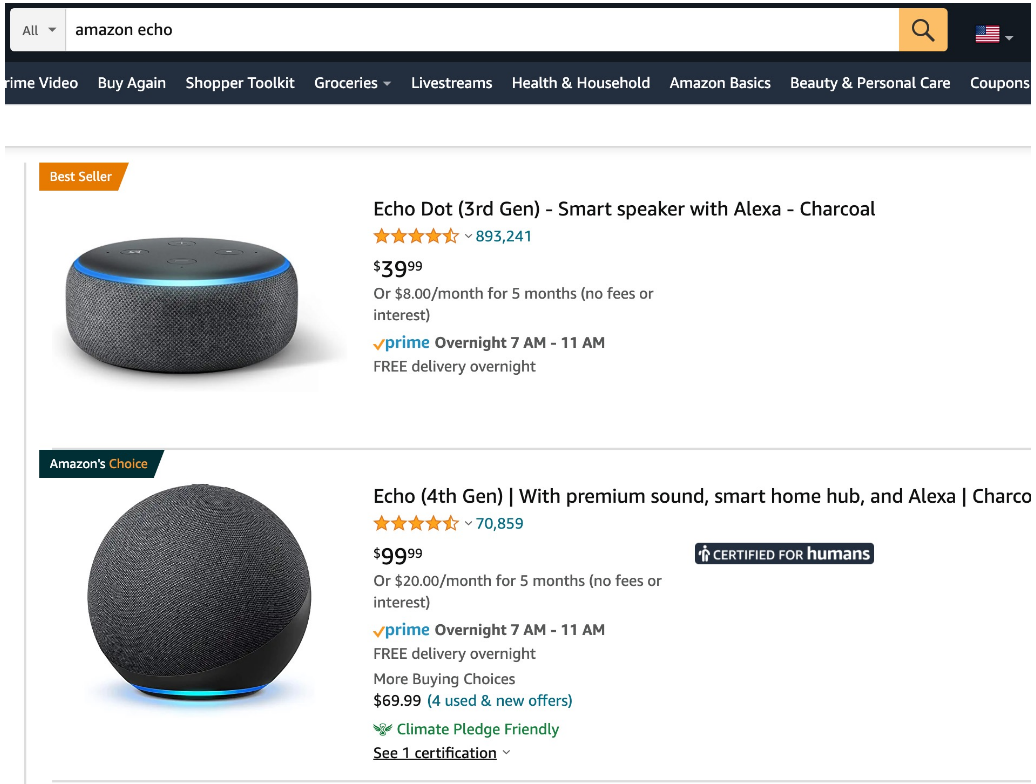
end for

Find the optimal list given the conditions, using combinatorial optimization algorithms.

Ref:
Z. Wen, B. Kveton, and A. Ashkan, "Efficient learning in large-scale combinatorial semi-bandits," in ICML, 2015

Slate Actions (position and diversity effects)

Return a ranked result list for user's search query



The goal is to maximize the total revenue per search result return, while you can track the revenue for each shown product.

Algorithm 2 Combinatorial Linear Thompson Sampling

Input: Combinatorial structure $(E, \mathcal{A})$, generalization matrix $\Phi \in \mathbb{R}^{L \times d}$, algorithm parameters $\lambda, \sigma > 0$, oracle ORACLE

Initialize $\Sigma_1 \leftarrow \lambda^2 I \in \mathbb{R}^{d \times d}$ and $\bar{\theta}_1 = 0 \in \mathbb{R}^d$
for all $t = 1, 2, \dots, n$ **do**
  Sample $\theta_t \sim N(\bar{\theta}_t, \Sigma_t)$
  Compute $A^t \leftarrow \texttt{ORACLE}(E, \mathcal{A}, \Phi\theta_t)$
  Choose set A^t, and observe $\mathbf{w}_t(e), \forall e \in A^t$
  Compute $\bar{\theta}_{t+1}$ and Σ_{t+1} based on Algorithm 1
end for

Find the optimal list given the conditions, using combinatorial optimization algorithms.

Ref:
Z. Wen, B. Kveton, and A. Ashkan, "Efficient learning in large-scale combinatorial semi-bandits," in ICML, 2015

Combinatorial Actions

Content layout on a webpage
for upselling membership/subscription

Challenges:

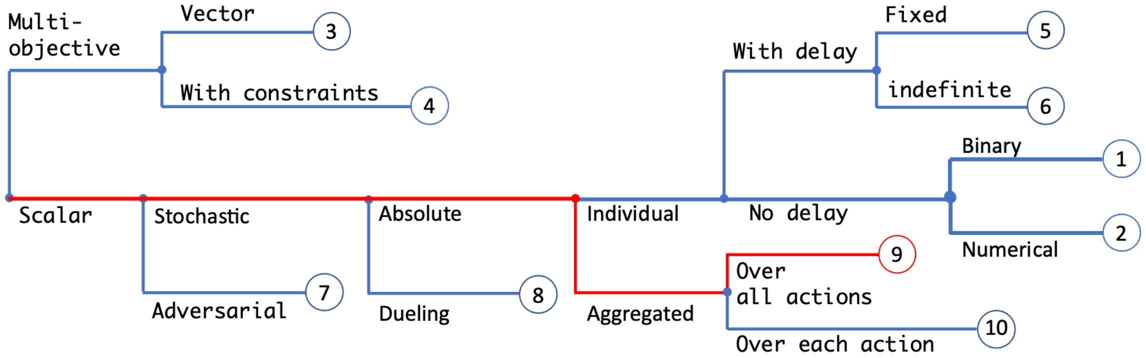
- Combinatorial explosions of actions
- Interaction effects between sub-actions

Combinatorial Actions

Content layout on a webpage
for upselling membership/subscription

Challenges:
- Combinatorial explosions of actions
- Interaction effects between sub-actions

⇔ Node 9; Reward Granularity: aggregated over all actions



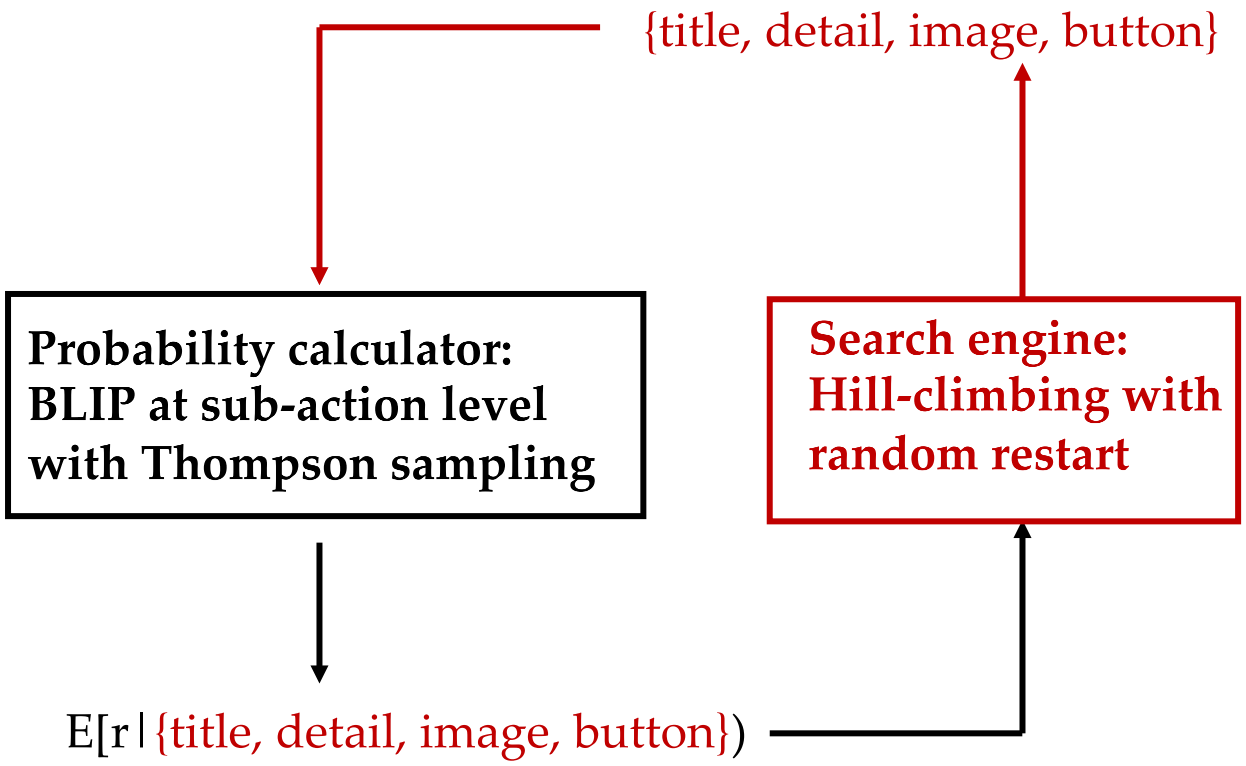
Combinatorial Actions

Content layout on a webpage
for upselling membership/subscription

Challenges:

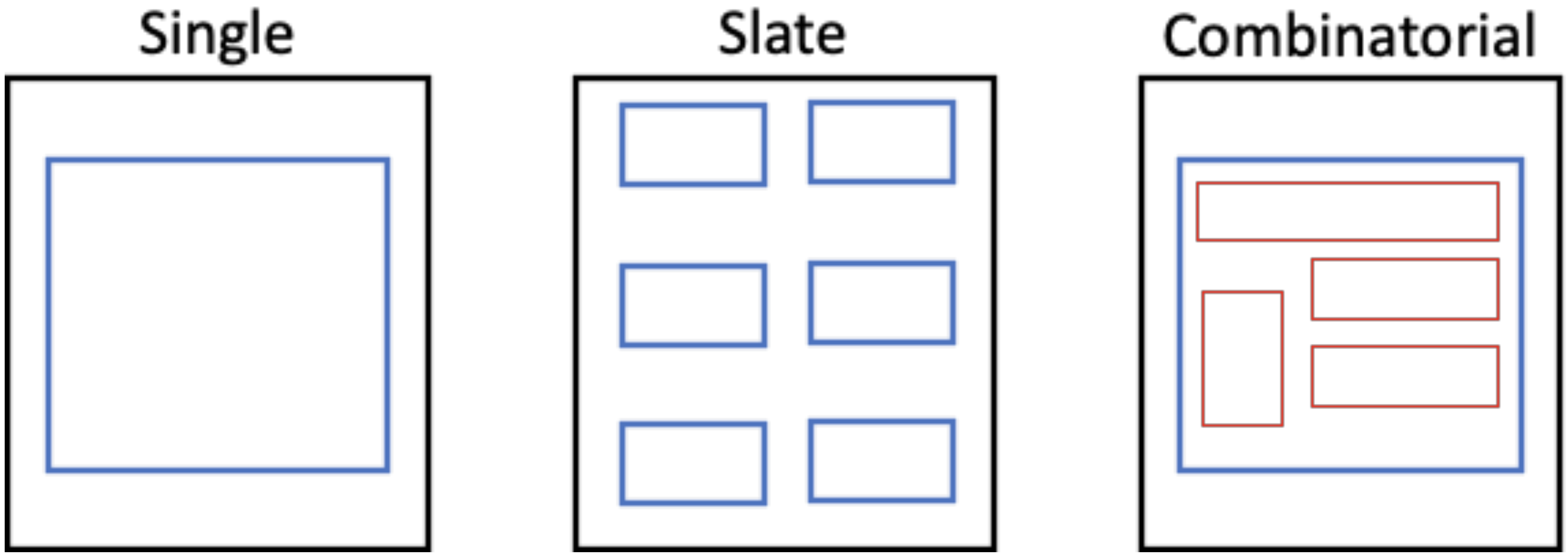
- Combinatorial explosions of actions
- Interaction effects between sub-actions

Algorithm: multivariate Bandit



Ref:
D. N. Hill, H. Nassif, Y. Liu, A. Iyer, and S. Vishwanathan, "An efficient bandit algorithm for realtime multivariate optimization," in KDD, pp. 1813–1821, 2017.

Common Action Types



Next question to ask:
Do we formulate Bandit differently if different sizes of the action set?

Action Set Size

Model actions as categorical variables.

Represent each action as a feature vector in the reward function.

- Discretize the action space.
- Continuous Bandit

Bandits with discrete actions

- Video recommendation
- Product recommendation
- Inventory buying

Bandits with continuous actions

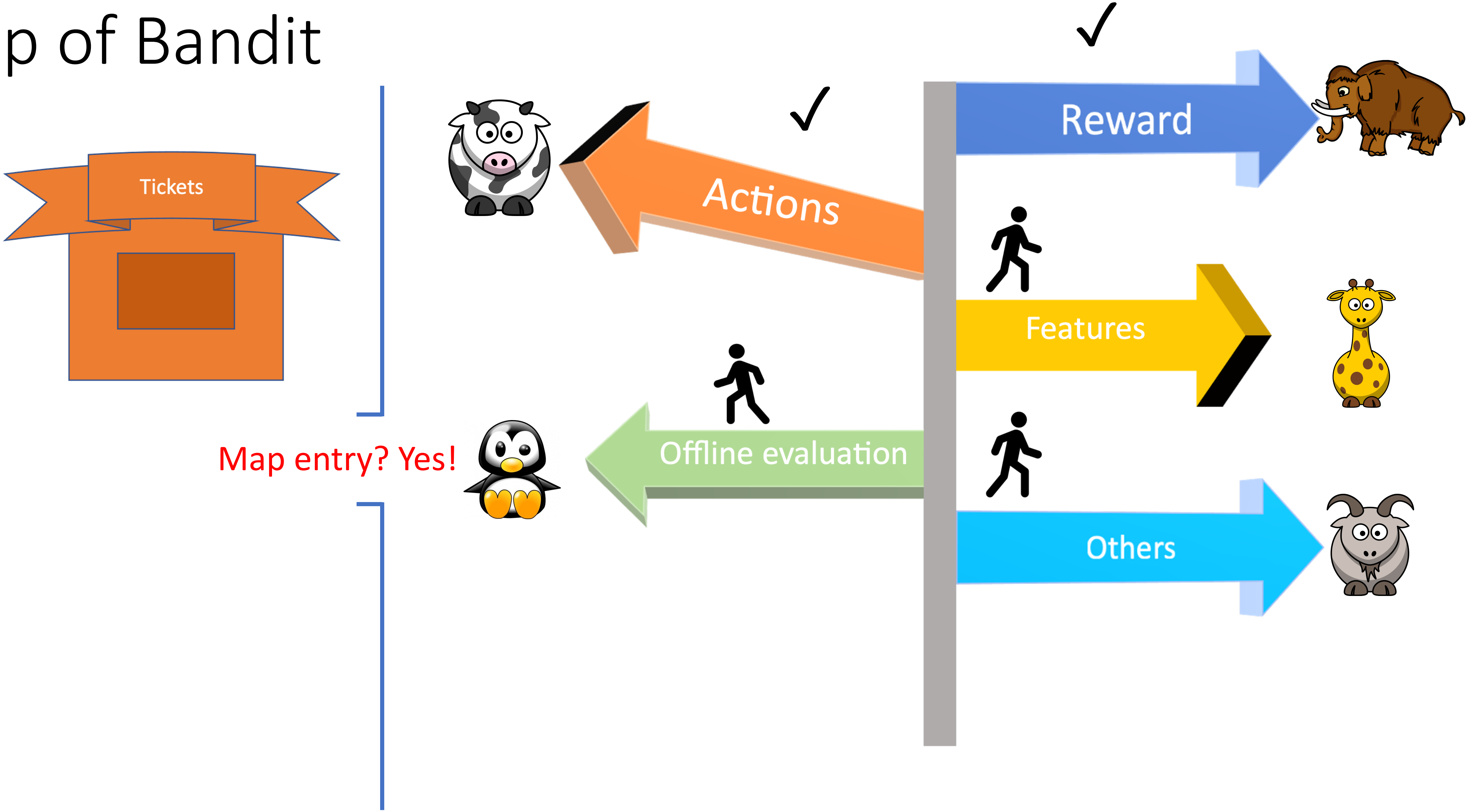
- Advertisement recommendation
- Fashion style recommendation
- Skill recommendation for virtual assistant

- Dynamic pricing
- Hyper parameter search

- Algorithm selection
- Marketing message recommendation

~100 or less than | ~ thousands | ~millions | infinite | Size of action set

A Map of Bandit



Feature Engineering

- Determining input features: $\boldsymbol{\phi_a}$ (for action), $\boldsymbol{\phi_x}$ (for context)
  - Needed for large action/context spaces
  - Used in modeling reward: $E[r] = f(\boldsymbol{\phi_a}, \boldsymbol{\phi_x})$, or policy
- Linear bandits examples
  - $E[r] = \boldsymbol{w} \cdot (\boldsymbol{\phi_a} \otimes \boldsymbol{\phi_x})$ with unknown weights $\boldsymbol{w}$
  - Learn lower-dimensional embeddings as features
- Nonlinear bandits examples
  - Kernelised Bandits Michal et al. “Finite-Time Analysis of Kernelised Contextual Bandits,” UAI, 2013.
  - Neural Bandits Zhou et al. “Neural contextual bandits with UCB-based exploration,” ICML, 2020.
  - …

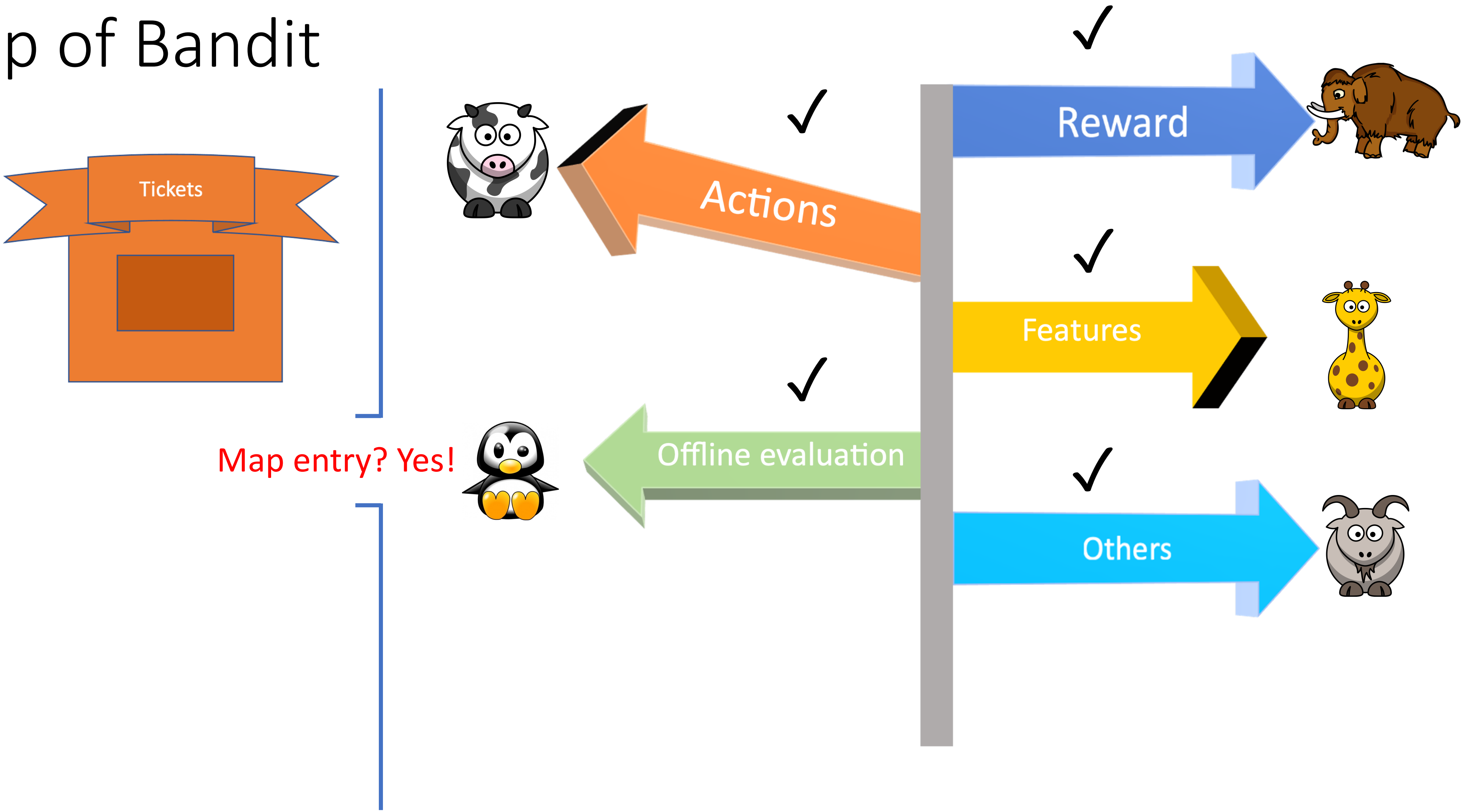
Offline (Off-policy) Policy Evaluation

- Often critical to evaluate a new policy offline before deploying it.
- Challenge: we don’t know user reaction to actions different from the log
- Similar to counterfactual analysis in causal inference.
- Usually, we assume stationary policy. Common methods:
  - Simulation: Bayir, et al. “Genie: An open box counterfactual policy estimator for optimizing sponsored search marketplace,” in WSDM, 2019.
  - Inverse propensity scoring and self-normalized variants: A. Swaminathan and T. Joachims, “The self-normalized estimator for counterfactual learning,” in NIPS, 2015.
  - Doubly robust evaluation: M. Dudik, J. Langford, and L. Li, “Doubly robust policy evaluation and learning,” in ICML, 2011.

Others

- Best-arm identification. the goal is not to maximize reward during an experiment, but to identify the best action (e.g., best marketing campaign strategy) at the end of the experiment.
- Privacy-preserving bandits. A system that updates local agents by collecting feedback from other local agents in a differentially-private manner.

A Map of Bandit



Which Bandit algorithms are for your problem?

Business problems with different characteristics $\xrightarrow{?}$ A zoo of Bandit algorithms

What marketing content to recommend to maximize offer signup?

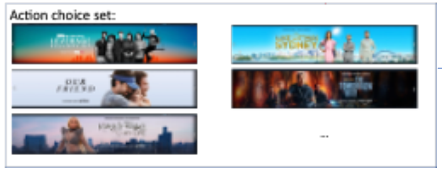
Beat-the-Mean
BLIP
LinTSPBMRank
OTFLinUCB
LinGreedy
NeuralUCB
C^2UCB
CombLinTS
Multivariate bandits

Use our paper as a map to find the answer ☺

Our map for your look up

<Your business problems> → <Your solutions>

Recommend a video to maximize watch time?



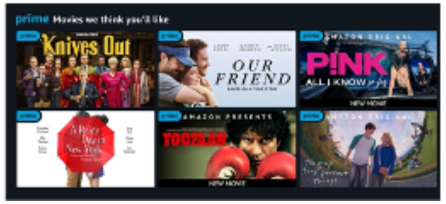
Numerical reward with delay

Single action

OTFLinUCB

C. Vernade, A. Carpentier, T. Lattimore, G. Zappella, B. Ermis, and M. Brueckner, "Linear bandits with stochastic delayed feedback," in ICML, 2020.

Recommend a subset of videos to maximize watch time?



Numerical reward

Slate Bandit

LinTSPBMRank

B. Ermis, P. Ernst, Y. Stein, and G. Zappella, "Learning to rank in the position based model with bandit feedback," in CIKM, 2020.

What marketing content to recommend to maximize offer signup?

Binary reward

Combinatorial action, 2 customer features

Multivariate bandits

D. N. Hill, H. Nassif, Y. Liu, A. Iyer, and S. Vishwanathan, "An efficient bandit algorithm for realtime multivariate optimization," in KDD, pp. 1813–1821, 2017.

…

… → …