

Online Learning from Human Preferences

Active Learning, Duelling Bandits,
and Contextual Comparisons

Suryanarayana Sankagiri
Wadhvani School of Data Science and AI
Indian Institute of Technology Madras

[Personal webpage](#) | [Email](#)

DA5453 — Learning from Human Preferences
First offered at IIT Madras, July–November 2026
[Course webpage](#)

Module 3 October 2026

AI-use disclaimer. This document was generated with substantial assistance from AI tools and may contain hallucinations, mathematical errors, omissions, or passages that are difficult to read. Nevertheless, I, Suryanarayana Sankagiri, accept full responsibility for the material presented here. Readers are encouraged to contact me about any suspected error or unclear passage.

About these notes

These notes cover Module 3 of DA5453, Learning from Human Preferences. In [the earlier modules](#), we studied how to learn from a given set of comparisons. Here, we also choose which comparisons to collect. We begin with active learning, where the aim is to find a good item using few queries. We then study duelling bandits, where the quality of the items shown during learning also matters. Finally, we use item features to share information across comparisons, leading to contextual linear bandits and contextual duelling bandits.

The chapters develop the models, algorithms, and their guarantees step by step. Each chapter ends with a summary of its main ideas and equations. The appendices give some model calculations and the probability and matrix results used in the bandit proofs; you can return to them when those results are first needed. Notation is introduced in the text and collected at the end for reference. These notes are intended both for students taking the course and for readers following the material independently.

Contents

About these notes	ii
1 Active learning from comparisons	1
1.1 Choosing which comparisons to collect	1
1.2 Repeated comparisons under noise	2
1.3 Allowing an approximately best item	3
1.4 A noisy knockout tournament	4
1.5 Tracking the best remaining item	4
1.6 Allocating accuracy and confidence across rounds	5
1.7 Geometric allocation and the final guarantee	6
1.8 Other pool sizes and ties	7
1.9 Chapter summary	8
2 Duelling bandits	9
2.1 The duelling bandit problem	9
2.2 Hoeffding's inequality and confidence intervals	10
2.3 The RUCB algorithm and its guarantee	11
2.4 A numerical example: from testing to using the winner	13
2.5 Proof of the regret guarantee	15
2.6 Models, winner definitions, and the regret scale	19
2.7 Ordinary UCB and the exploration bonus	19
2.8 Chapter summary	20
3 Contextual duelling bandits	21
3.1 A shared model of item utilities	21
3.2 Regret in utility units	22
3.3 Estimating the preference weights	22
3.4 The directions measured by our comparisons	23
3.5 Linear bandits and a fixed confidence radius	24
3.6 Confidence ellipsoids and a fixed logistic radius	27
3.7 From one parameter set to every pairwise interval	28
3.8 Plausible winners and MaxInP	28
3.9 Bounding the regret of the selected pair	29
3.10 Cumulative regret and the meaning of the bound	30
3.11 Changing contexts and available items	31
3.12 Chapter summary	32
A Model checks and regression geometry	33
A.1 BTL satisfies SST and STI	33
A.2 Fixed-design regression	33
B The elliptic-potential lemma	34
B.1 Rank-one growth and the general lemma	34
B.2 When the truncation can be removed	34
B.3 The effect of one query on a width	35
C Adaptive concentration: the Abbasi-Yadkori lemma	36
C.1 Conditional noise and predictable inputs	36
C.2 A proof through exponential mixtures	36
C.3 A deterministic horizon bound	37

D	A confidence proof for the constrained logistic estimator	38
D.1	Centred Bernoulli residuals	38
D.2	A uniform lower bound on likelihood curvature	38
D.3	Use constrained optimality, including the boundary	39
D.4	Interpreting the constants	39
E	Notation reference	40
	Sources and further reading	41

Active learning from comparisons

1.1 Choosing which comparisons to collect

So far, we have largely worked with a comparison dataset that is already available. In many settings, we can instead decide which comparisons to ask for. Suppose, for example, that we have several candidate responses to a prompt and want to identify the best one. Asking a person to compare every pair may be expensive. Once we have evidence that a response is poor, we might prefer to spend the remaining queries on the more promising responses.

This is the idea of *active learning*: the learner chooses which data to collect, using what it has learned so far. We will study a simple version of this problem—finding the best among a fixed collection of items—and ask how many comparisons are needed.

1.1.1 The comparison model

There are K fixed items, labelled $[K] = \{1, \dots, K\}$, with unknown comparison probabilities

$$p_{ij} = \mathbb{P}(i \text{ wins against } j), \quad p_{ji} = 1 - p_{ij}, \quad p_{ii} = \frac{1}{2}.$$

Thus p_{ij} is the chance that i wins when we compare it with j . These probabilities are fixed but unknown. Each query uses fresh noise: once the pair is chosen, its outcome has the stated probability, irrespective of earlier outcomes. The learner may use those earlier outcomes to choose its next pair.

Assume that there is a hidden order $\succeq$ of the items, with $i \succeq j$ implying $p_{ij} \geq 1/2$. In words, an item that is higher in the order wins at least as often as it loses against a lower item. The learner does not know this order. We will add two further assumptions when we analyse noisy selection.

1.1.2 A noiseless knockout tournament

Let us first consider the case in which a comparison always reveals the better item. The natural approach is a knockout tournament, just as in tennis or football.

Noiseless knockout

Repeat until only one item remains:

1. Pair all remaining items arbitrarily.
2. Compare each pair.
3. Keep the winners.

Return the last remaining item.

The best item never loses, so it survives every round. Since the procedure ends with one item, that item must be the best.

For now, take $K = 2^R$, so that every round can pair the remaining items without a bye. Section 1.8 handles other sizes. There are $R = \log_2 K$ rounds. Round r has $K/2^r$ comparisons, so the total is

$$\frac{K}{2} + \frac{K}{4} + \dots + 1 = K - 1.$$

This is optimal for noiseless exact selection: every other item must have lost at least once before we can rule it out as the maximum. One comparison supplies only one loss, so noiseless exact selection requires at least $K - 1$ comparisons.

Notice how the comparisons depend on what we have observed. Once an item loses, we know it cannot be the maximum, so we stop comparing it. Later rounds compare only the winners of earlier rounds. Finding the maximum requires much less information than recovering the full ranking.

1.2 Repeated comparisons under noise

When comparisons are noisy, a single knockout comparison can eliminate the best item. The simple way around this is to compare a pair repeatedly and take the majority outcome. We will call this entire contest a *match*: a match consists of m elementary comparisons, followed by a majority decision.

Suppose $i \succeq j$ and, for some known $\gamma > 0$,

$$p_{ij} \geq \frac{1}{2} + \gamma.$$

For example, $\gamma = .01$ covers $p_{ij} = .51$, but not $p_{ij} = .5001$. Let $\hat{p}_{ij}$ be the fraction of the m observations won by i . A wrong majority decision requires $\hat{p}_{ij} \leq 1/2$, including a possible empirical tie. To bound this probability, recall the one-sided form of Hoeffding's inequality. For any error margin $\varepsilon > 0$,

$$\mathbb{P}(\hat{p}_{ij} - p_{ij} \leq -\varepsilon) \leq e^{-2m\varepsilon^2}.$$

Here a wrong decision requires an error of at least γ , so

$$\mathbb{P}(\text{match returns } j) \leq \mathbb{P}(\hat{p}_{ij} - p_{ij} \leq -\gamma) \leq e^{-2m\gamma^2}.$$

To make the error probability of this one match at most δ , it suffices to choose

$$m \geq \frac{1}{2\gamma^2} \log \frac{1}{\delta}.$$

Take a ceiling when choosing the integer repetition count. Break an empirical tie arbitrarily. All logarithms are natural unless a base is displayed.

1.2.1 Exact selection with a known minimum gap

Fix $0 < \delta < 1/2$. Assume, temporarily, that *every* better item beats every worse one with a known margin of at least γ . There are $K - 1$ matches. To ensure that all matches are correct with probability at least $1 - \delta$, give each match an error allowance of $\delta/(K - 1)$. Substituting this allowance in the one-match calculation gives

$$m = \left\lceil \frac{1}{2\gamma^2} \log \frac{K - 1}{\delta} \right\rceil.$$

Conditionally on the previous outcomes, each match has error probability at most $\delta/(K - 1)$. A union bound then gives

$$\mathbb{P}(\text{some match is wrong}) \leq (K - 1) \frac{\delta}{K - 1} = \delta.$$

On the event that all matches are correct, the best item survives. Hence exact selection succeeds with probability at least $1 - \delta$, using

$$(K - 1)m = O\left(\frac{K}{\gamma^2} \log \frac{K}{\delta}\right)$$

elementary comparisons. This is a sufficient bound; it need not be the sharpest.

The difficulty is that we usually do not know a lower bound on the true gaps. If two relevant items are almost tied, distinguishing them reliably can require very many observations. Without such a gap bound, no finite query budget depending only on K and δ can guarantee exact selection uniformly over these comparison problems.

1.3 Allowing an approximately best item

Suppose that two responses are almost indistinguishable in quality. Must we spend a large number of queries deciding which is slightly better? For many purposes, either response would be acceptable. We will therefore relax the target: instead of insisting on the exact best item, we ask for an item that is sufficiently close to it.

The (ε, δ) -PAC formulation

Given $0 < \varepsilon < 1/2$ and $0 < \delta < 1/2$, return an item $\hat{i}$ such that

$$\mathbb{P}\left(p_{\hat{i}j} \geq \frac{1}{2} - \varepsilon \quad \text{for every } j \in [K]\right) \geq 1 - \delta.$$

PAC means *probably approximately correct*. The approximation tolerance is ε ; the failure probability is δ . For $\varepsilon = .05$, no competitor should beat the returned item with probability greater than .55. We do not require the returned item to be the unique best. The outer probability is over the observed feedback and any random choices made by the algorithm.

This target does not require a minimum gap between distinct items. To achieve it efficiently, however, we need some structure in how the comparison probabilities relate to the hidden order. First define the signed preference margin

$$\Delta(i, j) := p_{ij} - \frac{1}{2}, \quad \Delta(j, i) = -\Delta(i, j).$$

For items in the order $a \succeq b \succeq c$, assume:

The preference structure

1. **A hidden total order.** $a \succeq b$ implies $p_{ab} \geq 1/2$.
2. **Strong stochastic transitivity (SST).**

$$\Delta(a, c) \geq \max\{\Delta(a, b), \Delta(b, c)\}.$$

3. **Stochastic triangle inequality (STI).**

$$\Delta(a, c) \leq \Delta(a, b) + \Delta(b, c).$$

SST says that a comparison with a more distant item in the order has at least as large a margin. For example, the best item should have no less an advantage over the third item than over the second. STI gives a complementary upper bound: the margin across two successive steps cannot exceed the sum of the margins for those steps.

These assumptions will serve different purposes in the proof. SST will let us check whether a returned item is good enough by comparing it only with the best item. STI will let us add up the losses when one good item is replaced by another. The BTL model from Module 1 satisfies both properties; Appendix A checks this directly.

1.3.1 Checking the returned item against the best

Let i^* be the true best item. Under SST, it is enough to ensure

$$\Delta(i^*, \hat{i}) \leq \varepsilon.$$

Indeed, if $j \succeq \hat{i}$, SST gives $\Delta(j, \hat{i}) \leq \Delta(i^*, \hat{i}) \leq \varepsilon$. If $\hat{i} \succeq j$, then $p_{\hat{i}j} \geq 1/2$ already. Thus every competitor satisfies the original PAC target. Conversely, that target includes the competitor $j = i^*$.

1.4 A noisy knockout tournament

We will keep the knockout structure. The question is how many times to compare each pair. Early rounds contain many matches, while the final round contains only one. We need not spend the same effort on every match: let m_r be the number of elementary comparisons used in each match of round r . We will choose these repetition counts after understanding what the proof requires.

Noisy knockout, with repetition counts still to be chosen

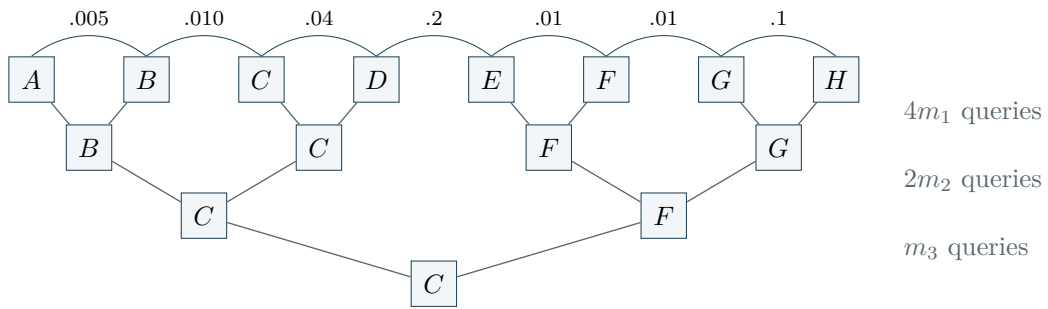
Start with $S_1 = [K]$, where $K = 2^R$. For $r = 1, \dots, R$:

1. Pair the items in S_r arbitrarily, before observing this round's outcomes.
2. Play each pair m_r times using fresh observations.
3. Advance each empirical majority winner to S_{r+1} ; break an empirical tie arbitrarily.

Return the sole item in S_{R+1} .

1.4.1 An eight-item example

Let $A \succ B \succ \dots \succ H$. The labels above adjacent items are *preference margins* Δ , not win probabilities. Below them is one possible set of empirical majority outcomes.



Suppose $\varepsilon = .05$. Then A, B, C are all good enough. In particular, STI gives

$$\Delta(A, C) \leq \Delta(A, B) + \Delta(B, C) = .005 + .010 = .015 < .05.$$

So eliminating A and then B need not be a failure. The sets of these good items remaining in the illustrated run are

$$\{A, B, C\} \longrightarrow \{B, C\} \longrightarrow \{C\} \longrightarrow \{C\}.$$

We need *some* sufficiently good item to remain; it need not always be the same one.

Let b_r denote the best item at the start of round r . In this example, $b_1 = A, b_2 = B, b_3 = C, b_4 = C$. The learner observes only outcomes and survivor sets; the identities b_r are used in the proof. The next step is to quantify how much the best remaining item can deteriorate.

1.5 Tracking the best remaining item

Let $b_r = \text{best}(S_r)$ be the highest-ranked member of S_r . Because each survivor set is a subset of the preceding set,

$$i^* = b_1 \succeq b_2 \succeq \dots \succeq b_{R+1} = \hat{i}.$$

Our goal is $\Delta(i^*, \hat{i}) \leq \varepsilon$. By repeated application of STI,

$$\Delta(i^*, \hat{i}) \leq \sum_{r=1}^R \Delta(b_r, b_{r+1}).$$

Choose a positive tolerance ε_r for each round. It is sufficient to ensure

$$\Delta(b_r, b_{r+1}) \leq \varepsilon_r, \quad \sum_{r=1}^R \varepsilon_r \leq \varepsilon.$$

The ε_r are *chosen tolerances*, not the unknown realised losses.

1.5.1 Controlling the loss in one round

Recall the majority calculation. Choose a failure allowance $0 < \delta_r < 1$ for round r , and give every match

$$m_r = \left\lceil \frac{1}{2\varepsilon_r^2} \log \frac{1}{\delta_r} \right\rceil$$

observations. If the better input beats the worse by more than ε_r , the match selects the worse one with probability at most δ_r . If the margin is at most ε_r , either output is close enough. Hence for better input u and returned winner W ,

$$\mathbb{P}(\Delta(u, W) > \varepsilon_r) \leq \delta_r.$$

Look at the match containing b_r . Write w_r for its winner. If b_r wins, it remains the best survivor: $b_{r+1} = b_r$, so the loss is zero. If it loses, its replacement w_r is still present, and the best survivor is at least as good as that replacement:

$$b_r \succeq b_{r+1} \succeq w_r.$$

SST therefore gives $\Delta(b_r, b_{r+1}) \leq \Delta(b_r, w_r)$. Combining this with the match guarantee yields

The one-round guarantee

$$\mathbb{P}(\Delta(b_r, b_{r+1}) > \varepsilon_r) \leq \delta_r.$$

Why did we not multiply the error allowance by the number of matches? We only needed to follow the match containing the current best item. Its winner remains available, whatever happens in the other matches. The algorithm does not know which match this is, so it gives every match the same repetition count. The proof needs the guarantee for just this one match.

The argument is conditional on the past and the chosen pairing, which fix b_r and its opponent. Fresh observations then justify the majority bound. Averaging over the past preserves the stated probability bound.

1.6 Allocating accuracy and confidence across rounds

Call round r bad if the best remaining item deteriorates by more than its allocated tolerance:

$$E_r^{\text{bad}} = \{\Delta(b_r, b_{r+1}) > \varepsilon_r\}.$$

The one-round argument gives $\mathbb{P}(E_r^{\text{bad}}) \leq \delta_r$. A union bound across rounds gives

$$\mathbb{P}\left(\bigcup_{r=1}^R E_r^{\text{bad}}\right) \leq \sum_{r=1}^R \delta_r.$$

No independence between rounds is required. If no round is bad, the total loss is at most $\sum_r \varepsilon_r$. Therefore the conditions

$$\sum_{r=1}^R \varepsilon_r \leq \varepsilon, \quad \sum_{r=1}^R \delta_r \leq \delta$$

suffice for the PAC guarantee. The first sum bounds how much quality we may lose across all the rounds. The second bounds the probability that any round loses more than its allowance. Keeping these two requirements separate will help us choose the repetition counts.

1.6.1 The total query cost

There are $K/2^r$ matches in round r . Ignoring integer ceilings for the moment, the algorithm design question is

Choose the round budgets

Choose ε_r, δ_r to make

$$\sum_{r=1}^R \underbrace{\frac{K}{2^r}}_{\text{number of matches}} \underbrace{\frac{1}{2\varepsilon_r^2} \log \frac{1}{\delta_r}}_{\text{observations per match}}$$

small, subject to $\sum_r \varepsilon_r \leq \varepsilon$ and $\sum_r \delta_r \leq \delta$.

1.6.2 Equal allocation works, but is wasteful

The simple choice

$$\varepsilon_r = \frac{\varepsilon}{R}, \quad \delta_r = \frac{\delta}{R}$$

satisfies both constraints. Each match then needs

$$m = \left\lceil \frac{R^2}{2\varepsilon^2} \log \frac{R}{\delta} \right\rceil$$

observations. Across $K - 1$ matches, this costs

$$O\left(\frac{KR^2}{\varepsilon^2} \log \frac{R}{\delta}\right), \quad R = \log_2 K.$$

The R^2 term gives a $(\log K)^2$ penalty, with an additional confidence logarithm. This allocation asks the numerous early matches to be just as accurate as the few late matches. The shrinking number of matches suggests a better allocation.

1.7 Geometric allocation and the final guarantee

Use larger tolerances in early rounds, and tighten them in later rounds when fewer matches remain. Choose geometrically decreasing budgets:

$$\varepsilon_r \propto \varepsilon q^{r-1}, \quad \delta_r \propto \delta \left(\frac{1}{2}\right)^{r-1}.$$

With the normalising constants included, the choice is

The Falahatgar schedule

$$q = 2^{-1/3}, \quad \varepsilon_r = (1 - q)\varepsilon q^{r-1}, \quad \delta_r = \frac{\delta}{2^r}.$$

In each round of our existing tournament, use

$$m_r = \left\lceil \frac{1}{2\varepsilon_r^2} \log \frac{1}{\delta_r} \right\rceil$$

observations per pair, then advance the empirical majority winners.

The two geometric sums obey

$$\sum_{r=1}^R \varepsilon_r = \varepsilon(1 - q^R) \leq \varepsilon, \quad \sum_{r=1}^R \delta_r = \delta(1 - 2^{-R}) \leq \delta.$$

So the correctness requirements are satisfied.

PAC maximum selection

Under the hidden-order, fresh comparison-noise, SST, and STI assumptions, the algorithm satisfies

$$\mathbb{P}\left(\widehat{p}_{ij} \geq \frac{1}{2} - \varepsilon \quad \text{for every } j \in [K]\right) \geq 1 - \delta,$$

using

$$O\left(\frac{K}{\varepsilon^2} \left(1 + \log \frac{1}{\delta}\right)\right)$$

elementary comparisons.

The known minimum gap has disappeared. We allow an approximately best item, preserve a good survivor in each round, and allocate effort according to how many matches remain. For fixed ε and δ , the comparison cost is linear in K .

Falahatgar, Orlitsky, Pichapati, and Suresh, *Maximum Selection and Ranking under Noisy Comparisons*, ICML 2017, Section 3. Here we use fixed-budget majority comparisons and the SST case of their result.

1.7.1 Counting the comparisons

The actual number of elementary comparisons, including integer rounding, is

$$Q = \sum_{r=1}^R \frac{K}{2^r} \left\lceil \frac{1}{2\varepsilon_r^2} \log \frac{1}{\delta_r} \right\rceil.$$

Using $\lceil x \rceil \leq x + 1$, the ceilings add at most $\sum_r K/2^r = K - 1$ observations. Substitute the geometric budgets. Since $q^3 = 1/2$,

$$\frac{K}{2^r} \frac{1}{2\varepsilon_r^2} = \frac{K}{4(1-q)^2 \varepsilon^2} q^{r-1}.$$

Writing $c_0 = 1/[4(1-q)^2]$, a constant independent of K, ε, δ , we obtain

$$\begin{aligned} Q &\leq (K - 1) + \frac{c_0 K}{\varepsilon^2} \sum_{r=1}^R q^{r-1} \left(\log \frac{1}{\delta} + r \log 2 \right) \\ &\leq (K - 1) + \frac{c_0 K}{\varepsilon^2} \left[\frac{1}{1-q} \log \frac{1}{\delta} + \frac{\log 2}{(1-q)^2} \right]. \end{aligned}$$

Here we used $\sum_{r \geq 1} q^{r-1} = 1/(1-q)$ and $\sum_{r \geq 1} r q^{r-1} = 1/(1-q)^2$. The match count halves each round, while the accuracy factor $1/\varepsilon_r^2$ grows by $2^{2/3} < 2$. The resulting geometric decrease absorbs the linear growth of the confidence logarithm and gives

$$Q = O\left(\frac{K}{\varepsilon^2} \left(1 + \log \frac{1}{\delta}\right)\right).$$

If confidence is allowed to depend on K , for example $\delta = 1/K$, its logarithm introduces a $\log K$ factor. The bound is linear in K when ε, δ are fixed.

1.8 Other pool sizes and ties

If $K \geq 2$ is not a power of two, pad to $N = 2^{\lceil \log_2 K \rceil} < 2K$ using dummy items below every real item. A real item beats a dummy deterministically; dummy–dummy comparisons have probability $1/2$. These probabilities preserve SST and STI. Simulate dummy comparisons without asking a human. The bound for N changes only the constant, and the winner is real. This is a bracket with byes. For $K = 1$, return the sole item without a query.

Use a fixed ordering to select a representative best item when true ties occur. The proof only uses weak inequalities. Empirical majority ties can also be broken arbitrarily; the Hoeffding argument already includes the event $\hat{p} \leq 1/2$.

The algorithm needs only K, ε, δ and observed outcomes. The hidden order, true gaps, and best-survivor identities are used in the analysis.

In the eight-item example, the second round loses .010, more than the schedule's tolerance $\varepsilon_2 \approx .0082$. Its total loss is still at most $.015 < .05$. Controlling every round's loss is a sufficient condition for success; a particular run can succeed even if that condition fails.

1.9 Chapter summary

We began with a knockout tournament that finds the best item in $K - 1$ noiseless comparisons. With noise, repeated comparisons make each match more reliable, but exact selection requires enough repetitions to resolve the relevant true gaps. Allowing an approximately best item lets us choose the required accuracy ourselves.

The noisy tournament need only retain some good item after each round. SST lets us follow the best remaining item, and STI lets us add the losses as that item changes. Giving greater accuracy to the later, less numerous matches keeps the total query cost linear in K for fixed accuracy and confidence.

Principal equations

$$\Delta(i, j) = p_{ij} - \frac{1}{2}, \quad \mathbb{P}(\Delta(i^*, \hat{i}) \leq \varepsilon) \geq 1 - \delta.$$

$$\varepsilon_r = (1 - 2^{-1/3})\varepsilon 2^{-(r-1)/3}, \quad \delta_r = \delta/2^r, \quad Q = O\left(\frac{K}{\varepsilon^2} \left(1 + \log \frac{1}{\delta}\right)\right).$$

Every query has counted equally so far. If the queried items are themselves experienced by a user, a poor comparison can cost more than a good one. The next chapter keeps adaptive pair selection but evaluates the whole sequence of actions.

Duelling bandits

2.1 The duelling bandit problem

In the preceding chapter, we wanted to find a good item using as few comparisons as possible. We chose the pairs adaptively and used Hoeffding's inequality to decide how many observations were needed. We will carry both ideas into the duelling bandit problem.

The setting remains broadly the same: we have K items and learn about them through noisy comparisons. Here, however, we also care about the quality of the intermediate choices. In active selection, every query counted equally. We will now develop a model in which comparing two poor items costs more than comparing two good ones.

Why is this a good model? Consider an assistant that shows two candidate explanations and asks a student which is more helpful. Both explanations have already been read by the time the student supplies a preference. A comparison involving a confusing explanation can therefore have a cost even if its label helps identify the best explanation later. We want the system to gather useful evidence while increasingly showing good items. It must learn from the same choices whose quality we are evaluating.

In the preceding paragraphs, we talked about the best item and about poor items. How do we define these notions precisely? One option is to assume that comparisons come from the BTL model we studied in Module 1. Each item has a latent parameter θ_i , and

$$\mathbb{P}(i \text{ beats } j) = \frac{e^{\theta_i}}{e^{\theta_i} + e^{\theta_j}} = \sigma(\theta_i - \theta_j), \quad \sigma(v) = \frac{1}{1 + e^{-v}}.$$

A larger parameter means that the item is more likely to win. If one item has a strictly larger parameter than all the others, it is the best item; we could measure how poor another item is by the difference between their parameters.

However, for the purpose of understanding duelling bandits, we do not need to assume the BTL model. We can make these notions precise with much weaker assumptions. Let us first describe comparisons without assuming that they arise from latent parameters. We call each item an *arm*, as is customary in bandit problems, and write

$$p_{ij} = \mathbb{P}(i \text{ beats } j), \quad p_{ji} = 1 - p_{ij}, \quad p_{ii} = \frac{1}{2}. \quad (2.1)$$

The matrix $P = (p_{ij})$ is fixed and unknown. On round t , we choose the pair (i_t, j_t) and observe which arm wins. Each comparison uses fresh, independent noise with the chosen pair's win probability. For a given pair, recording a win as 1 and a loss as 0 gives independent Bernoulli samples with the same mean. We can still choose when to compare that pair using the observations collected so far.

We have not assumed any ordering of the arms or any further relation between different pairs' probabilities. In particular, we could have

$$p_{AB} = p_{BC} = p_{CA} = 0.7.$$

Then A tends to beat B , B tends to beat C , and C tends to beat A . This is a *Condorcet cycle*; among these three arms, none beats both of the others more often than it loses.

For the main guarantee in this chapter, we assume that there is a *strict Condorcet winner*: an arm i^* satisfying

$$p_{i^*j} > \frac{1}{2} \quad \text{for every } j \neq i^*. \quad (2.2)$$

In words, this arm beats each rival more often than it loses. There can be only one such arm: two of them would each have to beat the other with probability greater than $1/2$, which is impossible. This gives us the best arm we need, without requiring a BTL model or even an ordering of all the other arms. We will return to these modelling choices later.

2.1.1 Regret and the learning objective

We measure an arm's shortfall by the winner's advantage over it. Define

$$\Delta_i = p_{i^*i} - \frac{1}{2}. \quad (2.3)$$

For example, if the winner beats arm i with probability 0.70, then $\Delta_i = 0.20$. The winner has gap zero, and every other arm has a positive gap. Since both items in a pair are shown to the user, we define *cumulative regret* over T rounds as

$$R_T = \sum_{t=1}^T \frac{\Delta_{i_t} + \Delta_{j_t}}{2}. \quad (2.4)$$

Each summand is the *instantaneous regret*, $r(i_t, j_t) = (\Delta_{i_t} + \Delta_{j_t})/2$. It lies between 0 and $1/2$, and is zero if and only if both chosen arms are the winner. We allow this in duelling bandits: choosing (i, i) means using arm i alone, with no informative comparison required. In particular, once we have found the winner, we can use it alone and incur no further regret.

The learning problem

We are given $K \geq 2$ arms and a fixed horizon $T \geq 2$, known in advance. The unknown preference matrix satisfies Equation (2.1) and has a strict Condorcet winner. On rounds $t = 1, \dots, T$, choose (i_t, j_t) using previous outcomes, and observe one fresh comparison outcome if the arms differ. The objective is to minimise

$$\mathbb{E}[R_T] = \mathbb{E}\left[\sum_{t=1}^T \frac{\Delta_{i_t} + \Delta_{j_t}}{2}\right].$$

We seek a guarantee for which $\mathbb{E}[R_T/T] \rightarrow 0$ as T grows.

The learner observes which arm wins each comparison, not the regret. Calculating regret would require the true winner and the gaps; we use these unknown quantities to evaluate its choices. The expectation averages over comparison noise and any random choices made by the algorithm.

Vanishing average regret does not require the total regret to vanish, or every later round to be optimal. Occasional comparisons may still be needed, but their total cost should grow more slowly than T . Here the arms and their comparison probabilities stay fixed as T grows.

Choosing both arms independently and uniformly on every round does not achieve this. In that case,

$$\mathbb{E}[R_T] = \frac{T}{K} \sum_{i=1}^K \Delta_i.$$

The coefficient of T is positive, so the average regret stays constant however long we run. The algorithm must use what it has learned to change which comparisons it makes.

As before, we can estimate a comparison probability by the fraction of times an arm wins. But an estimate based on five comparisons should not be treated in the same way as one based on five hundred. We will keep a *confidence interval* around each estimate, and use these intervals to decide which arms could still be the winner and which comparison to make next. Let us first see how Hoeffding's inequality gives us these intervals.

2.2 Hoeffding's inequality and confidence intervals

For one pair, write p for the probability that the first arm wins and $\hat{p}_n$ for its fraction of wins in n comparisons. For any error margin $\varepsilon > 0$, Hoeffding's inequality gives

$$\mathbb{P}(|\hat{p}_n - p| > \varepsilon) \leq 2e^{-2n\varepsilon^2}. \quad (2.5)$$

To allow a failure probability $\delta \in (0, 1)$, set the right-hand side equal to δ and solve for the error margin:

$$\varepsilon_n = \sqrt{\frac{\log(2/\delta)}{2n}}.$$

Thus, with probability at least $1 - \delta$, p lies in the confidence interval $[\hat{p}_n - \varepsilon_n, \hat{p}_n + \varepsilon_n]$. Its width is

$$2\varepsilon_n = \sqrt{\frac{2\log(2/\delta)}{n}}.$$

At a fixed confidence level, the width decreases as $1/\sqrt{n}$. Four times as many comparisons halve the width; to reduce it to a quarter, we need sixteen times as many comparisons. Equivalently, an interval of width at most 2ε requires

$$n \geq \frac{\log(2/\delta)}{2\varepsilon^2}.$$

This is why a close comparison needs more samples: when p is near $1/2$, we need a narrow interval to establish which side of $1/2$ it lies on.

The width is determined by the sample count and chosen confidence level. The centre, $\hat{p}_n$, depends on the observed outcomes and moves as new comparisons arrive. It need not move steadily in one direction, even though the interval gets narrower.

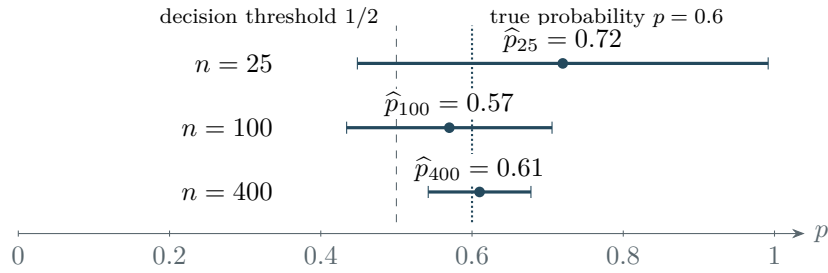


Figure 2.1: Narrower intervals with changing empirical estimates. The centres correspond to 18 wins in 25 comparisons, 57 in 100, and 244 in 400, a possible sequence of observations when the true win probability is $p = 0.6$. All three intervals cover this ground truth. The separate line at $1/2$ is the decision threshold. Each interval uses $\delta = 0.05$; their widths are approximately 0.543, 0.272, and 0.136. Only the last interval lies wholly above $1/2$.

2.3 The RUCB algorithm and its guarantee

The Relative Upper Confidence Bound (RUCB) algorithm chooses a plausible winner and compares it with a challenger. We present a simplified version of the algorithm developed by Zoghi et al. [Zoghi et al. \(2014\)](#), and prove its regret guarantee in Section 2.5. The published algorithm has additional rules for selecting the first candidate; the guarantee here is for the version we state below.

2.3.1 Notation and confidence bounds

For each round t , we use the following quantities:

- $N_{ij}(t)$ is the number of comparisons of the unordered pair $\{i, j\}$ before round t . Thus $N_{ji}(t) = N_{ij}(t)$.
- $\hat{p}_{ij}(t)$ is the fraction of those comparisons won by i , defined when $N_{ij}(t) > 0$. The reverse fraction is $\hat{p}_{ji}(t) = 1 - \hat{p}_{ij}(t)$.
- $\delta \in (0, 1)$ is our total failure allowance. Write $M = \binom{K}{2}$ for the number of distinct unordered pairs, and $\Lambda = \log(2MT/\delta)$.

- $b_{ij}(t)$ is the error margin added to and subtracted from $\hat{p}_{ij}(t)$ to obtain a confidence interval:

$$b_{ij}(t) = \sqrt{\frac{\Lambda}{2N_{ij}(t)}} \quad \text{when } N_{ij}(t) > 0. \quad (2.6)$$

The interval width is $2b_{ij}(t)$.

- $U_{ij}(t)$ is the upper confidence bound for the probability that i beats j :

$$U_{ij}(t) = \begin{cases} 1/2, & i = j, \\ 1, & i \neq j, N_{ij}(t) = 0, \\ \min\{1, \hat{p}_{ij}(t) + b_{ij}(t)\}, & i \neq j, N_{ij}(t) > 0. \end{cases} \quad (2.7)$$

An unobserved pair has upper bound 1. A self-pair has known value $p_{ii} = 1/2$ and needs no estimation.

The choice of b_{ij} comes from Hoeffding's inequality. For a fixed pair $\{i, j\}$ and fixed sample count $n \geq 1$, let $\hat{p}_{ij,n}$ be the fraction of its first n outcomes won by i . Then

$$\mathbb{P}\left(p_{ij} \in \left[\hat{p}_{ij,n} - \sqrt{\frac{\Lambda}{2n}}, \hat{p}_{ij,n} + \sqrt{\frac{\Lambda}{2n}}\right]\right) \geq 1 - \frac{\delta}{MT}.$$

The algorithm uses this error margin at the observed count $n = N_{ij}(t)$. We choose $\delta/(MT)$ to allow a union bound later. Section 2.5 explains that allocation and shows why the intervals remain valid together when the algorithm chooses the comparison counts adaptively.

Simplified RUCB

Input: $K \geq 2$ arms, horizon $T \geq 2$, failure allowance $\delta \in (0, 1)$.

- 1: Set all comparison and win counts to 0;
 set $M = \binom{K}{2}$ and $\Lambda = \log(2MT/\delta)$.
- 2: **for** $t = 1, \dots, T$ **do**
- 3: Compute $U(t)$ using Equation (2.7).
- 4: Construct $\mathcal{C}_t = \{i : U_{ij}(t) \geq 1/2 \text{ for every } j\}$.
- 5: Pick $i_t \in \mathcal{C}_t$ arbitrarily; if $\mathcal{C}_t = \emptyset$, pick any arm.
- 6: Pick $j_t \in \arg \max_{j \in \{1, \dots, K\}} U_{ji_t}(t)$, breaking ties arbitrarily.
- 7: Compare (i_t, j_t) and update the pair's counts and win fractions.
 If $i_t = j_t$, use that arm alone and leave all counts unchanged.
- 8: **end for**

There are two important steps: constructing $\mathcal{C}_t$, the set of plausible winners, and choosing a challenger to a plausible winner. To construct $\mathcal{C}_t$, read the *rows* of U . If $U_{ij} < 1/2$ for even one opponent j , arm i cannot be the winner when the bounds are valid. The remaining arms are plausible winners, and we may choose any one of them as i_t . This arm is sometimes called the champion.

To choose its challenger, read *column* i_t and maximise U_{ji_t} . A large upper bound may reflect a high observed win fraction, or too few comparisons to rule out that opponent. The challenger need not belong to $\mathcal{C}_t$: an arm that cannot itself be the winner may still help test another candidate. The self-pair, with upper bound $1/2$, is also an option.

The fallback for an empty $\mathcal{C}_t$ makes the algorithm well-defined even when estimates are misleading. It is never needed when all the confidence bounds hold, as we will show in the proof.

2.3.2 The performance guarantee

The regret bound depends on how hard it is to distinguish the winner from its closest rival. Write $\Delta_{\min} = \min_{i \neq i^*} \Delta_i > 0$ for the winner's smallest advantage. We use this quantity to state the guarantee; the algorithm does not need to know it.

Regret of simplified RUCB

With probability at least $1 - \delta$,

$$R_T \leq \frac{M}{2} \left(1 + \frac{2 \log(2MT/\delta)}{\Delta_{\min}^2} \right). \quad (2.8)$$

Consequently,

$$\mathbb{E}[R_T] \leq \frac{M}{2} \left(1 + \frac{2 \log(2MT/\delta)}{\Delta_{\min}^2} \right) + \frac{\delta T}{2}. \quad (2.9)$$

In particular, running the algorithm with $\delta = 1/T$ gives $\mathbb{E}[R_T] = O(\log T)$: it grows at most logarithmically with T , when the arms and their comparison probabilities stay the same.

The main conclusion is that regret grows only logarithmically with T when we keep the comparison problem fixed. Dividing by T , the average regret therefore tends to zero, as we wanted. The other terms tell us what makes learning harder: smaller gaps require more accurate estimates, and more arms give us more pairs to consider. This is a loose bound and can exceed the trivial bound $T/2$ for a short run. It is enough for our present purpose; the published RUCB algorithm improves the leading dependence on the number of arms through an additional selection mechanism. Before proving it, let us follow the algorithm through an example.

2.4 A numerical example: from testing to using the winner

We will follow one possible run with three arms A, B, C , horizon $T = 1000$, and $\delta = 0.1$. Thus $M = 3$ and $\Lambda = \log(60000) \approx 11.0021$. The calculations use the exact logarithm; displayed values are rounded.

Choice and tie-breaking rules for this example. Use the order A, B, C : choose the first arm in $\mathcal{C}_t$, and choose the first arm in this order among challengers tied for the largest upper bound. Other orders or ways of resolving these choices can also work, but may produce a different sequence of comparisons. The proof only needs the first arm to be in $\mathcal{C}_t$ when it is nonempty, and the challenger to maximise its column.

Suppose the underlying preferences are

$$P = \begin{array}{c|ccc} & A & B & C \\ \hline A & 0.50 & 0.30 & 0.20 \\ B & 0.70 & 0.50 & 0.65 \\ C & 0.80 & 0.35 & 0.50 \end{array}.$$

The row arm beats the column arm with the displayed probability. Thus B is the winner and $(\Delta_A, \Delta_B, \Delta_C) = (0.20, 0, 0.15)$. The learner does not see this matrix. We use it to check the costs:

$$r(A, B) = 0.10, \quad r(A, C) = 0.175, \quad r(B, C) = 0.075.$$

The learner starts with no observations. We will give the outcomes as we proceed and use the resulting bounds to explain each change of pair. All counts below refer to observations made before the stated round.

2.4.1 Rounds 1–64: testing the first candidate

At round 1, the upper-bound matrix is

$$U(1) = \begin{array}{c|ccc} & A & B & C \\ \hline A & 0.5 & 1 & 1 \\ B & 1 & 0.5 & 1 \\ C & 1 & 1 & 0.5 \end{array}.$$

Every arm is therefore a plausible winner. The first arm is A ; in its column, B and C tie for the largest bound. We therefore compare (A, B) . In our sequence, B wins. Arm A has lost its only comparison, so its empirical win fraction is zero. However, the error margin is $b_{AB} = \sqrt{\Lambda/2} > 1$, giving $U_{AB} = U_{BA} = 1$ after capping the endpoints. We have too little evidence to discard A . At round 2, A is still the first candidate, and B again wins the challenger tie with C .

In the outcome sequence we are following, this continues through the first 63 observations: A stays in the candidate set and U_{BA} remains capped at 1. The unobserved pair CA also has $U_{CA} = 1$, so our tie rule keeps selecting B to challenge A . This explains why the first 64 rounds all compare AB .

2.4.2 Round 65: an unobserved challenger takes over

After the 64th comparison, A has won 19 times, so

$$\hat{p}_{AB} = \frac{19}{64} = 0.296875, \quad b_{AB} = \sqrt{\frac{\Lambda}{128}} \approx 0.2932.$$

Thus $U_{AB} \approx 0.5901$ still allows A to be a candidate, but $U_{BA} \approx 0.9963$ has now fallen below 1. Column A has entries $(0.50, 0.9963, 1)$: C is the unique maximiser, so round 65 compares (A, C) . The tie that previously favoured B has been broken by evidence. We have not seen C beat A ; in fact, we have not compared them at all. The algorithm selects C because this possibility is still unresolved. This is how the upper bound leads it to explore.

2.4.3 Round 123: evidence changes the candidate

The algorithm next keeps testing A against C . After 58 such comparisons, A has won 11. At the start of round 123, its upper bound is

$$U_{AC} = \frac{11}{58} + \sqrt{\frac{\Lambda}{116}} \approx 0.4976 < \frac{1}{2}.$$

Even the upper bound for A beating C is now below $1/2$, so A fails its row test. The new candidate set is $\{B, C\}$ and the first arm is B . Column B has entries $(0.5901, 0.50, 1)$, so we compare (B, C) . Notice that C , which is not the true winner, supplied the evidence that ruled out A . We did not need to know the winner first.

1. Find plausible winners				2. Choose the challenger			
$U(123)$	A	B	C	$U(123)$	A	B	C
A	0.5000	0.5901	0.4976	A	0.5000	0.5901	0.4976
B	0.9963	0.5000	1	B	0.9963	0.5000	1
C	1	1	0.5000	C	1	1	0.5000
$\mathcal{C}_t = \{B, C\}$: choose $i_t = B$				$U_{CB} = 1$ is largest: choose $j_t = C$			

Figure 2.2: The upper confidence bound matrix $U(123)$, shown twice to illustrate the two selection steps. Each entry bounds the probability that the row arm beats the column arm; this is not the true preference matrix P . On the left, $U_{AC} < 1/2$ rules out A , leaving B, C as plausible winners. Our example chooses B . On the right, the largest entry in column B selects C as challenger. Values use $\Lambda = \log(60000)$ and are rounded to four decimal places.

2.4.4 Round 220: an excluded arm is useful again

At round 220, B has won 63 of its 97 comparisons with C . Thus

$$U_{CB} = \frac{34}{97} + \sqrt{\frac{\Lambda}{194}} \approx 0.5887.$$

The bound $U_{AB} \approx 0.5901$ is slightly larger. The first arm remains B , but the challenger is now A , even though $A \notin \mathcal{C}_t$. We ruled out A using its losses to C . We still need evidence that B beats A , and the interval for that pair is not yet narrow enough. Over the next rounds, the algorithm alternates between testing B against A and against C , according to which upper bound is larger.

2.4.5 Round 439: using the winner alone

By the start of round 439, the pair AB has been compared 132 times, with 39 wins for A , and BC has been compared 248 times, with 161 wins for B . Substituting these counts into the upper bounds against B gives

$$U_{AB} \approx 0.49960, \quad U_{BB} = 0.50000, \quad U_{CB} \approx 0.49974.$$

Both possible challengers now have upper bounds below $1/2$, so the algorithm chooses (B, B) . This supplies no new data. Since our confidence margin uses a fixed horizon, the bounds stay unchanged and the algorithm uses B alone for the rest of this run.

2.4.6 Recap of the run

The table collects the stages we have just worked through. Each N is a count before round t , and W counts wins by the first named arm. The comparison at that round is listed in the final column.

t	A, B		A, C		B, C		$\mathcal{C}_t$	(i_t, j_t)
	N	W	N	W	N	W		
1	0	0	0	0	0	0	$\{A, B, C\}$	(A, B)
65	64	19	0	0	0	0	$\{A, B, C\}$	(A, C)
123	64	19	58	11	0	0	$\{B, C\}$	(B, C)
220	64	19	58	11	97	63	$\{B, C\}$	(B, A)
439	132	39	58	11	248	161	$\{B\}$	(B, B)

To specify this example completely, take an outcome sequence in which the first arm's win counts after n observations are $\lfloor 0.30n \rfloor$, $\lfloor 0.20n \rfloor$, and $\lfloor 0.65n \rfloor$ for AB, AC, BC , respectively. This is one possible sequence of noisy outcomes, chosen so that we can follow the calculations. The true probabilities do not force these counts; a different run can look different.

In total, this run makes 132 comparisons of AB , 58 of AC , and 248 of BC . Its cumulative regret is

$$132(0.10) + 58(0.175) + 248(0.075) = 41.95.$$

For comparison, uniform random choices on this instance have expected regret $1000(0.20 + 0 + 0.15)/3 \approx 116.67$. The example shows how the choices can improve, but one run cannot establish the guarantee. We now need to show that a similar restriction on costly choices holds whenever the confidence bounds are valid.

2.5 Proof of the regret guarantee

Each comparison costs at most $1/2$, so it is enough to bound the number of rounds with positive regret. We will do this in three steps. First, show that all our confidence bounds hold together with high probability. Second, show that when these bounds hold, the algorithm never uses a wrong arm alone. Third, show that a distinct pair can be chosen only while its confidence interval is sufficiently wide. Counting how often this can happen will give the regret bound.

2.5.1 Step 1: make all confidence bounds valid together

Hoeffding gives a guarantee for a fixed pair after a fixed number of samples. The count $N_{ij}(t)$, however, depends on earlier outcomes. For example, the algorithm may stop comparing a pair just when its empirical win fraction becomes small enough. A guarantee for a fixed sample size does not automatically apply to this selected sample size. Our way around this is to make the bound hold for every possible sample size at once.

Imagine that, before the run starts, an independent sequence of outcomes has been generated for each unordered pair. Whenever we query a pair, we reveal its next outcome. This gives exactly the same comparison model; it just lets us talk about outcomes that have not yet been observed. Let $\hat{p}_{ij,n}$ be the fraction of wins for i in the first n outcomes for $\{i, j\}$, whether or not the algorithm reveals that many. Write

$$\varepsilon_n = \sqrt{\frac{\Lambda}{2n}}, \quad \Lambda = \log \frac{2MT}{\delta}.$$

We want every pair's estimation error to be within this margin at every count from 1 to T . Call that event $\mathcal{E}$:

$$\mathcal{E} = \bigcap_{1 \leq i < j \leq K} \bigcap_{n=1}^T \{|\hat{p}_{ij,n} - p_{ij}| \leq \varepsilon_n\}. \quad (2.10)$$

An event is a set of possible runs. Here $\mathcal{E}$ is the set of runs in which every listed estimation error is small enough. Its complement $\mathcal{E}^c$ consists of runs where at least one bound fails.

Lemma 2.5.1 (Simultaneous confidence). *The event $\mathcal{E}$ has probability at least $1 - \delta$. Whenever it holds, every observed pair at every round $t \leq T$ satisfies $|\hat{p}_{ij}(t) - p_{ij}| \leq b_{ij}(t)$.*

Proof. For a fixed pair and fixed $n \in \{1, \dots, T\}$, Hoeffding gives

$$\mathbb{P}(|\hat{p}_{ij,n} - p_{ij}| > \varepsilon_n) \leq 2e^{-2n\varepsilon_n^2} = 2e^{-\Lambda} = \frac{\delta}{MT}. \quad (2.11)$$

A failure of $\mathcal{E}$ means that at least one of these MT bounds fails. Taking a union bound,

$$\begin{aligned} \mathbb{P}(\mathcal{E}^c) &\leq \sum_{1 \leq i < j \leq K} \sum_{n=1}^T \mathbb{P}(|\hat{p}_{ij,n} - p_{ij}| > \varepsilon_n) \\ &\leq \sum_{1 \leq i < j \leq K} \sum_{n=1}^T \frac{\delta}{MT} = MT \frac{\delta}{MT} = \delta. \end{aligned}$$

No independence between these failure events is needed. We count only unordered pairs: reversing a pair complements both the estimate and its mean, leaving the absolute error unchanged.

At any round, an observed pair has some count $n = N_{ij}(t)$ between 1 and T . The event covers that count, and $\varepsilon_n = b_{ij}(t)$. Thus the bounds hold at the actual counts, however the algorithm chose them. $\square$

This union bound is deliberately loose. There are only T rounds in total, so the algorithm does not query all M pairs T times each. We have protected every pair at every possible count, including counts it will never reach. This lets us use a simple argument without analysing when the algorithm stops sampling each pair. We are bounding MT possible estimation errors, not claiming that MT comparisons are made. There is no need to sum over rounds again: every estimate used on a round is already covered by $\mathcal{E}$.

	Number of samples of this pair, n							
	1	2	3	4	5	6	7	
AB								Used in this run
AC								
BC								Covered but unused

Figure 2.3: The union bound covers possible estimates, not just the ones we use. With three pairs and $T = 7$, it protects all $3 \times 7 = 21$ cells. A run making four comparisons of AB , two of AC , and one of BC gives only the seven shaded estimates. Cell $(AB, 4)$ refers to the mean of the first four AB outcomes, not to round 4.

2.5.2 Step 2: rule out using a wrong arm alone

For the next two lemmas, suppose we are following a run in which $\mathcal{E}$ holds. In other words, all the error bounds from Lemma 2.5.1 are valid throughout the run. We can use them to reason about each choice the algorithm makes. At the end we will account for the runs where a bound fails.

For every observed pair and round in this run, $|\hat{p}_{ij}(t) - p_{ij}| \leq b_{ij}(t)$. Therefore

$$p_{ij} \leq U_{ij}(t) \leq p_{ij} + 2b_{ij}(t), \quad (2.12)$$

$$U_{ij}(t) + U_{ji}(t) \leq 1 + 2b_{ij}(t). \quad (2.13)$$

For the first line, add b_{ij} to the estimation inequalities and note that clipping at 1 preserves the bounds because $p_{ij} \leq 1$. For the second, add the two unclipped upper endpoints and use $\hat{p}_{ij} + \hat{p}_{ji} = 1$; clipping can only reduce their sum. An unobserved pair still satisfies $p_{ij} \leq U_{ij} = 1$.

Lemma 2.5.2 (The winner stays plausible; wrong self-pairs are excluded). *If $\mathcal{E}$ holds, then $i^* \in \mathcal{C}_t$ at every round. Moreover, whenever the algorithm chooses $i_t = j_t$, both arms equal i^* , so the round has zero regret.*

Proof. For every rival j , we have $U_{i^*j} \geq p_{i^*j} > 1/2$, and the diagonal entry is $1/2$. The winner therefore passes every test in its row and belongs to $\mathcal{C}_t$. In particular, this set is never empty, so the algorithm always chooses its first arm from it.

Now suppose $i_t = i \neq i^*$. The true winner is one possible challenger, and

$$U_{i^*i}(t) \geq p_{i^*i} = \frac{1}{2} + \Delta_i > \frac{1}{2} = U_{ii}(t).$$

Since the challenger maximises this column, it cannot be i itself. Thus a wrong arm cannot be chosen alone. Any self-pair must be (i^*, i^*) , which has zero regret. $\square$

This removes a possible obstacle: the algorithm cannot keep paying regret by using a wrong arm alone without collecting any new evidence. It remains to count comparisons between distinct arms.

2.5.3 Step 3: a costly comparison needs a wide interval

Lemma 2.5.3 (Width of a selected pair's confidence interval). *Suppose $\mathcal{E}$ holds. If the algorithm selects a distinct pair (i_t, j_t) that has already been observed, then*

$$2b_{i_t j_t}(t) \geq \Delta_{\min}. \quad (2.14)$$

The left side is the interval width. It decreases each time this pair is sampled. The lemma therefore says that, once its interval is narrower than $\Delta_{\min}$, the pair cannot be selected again on this run.

Proof. There are two cases, depending on whether the first arm is the winner.

Case 1: the first arm is not the winner. Write $i = i_t \neq i^*$ and $j = j_t$. We do not know which challenger the algorithm chose, but we do know that its upper bound is at least that of every other challenger, including the true winner. Hence

$$U_{ji}(t) \geq U_{i^*i}(t) \geq \frac{1}{2} + \Delta_i.$$

On the other hand, i was selected from $\mathcal{C}_t$, so its row passed the candidate test against this same arm j :

$$U_{ij}(t) \geq \frac{1}{2}.$$

Add these two inequalities. If the pair has already been observed, Equation (2.13) gives

$$1 + \Delta_i \leq U_{ji}(t) + U_{ij}(t) \leq 1 + 2b_{ij}(t).$$

Hence $2b_{ij}(t) \geq \Delta_i \geq \Delta_{\min}$, as required.

This is the restriction we needed on the pair actually compared. Notice why both selection steps mattered: choosing i as a candidate gave us $U_{ij} \geq 1/2$, and choosing the challenger gave us the stronger bound in the opposite direction. Their sum forces the confidence interval for this pair to be sufficiently wide. We never needed to assume that j itself was the winner.

Case 2: the first arm is the winner. Now let $i_t = i^*$ and $j = j_t \neq i^*$. The self-pair was an option in the maximisation, so

$$U_{ji^*}(t) \geq U_{i^*i^*}(t) = \frac{1}{2}.$$

For an observed pair, Equation (2.12) gives the other side:

$$U_{ji^*}(t) \leq p_{ji^*} + 2b_{ji^*}(t) = \frac{1}{2} - \Delta_j + 2b_{ji^*}(t).$$

Combining them yields $2b_{ji^*}(t) \geq \Delta_j \geq \Delta_{\min}$. So this case gives the same restriction. Although j really loses to the winner more often than it wins, its upper bound must still be at least $1/2$ for the algorithm to select it. Once that bound drops below $1/2$, the self-pair has a larger index and j cannot be chosen against the winner.

□

2.5.4 Complete the proof by counting comparisons

Let $Q_{ij}(T)$ count the rounds up to T on which the unordered distinct pair $\{i, j\}$ is selected. Lemma 2.5.2 says that all other rounds have zero regret. Since every round costs at most $1/2$,

$$R_T \leq \frac{1}{2} \sum_{i < j} Q_{ij}(T) \quad \text{when } \mathcal{E} \text{ holds.} \quad (2.15)$$

It is therefore sufficient to bound each pair's count. At the m th selection of $\{i, j\}$, its previous count is exactly $m - 1$, regardless of when those observations were collected. Its first selection is allowed without any prior data. If $m \geq 2$, Lemma 2.5.3 gives

$$2\sqrt{\frac{\Lambda}{2(m-1)}} \geq \Delta_{\min} \implies m - 1 \leq \frac{2\Lambda}{\Delta_{\min}^2}.$$

Thus

$$Q_{ij}(T) \leq 1 + \frac{2\Lambda}{\Delta_{\min}^2}. \quad (2.16)$$

Substitute this into Equation (2.15) and sum over all M pairs to obtain

$$R_T \leq \frac{M}{2} \left(1 + \frac{2\Lambda}{\Delta_{\min}^2} \right).$$

This bound holds whenever $\mathcal{E}$ does. Lemma 2.5.1 gives $\mathbb{P}(\mathcal{E}) \geq 1 - \delta$, proving the high-probability guarantee in Equation (2.8).

2.5.5 Include confidence failures in expected regret

We have so far bounded regret only for runs in $\mathcal{E}$. To bound the expectation, we must also include the runs where some confidence interval fails. These have probability at most δ , and even on such a run we have $R_T \leq T/2$. The symbol $\mathbf{1}_{\mathcal{E}}$ is 1 when $\mathcal{E}$ holds and 0 otherwise. We can therefore separate runs with valid bounds from runs with a failure:

$$\begin{aligned} \mathbb{E}[R_T] &= \mathbb{E}[R_T \mathbf{1}_{\mathcal{E}}] + \mathbb{E}[R_T \mathbf{1}_{\mathcal{E}^c}] \\ &\leq \frac{M}{2} \left(1 + \frac{2\Lambda}{\Delta_{\min}^2} \right) + \frac{T}{2} \mathbb{P}(\mathcal{E}^c) \\ &\leq \frac{M}{2} \left(1 + \frac{2 \log(2MT/\delta)}{\Delta_{\min}^2} \right) + \frac{\delta T}{2}. \end{aligned}$$

There is still a term $\delta T/2$ in this bound. If we kept δ fixed as T grew, this term would grow linearly. We can avoid that: since T is known in advance, choose $\delta = 1/T$ *when running the algorithm*. The failure term becomes $1/2$, while the cost of the wider intervals enters only through the logarithm. We get

$$\mathbb{E}[R_T] \leq \frac{M}{2} \left(1 + \frac{2 \log(2MT^2)}{\Delta_{\min}^2} \right) + \frac{1}{2}.$$

For the same arms and comparison probabilities, this grows at most logarithmically with T . Dividing by T shows that the expected average regret tends to zero. We have paid for confidence failures as well as for the comparisons made when the bounds are valid. The proof required neither a ranking of the nonwinning arms nor gaps between them: they may even be tied or form cycles.

2.6 Models, winner definitions, and the regret scale

We have proved the regret guarantee using only the strict Condorcet assumption. Let us now compare that assumption with the models we have seen earlier, and consider how the objective changes when there is no Condorcet winner.

The tournament in Chapter 1 assumed a hidden order, SST, and STI. The RUCB proof needs less structure: nonwinning items may tie or cycle. For example, A can beat B, C, D with probability .60, while $p_{BC} = p_{CD} = p_{DB} = .70$. The RUCB proof still applies, because it compares choices with A , without ordering the other arms.

Under BTL, the item with strictly largest utility is a strict Condorcet winner. The probability deficit is then $\Delta_i = \sigma(\theta_{i^*} - \theta_i) - 1/2$. A different objective would charge the utility deficit $\theta_{i^*} - \theta_i$. Both prefer the same item, but the scales differ. In Chapter 3 we use utility regret explicitly, because shared linear scores provide those units.

2.6.1 Other notions of winner

A strict Condorcet winner need not exist. We could instead assign each item a score and choose an item with the largest score. Different scores capture different meanings of “best”. The *Borda score*, for example, is the chance of beating a uniformly selected different item:

$$B_i = \frac{1}{K-1} \sum_{j \neq i} p_{ij}.$$

A maximiser always exists, even when the probabilities form cycles.

A Borda winner can also differ from an existing Condorcet winner. Consider three items with $p_{AB} = p_{AC} = .51$ and $p_{BC} = .90$. Here A is the Condorcet winner, but $B_A = .51$, $B_B = .695$, and $B_C = .295$. Winning narrowly against every rival differs from having the largest average win probability.

The *Copeland score* instead counts the opponents beaten with probability greater than $1/2$. A strict Condorcet winner beats all rivals and uniquely maximises this count; in a cycle several arms may tie. Learning either kind of winner requires its own analysis.

Our objective charges both arms, often called *strong regret*. The alternative *weak regret* charges the smaller deficit, $\min\{\Delta_i, \Delta_j\}$, and is zero whenever a pair contains the winner. We continue to charge both arms in the next chapter.

2.7 Ordinary UCB and the exploration bonus

In an ordinary stochastic bandit, choosing arm i gives a numerical reward in $[0, 1]$ from a fixed distribution with unknown mean μ_i . Unlike a comparison label, its interpretation does not depend on an opponent. The expected shortfall of selecting i is $\mu_{\star} - \mu_i$, with $\mu_{\star} = \max_j \mu_j$.

As with noisy comparisons, the largest empirical mean need not belong to the best arm. An arm sampled only a few times may look poor because of unlucky outcomes. UCB adds a confidence margin to each empirical mean. An arm with little data receives a larger margin, giving the algorithm a reason to sample it again. UCB1 first samples every arm once, then uses (Auer et al., 2002)

$$N_i(t) = \#\{s < t : a_s = i\}, \quad b_i(t) = \sqrt{\frac{2 \log t}{N_i(t)}}, \quad a_t \in \arg \max_i [\hat{\mu}_i(t) + b_i(t)].$$

Suppose at the current round the errors obey $|\hat{\mu}_i(t) - \mu_i| \leq b_i(t)$ for every arm. For a best arm i^* and selected arm a_t ,

$$\mu_{i^*} \leq \hat{\mu}_{i^*}(t) + b_{i^*}(t) \leq \hat{\mu}_{a_t}(t) + b_{a_t}(t) \leq \mu_{a_t} + 2b_{a_t}(t).$$

Subtracting μ_{a_t} shows that the chosen arm's regret is at most $2b_{a_t}(t)$. Thus, while the intervals are valid, an arm can be chosen despite being poor only if its interval is still wide enough. Sampling that arm gives another observation and reduces this width.

This rule is commonly described as *optimism in the face of uncertainty*: each arm is scored by the upper end of its confidence interval. The inequality above explains why the upper end is useful. It bounds the cost of the action we take by uncertainty that this same action helps reduce.

This one-round argument assumes valid intervals; it is not a substitute for a simultaneous probability theorem. Chapter 3 gives the corresponding complete argument for linear rewards and then for linear-score comparisons.

2.8 Chapter summary

The duelling bandit objective accounts for the quality of the whole sequence of comparisons. Simplified RUCB chooses a plausible winner and tests the opponent with the largest upper bound against it. A large upper bound can reflect either evidence of a strong opponent or a lack of evidence about that pair. The proof first makes all the confidence intervals valid together. For any run where they are valid, the algorithm never uses a wrong arm alone, and a distinct pair can be selected only while its interval is wide enough. Every comparison narrows that pair's interval. Counting these comparisons gives logarithmic regret for fixed arms and probabilities.

Principal equations

$$\begin{aligned} \Delta_i &= p_{i^*i} - \frac{1}{2}, \quad R_T = \sum_{t=1}^T \frac{\Delta_{i_t} + \Delta_{j_t}}{2}, \quad \Lambda = \log \frac{2 \binom{K}{2} T}{\delta}. \\ \mathcal{C}_t &= \{i : U_{ij}(t) \geq \frac{1}{2} \forall j\}, \quad i_t \in \mathcal{C}_t, \quad j_t \in \arg \max_j U_{ji_t}(t). \\ \mathbb{E}R_T &\leq \frac{\binom{K}{2}}{2} \left(1 + \frac{2\Lambda}{\Delta_{\min}^2}\right) + \frac{\delta T}{2}. \end{aligned}$$

The next chapter shares information through feature vectors. This requires a stronger model, but its confidence argument depends on the number of features rather than the number of separate pairs.

Contextual duelling bandits

The preceding chapter treated each pair's comparison probability as a separate unknown. Its regret bound counted all $\binom{K}{2}$ possible pairs. This is reasonable when we have no information relating one item to another. But suppose our items are news articles, and we know their topics and writing styles. Learning that a reader prefers a detailed sports article to a brief finance article should help us assess other articles with similar attributes.

We will represent this relationship through item features and a shared preference parameter. The number of unknown parameters can then be much smaller than the number of items. We begin with one reader and a fixed collection of articles. Once the algorithm and its analysis are in place, we will allow the reader, the request, and the available articles to change from round to round.

3.1 A shared model of item utilities

There are K items. Item i has a known feature vector $x_i \in \mathbb{R}^d$. We assume that its utility is

$$u(i) = x_i^\top \theta^*,$$

where $\theta^* \in \mathbb{R}^d$ is a fixed but unknown vector of preference weights. The learner knows the features and must learn the weights from comparisons. The BTL model now gives

$$p_{ij} = \sigma((x_i - x_j)^\top \theta^*), \quad \sigma(v) = \frac{1}{1 + e^{-v}}. \quad (3.1)$$

On round t , we choose i_t, j_t and observe Y_t , with $Y_t = 1$ when the first item wins and $Y_t = 0$ when the second wins. Conditional on the history and the chosen pair, the new label has the probability in Equation (3.1).

Compared with the earlier K -armed problem, we have made a stronger assumption. All pairwise probabilities must come from the same linear utility model. In return, evidence about one pair tells us about the shared weights and hence about other pairs, including pairs we have never compared.

Example 3.1.1 (One reader and four articles). Let the two features describe topic balance (sports versus finance) and depth (detailed versus brief). Use the following centred values:

Item	Description	Feature vector
A	Sports; average depth	$(.5, 0)$
B	Finance; average depth	$(-.5, 0)$
C	Balanced topics; detailed	$(0, .5)$
D	Balanced topics; brief	$(0, -.5)$

Suppose, for illustration, that the reader's weights are $\theta^* = (1, .2)$. The utilities are then $.5, -.5, .1, -.1$, respectively, so A is best. Comparing A with B measures the reader's preference for sports over finance. Comparing C with D measures the preference for depth.

The two comparisons also help predict other pairs. For example, the utility difference between A and C is $(\theta_1^* - \theta_2^*)/2$. Once we have learned about both weights, we can predict this difference without collecting separate data for A, C . The learner is given the feature table, but it is not given the hypothetical weights used in this illustration.

3.1.1 Where the features come from

Features might be editorial tags, article length, or a representation produced by a fixed neural encoder. As with the fixed-representation reward models in Module 2, comparisons train the linear scoring weights. More features may describe preferences better, but also give us more weights to estimate. Our guarantees assume that the chosen features and linear scores correctly describe the comparison probabilities.

3.2 Regret in utility units

An item with the largest utility also has the largest probability of winning against every other item. Let

$$i^* \in \arg \max_{i \in [K]} u(i).$$

There may be several maximisers; any one can serve as the benchmark. We again charge both items shown in a round. This time, since the model gives numerical utilities, we measure the shortfall in utility units:

$$\Delta_i = u(i^*) - u(i), \quad r_t = \frac{\Delta_{i_t} + \Delta_{j_t}}{2}, \quad R_T = \sum_{t=1}^T r_t. \quad (3.2)$$

The notation Δ_i is the same as in Chapter 2, but its meaning has changed. Previously it was a probability margin; here it is a utility difference. Under BTL, the earlier margin would be $\sigma(\Delta_i) - 1/2$, which is generally a different number.

We take the horizon T to be known in advance and seek a bound on regret that grows more slowly than T . We hope that the main dependence will be on the feature dimension d , rather than on the number of items. The learner observes only the comparison labels, so it cannot directly calculate the utilities or regret.

As before, selecting the same item twice means using that item alone. Its feature difference is zero, so the comparison supplies no information. Using a best item alone has zero regret. If we forced distinct items, even knowing the best item would generally leave a positive cost in every round.

3.3 Estimating the preference weights

After observing some comparisons, we can fit the shared parameter by maximum likelihood, just as in Module 1. For each previous round s , write

$$z_s = x_{i_s} - x_{j_s}.$$

The likelihood of the labels observed before round t is

$$\mathcal{L}_t(\theta) = \prod_{s < t} \sigma(z_s^\top \theta)^{Y_s} [1 - \sigma(z_s^\top \theta)]^{1-Y_s}.$$

Taking minus the logarithm gives the logistic-regression objective

$$L_t(\theta) = -\log \mathcal{L}_t(\theta) = \sum_{s < t} \left[\log(1 + e^{z_s^\top \theta}) - Y_s z_s^\top \theta \right]. \quad (3.3)$$

The product uses the conditional probability of each new label given the past and the chosen pair. Thus we can use this likelihood even though earlier labels influenced which pairs we chose next.

3.3.1 Regularisation and a bounded parameter set

For the confidence analysis, assume a known bound $S > 0$ and normalise the features so that

$$\|x_i\|_2 \leq \frac{1}{2} \quad \text{for every } i, \quad \|\theta^*\|_2 \leq S. \quad (3.4)$$

Then every comparison direction has $\|z_s\|_2 \leq 1$, and its utility difference has absolute value at most S .

We use a ridge penalty and constrain the estimate to the same parameter ball:

$$\hat{\theta}_t = \arg \min_{\|\theta\|_2 \leq S} F_t(\theta), \quad F_t(\theta) = L_t(\theta) + \frac{\lambda}{2} \|\theta\|_2^2, \quad \lambda \geq 1. \quad (3.5)$$

The penalty makes the objective strictly convex, and the compact constraint set ensures that a minimiser exists. Hence the estimate is unique, including when the comparisons so far have missed some directions. Before any labels arrive, the estimate is zero. The condition $\lambda \geq 1$ will simplify the later regret calculation.

We will refer to this as a penalised likelihood estimate. An unregularised MLE can be nonunique or fail to be finite, so we use precisely the estimator above throughout the confidence proof.

3.3.2 Why an estimate alone is not enough

The fitted utilities are $\hat{u}_t(i) = x_i^\top \hat{\theta}_t$. A natural first attempt is to compare the two items with the largest fitted utilities. But how reliable are those utilities? We may have learned a great deal about the reader's preference for topic and very little about the preference for depth. An article with an apparently modest fitted utility could still be best.

Repeatedly comparing the same two estimated leaders may leave precisely this uncertainty unresolved. To choose comparisons well, we need more than a point estimate. We need to know which utility differences are already well measured and which ones remain uncertain.

3.4 The directions measured by our comparisons

Let us return to the four articles. Every A, B comparison depends only on the topic coefficient, since $x_A - x_B = (1, 0)$. However many such comparisons we make, they tell us nothing about the depth coefficient. The number of observations alone therefore cannot describe what we have learned. We also need to record their directions.

To see how to do this, consider changing a parameter from θ to $\theta + h$. The utility difference in observation s changes by $z_s^\top h$. If this is zero for every previous observation, all their comparison probabilities remain unchanged. The data cannot distinguish the two parameters. The matrices

$$G_t = \sum_{s < t} z_s z_s^\top, \quad V_t = \lambda I + G_t.$$

record these directions. We call V_t the *regularised design matrix*, following common bandit terminology. The term λI keeps it invertible even when some directions have not been compared.

In particular,

$$h^\top G_t h = \sum_{s < t} (z_s^\top h)^2.$$

A large value means that changing the parameter by h changes many of the gaps we have measured, or changes some of them substantially. A zero value means that this change is invisible in our observations. The matrix uses the selected directions, while the labels determine the fitted parameter. After comparing a new pair, we update it by

$$V_{t+1} = V_t + z_t z_t^\top, \quad z_t = x_{i_t} - x_{j_t}.$$

Reversing the order of a pair negates z_t , but leaves its outer product unchanged.

3.4.1 Coordinate queries and missing directions

For the running articles, $x_A - x_B = e_1$ and $x_C - x_D = e_2$. After n_1 comparisons of the first pair and n_2 of the second,

$$G_t = \text{diag}(n_1, n_2), \quad V_t = \text{diag}(\lambda + n_1, \lambda + n_2).$$

If $n_2 = 0$, increasing n_1 supplies no evidence about depth. The ridge term keeps the inverse finite but does not supply a depth observation.

3.4.2 Mixed queries do not necessarily cover all coordinates

Comparing only A with C gives $z = (.5, -.5)$. After n such observations,

$$G_t = \frac{n}{4} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \quad z^\top \theta = \frac{\theta_1 - \theta_2}{2}.$$

The observed direction $(1, -1)/\sqrt{2}$ has eigenvalue $n/2$, while $(1, 1)/\sqrt{2}$ has eigenvalue zero. Both coordinates occur in each row, but their sum is still unmeasured. Off-diagonal entries matter: counts for each coordinate do not by themselves describe coverage.

3.5 Linear bandits and a fixed confidence radius

We want to use the design matrix to describe uncertainty in the fitted parameter. It is helpful to first see how this works with numerical feedback, where the estimation calculation is simpler. We will develop the linear bandit algorithm in this setting, then return to comparison labels.

In a linear bandit, we choose one feature vector x_t and observe a noisy numerical measurement of its utility:

$$Y_t = x_t^\top \theta^* + \eta_t.$$

For this section, Y_t is a real number. Assume $\|x_t\|_2 \leq 1$, $\|\theta^*\|_2 \leq S$, and that the noise has conditional mean zero. We also assume that it is τ -sub-Gaussian, meaning that

$$\mathbb{E}[e^{a\eta_t} \mid \text{information before the response}] \leq e^{a^2\tau^2/2} \quad \text{for every } a \in \mathbb{R}.$$

This condition controls the noise tails on a scale set by τ . Independent Gaussian noise of standard deviation τ satisfies it. The conditional form lets the action depend on earlier observations: once the action is chosen, the new noise still has the stated bound. Appendix C develops the probability result we use under this assumption.

3.5.1 Ridge estimation and the regularised design

For $\lambda \geq 1$, minimise squared error plus a ridge penalty:

$$\hat{\theta}_t^{\text{lin}} = \arg \min_{\theta \in \mathbb{R}^d} \left\{ \frac{1}{2} \sum_{s < t} (Y_s - x_s^\top \theta)^2 + \frac{\lambda}{2} \|\theta\|_2^2 \right\}.$$

Differentiating gives the explicit solution

$$V_t^{\text{lin}} = \lambda I + \sum_{s < t} x_s x_s^\top, \quad \hat{\theta}_t^{\text{lin}} = (V_t^{\text{lin}})^{-1} \sum_{s < t} x_s Y_s. \quad (3.6)$$

This is the same kind of design matrix as for comparisons. Here we add the chosen vector x_s , because the response measures its utility directly; a comparison measures the difference z_s instead. At $t = 1$, the estimate is zero and $V_1^{\text{lin}} = \lambda I$.

3.5.2 From the design matrix to an error ellipsoid

For intuition, suppose temporarily that we fix all query vectors in advance, omit the penalty, and use independent noise with variance τ^2 . If $G = \sum_s x_s x_s^\top$ is invertible, the ordinary least-squares error has covariance $\tau^2 G^{-1}$. Appendix A derives this identity.

For example, query e_1 a total of n_1 times and e_2 a total of n_2 times. The two coefficients are estimated by averaging their respective responses. Their error variances are τ^2/n_1 and τ^2/n_2 . More queries in the first direction make its error smaller, while leaving the second direction's error unchanged. A natural error region therefore allows different amounts of error in different directions.

For a positive definite matrix V , define the weighted norm

$$\|h\|_V = \sqrt{h^\top V h}.$$

When $V = \text{diag}(n_1, n_2)$, a bound $\|h\|_V \leq c$ becomes $n_1 h_1^2 + n_2 h_2^2 \leq c^2$, an ellipse. Its axes shorten as the corresponding query counts grow. In d dimensions, the same quadratic inequality describes an ellipsoid. The fixed-design covariance identity also gives $\mathbb{E} \|\hat{\theta} - \theta^*\|_G^2 = \tau^2 d$, so its natural radius is of order $\tau\sqrt{d}$.

This calculation explains the shape we should expect. To obtain a confidence guarantee for our regularised estimate and adaptively chosen queries, we need the following error calculation and probability bound.

Write $M_t^{\text{lin}} = \sum_{s \leq t} \eta_s x_s$ for the accumulated noise in the sampled directions. Substituting the response model into Equation (3.6) gives

$$\hat{\theta}_t^{\text{lin}} - \theta^* = (V_t^{\text{lin}})^{-1} (M_t^{\text{lin}} - \lambda \theta^*).$$

Consequently,

$$\|\hat{\theta}_t^{\text{lin}} - \theta^*\|_{V_t^{\text{lin}}} \leq \|M_t^{\text{lin}}\|_{(V_t^{\text{lin}})^{-1}} + \sqrt{\lambda} S.$$

The first term comes from the observation noise. The second comes from the penalty's tendency to shrink the estimate towards zero. To obtain it, we used $V_t^{\text{lin}} \succeq \lambda I$, meaning that $V_t^{\text{lin}} - \lambda I$ is positive semidefinite, together with $\|\theta^*\|_2 \leq S$.

3.5.3 A confidence radius chosen before the run

The self-normalized inequality of Abbasi-Yadkori et al. bounds the noise term at all times together. Using it with the preceding error inequality gives a parameter-confidence bound. We will use one radius throughout the run. For the known horizon T , choose

$$\beta_{\text{lin}} = \tau \sqrt{d \log \left(1 + \frac{T-1}{\lambda d} \right) + 2 \log \frac{1}{\delta}} + \sqrt{\lambda} S. \quad (3.7)$$

With probability at least $1 - \delta$, for every $t \leq T$,

$$\|\hat{\theta}_t^{\text{lin}} - \theta^*\|_{V_t^{\text{lin}}} \leq \beta_{\text{lin}}. \quad (3.8)$$

The usual bound has $t-1$ in place of $T-1$. Since it increases with t , substituting the horizon gives a valid bound for every $t \leq T$. Thus β_{lin} is constant during this run, although a longer planned run requires a larger radius. The design matrix still changes, so the confidence ellipsoid can become narrower even with this fixed radius.

3.5.4 An interval for each candidate's utility

The parameter bound is useful because it gives an interval for any linear prediction. To assess a candidate vector x , apply Cauchy–Schwarz in the weighted norms:

$$|x^\top (\hat{\theta}_t^{\text{lin}} - \theta^*)| \leq \|x\|_{(V_t^{\text{lin}})^{-1}} \|\hat{\theta}_t^{\text{lin}} - \theta^*\|_{V_t^{\text{lin}}}.$$

On the confidence event, the true expected reward therefore belongs to

$$x^\top \hat{\theta}_t^{\text{lin}} \pm \beta_{\text{lin}} \|x\|_{(V_t^{\text{lin}})^{-1}}.$$

The shorthand $[m \pm b]$ denotes the interval $[m - b, m + b]$. The half-width is larger when x points in a direction that is poorly measured by the design. One parameter bound gives these intervals for every candidate at once. We do not need to assign a separate failure allowance to every item.

3.5.5 The linear UCB algorithm

We can now apply the UCB idea from the preceding chapter. On each round, fit the ridge estimate using the previous responses, then select an available vector with the largest upper utility bound:

$$x_t \in \arg \max_{x \text{ available at } t} \left[x^\top \hat{\theta}_t^{\text{lin}} + \beta_{\text{lin}} \|x\|_{(V_t^{\text{lin}})^{-1}} \right]. \quad (3.9)$$

After receiving its response, add $x_t x_t^\top$ to the design and use the new data in the next fit. Let x_t^* maximise true utility over the same available set. Suppose that the parameter-confidence bound holds throughout the run. The three UCB inequalities are

$$\begin{aligned} (x_t^*)^\top \theta^* &\leq (x_t^*)^\top \hat{\theta}_t^{\text{lin}} + \beta_{\text{lin}} \|x_t^*\|_{(V_t^{\text{lin}})^{-1}} \\ &\leq x_t^\top \hat{\theta}_t^{\text{lin}} + \beta_{\text{lin}} \|x_t\|_{(V_t^{\text{lin}})^{-1}} \\ &\leq x_t^\top \theta^* + 2\beta_{\text{lin}} \|x_t\|_{(V_t^{\text{lin}})^{-1}}. \end{aligned}$$

The first and last inequalities use the confidence intervals; the middle one uses our selection rule. Define the round's regret as $r_t^{\text{lin}} = (x_t^* - x_t)^\top \theta^*$. Subtracting the selected utility from both ends gives

$$r_t^{\text{lin}} \leq 2\beta_{\text{lin}} \|x_t\|_{(V_t^{\text{lin}})^{-1}}.$$

Thus the regret is bounded by the width in the direction we actually query.

To sum these widths, we use the elliptic-potential lemma, proved in Appendix B. It says that

$$\sum_{t=1}^T \|x_t\|_{(V_t^{\text{lin}})^{-1}}^2 \leq 2d \log \left(1 + \frac{T}{\lambda d} \right).$$

Each observation increases the design matrix in its chosen direction. The lemma quantifies how this limits the total squared widths over the run, even when the directions keep changing. Cauchy–Schwarz then gives, for $R_T^{\text{lin}} = \sum_{t=1}^T r_t^{\text{lin}}$,

$$R_T^{\text{lin}} \leq 2\beta_{\text{lin}} \sqrt{2dT \log \left(1 + \frac{T}{\lambda d} \right)}. \quad (3.10)$$

This is a $\tilde{O}(d\sqrt{T})$ bound for fixed S, λ, τ , where the tilde suppresses logarithmic factors. The entire calculation uses the fixed multiplier from Equation (3.7). Since the mean utility of a vector lies in $[-S, S]$, regret is at most $2ST$ even when the confidence bound fails. Adding $2\delta ST$ to the right side of Equation (3.10) therefore gives an expectation bound. For $T \geq 2$, choosing $\delta = 1/T$ before the run makes this extra term $2S$.

3.5.6 Returning to comparison feedback

We now have a useful pattern: a parameter-confidence bound gives utility intervals, a selection rule bounds regret by the queried width, and the potential lemma controls the sum of those widths.

For comparisons, the conditional mean of the label is $\sigma(z_t^\top \theta^*)$, so we return to the penalised logistic estimate in Equation (3.5). We need a confidence theorem for that estimate. We will also need a rule for choosing two items whose combined regret is controlled by the uncertainty of the comparison between them.

3.6 Confidence ellipsoids and a fixed logistic radius

For logistic regression, we will use the same weighted norm with $V_t = \lambda I + \sum_{s < t} z_s z_s^\top$. The design again describes which directions have been measured. To obtain a confidence radius, we must also account for how a change in utility affects a binary label.

The logistic function has derivative $\sigma'(v) = \sigma(v)[1 - \sigma(v)]$. Its slope is smaller when the win probability is close to zero or one: a change in the utility gap then produces only a small change in the probability. Our bound on the parameter and features ensures that every feasible gap lies in $[-S, S]$. The smallest slope on this interval is

$$\kappa = \min_{|v| \leq S} \sigma'(v) = \sigma(S)[1 - \sigma(S)] > 0.$$

The confidence proof compares the curvature of the logistic loss with the design matrix using this lower bound. For a known horizon T and failure allowance $\delta \in (0, 1)$, choose

$$\beta = \beta_T = \frac{1}{\kappa} \left[\frac{1}{2} \sqrt{d \log \left(1 + \frac{T-1}{\lambda d} \right)} + 2 \log \frac{1}{\delta} + \sqrt{\lambda} S \right]. \quad (3.11)$$

As in the linear bandit, β_T is the usual time-dependent radius evaluated at the horizon. We use this one value throughout the run. The factor $1/2$ in its noise term comes from the fact that a centred Bernoulli label is $1/2$ -sub-Gaussian. Appendix D proves the following result.

Theorem 3.6.1 (Simultaneous logistic confidence). *Under the comparison model (3.1), bounds (3.4), and estimator (3.5), with probability at least $1 - \delta$,*

$$\left\| \hat{\theta}_t - \theta^* \right\|_{V_t} \leq \beta \quad \text{for every } 1 \leq t \leq T.$$

This holds when the queried pairs depend on earlier labels.

The confidence ellipsoid we use in the algorithm is

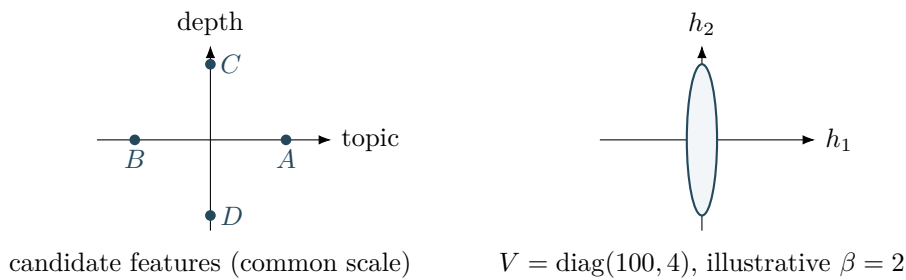
$$\mathcal{E}_t = \{ \theta : \left\| \theta - \hat{\theta}_t \right\|_{V_t} \leq \beta \}.$$

Its centre is the estimate, its shape comes from V_t , and its size is set by β . With probability at least $1 - \delta$, the true parameter belongs to all these ellipsoids up to round T . The parameter itself is fixed; it is the observed labels, estimates, and confidence sets that vary from run to run.

3.6.1 The axes of the ellipsoid

If V_t has eigenvalues $\nu_k > 0$ and orthonormal eigenvectors e_k , its quadratic form weights the error in direction e_k by ν_k . The ellipsoid's semiaxis in that direction has length $\beta / \sqrt{\nu_k}$. Larger eigenvalues therefore give shorter axes.

With $\lambda = 1$, 99 topic comparisons and three depth comparisons give $V_t = \text{diag}(100, 4)$. Writing $h = \theta - \hat{\theta}_t$, the ellipsoid is described by $100h_1^2 + 4h_2^2 \leq \beta^2$. Its depth axis is five times as long as its topic axis.



The picture and the numerical examples below use $\beta = 2$ to make the geometry easy to see. This is an illustrative radius, rather than the value given by the conservative formula above. The calculations show how to use an ellipsoid of a given size; a statistical guarantee requires a radius justified by the confidence theorem.

3.7 From one parameter set to every pairwise interval

We can now answer the question that motivated the uncertainty analysis: how reliable is the fitted utility of an item, or the fitted difference between two items? The weighted Cauchy–Schwarz calculation from the linear-bandit section applies again. For an item’s utility, it gives

$$u(i) \in x_i^\top \hat{\theta}_t \pm \beta \|x_i\|_{V_t^{-1}}.$$

To compare two items, apply the same calculation to their feature difference. We will use two quantities:

$$\hat{g}_t(i, j) = (x_i - x_j)^\top \hat{\theta}_t, \quad b_t(i, j) = \beta \|x_i - x_j\|_{V_t^{-1}}. \quad (3.12)$$

$\hat{g}_t(i, j)$ is the fitted utility gap, and $b_t(i, j)$ is its confidence half-width. Whenever the parameter bound holds,

$$u(i) - u(j) \in [\hat{g}_t(i, j) - b_t(i, j), \hat{g}_t(i, j) + b_t(i, j)]. \quad (3.13)$$

One ellipsoid therefore gives intervals for all pairs together. This is why the confidence radius need not contain a union-bound factor for the number of items. To obtain an interval for the win probability, apply the increasing function σ to both endpoints.

If the whole gap interval is positive, i is better than j whenever the bound holds. If it contains zero, the interval alone does not establish their order. We will again write $U_{ij}(t)$ for the upper endpoint:

$$U_{ij}(t) = \hat{g}_t(i, j) + b_t(i, j).$$

Here it bounds a utility gap, whereas in Chapter 2 it bounded a win probability. The threshold for ruling out a negative utility gap is zero.

3.7.1 Comparing differences directly

We could subtract the two separate utility intervals, but this can give a much wider interval than we need. Their half-widths add to $\beta[\|x_i\|_{V_t^{-1}} + \|x_j\|_{V_t^{-1}}]$. The direct gap interval uses $\beta\|x_i - x_j\|_{V_t^{-1}}$, which is no larger by the triangle inequality. The two utilities use the same parameter, so common sources of error can cancel.

For example, consider two more articles with features $x_P = (.3, .3)$ and $x_Q = (-.3, .3)$. Use $\hat{\theta}_t = (1, .2)$, $V_t = \text{diag}(100, 4)$, and illustrative $\beta = 2$. Each utility interval has half-width approximately .306. Subtracting the intervals gives a gap interval $.6 \pm .612$, which contains zero. But $x_P - x_Q = (.6, 0)$, so the direct interval is $.6 \pm .12$, entirely above zero. Whenever the illustrative ellipsoid contains the true parameter, this narrower interval establishes the order. Both articles have the same depth feature, so uncertainty about the depth coefficient cancels from their difference.

3.8 Plausible winners and MaxInP

We have intervals for all utility differences. Let us use them to decide which items could still be best. If the upper bound for $u(i) - u(j)$ is negative, then j is better than i whenever the bound is valid. We can rule out i for this round. Keeping the items that pass every such test gives

$$\mathcal{C}_t = \{i \in [K] : U_{ij}(t) \geq 0 \text{ for every } j \in [K]\}. \quad (3.14)$$

This is the same kind of row test as in RUCB. The threshold is now zero, because the intervals bound utility differences.

The set is always nonempty. An item with the largest estimated utility has nonnegative estimated gaps against every rival, so it passes all the tests. When the confidence intervals are valid, every true best item also survives. We will use that fact in the proof.

Among these plausible winners, choose the pair with the widest gap interval:

$$(i_t, j_t) \in \arg \max_{i, j \in \mathcal{C}_t} b_t(i, j) = \arg \max_{i, j \in \mathcal{C}_t} \|x_i - x_j\|_{V_t^{-1}}. \quad (3.15)$$

The reason for this choice is that an unresolved comparison between plausible winners is relevant to deciding which item is best. Restricting both items to $\mathcal{C}_t$ also controls the cost of both items we show. In RUCB we could use an excluded arm as the challenger; here, both items must pass the candidate test.

The common multiplier β does not change which pair is widest. It does change which items survive the test defining $\mathcal{C}_t$.

Regularised MaxInP with fixed confidence radius

Input: Item features, horizon T , parameter bound S , ridge strength $\lambda \geq 1$, and failure allowance $\delta \in (0, 1)$.

- 1: Set $V_1 = \lambda I$, $\hat{\theta}_1 = 0$; compute β from Equation (3.11).
- 2: **for** $t = 1, \dots, T$ **do**
- 3: Fit $\hat{\theta}_t$ using the labels observed before t .
- 4: Compute the gap bounds and construct $\mathcal{C}_t$ using Equation (3.14).
- 5: Select a widest pair in $\mathcal{C}_t$ using Equation (3.15); break ties by a fixed rule.
- 6: Compare the items, store $z_t = x_{i_t} - x_{j_t}$ and its label, and update $V_{t+1} = V_t + z_t z_t^\top$.
- 7: **end for**

If only one candidate remains, the rule selects it twice. No label is needed and the design matrix stays unchanged. More generally, a self-pair has $z_t = 0$ and can be omitted from the likelihood fit. Recompute $\mathcal{C}_t$ every round: the fitted parameter changes with new labels, so an item ruled out on one round is not permanently deleted from consideration.

The selection idea is Saha’s Maximum-Informative-Pair rule (Saha, 2021). We analyse a regularised version using the constrained estimate above and the non-strict candidate inequalities $U_{ij}(t) \geq 0$. The published version begins with initial exploration and uses an unregularised design and MLE. The proof below is for the rule stated here.

3.8.1 A round with the four articles

Take $\hat{\theta}_t = (.4, 0)$, $V_t = \text{diag}(100, 4)$, and illustrative $\beta = 2$. The A – B interval is $[.2, .6]$, so B is ruled out. The A – C and A – D intervals are each approximately $[-.310, .710]$, so both C and D remain possible winners. The C – D interval is $[-1, 1]$. Hence

$$\mathcal{C}_t = \{A, C, D\}, \quad b_t(A, C) = b_t(A, D) = 2\sqrt{.065} \approx .510, \quad b_t(C, D) = 1.$$

MaxInP compares C with D , although A has the largest estimated utility. The depth coefficient is still poorly known. A sufficiently positive value could make the detailed article C best; a sufficiently negative value could make the brief article D best. Comparing these two articles directly measures this uncertainty.

3.9 Bounding the regret of the selected pair

Suppose that the confidence bound in Theorem 3.6.1 holds throughout the run. We will first bound the regret of one round, then sum these bounds. Consider any surviving item $a \in \mathcal{C}_t$. Its gap against a true best item i^* is $u(a) - u(i^*) = -\Delta_a$. The lower end of its interval is no greater than this true gap. The upper end is nonnegative, since a passed the candidate test. Together, these observations give

$$\begin{aligned} -\Delta_a &\geq \hat{g}_t(a, i^*) - b_t(a, i^*) \\ &\geq -2b_t(a, i^*), \end{aligned}$$

where the last inequality uses $\hat{g}_t(a, i^*) + b_t(a, i^*) \geq 0$. Equivalently,

$$\Delta_a \leq 2b_t(a, i^*).$$

The optimum also belongs to $\mathcal{C}_t$. Since MaxInP maximises the width of *every* pair in that set, for both $a = i_t$ and $a = j_t$,

$$\Delta_a \leq 2b_t(a, i^*) \leq 2b_t(i_t, j_t).$$

Taking the average of the two deficits yields the key bound

$$\boxed{r_t \leq 2b_t(i_t, j_t) = 2\beta \|z_t\|_{V_t^{-1}}}. \quad (3.16)$$

Thus the cost of both selected items is bounded by uncertainty in the comparison we actually make. That comparison adds an observation in direction z_t , helping reduce the same uncertainty. The proof uses the true best item to establish the bound; the algorithm does not need to know its identity or any minimum utility gap.

The restriction to plausible winners is essential. For example, take two one-dimensional features $+0.5, -0.5$ and true weight $\theta^* = 1$. Comparing the distinct items has utility regret $.5$ in every round. Yet this pair is always wider than a self-pair while V_t is finite. A rule that keeps choosing the widest pair from the whole collection would therefore incur linear regret, even after learning which item is better. The candidate test lets us stop showing an item once it is confidently worse.

3.10 Cumulative regret and the meaning of the bound

The one-round bound reduces our task to summing the widths of the queried comparisons. Each query adds $z_t z_t^\top$ to the design matrix, making it better measured in that direction. The same elliptic-potential lemma used for linear bandits gives

$$\sum_{t=1}^T \|z_t\|_{V_t^{-1}}^2 \leq 2 \log \frac{\det V_{T+1}}{\det V_1} \leq 2d \log \left(1 + \frac{T}{\lambda d} \right).$$

This is a deterministic matrix calculation. Adaptive selection does not change it. Combining it with Equation (3.16) gives

$$\begin{aligned} R_T &\leq 2\beta \sum_{t=1}^T \|z_t\|_{V_t^{-1}} \\ &\leq 2\beta \sqrt{T \sum_{t=1}^T \|z_t\|_{V_t^{-1}}^2} \\ &\leq 2\beta \sqrt{2dT \log \left(1 + \frac{T}{\lambda d} \right)}. \end{aligned}$$

Theorem 3.10.1 (Regret of the stated regularised MaxInP rule). *Under the fixed-feature, fixed-parameter conditional logistic model and bounds (3.4), run the algorithm of Section 3.8 with $\lambda \geq 1$ and fixed $\beta = \beta_T$ from Equation (3.11). With probability at least $1 - \delta$,*

$$R_T \leq 2\beta \sqrt{2dT \log \left(1 + \frac{T}{\lambda d} \right)}.$$

In expectation,

$$\mathbb{E}R_T \leq 2\beta \sqrt{2dT \log \left(1 + \frac{T}{\lambda d} \right)} + \delta ST.$$

To obtain the expectation bound, we account for the runs where a confidence interval fails, just as in the RUCB proof. The feature and parameter bounds give $0 \leq r_t \leq S$, so even such a run has $R_T \leq ST$. These runs have probability at most δ , giving the term δST .

If $T \geq 2$, choose $\delta = 1/T$ before computing the radius and running the algorithm. The failure term then becomes S , while the smaller failure allowance affects β only through a logarithm.

For fixed S, λ , and with this confidence choice, expected regret has order $\tilde{O}(d\sqrt{T})$, so average regret tends to zero. The constants still matter: $1/\kappa$ can be large when the model permits probabilities close to zero or one. Increasing λ decreases the geometric widths but increases the ridge-bias term in β .

There is no explicit number-of-items factor in the regret bound. Computing all candidate tests and the widest surviving pair can still require $O(K^2)$ pair evaluations in a round.

3.11 Changing contexts and available items

So far, we have kept one reader and one collection of articles fixed. Now suppose that each round brings a different reader or request, and that new articles become available. A comparison made for one reader should help with another reader only to the extent that our features describe what they have in common.

We call the information available about the current reader or request the *context*, and write it as q_t . Let $\mathcal{A}_t$ be the finite, nonempty set of items available on round t . A fixed feature map ϕ combines the context with an item:

$$x_{t,i} = \phi(q_t, i) \in \mathbb{R}^d, \quad i \in \mathcal{A}_t.$$

These feature vectors are observed before we choose a pair. For news, the context might describe the reader's interests or the kind of article requested. For response evaluation, it might be the prompt, with each candidate item a response to that prompt.

We continue to use one fixed unknown parameter θ^* :

$$u_t(i) = x_{t,i}^\top \theta^*, \quad \mathbb{P}(Y_t = 1 \mid \text{past, current features, selected pair}) = \sigma((x_{t,i_t} - x_{t,j_t})^\top \theta^*).$$

The features can change, so the best item can change too. The common parameter is what allows previous comparisons to help us assess the current items.

3.11.1 Features that express context dependence

As in Module 2's reward models, the features must capture how an item relates to its context. Let $s \in \{0, 1\}$ indicate a reader's interest in sports and $c \in \{0, 1\}$ indicate a sports article. An additive score $\alpha s + \gamma c$ gives a sports-finance gap of γ for every reader: the reader-only term cancels. Adding the interaction ηsc makes the gap $\gamma + \eta s$, allowing different rankings with the same shared weights. A joint prompt-response encoder can provide richer interactions of this kind.

3.11.2 The algorithm and guarantee in the contextual setting

On each round, apply MaxInP to the current set $\mathcal{A}_t$ and features $x_{t,i}$. The estimate and design still use all previous observations:

$$z_s = x_{s,i_s} - x_{s,j_s}, \quad V_t = \lambda I + \sum_{s < t} z_s z_s^\top.$$

The current gaps and widths use the current vectors, and the candidate test ranges over the current items. An item need not have appeared before for these calculations to apply.

The regret benchmark must also refer to this round's items. Define

$$i_t^* \in \arg \max_{i \in \mathcal{A}_t} u_t(i), \quad \Delta_t(i) = u_t(i_t^*) - u_t(i), \quad r_t = \frac{\Delta_t(i_t) + \Delta_t(j_t)}{2}.$$

Under the same uniform bounds $\|x_{t,i}\|_2 \leq 1/2$ and $\|\theta^*\|_2 \leq S$, Theorem 3.10.1 holds with exactly the same radius and regret bound.

To see why, revisit the three parts of the proof. The concentration theorem allows directions chosen using the information available before each label. The candidate test keeps a best item from the current set, and the one-round argument compares both selected items with that current best item. Finally, the potential lemma applies to the whole sequence of observed differences, however those differences were chosen.

With numerical feedback, we instead apply linear UCB to the current vectors $x_{t,i}$. This is the contextual linear bandit problem: the available actions and their features depend on the observed context. The proof in Section 3.5 already compares the selected vector with the best vector available on that round, so Equation (3.10) continues to hold, using the same fixed β_{lin} .

Thus we do not need the contexts to be independent or identically distributed. We do need the stated conditional comparison model with a fixed feature map and fixed parameter. Changing the recorded context is accounted for by the features; a changing unknown preference parameter would require another model and analysis.

3.12 Chapter summary

Shared features let us learn about many items through one parameter vector. The design matrix records the directions measured by our comparisons. A confidence ellipsoid then gives intervals for the utility of any item and, more directly, for the utility difference of any pair.

MaxInP keeps items that could still be best and compares the widest pair among them. The candidate test bounds each selected item's shortfall, while the widest-pair rule makes both shortfalls no larger than twice the half-width of the comparison we query. The potential lemma controls the sum of these queried widths.

Principal equations

$$\begin{aligned}
 p_t(i, j) &= \sigma((x_{t,i} - x_{t,j})^\top \theta^*), & V_t &= \lambda I + \sum_{s < t} z_s z_s^\top, & \beta &= \beta_T. \\
 b_t(i, j) &= \beta \|x_{t,i} - x_{t,j}\|_{V_t^{-1}}, & \mathcal{C}_t &= \{i \in \mathcal{A}_t : \hat{g}_t(i, j) + b_t(i, j) \geq 0 \text{ for all } j \in \mathcal{A}_t\}. \\
 (i_t, j_t) &\in \arg \max_{i, j \in \mathcal{C}_t} b_t(i, j), & R_T &\leq 2\beta \sqrt{2dT \log(1 + T/(\lambda d))}.
 \end{aligned}$$

The regret inequality holds with probability at least $1 - \delta$, using the radius chosen for the horizon. The same reasoning applies to changing contexts because the current features are known before the next comparison. In both the linear and duelling problems, the algorithm's regret is controlled by uncertainty that its own observations help reduce.

Model checks and regression geometry

We verify the BTL properties used in Chapter 1, then derive the fixed-design covariance identity used to motivate the confidence ellipsoids in Chapter 3.

A.1 BTL satisfies SST and STI

Let $p_{ij} = \sigma(\theta_i - \theta_j)$ with $\sigma(t) = 1/(1 + e^{-t})$. For $a \succeq b \succeq c$, set $x = \theta_a - \theta_b \geq 0$, $y = \theta_b - \theta_c \geq 0$, and $f(t) = \sigma(t) - 1/2$ for $t \geq 0$. The relevant margins are $f(x), f(y), f(x + y)$. Because f is increasing, $f(x + y) \geq \max\{f(x), f(y)\}$: this is SST. Also

$$f'(t) = \sigma(t)(1 - \sigma(t)), \quad f''(t) = \sigma(t)(1 - \sigma(t))(1 - 2\sigma(t)) \leq 0.$$

Thus f' is nonincreasing on $[0, \infty)$. As $f(0) = 0$,

$$f(x + y) - f(x) = \int_0^y f'(x + t) dt \leq \int_0^y f'(t) dt = f(y),$$

which gives STI. The margins are differences from 1/2 in *probability*; they are not differences of latent utilities.

A.2 Fixed-design regression

Consider numerical observations $Y = X\theta^* + \varepsilon$ with fixed X , whose rows are the sampled feature vectors. Assume centred noise with covariance $\tau^2 I$. If $G = X^\top X$ is invertible, ordinary least squares gives

$$\hat{\theta} - \theta^* = G^{-1}X^\top \varepsilon, \quad \text{Cov}(\hat{\theta} - \theta^*) = \tau^2 G^{-1}.$$

Thus

$$\mathbb{E} \left\| \hat{\theta} - \theta^* \right\|_G^2 = \text{tr}(G\tau^2 G^{-1}) = \tau^2 d.$$

For ridge with $V = G + \lambda I$, the error has mean $-\lambda V^{-1}\theta^*$ and covariance $\tau^2 V^{-1}GV^{-1}$, which generally differs from $\tau^2 V^{-1}$. A parameter-confidence radius must include the bias. For adaptive designs, conditioning on a realised X need not preserve the simple fixed-design noise law, since earlier outcomes helped choose the rows. Appendix C supplies the adaptive argument.

The elliptic-potential lemma

Both the linear-bandit and MaxInP analyses reduce regret to a sum of directional uncertainties. This appendix proves the deterministic calculation that controls their accumulation. The term *elliptic potential* refers to the log-determinant of the design, which tracks growth across all its axes.

B.1 Rank-one growth and the general lemma

Let V_1 be positive definite and let $z_1, \dots, z_T \in \mathbb{R}^d$ be any sequence, deterministic or chosen adaptively. Define

$$V_{t+1} = V_t + z_t z_t^\top, \quad q_t^{\text{geom}} = z_t^\top V_t^{-1} z_t.$$

The matrix determinant lemma gives

$$\det V_{t+1} = \det V_t (1 + q_t^{\text{geom}}). \quad (\text{B.1})$$

Taking logarithms and summing telescopes:

$$\sum_{t=1}^T \log(1 + q_t^{\text{geom}}) = \log \frac{\det V_{T+1}}{\det V_1}. \quad (\text{B.2})$$

For every $q \geq 0$, $\min\{1, q\} \leq 2 \log(1 + q)$. When $0 \leq q \leq 1$, one can use $\log(1 + q) \geq q/(1 + q) \geq q/2$; for $q \geq 1$, use $2 \log(1 + q) \geq 2 \log 2 > 1$. Therefore:

Lemma B.1.1 (Elliptic potential, general form). *For the sequence above,*

$$\sum_{t=1}^T \min\{1, \|z_t\|_{V_t^{-1}}^2\} \leq 2 \log \frac{\det V_{T+1}}{\det V_1}.$$

The proof is Equation (B.2) and the scalar inequality. No labels, probabilities, or independence assumptions enter it.

B.2 When the truncation can be removed

In the main chapters, $V_1 = \lambda I$, $\lambda \geq 1$, and $\|z_t\|_2 \leq 1$. Since $V_t \succeq \lambda I$,

$$0 \leq q_t^{\text{geom}} \leq \frac{\|z_t\|_2^2}{\lambda} \leq 1.$$

All terms are already below the truncation threshold. In addition, $\text{tr } V_{T+1} \leq \lambda d + T$. If $\nu_1, \dots, \nu_d$ are the eigenvalues of V_{T+1} , AM-GM gives

$$\det V_{T+1} = \prod_k \nu_k \leq \left(\frac{\sum_k \nu_k}{d} \right)^d \leq (\lambda + T/d)^d.$$

Since $\det V_1 = \lambda^d$, we obtain the exact form used earlier:

Lemma B.2.1 (Normalised form used in the notes). *If $\lambda \geq 1$ and $\|z_t\|_2 \leq 1$, then*

$$\sum_{t=1}^T \|z_t\|_{V_t^{-1}}^2 \leq 2 \log \frac{\det V_{T+1}}{\lambda^d} \leq 2d \log \left(1 + \frac{T}{\lambda d} \right).$$

This also applies to the directly observed action vectors in the linear-bandit section. With a general bound $\|z_t\|_2 \leq L$, the trace expression becomes $\lambda d + TL^2$, and $\lambda \geq L^2$ suffices to remove truncation. Dropping the $\min\{1, q\}$ without such a bound would not be valid.

B.3 The effect of one query on a width

For a queried direction z , write $q = z^\top V_t^{-1} z$. Sherman–Morrison gives

$$(V_t + zz^\top)^{-1} = V_t^{-1} - \frac{V_t^{-1} z z^\top V_t^{-1}}{1 + q}.$$

Consequently, the same direction’s squared geometric width changes to

$$z^\top V_{t+1}^{-1} z = \frac{q}{1 + q}.$$

Since our multiplier is fixed, the squared confidence half-width changes from $\beta^2 q$ to $\beta^2 q/(1 + q)$. The decrease $\beta^2 q^2/(1 + q)$ grows with q . The interval’s centre can move when we refit the estimate. The determinant increases by $1 + q$ under the same update, linking this calculation to the potential lemma.

Adaptive concentration: the Abbasi-Yadkori lemma

The previous appendix controls matrix growth. We now need a probability theorem for the noise accumulated along an adaptive sequence of directions. The self-normalized inequality of Abbasi-Yadkori, Pál, and Szepesvári (Abbasi-Yadkori et al., 2011) provides that theorem. We use the indexing of these notes, so M_t, V_t contain observations strictly before round t .

C.1 Conditional noise and predictable inputs

Let $(\mathcal{F}_t)$ be increasing information sets (a filtration), and let z_s be known before response s , meaning that it is $\mathcal{F}_{s-1}$ -measurable. Current context, candidate features, and the learner's selected pair are included in that information. The response noise ϵ_s is revealed next and is conditionally τ -sub-Gaussian for $\tau > 0$:

$$\mathbb{E}[\exp(a\epsilon_s) \mid \mathcal{F}_{s-1}] \leq \exp(a^2\tau^2/2) \quad (a \in \mathbb{R}). \quad (\text{C.1})$$

One can include the next observed context and action choice in the next information set. The essential requirement is the displayed conditional law after the present direction has been selected. Independent fresh noise with the stated distribution suffices; conditional centring and the moment bound are what the theorem actually uses.

For a fixed positive definite V_1 , set

$$M_t = \sum_{s < t} \epsilon_s z_s, \quad V_t = V_1 + \sum_{s < t} z_s z_s^\top.$$

The matrix used to measure the noise depends on the same chosen inputs that generated it. This is why the bound is called self-normalized.

Theorem C.1.1 (Self-normalized martingale inequality). *For every $\delta \in (0, 1)$, with probability at least $1 - \delta$, simultaneously for every $t \geq 1$,*

$$\|M_t\|_{V_t^{-1}}^2 \leq \tau^2 \left[\log \frac{\det V_t}{\det V_1} + 2 \log \frac{1}{\delta} \right]. \quad (\text{C.2})$$

This statement already holds at all times. We do not need to allocate δ/T to each round or add a separate union bound over current feature vectors. The determinant records how much of the input space has been explored. The theorem does not require the inputs to be IID or independent of previous noise.

C.2 A proof through exponential mixtures

The following derivation explains both the determinant and uniform coverage. For a fixed vector $u \in \mathbb{R}^d$, define

$$Z_t(u) = \exp\left(\frac{u^\top M_t}{\tau} - \frac{1}{2} u^\top (V_t - V_1) u\right).$$

When the next observation arrives, the multiplying factor is

$$\exp\left((u^\top z_t) \epsilon_t / \tau - (u^\top z_t)^2 / 2\right).$$

Its conditional expectation is at most one by Equation (C.1). Thus $Z_t(u)$ is a nonnegative supermartingale starting at one: on average, conditional on what is already known, its next value does not exceed its current value.

Average these processes over the *fixed* Gaussian distribution $u \sim \mathcal{N}(0, V_1^{-1})$. Nonnegativity permits interchanging the integral and conditional expectation, so the mixture $\bar{Z}_t = \mathbb{E}_u Z_t(u)$ is also a nonnegative supermartingale starting at one. Completing the square in the Gaussian integral gives

$$\begin{aligned}\bar{Z}_t &= \frac{\det(V_1)^{1/2}}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} \exp\left(\frac{u^\top M_t}{\tau} - \frac{1}{2} u^\top V_t u\right) du \\ &= \left(\frac{\det V_1}{\det V_t}\right)^{1/2} \exp\left(-\frac{\|M_t\|_{V_t^{-1}}^2}{2\tau^2}\right).\end{aligned}$$

The completed square is centred at $V_t^{-1}M_t/\tau$; its normalising integral contributes $(2\pi)^{d/2}/\sqrt{\det V_t}$. This explains why the determinant appears rather than an arbitrary count of observations.

C.2.1 Why the guarantee is uniform in time

For a nonnegative supermartingale starting at one, the probability of ever reaching $1/\delta$ is at most δ (Ville's inequality). For completeness, stop at the first crossing or at a fixed terminal time. The stopped expectation is at most one, while a crossing contributes at least $1/\delta$. Hence the probability of a crossing before that time is at most δ . Letting the terminal time increase proves the ever-crossing statement.

Outside this crossing event, $\bar{Z}_t \leq 1/\delta$ for every t . Substituting the integral formula and taking logarithms gives Equation (C.2). This also shows why adaptively stopping or selecting a round does not invalidate the same bound.

C.3 A deterministic horizon bound

When $V_1 = \lambda I$ and $\|z_s\|_2 \leq 1$, $\text{tr } V_t \leq \lambda d + t - 1$, so AM–GM gives

$$\log \frac{\det V_t}{\lambda^d} \leq d \log \left(1 + \frac{t-1}{\lambda d}\right).$$

The theorem therefore implies, on a single event of probability at least $1 - \delta$,

$$\|M_t\|_{V_t^{-1}} \leq \tau \sqrt{d \log(1 + (t-1)/(\lambda d)) + 2 \log(1/\delta)}.$$

For $t \leq T$, replace $t-1$ by $T-1$ to obtain a single constant. Adding the ridge-bias term proves the linear confidence statement in Section 3.5. The logistic estimator needs an extra step, supplied in the next appendix.

A confidence proof for the constrained logistic estimator

This appendix completes Theorem 3.6.1. It uses an unweighted regularised design, a global lower curvature bound on a known parameter ball, and the self-normalized noise inequality. It is a conservative finite-sample argument for the estimator used in Chapter 3.

D.1 Centred Bernoulli residuals

Under the conditional logistic model, define

$$\epsilon_s = Y_s - \sigma(z_s^\top \theta^*), \quad M_t = \sum_{s < t} \epsilon_s z_s.$$

For unlabelled self-pairs, we may introduce independent fair dummy labels in this calculation. Their directions are zero, so their contributions to the noise sum, likelihood derivatives, and design are all zero. The conditional mean of ϵ_s is zero. Given the direction and past, its support is contained in an interval of length one. Conditional Hoeffding's lemma gives

$$\mathbb{E}[e^{a\epsilon_s} \mid \mathcal{F}_{s-1}] \leq e^{a^2/8} = e^{a^2(1/2)^2/2}.$$

Thus $\tau = 1/2$ in Theorem C.1.1. With probability at least $1 - \delta$, simultaneously for all t ,

$$\|M_t\|_{V_t^{-1}} \leq \frac{1}{2} \sqrt{\log \frac{\det V_t}{\lambda^d} + 2 \log \frac{1}{\delta}}. \quad (\text{D.1})$$

The residuals need not be identically distributed: their Bernoulli means vary with the selected pairs. Their conditional moment bound remains valid.

D.2 A uniform lower bound on likelihood curvature

For the penalised objective in Equation (3.5),

$$H_t(\theta) = \nabla^2 F_t(\theta) = \lambda I + \sum_{s < t} \sigma'(z_s^\top \theta) z_s z_s^\top, \quad \sigma'(v) = \sigma(v)[1 - \sigma(v)].$$

All θ in the feasible ball obey $|z_s^\top \theta| \leq S$. Since $0 < \kappa \leq 1/4$,

$$H_t(\theta) \succeq \lambda I + \kappa \sum_{s < t} z_s z_s^\top \succeq \kappa V_t. \quad (\text{D.2})$$

The bound applies to every point of the segment between θ^* and $\hat{\theta}_t$, because both are in a convex ball. This is an exact curvature inequality, not a quadratic approximation near the fitted parameter.

Let $h = \hat{\theta}_t - \theta^*$. Integrating the Hessian along the segment gives

$$[\nabla F_t(\hat{\theta}_t) - \nabla F_t(\theta^*)]^\top h = \int_0^1 h^\top H_t(\theta^* + ah) h da \geq \kappa \|h\|_{V_t}^2. \quad (\text{D.3})$$

D.3 Use constrained optimality, including the boundary

The minimiser satisfies $\nabla F_t(\hat{\theta}_t)^\top (\theta - \hat{\theta}_t) \geq 0$ for every feasible θ . Substituting $\theta = \theta^*$ gives $\nabla F_t(\hat{\theta}_t)^\top h \leq 0$. At the true parameter,

$$\nabla F_t(\theta^*) = -M_t + \lambda \theta^*.$$

Combining this with Equation (D.3) gives

$$\begin{aligned} \kappa \|h\|_{V_t}^2 &\leq [\nabla F_t(\hat{\theta}_t) - \nabla F_t(\theta^*)]^\top h \\ &\leq -\nabla F_t(\theta^*)^\top h = (M_t - \lambda \theta^*)^\top h \\ &\leq (\|M_t\|_{V_t^{-1}} + \lambda \|\theta^*\|_{V_t^{-1}}) \|h\|_{V_t} \\ &\leq (\|M_t\|_{V_t^{-1}} + \sqrt{\lambda} S) \|h\|_{V_t}. \end{aligned}$$

The third line uses weighted Cauchy–Schwarz and the triangle inequality. The last uses $V_t \succeq \lambda I$. If $h = 0$, the desired bound already holds; otherwise divide by $\kappa \|h\|_{V_t}$.

Insert Equation (D.1) and the trace bound from Appendix C to get

$$\|\hat{\theta}_t - \theta^*\|_{V_t} \leq \frac{1}{\kappa} \left[\frac{1}{2} \sqrt{d \log(1 + (t-1)/(\lambda d)) + 2 \log(1/\delta)} + \sqrt{\lambda} S \right] = \beta_t.$$

This defines the time-dependent radius β_t . It increases with t , so on the same event the error is at most β_T for every $t \leq T$. This is the fixed radius used in Equation (3.11). The proof also covers estimates on the boundary of the parameter ball, where the fitted gradient need not vanish.

D.4 Interpreting the constants

The $\sqrt{\lambda} S$ term accounts for regularisation, including unobserved directions at the start of the run. The determinant, or its trace upper bound, accounts for the geometry of adaptively accumulated noise. Finally, $1/\kappa$ converts score noise into parameter control using the least curvature allowed in the known ball. This last step makes the bound conservative when S is large.

Notation reference

The table collects notation already introduced in the text. All logarithms are natural unless another base is displayed. In a probability matrix, the row item beats the column item with the probability shown.

Symbol	Meaning
$[K], i, j$	Item or arm indices; K is the number of items in a fixed pool.
p_{ij}	Probability that i beats j .
Y_t	Comparison label: one when the first selected item wins. In the linear-bandit section, a numerical utility measurement instead.
$i^*, \hat{i}$	A true best item and a returned item. The RUCB chapter assumes a unique strict Condorcet winner.
$\Delta(i, j) = p_{ij} - 1/2$	Signed probability margin in the active-learning chapter.
Δ_i	Chapter 2: probability margin $p_{i^*i} - 1/2$. Chapter 3 with fixed features: utility deficit $u(i^*) - u(i)$.
$\Delta_t(i)$	Utility deficit against the best available item on round t , when contexts and candidate sets change.
Q, T, r_t, R_T	Pure-exploration query count; bandit horizon; one-round and cumulative regret.
ε, δ	Approximation tolerance and total failure allowance.
$N_{ij}(t), \hat{p}_{ij}(t)$	Number of unordered-pair comparisons and win fraction before round t .
$\mathcal{C}_t, U_{ij}(t)$	Current plausible-winner set and upper comparison bound. Probability bounds in Chapter 2, utility-gap bounds in Chapter 3.
$q_t, \mathcal{A}_t, x_{t,i}$	Observed context, available item set, and fixed-map feature vector $\phi(q_t, i)$. For a fixed pool, write x_i .
$\theta^*, \hat{\theta}_t, S, d$	True weight vector, estimate before round t , known parameter-norm bound, and feature dimension.
$z_t = x_{t,i_t} - x_{t,j_t}$	Queried feature difference in a duel.
$\mathcal{L}_t, L_t = -\log \mathcal{L}_t, F_t$	Likelihood, negative log-likelihood, and its ridge-penalised version.
X, G_t, V_t	Fixed regression design in Appendix A, unregularised Gram matrix, and regularised design. In duelling, $V_t = \lambda I + \sum_{s < t} z_s z_s^\top$.
$\ h\ _V, V \succeq W$	$\sqrt{h^\top V h}$; $V - W$ is positive semidefnite.
λ, β, κ	Ridge strength, fixed horizon-dependent confidence radius, and lower logistic curvature.
$b_t(i, j), q_t^{\text{geom}}$	Pairwise confidence half-width and squared geometric width of the selected pair.
$\sigma, \mathbb{P}, \mathbb{E}, \log$	Logistic function, probability, expectation, and natural logarithm.

The quantity $b_t(i, j)$ is a confidence half-width; the full interval width is $2b_t(i, j)$. Counts, estimates, and matrices indexed by t use observations strictly before round t . Symbols whose meanings change across models are defined again at their first use.

Sources and further reading

For the background on choice models, maximum likelihood, and reward modelling, see the Module 1 and 2 readers linked on the [course webpage](#). The papers below develop the algorithms and probability results used here. The active-learning chapter follows Falahatgar et al.'s maximum-selection argument. The RUCB and MaxInP chapters analyse the simplified and regularised versions stated in these notes. For a broader treatment of linear bandits and their concentration tools, see Chapters 19–21 of Lattimore and Szepesvári's book.

References

- M. Falahatgar, A. Orlitsky, V. Pichapati, and A. T. Suresh. *Maximum Selection and Ranking under Noisy Comparisons*. ICML, PMLR 70:1088–1096, 2017. [Paper and supplementary proofs](#).
- M. Zoghi, S. Whiteson, R. Munos, and M. de Rijke. *Relative Upper Confidence Bound for the K-Armed Dueling Bandit Problem*. ICML, PMLR 32(2):10–18, 2014. [Paper and supplementary material](#).
- P. Auer, N. Cesa-Bianchi, and P. Fischer. *Finite-time Analysis of the Multiarmed Bandit Problem*. Machine Learning 47:235–256, 2002. [Journal article](#).
- Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. *Improved Algorithms for Linear Stochastic Bandits*. Advances in Neural Information Processing Systems 24, 2011. [Paper](#), especially Theorems 1 and 2.
- A. Saha. *Optimal Algorithms for Stochastic Contextual Preference Bandits*. Advances in Neural Information Processing Systems 34, 2021. [Paper](#), especially Algorithm 1 (MaxInP).
- T. Lattimore and C. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020. [Author-hosted book](#), Chapters 19–21.