

---

# **Topics of Statistical Learning**

**Chi-Kuang Yeh**

**Oct 21, 2026**

# Contents

|                                                     |           |
|-----------------------------------------------------|-----------|
| <b>Course overview</b>                              | <b>6</b>  |
| A common reasoning route                            | 6         |
| Four connected questions                            | 7         |
| Choose a starting point                             | 8         |
| What you should be able to do                       | 8         |
| Course materials                                    | 8         |
| <b>Course guide and study route</b>                 | <b>9</b>  |
| Course information                                  | 9         |
| Learning outcomes                                   | 9         |
| A 15-week teaching route                            | 10        |
| Independent-study route                             | 11        |
| Proposed assessment for a taught offering           | 12        |
| Working principles                                  | 12        |
| <b>Computational workflow and reproducibility</b>   | <b>13</b> |
| 1. Run a worked experiment                          | 13        |
| 2. Predict, reproduce, change, interpret            | 13        |
| 3. Keep the target outside the learner              | 14        |
| 4. Quantify simulation error                        | 14        |
| 5. Make a small reproducible report                 | 14        |
| 6. Optional clustering-inference packages           | 15        |
| Readiness check                                     | 15        |
| <b>Assignments and paper discussion</b>             | <b>16</b> |
| Assignment 1 · Learning without complete labels     | 16        |
| Assignment 2 · Comparisons and sequential decisions | 17        |
| Assignment 3 · Inference after clustering           | 18        |
| Paper discussion and reading note                   | 20        |
| <b>Research project</b>                             | <b>21</b> |
| Choose a statistical target first                   | 21        |
| Milestones                                          | 22        |
| Required report structure                           | 22        |
| Report rubric                                       | 23        |
| Scope decisions that help                           | 23        |
| A completed study should answer these questions     | 23        |

|                                                                  |           |
|------------------------------------------------------------------|-----------|
| <b>Interactive demonstrations</b>                                | <b>24</b> |
| 1. Recovering risk from incomplete labels . . . . .              | 24        |
| 2. Exploration with one-arm feedback . . . . .                   | 24        |
| 3. Selection changes the reference distribution . . . . .        | 25        |
| <b>Reading guide</b>                                             | <b>26</b> |
| Weak supervision and incomplete labels . . . . .                 | 26        |
| Preference learning and sequential decisions . . . . .           | 27        |
| Inference after selection and clustering . . . . .               | 28        |
| A one-page reading record . . . . .                              | 28        |
| <b>1. Statistical targets and observation mechanisms</b>         | <b>29</b> |
| The question before the algorithm . . . . .                      | 29        |
| Population risk and the prediction target . . . . .              | 29        |
| Learning a rule and evaluating it . . . . .                      | 30        |
| Which population and which sampling unit? . . . . .              | 31        |
| Identification, estimation, and guarantees . . . . .             | 31        |
| A decision can change when the cost changes . . . . .            | 32        |
| Laboratory — Training error and future prediction risk . . . . . | 32        |
| Exercises . . . . .                                              | 35        |
| Primary readings . . . . .                                       | 36        |
| <b>2. What weak supervision tells us</b>                         | <b>37</b> |
| Two ways aggregate information identifies risk . . . . .         | 38        |
| Record the observation, not an imagined complete label . . . . . | 38        |
| A map of observation settings . . . . .                          | 39        |
| Class-conditional label noise . . . . .                          | 39        |
| Correcting the loss instead of the posterior . . . . .           | 40        |
| Candidate sets and complementary labels . . . . .                | 40        |
| Several weak sources and the evaluation sample . . . . .         | 41        |
| Quantify what an inverse does to error . . . . .                 | 41        |
| Laboratory — Correcting a noisy risk curve . . . . .             | 42        |
| Exercises . . . . .                                              | 45        |
| Primary readings . . . . .                                       | 46        |
| <b>3. Learning from positives and unlabelled observations</b>    | <b>47</b> |
| What is missing? . . . . .                                       | 47        |
| The two-sample model . . . . .                                   | 47        |
| Deriving the PU risk identity . . . . .                          | 48        |
| A logistic objective that can be implemented directly . . . . .  | 49        |
| A different design: labels revealed within one sample . . . . .  | 49        |
| What the assumptions do not supply . . . . .                     | 50        |
| Plan the positive and unlabelled sample sizes . . . . .          | 50        |
| Laboratory — A linear PU classifier . . . . .                    | 50        |
| Exercises . . . . .                                              | 54        |
| Primary readings . . . . .                                       | 55        |
| <b>4. PU estimation, overfitting, and sensitivity</b>            | <b>56</b> |

|                                                                    |           |
|--------------------------------------------------------------------|-----------|
| An unbiased objective can still be difficult to minimize . . . . . | 56        |
| The non-negative risk correction . . . . .                         | 56        |
| Bias, variance, and consistency are different properties . . . . . | 57        |
| Sensitivity to the supplied class prior . . . . .                  | 58        |
| Feature-dependent positive selection . . . . .                     | 58        |
| Designing a useful evaluation . . . . .                            | 59        |
| Separate three sources of failure in an experiment . . . . .       | 59        |
| Laboratory — Fixed-rule bias and fitted-rule behaviour . . . . .   | 59        |
| Exercises . . . . .                                                | 63        |
| Primary readings . . . . .                                         | 64        |
| <b>5. Learning from two unlabelled mixtures</b>                    | <b>65</b> |
| Where the supervision comes from . . . . .                         | 65        |
| The two-mixture model and target population . . . . .              | 65        |
| Inverting the mixtures . . . . .                                   | 66        |
| Why small prior separation is a statistical problem . . . . .      | 67        |
| Identification failures and links to PU learning . . . . .         | 68        |
| Allocate a budget according to variance contributions . . . . .    | 68        |
| Laboratory — Prior separation and assumption failure . . . . .     | 68        |
| Exercises . . . . .                                                | 72        |
| Primary readings . . . . .                                         | 72        |
| <b>6. Pairwise preference learning</b>                             | <b>74</b> |
| From a comparison design to a target . . . . .                     | 75        |
| 6.1 · Specify the observation and target . . . . .                 | 75        |
| 6.2 · Derive the likelihood . . . . .                              | 76        |
| 6.3 · Identification is different from a finite estimate . . . . . | 76        |
| 6.4 · Why the comparison schedule matters . . . . .                | 77        |
| Ask what a new comparison would repair . . . . .                   | 77        |
| 6.5 · Execute the comparison experiment . . . . .                  | 78        |
| 6.6 · Exercises . . . . .                                          | 81        |
| Reading and transition . . . . .                                   | 82        |
| <b>7. Contextual preferences, judges, and reward learning</b>      | <b>83</b> |
| 7.1 · A contextual comparison model . . . . .                      | 83        |
| 7.2 · Presentation and sampling matter . . . . .                   | 84        |
| 7.3 · Repeated judges are not independent rows . . . . .           | 84        |
| 7.4 · From preference likelihood to a policy . . . . .             | 85        |
| Design the holdout around the intended generalization . . . . .    | 86        |
| 7.5 · Execute the judge and policy experiment . . . . .            | 86        |
| 7.6 · Exercises . . . . .                                          | 89        |
| Reading and transition . . . . .                                   | 90        |
| <b>8. Online learning and regret</b>                               | <b>91</b> |
| The information available at each round . . . . .                  | 92        |
| 8.1 · Write the protocol before the algorithm . . . . .            | 92        |
| 8.2 · Learn from experts with Hedge . . . . .                      | 93        |
| 8.3 · Derive projected online gradient descent . . . . .           | 93        |

|                                                                          |            |
|--------------------------------------------------------------------------|------------|
| 8.4 · Regret is not automatically statistical risk . . . . .             | 94         |
| Zero static regret can coexist with a large tracking error . . . . .     | 94         |
| 8.5 · Execute the full-information experiment . . . . .                  | 95         |
| 8.6 · Exercises . . . . .                                                | 98         |
| Reading and transition . . . . .                                         | 99         |
| <b>9. Bandits: learning from chosen actions</b>                          | <b>100</b> |
| The bandit information loop . . . . .                                    | 101        |
| 9.1 · Define the stochastic model . . . . .                              | 101        |
| 9.2 · Why greedy allocation can fail . . . . .                           | 102        |
| 9.3 · Upper confidence bounds . . . . .                                  | 102        |
| 9.4 · Cumulative reward and final identification differ . . . . .        | 103        |
| Read the allocation and the regret together . . . . .                    | 103        |
| 9.5 · Execute a stationary bandit experiment . . . . .                   | 104        |
| 9.6 · Exercises . . . . .                                                | 106        |
| Reading and transition . . . . .                                         | 107        |
| <b>10. Duelling bandits and adaptive comparisons</b>                     | <b>108</b> |
| 10.1 · The preference matrix . . . . .                                   | 108        |
| 10.2 · Declare what winning means . . . . .                              | 109        |
| 10.3 · A declared regret convention . . . . .                            | 109        |
| 10.4 · Connect adaptive comparison to optimal design . . . . .           | 110        |
| A model restriction predicts comparisons you did not observe . . . . .   | 111        |
| 10.5 · A transparent explore-then-commit baseline . . . . .              | 111        |
| 10.6 · Exercises . . . . .                                               | 114        |
| Reading and transition . . . . .                                         | 115        |
| <b>11. Selective inference: asking a question after seeing the data</b>  | <b>116</b> |
| From a selected question to a conditional reference . . . . .            | 117        |
| Learning objectives . . . . .                                            | 117        |
| 11.1 A selected parameter is still a parameter . . . . .                 | 118        |
| 11.2 What probability should be controlled? . . . . .                    | 118        |
| 11.3 Deriving an exact selected-Gaussian pivot . . . . .                 | 119        |
| 11.4 From a one-dimensional truncation to polyhedral selection . . . . . | 120        |
| 11.5 What splitting does and does not provide . . . . .                  | 120        |
| Translate the conditioning event into a rejection threshold . . . . .    | 121        |
| 11.6 Reproducible Laboratory . . . . .                                   | 121        |
| Exercises . . . . .                                                      | 124        |
| Solutions . . . . .                                                      | 124        |
| Reading and next step . . . . .                                          | 125        |
| <b>12. Selective inference after hierarchical clustering</b>             | <b>126</b> |
| Learning objectives . . . . .                                            | 126        |
| 12.1 State the model before interpreting a cluster . . . . .             | 126        |
| 12.2 Why the ordinary reference distribution fails . . . . .             | 127        |
| 12.3 Reduce the problem to one varying distance . . . . .                | 127        |
| 12.4 The truncation set remembers the algorithm . . . . .                | 128        |
| 12.5 What must be fixed or accounted for? . . . . .                      | 129        |

|                                                                                     |            |
|-------------------------------------------------------------------------------------|------------|
| 12.6 Why ordinary row splitting can still fail . . . . .                            | 129        |
| Equal average means do not require identical clusters . . . . .                     | 129        |
| 12.7 Reproducible Laboratory . . . . .                                              | 130        |
| Exercises . . . . .                                                                 | 133        |
| Solutions . . . . .                                                                 | 133        |
| Reading and next step . . . . .                                                     | 134        |
| <b>13. Selective inference after k-means</b>                                        | <b>135</b> |
| Learning objectives . . . . .                                                       | 135        |
| 13.1 Lloyd's algorithm is part of the statistical procedure . . . . .               | 135        |
| 13.2 Why the fitting criterion rewards the naïve test statistic . . . . .           | 136        |
| 13.3 Conditioning on the assignment history . . . . .                               | 137        |
| 13.4 The choices outside the path also matter . . . . .                             | 137        |
| 13.5 Three different assessments of a clustering . . . . .                          | 138        |
| Keep a record of the choices that define the event . . . . .                        | 138        |
| 13.6 Reproducible Laboratory . . . . .                                              | 139        |
| Exercises . . . . .                                                                 | 143        |
| Solutions . . . . .                                                                 | 143        |
| Reading and next step . . . . .                                                     | 144        |
| <b>14. Research synthesis: assumptions, validation, and independent information</b> | <b>145</b> |
| Separate discovery and inference information . . . . .                              | 146        |
| Learning objectives . . . . .                                                       | 146        |
| 14.1 Start with the target and the observation mechanism . . . . .                  | 147        |
| 14.2 A focused extension: independent information for the same units . . . . .      | 147        |
| 14.3 Inference conditional on the discovered groups . . . . .                       | 148        |
| 14.4 A sensitivity study must challenge an assumption . . . . .                     | 149        |
| 14.5 Design a small but interpretable experiment . . . . .                          | 149        |
| 14.6 What the final project should contain . . . . .                                | 150        |
| Information reserved for inference has a measurable cost . . . . .                  | 150        |
| 14.7 Reproducible Laboratory . . . . .                                              | 151        |
| Exercises . . . . .                                                                 | 154        |
| Solutions . . . . .                                                                 | 154        |
| Reading and course conclusion . . . . .                                             | 155        |

# Course overview

This course develops **preference learning, online learning, positive-unlabelled (PU) and unlabelled-unlabelled (UU) learning, other forms of weak supervision, and selective inference after clustering**. The unifying habit is to specify the observation process, the statistical target, the assumptions that identify it, and the guarantee a method actually provides.

Each of the 14 chapters contains explanations, mathematical arguments, worked examples, a downloadable experiment, exercises, and separate worked solutions. Read the example first, predict what the experiment will show, and then change an assumption. The *interactive demonstrations* (page 24) offer a quick way to see three of these mechanisms in action.

**Instructor:** Chi-Kuang Yeh · [cyeh@gsu.edu](mailto:cyeh@gsu.edu)

**Level:** advanced master's / doctoral · **Format:** independent study or a proposed 15-week graduate elective

## A common reasoning route

### Core idea - match the conclusion to the information

Start with the quantity you want to learn, describe how the observations were produced, and identify the assumptions connecting them. Then choose an estimator or decision rule and an evaluation that answers that same question. Prediction error, regret, and selective error control are different guarantees.

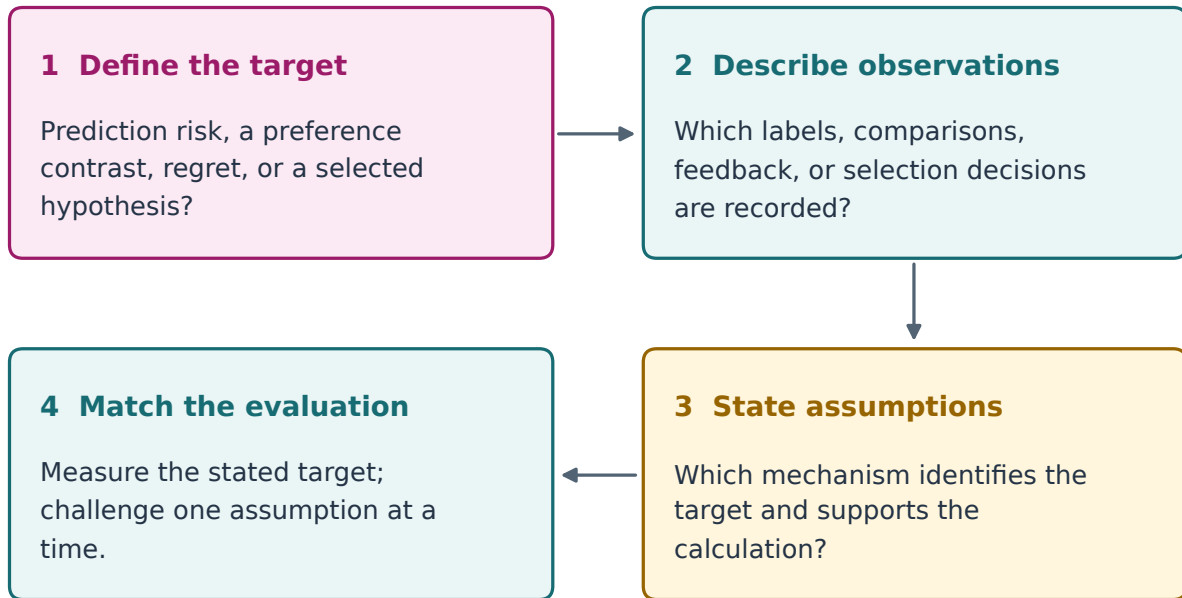


Fig. 1: The course's common reasoning route. Each chapter fills in these steps for a different observation or selection mechanism.

## Four connected questions

| Learning route                  | Central question                                                                           | Chapters                   |
|---------------------------------|--------------------------------------------------------------------------------------------|----------------------------|
| Incomplete and weak supervision | What can be learned when the labels are incomplete, noisy, or only available in aggregate? | <b>1-5</b><br>(page 29)    |
| Preference learning             | What does a comparison identify, and which comparisons should we collect?                  | <b>6-7</b><br>(page 74)    |
| Online learning and bandits     | How should a learner act when data arrive sequentially and feedback may be partial?        | <b>8-10</b><br>(page 91)   |
| Inference after selection       | How can we make a valid statement about a structure selected using the same data?          | <b>11-14</b><br>(page 116) |

The last route studies inference after a data-dependent choice. It connects to the learning routes through the observation and selection mechanisms; it is a distinct inferential problem from learning with weak labels.

## Choose a starting point

- **Read in order:** begin with *statistical targets* (page 29), then follow the chapter links. The *course guide* (page 9) maps the book to 15 teaching weeks.
- **Focus on a research problem:** use the *reading guide* (page 26) to find the relevant model, theorem, and implementation. Revisit the first chapter whenever the target becomes unclear.
- **Learn through computation:** follow the *computational workflow* (page 13), run a chapter experiment, then attempt the exercises before opening the solutions.
- **Build a research project:** use the *assignments* (page 16) for focused practice and the *project guide* (page 21) for a reproduction plus a consequential extension.

## What you should be able to do

By the end, you should be able to derive PU and UU risk identities; distinguish several weak-label mechanisms; fit and critique a pairwise preference model; formulate a sequential decision problem with an explicit comparator; derive a basic regret bound; explain conditional validity after selection; and design a reproducible experiment that probes a method's assumptions.

You will need graduate probability, mathematical statistics, linear algebra, basic optimisation, and some programming experience. *The workflow page* (page 13) includes a short readiness check and a minimal reproducible-report template.

### Note

This is a developed topic-course resource. The accompanying graduate-course proposal uses **STAT 8XXX**, with the number and offering to be assigned. The 15-week route and assessment scheme are proposed teaching arrangements; meeting dates, registration details, and institution-specific policies will be set for an actual offering.

## Course materials

[Download the complete course book](#) · [Download all labs](#) · [Course proposal](#) · [Source and build instructions](#)

# Course guide and study route

**Topics of Statistical Learning** studies the statistical structure behind learning with incomplete information and inference after a data-dependent choice. The course combines mathematical derivations, small computational investigations, critical reading, and a focused research project.

## Course information

| Item                    | Proposed arrangement                                                                                                     |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------|
| Instructor              | Chi-Kuang Yeh · <a href="mailto:cyeh@gsu.edu">cyeh@gsu.edu</a>                                                           |
| Course number           | STAT 8XXX — number to be assigned                                                                                        |
| Level and credits       | Advanced master's / doctoral elective; 3 proposed credit hours                                                           |
| Teaching format         | Lecture, seminar discussion, and guided computational investigations                                                     |
| Duration                | 15 teaching weeks; also suitable for independent study                                                                   |
| Preparation             | Graduate probability and inference, linear algebra, basic optimisation, statistical learning, and programming experience |
| Independent preparation | Approximately 6-9 hours per week during a taught offering                                                                |
| Offering information    | Term, CRN, room, meeting times, and office hours to be announced                                                         |

The public book is usable immediately. Course registration, dated deadlines, and institutional policies are supplied separately when an offering is confirmed.

## Learning outcomes

1. Specify the observed variables, latent variables, population target, and observation mechanism for a learning problem.
2. Derive PU and UU risk representations and explain the assumptions behind identification and stable estimation.

3. Distinguish noisy, complementary, partial, bag-level, and multiple-source labels, and recognise how their mechanisms differ.
4. Analyse identification, separation, comparison design, and dependence in preference models.
5. Formulate full-information and bandit problems, define an appropriate comparator, and derive a basic regret guarantee.
6. Distinguish ordinary, selective, and simultaneous inferential claims, including claims made after clustering.
7. Conduct and communicate a reproducible methodological investigation with a baseline, a sensitivity analysis, and uncertainty assessment.

## **A 15-week teaching route**

The chapters are learning units. The schedule allocates extra time to the assumptions and computations that benefit from discussion.

| Wk | Question and material                                       | Reading focus                                 | Suggested milestone                                       |
|----|-------------------------------------------------------------|-----------------------------------------------|-----------------------------------------------------------|
| 1  | <i>Targets, risks, and observation mechanisms</i> (page 29) | The course's model-target-mechanism framework | Computational readiness check                             |
| 2  | <i>Forms of weak supervision</i> (page 37)                  | Natarajan et al.; optional Ishida and Ratner  | Mechanism comparison                                      |
| 3  | <i>Positive-unlabelled learning</i> (page 47)               | Elkan-Noto; PU risk derivation                | Start Assignment 1                                        |
| 4  | <i>PU estimation and sensitivity</i> (page 56)              | Kiryo et al.; selected Bekker sections        | Prior and selection experiment                            |
| 5  | <i>Two unlabelled mixtures</i> (page 65)                    | Lu et al.; inversion and conditioning         | Assignment 1                                              |
| 6  | <i>Pairwise preferences</i> (page 74)                       | Turner-Firth; identification and estimation   | Project proposal                                          |
| 7  | <i>Context and comparison design</i> (page 83)              | Preference model extensions; optional DPO     | Start Assignment 2                                        |
| 8  | <i>Online learning and regret</i> (page 91)                 | Selected Hazan sections                       | Work through the OGD bound                                |
| 9  | <i>Bandit feedback</i> (page 100)                           | Selected Lattimore-Szepesvári chapters        | Evaluate exploration                                      |
| 10 | <i>Duelling bandits</i> (page 108)                          | Yue et al.; winner criteria                   | Assignment 2                                              |
| 11 | <i>Selection and conditional inference</i> (page 116)       | Lee et al.; truncated reference distributions | Reproduction checkpoint                                   |
| 12 | <i>Inference after hierarchical clustering</i> (page 126)   | Gao-Bien-Witten                               | Start Assignment 3                                        |
| 13 | <i>Inference after k-means</i> (page 135)                   | Chen-Witten; optional data thinning           | Audit algorithm choices                                   |
| 14 | <i>Research synthesis</i> (page 145)                        | Project-specific primary papers               | Assignment 3; project clinic                              |
| 15 | Project presentations and discussion                        | Revisit the target and assumptions            | Presentation; final report in the designated final period |

## Independent-study route

Work through one chapter at a time. For each chapter, write down the model and target before reading the derivation. Run the supplied experiment unchanged, reproduce its main result, then alter one parameter or assumption. Answer at least one derivation question and one experimental question before revealing a solution. Keep a short record of what the experiment can and cannot establish.

For a narrower reading group, Chapters 1–5 form a weak-supervision block; Chapters 6–10 form a preference and sequential-learning block; Chapters 11–13 form an inference-after-selection block. Chapter 1 supplies notation for all routes, and Chapter 14 provides a shared project framework.

## Proposed assessment for a taught offering

| Component                                    | Weight              | Timing                             |
|----------------------------------------------|---------------------|------------------------------------|
| Three integrated assignments                 | 45% total; 15% each | Weeks 5, 10, 14                    |
| Paper discussion and a two-page reading note | 10%                 | Assigned during the term           |
| Project proposal                             | 5%                  | Week 6                             |
| Reproduction checkpoint                      | 10%                 | Week 11                            |
| Final project report                         | 20%                 | Designated final-assessment period |
| Project presentation                         | 10%                 | Week 15                            |

There is no proposed written midterm or final examination. A well-designed reproduction with one consequential extension is sufficient for the project. A new theorem or publication is not required. Details and rubrics are on the *assignment* (page 16) and *project* (page 21) pages.

## Working principles

**Reproducibility.** Record random seeds, software versions, preprocessing, data provenance, and an execution command. Separate training, method selection, and final evaluation. Report Monte Carlo uncertainty for simulated performance.

**Collaboration.** Discuss ideas and compare understanding. Identify collaborators and external sources, and write your own derivations, code, and interpretation. Any offering-specific collaboration policy takes precedence for assessed work.

**AI tools.** They may assist with coding, editing, or checking a line of reasoning when permitted by the offering. Record substantive use and verify the result. An apparently plausible derivation, citation, or executable program still needs independent checking. Do not put restricted or confidential data into an external service.

**Accessibility and policies.** Students should use the official accommodations and support channels at their institution. The final offering syllabus will include the applicable attendance, late-work, academic-integrity, accessibility, and support policies. This public study guide does not substitute for that document.

# Computational workflow and reproducibility

The hierarchical-clustering and k-means chapters include calculations using the authors' specialist packages. Those calculations run when the packages are installed, and the recorded output states whether they actually ran.

## 1. Run a worked experiment

Install R from [the R Project](#) or use an existing installation. RStudio is convenient but optional. Download a chapter's .R file, open a fresh R session in that folder, and run:

```
source("03-pu-learning.R", echo = FALSE)
sessionInfo()
```

From a terminal, the same calculation is:

```
Rscript --vanilla 03-pu-learning.R
```

Each script sets its own seed and prints its results. The outputs embedded in this book come from running those scripts, rather than from manually typed illustrative results. Small floating-point differences between platforms are possible. The lab bundle contains all scripts, recorded outputs, and an execution manifest with the R version and file hashes.

If a script honours `COURSE_FIGURE_DIR`, it can also save its figure:

```
dir.create("figures", showWarnings = FALSE)
Sys.setenv(COURSE_FIGURE_DIR = normalizePath("figures"))
source("11-selective-inference.R")
```

## 2. Predict, reproduce, change, interpret

For every lab, use the same four-step workflow.

1. **Predict:** state the expected direction of a result and the assumption supporting that prediction.

2. **Reproduce:** run the worked script unchanged and check the printed diagnostic.
3. **Change:** alter one data-generating feature, comparator, or selection rule. Keep the remaining parts fixed.
4. **Interpret:** identify whether the change affected identification, estimation error, decision performance, or inferential validity.

Keep a worked example separate from your independent solution. Passing a numerical check establishes only what that check measures. A transfer experiment and a statistical explanation remain part of the exercise.

### 3. Keep the target outside the learner

In a simulation, the evaluator knows the true class labels, class prior, arm means, or null hypothesis. The learner receives only the information allowed by the observation model. For example, UCB may see the reward from the arm it selects; its action rule cannot read the true arm means used later to calculate pseudo-regret.

For a PU experiment, generate the labelled positive sample and the unlabelled sample under the stated mechanism. Use a separately generated labelled test sample for evaluation. Do not treat missing labels as negative labels without recognising that this changes the target.

### 4. Quantify simulation error

If a rejection rate is estimated from independent repetitions, a basic Monte Carlo standard error is:

$$\widehat{\text{se}}(\hat{r}) = \sqrt{\frac{\hat{r}(1 - \hat{r})}{B}}.$$

At a true rejection rate near 0.05 and  $B = 500$ , this is about 0.0097. A result of 0.06 is therefore not strong evidence of a small systematic calibration defect by itself. State the repetition count and include an interval or an uncertainty bar. When comparing two procedures, using the same simulated dataset for both gives a paired comparison; analyse the paired differences.

### 5. Make a small reproducible report

Use a transparent report format that keeps calculations and interpretation together. A minimal source is available as `report-template.qmd`.

Your report should identify:

- the observed data, target, and assumptions;
- the method and a meaningful baseline;
- the simulation or data source and all preprocessing;

- training and evaluation separation;
- results with uncertainty and one failure or sensitivity analysis;
- the command needed to regenerate the report and figures.

Record the software session with `sessionInfo()`. For a larger project, an `renv` lockfile can record R package versions. Do not include private datasets or credentials in a public repository.

## 6. Optional clustering-inference packages

Read the authors' installation instructions and package tutorials before using them. These procedures encode specific algorithms and selection events.

- `clusterpval`: hierarchical clustering, with a [technical explanation](#).
- `KmeansInference`: selected contrasts after a documented k-means procedure, with [worked tutorials](#).

An ordinary `hclust` or `kmeans` result does not automatically satisfy another procedure's conditioning requirements. Specify the distance, linkage or update rule, initialisation, number of clusters, and rule for selecting a pair to test.

## Readiness check

Before starting the course, verify that you can simulate a normal mixture, fit a logistic regression, write a function with explicit inputs, summarise repeated simulations, and derive a conditional expectation by partitioning on a binary label.

Try this check without looking at the answer: simulate  $Y \sim \text{Bernoulli}(0.3)$  and  $X | Y \sim N(2Y - 1, 1)$ . Estimate the error of the threshold rule  $\hat{Y} = 1\{X > 0\}$ . How does the theoretical error depend on the class prior?

### Readiness-check answer

The two class-specific errors are both  $\Phi(-1)$ , so the population error is  $0.3\Phi(-1) + 0.7\Phi(-1) = \Phi(-1) \approx 0.1587$ . The prior cancels because this particular model and threshold are symmetric. Moving the threshold, changing the variances, or changing the separation can remove that cancellation.

# Assignments and paper discussion

These three investigations connect the derivations, experiments, and interpretation developed in the chapters. They may be completed independently. For a taught offering, the proposed allocation is **15% per assignment**, with the final calendar and submission arrangements supplied by the instructor.

Use the report template or an equivalent reproducible format. A typical submission is **5-7 pages**, excluding references and code. Prefer a small number of carefully explained figures to a large collection of unexplained simulation outputs.

## Assignment 1 · Learning without complete labels

**Suggested timing:** begin in Week 3; complete in Week 5.

**Preparation:** *Chapters 1-5* (page 29).

### Question

When can we estimate and minimise classification risk without a fully labelled training sample, and how does the answer depend on the observation mechanism?

### Investigation

1. **Derive the PU risk.** Define  $P_+$ ,  $P_-$ , the marginal feature distribution  $P_U$ , the target class prior  $\pi$ , and a loss. Derive the PU risk identity by substituting the mixture representation. State why your positive sample is representative of  $P_+$  and why your unlabelled sample represents  $P_U$ .
2. **Build a controlled experiment.** Use a binary Gaussian model or another fully specified distribution. Generate independent training and labelled evaluation samples. Compare a supervised oracle, a naive method that treats unlabelled cases as negative, an unbiased PU estimate, and its non-negative correction. Keep the hypothesis class and optimisation budget comparable. An oracle is a simulation benchmark with extra information, not an available PU procedure.
3. **Probe one assumption.** Vary either the supplied class prior or the positive-label selection mechanism. Explain which identity or estimation argument is

affected. If you introduce feature-dependent selection, do not apply a SCAR identity as though the observed positive sample remained representative.

4. **Study UU conditioning.** For a fixed classifier, generate two unlabelled mixtures with shared class-conditionals and known different priors. Estimate its target risk using the UU identity. Change the gap between the mixture priors while keeping the target prior fixed. Report bias, variability, and Monte Carlo uncertainty. Explain what happens when the two priors are equal.
5. **Write a conclusion.** Distinguish identification, variance, optimisation, and evaluation error. Explain one result that initially surprised you.

### Required evidence

Include the risk derivation, the observation-mechanism diagram or table, one comparison of predictive risk, one sensitivity figure, and a compact table of Monte Carlo uncertainty. Use a fixed classifier in the UU subexperiment to isolate risk-estimation variability from model-selection effects.

### Evaluation criteria

| Criterion                                           | %   |
|-----------------------------------------------------|-----|
| Correct target, mechanism, and derivation           | 30% |
| Fair experimental design and independent evaluation | 30% |
| Sensitivity analysis and statistical interpretation | 25% |
| Reproducibility and communication                   | 15% |

## Assignment 2 · Comparisons and sequential decisions

**Suggested timing:** begin in Week 7; complete in Week 10.

**Preparation:** *Chapters 6–10* (page 74).

### Question

What do pairwise comparisons identify, and how does restricted or adaptive feedback change the design and evaluation of a learning procedure?

### Investigation

1. **Identify the preference model.** Write the Bradley–Terry likelihood and choose an identification constraint. Explain the difference between the connectivity needed to identify contrasts and the outcome pattern needed for a finite unpenalised estimate.

2. **Construct three comparison datasets.** Use a connected graph with wins in both directions, a disconnected graph, and a connected graph with separation. Examine the objective, estimates, and numerical behaviour. If you use a penalty, state the penalised target and show how the result changes with penalty strength.
3. **Evaluate a comparison design.** Compare two pair-allocation schemes with the same comparison budget. State whether the goal is predicting pair outcomes, estimating a particular contrast, or recovering a ranking. Use the information structure from *Chapter 7* (page 83) to motivate the design. Simulate repeated datasets and evaluate the chosen objective.
4. **Implement a reward-bandit comparison.** In a separate stationary Bernoulli bandit model, compare uniform exploration and UCB. Initialise each arm once. Record action counts and cumulative pseudo-regret against the best fixed mean. The simulator may use true means for evaluation; the learner may not use them to choose an arm.
5. **Connect the settings carefully.** Explain what would have to change to turn the comparison problem into a duelling-bandit problem. Specify a winner criterion and a feedback model. A reward-bandit implementation is not automatically a duelling algorithm.

### Required evidence

Include the model and graph examples, an analysis of finite estimation, a budget-matched preference-design comparison, repeated bandit trajectories with uncertainty, and a paragraph explaining the two different feedback structures. If observations share an annotator or prompt, explain the dependence and evaluation unit.

### Evaluation criteria

| Criterion                                              | %   |
|--------------------------------------------------------|-----|
| Preference formulation, identification, and estimation | 30% |
| Comparison-design reasoning and experiment             | 25% |
| Sequential implementation and regret evaluation        | 30% |
| Reproducibility and communication                      | 15% |

## Assignment 3 · Inference after clustering

**Suggested timing:** begin in Week 12; complete in Week 14.

**Preparation:** *Chapters 11–13* (page 116).

## Question

How does using the data to choose clusters change the reference distribution of a test of their mean contrast?

## Investigation

1. **State the model and selected null.** Begin with independent Gaussian observations, a known common noise scale, and a single homogeneous cloud. Specify the dimension, sample size, clustering procedure, number of clusters, and tested pair. Write the null in terms of the selected groups' average observation-specific means.
2. **Freeze the selection procedure.** Use one documented hierarchical-clustering method supported by the authors' implementation. Keep the distance, linkage, preprocessing, cut rule, and pair-selection rule fixed. Record every step that uses the data.
3. **Compare two tests.** Compute a naive fixed-group test and the appropriate selective test from `clusterpval`. Reuse the same dataset for the comparison. Report the number of successful computations and any numerical failures. Do not silently exclude difficult datasets.
4. **Repeat under the null.** Use at least 500 repetitions when computationally feasible. Plot p-value distributions and report rejection rates with binomial or Monte Carlo uncertainty. Explain why apparent cluster separation in one sample is insufficient evidence of distinct population components.
5. **Investigate one extension.** Change separation under a stated alternative, the dimension, or an assumption about the noise. Distinguish a power experiment under the model from a robustness experiment outside it. Do not claim exact validity for an estimated-scale or non-Gaussian extension unless the chosen method supports that claim.
6. **Audit the inferential statement.** Explain why a selected-pair p-value does not by itself control errors over every pair, and why clustering stability answers a different question.

## Required evidence

Include an explicit selected null, an algorithm specification, an executed author-package call, null rejection results with uncertainty, and a failure or sensitivity analysis. The elementary Gaussian selection demo is useful for understanding the mechanism, but it does not replace the required clustering-inference computation.

## Evaluation criteria

| Criterion                                         | %   |
|---------------------------------------------------|-----|
| Model, selected target, and conditioning argument | 30% |
| Faithful implementation and selection audit       | 25% |
| Simulation design, uncertainty, and sensitivity   | 30% |
| Reproducibility and communication                 | 15% |

## Paper discussion and reading note

Choose a primary paper in consultation with the instructor for a taught offering, or use the *reading guide* (page 26) for independent study. Prepare a two-page note and a short discussion organised around six questions:

1. What is observed, and what is the target?
2. Which assumptions identify the target?
3. What does the main theorem guarantee, and for what procedure?
4. Which implementation choices matter to the theorem?
5. What is the most convincing experiment, and what comparison is missing?
6. What would you change or investigate next?

The proposed 10% discussion component evaluates accuracy, explanation of assumptions, critical analysis, and engagement with other participants. A concise derivation or a carefully chosen counterexample often contributes more than a long summary of every section.

# Research project

The project develops one question far enough that another researcher can reproduce the result and understand the assumptions behind it. A strong project reproduces a meaningful result, establishes a fair baseline, and investigates one consequential extension or sensitivity. It need not invent a new algorithm or prove a new theorem.

## Choose a statistical target first

Write one sentence of the form: **“Under this observation mechanism, we will estimate, predict, optimise, or test this target, and evaluate success by this criterion.”** This keeps an interesting implementation from becoming a collection of unrelated experiments.

| Direction                    | Example research question                                                                     | A useful sensitivity analysis                                                                  |
|------------------------------|-----------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| PU or UU learning            | How accurately can a fixed hypothesis class estimate target risk with incomplete labels?      | Misspecified class prior, feature-dependent positive selection, or shrinking mixture-prior gap |
| Preference models and design | Which comparison allocation best estimates a scientifically meaningful contrast?              | Comparison graph, separation, prompt heterogeneity, or repeated annotators                     |
| Online learning and bandits  | How does an algorithm balance information gathering and reward in a specified feedback model? | Horizon, reward gap, nonstationarity as an explicit model departure, or a different comparator |
| Inference after clustering   | How does conditioning on a specified selection event affect calibration and power?            | Signal strength, dimension, noise assumptions, or the selection procedure                      |

These are starting points. A project may combine topics when the connection is statistically explicit, such as comparison allocation through Fisher information or calibration after a data-dependent model choice.

## Milestones

### Week 6 · Proposal

Submit approximately two pages containing the question, observed data, target, assumptions, primary reference, baseline, and one planned extension. Identify the computational requirements and a minimal result that can be completed even if the extension proves difficult. This is **5%** of the proposed course assessment.

### Week 11 · Reproduction checkpoint

Provide working code, one reproduced table or figure, an explanation of any discrepancy, and a revised plan. Include the software environment and execution instructions. An honest diagnosis of an unsuccessful reproduction is useful evidence when it is specific and supported. This is **10%**.

### Week 15 · Presentation

Present for **12 minutes**, followed by questions. Explain the target before the algorithm, show the most informative evidence, and state one limitation. Every group member should be able to explain the assumptions and interpretation. This is **10%**.

### Final-assessment period · Report

Submit **8-10 pages**, excluding references and code, with a reproducible supplement. This is **20%**. The exact date is set for the offering. Independent-study readers can use the same structure as a research note.

## Required report structure

1. **Question and contribution.** What did you investigate, and why does the answer matter?
2. **Model and target.** Define the observed and latent variables, the data-generating or sampling mechanism, and the estimand or performance criterion.
3. **Method and guarantee.** Explain the method at the level needed to understand the result. State the relevant assumptions and distinguish a theorem from an empirical observation.
4. **Reproduction.** Identify the original result, what you matched, and any necessary departures from the reference implementation.
5. **Extension or sensitivity.** Change one consequential aspect of the setting. Explain the predicted effect and compare it with the observed result.
6. **Evaluation and uncertainty.** Use suitable baselines, independent or appropriately structured evaluation, repeated runs where relevant, and uncertainty measures.

7. **Interpretation and limitations.** State what the evidence supports and what it leaves unresolved.
8. **Reproducibility.** Supply code, versions, seeds, data provenance, execution commands, and the computational budget.

## Report rubric

| Criterion                            | %   | Evidence of strong work                                                                         |
|--------------------------------------|-----|-------------------------------------------------------------------------------------------------|
| Statistical formula-<br>tion         | 25% | A precise target, observation mechanism, and conse-<br>quential assumptions                     |
| Methodological rea-<br>soning        | 25% | A correct derivation, theorem interpretation, or cri-<br>tique tied to the question             |
| Empirical design<br>and uncertainty  | 30% | Fair comparisons, meaningful sensitivity, appropriate<br>evaluation, and uncertainty            |
| Reproducibility and<br>communication | 20% | Executable work, clear figures and tables, documented<br>provenance, and a coherent explanation |

## Scope decisions that help

An assignment may supply the baseline for the project, provided the project adds a clearly identified question and deeper analysis. A small simulated setting often makes the assumptions easier to examine than a large application with an unclear observation process. Real data can add value when their sampling and annotation mechanisms are sufficiently documented.

Use ordinary laptop resources when possible. No paid API service or GPU is required. If the project involves modern language-model preferences, a finite collection of precomputed comparisons can support a substantial statistical investigation without training a model.

## A completed study should answer these questions

Could someone reproduce the principal figure from a fresh session? Does the learner receive only the information permitted by the model? Was the final evaluation used to choose the method? Does the claimed guarantee concern prediction, regret, parameter estimation, selective validity, or simultaneous error control? What happens when one key assumption changes?

The final chapter, *Research synthesis* (page 145), works through this audit in a small reproducible investigation.

# Interactive demonstrations

Change a setting, predict its effect, and then examine the result. These small browser experiments expose a mechanism; the chapters develop the assumptions and proofs. Every experiment uses simulated data and a visible random seed. The displays run locally in your browser.

## 1. Recovering risk from incomplete labels

Let  $X \mid Y = +1 \sim N(\delta/2, 1)$  and  $X \mid Y = -1 \sim N(-\delta/2, 1)$ . For the threshold classifier  $g_t(x) = 1\{x > t\}$ , the true misclassification risk is

$$R(t) = \pi\Phi(t - \delta/2) + (1 - \pi)\{1 - \Phi(t + \delta/2)\}.$$

The true curve is available because this is a simulation. A PU learner receives a representative positive sample and an independent sample from the full feature marginal. A UU learner receives two unlabelled mixtures with shared class-conditionals and known priors  $a$  and  $b$ .

Try a wrong supplied prior in PU mode. In UU mode, move the two mixture priors together. At  $a = b$ , the two-mixture inversion is unidentified. An empirical corrected risk can lie outside  $[0, 1]$  even though a population misclassification probability cannot.

The PU display uses  $\hat{\pi}\hat{E}_{+|\ell_+} + \hat{E}_{U|\ell_-} - \hat{\pi}\hat{E}_{+|\ell_-}$ . Its non-negative version truncates only the estimated negative-class contribution. The UU display uses the mixture inversion from [Chapter 5](#) (page 65), with the specified target prior. See [Chapters 3–4](#) (page 47) for PU assumptions and finite-sample qualifications.

## 2. Exploration with one-arm feedback

Three stationary Bernoulli arms have means  $(0.30, 0.45, \mu_3)$ . Each round reveals only the reward from the selected arm. Compare round-robin uniform exploration with UCB, which first pulls each arm once and then maximises

$$\hat{\mu}_k(t-1) + \sqrt{2\log(t)/N_k(t-1)}.$$

The plot shows cumulative **pseudo-regret**  $\sum_{t=1}^T (\mu^* - \mu_{A_t})$ . Only the simulator uses the true means. Both procedures use the same pre-generated reward stream for each arm, indexed by that arm's pull count, to support a paired comparison.

This is a reward-bandit demonstration. A duelling-bandit learner observes a comparison outcome instead of an arm reward; it needs an explicit winner criterion and a corresponding performance definition. See [Chapter 9](#) (page 100) and [Chapter 10](#) (page 108).

### 3. Selection changes the reference distribution

Consider independent  $Z_1, \dots, Z_p \sim N(0, 1)$  and select  $J = \arg \max_j |Z_j|$ . Conditional on the remaining coordinates, the selected statistic must satisfy  $|Z_J| \geq m$ , where  $m = \max_{k \neq J} |Z_k|$ . The selected two-sided p-value is

$$p_{\text{sel}} = \frac{1 - \Phi(|Z_J|)}{1 - \Phi(m)}, \quad p_{\text{naive}} = 2\{1 - \Phi(|Z_J|)\}.$$

For  $p = 1$ , take  $m = 0$  and the two p-values agree. Under this independent all-null model, selecting the smallest naive p-value produces rejection probability  $1 - (1 - \alpha)^p$ . The repeated simulation reports a Monte Carlo standard error alongside each rejection rate.

The display numerically evaluates the normal tail and repeats the null experiment 3,000 times. [Chapter 11](#) (page 116) develops the conditional calculation. [Chapters 12–13](#) (page 126) explain what changes for selected cluster contrasts.

# Reading guide

Read each paper with a model-target-assumption-guarantee table beside you. The aim is to understand a small number of central arguments and reproduce an informative result, rather than to read every page of every reference. The links below point to original papers, books, or the authors' implementations.

## Weak supervision and incomplete labels

**Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. (2013).** [Learning with noisy labels](#). *NeurIPS*.

**Read for:** the relation between clean and corrupted losses under class-conditional noise. Identify the noise rates, the invertibility condition, and the consequence of estimating those rates. Used in Chapter 2.

**Elkan, C., and Noto, K. (2008).** [Learning classifiers from only positive and unlabeled data](#). *KDD*.

**Read for:** the positive-label selection model and its role in relating observed labels to latent classes. Distinguish an assumption that enables a relationship from an argument that identifies an unknown constant. Used in Chapter 3.

**du Plessis, M. C., Niu, G., and Sugiyama, M. (2015).** [Convex formulation for learning from positive and unlabeled data](#). *ICML, PMLR 37*, 1386–1394.

**Read for:** risk rewriting and the choice of loss. Follow where the positive distribution and the feature marginal enter the objective. Chapters 3–4.

**Kiryo, R., Niu, G., du Plessis, M. C., and Sugiyama, M. (2017).** [Positive-unlabeled learning with non-negative risk estimator](#). *NeurIPS*.

**Read for:** why an unbiased empirical component can become negative, how the correction changes the objective, and the difference between population identities and finite-sample behaviour. Chapter 4.

**Lu, N., Niu, G., Menon, A. K., and Sugiyama, M. (2019).** [On the minimal supervision for training any binary classifier from only unlabeled data](#). *ICLR*.

**Read for:** two-mixture identification, risk rewriting, and the roles of distinct mixture priors and a target prior. Derive the two-by-two inversion yourself before reading the general formulation. Chapter 5.

## Extensions

**Bekker, J., Robberechts, P., and Davis, J. (2019).** *Beyond the selected completely at random assumption for learning from positive and unlabeled data.* *ECML PKDD*.

Feature-dependent positive selection requires an explicit mechanism and additional identification conditions. Use this after the introductory SCAR setting.

**Ishida, T., Niu, G., Menon, A. K., and Sugiyama, M. (2019).** *Complementary-label learning for arbitrary losses and models.* *ICML, PMLR 97*, 2971–2980.

Read the label-generation model carefully: a complementary label rules out a class, and its distribution across excluded classes matters. In a binary problem, a correct complementary label reveals the true class.

**Ratner, A. J., De Sa, C. M., Wu, S., Selsam, D., and Ré, C. (2016).** *Data programming: creating large training sets, quickly.* *NeurIPS*.

Use as an introduction to combining multiple noisy supervision sources. Ask which dependencies are modelled and how source accuracy is identified.

## Preference learning and sequential decisions

**Bradley, R. A., and Terry, M. E. (1952).** *Rank analysis of incomplete block designs: I. The method of paired comparisons.* *Biometrika*, 39, 324–345.

**Read for:** the original paired-comparison model and its interpretation. Chapter 6.

**Turner, H., and Firth, D. (2012).** *Bradley-Terry models in R: the BradleyTerry2 package.* *Journal of Statistical Software*, 48(9), 1–21.

**Read for:** fitting, identification, model extensions, and practical interpretation. Distinguish structural graph connectivity from observed-data separation. Chapters 6–7.

**Hazan, E. (2022).** *Introduction to online convex optimization*, second edition. MIT Press.

**Read for:** the online protocol, static regret, projected gradient descent, and one complete regret proof. Start with the introductory and online-gradient-descent sections. Chapter 8.

**Lattimore, T., and Szepesvári, C. (2020).** *Bandit algorithms.* Cambridge University Press.

**Read for:** stochastic finite-armed bandits, confidence bounds, and the difference between cumulative reward and best-arm identification. Select the introductory stochastic-bandit and UCB material; the whole book is not required. Chapter 9.

**Yue, Y., Broder, J., Kleinberg, R., and Joachims, T. (2012).** *The k-armed dueling bandits problem.* *Journal of Computer and System Sciences*, 78(5), 1538–1556.

**Read for:** comparative feedback and the assumptions needed to define and learn a winner. Check which regret definition the result controls. Chapter 10.

## Optional connection to generative AI

**Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. (2023).** *Direct preference optimization: your language model is secretly a reward model.* *NeurIPS*.

Read after the paired-comparison model. Identify the policy, reference distribution, regularisation, and assumptions connecting a preference model to the optimisation objective. The course does not require language-model training.

## Inference after selection and clustering

**Lee, J. D., Sun, D. L., Sun, Y., and Taylor, J. E. (2016).** *Exact post-selection inference, with application to the lasso.* *Annals of Statistics*, 44(3), 907–927.

**Read for:** the conditional-inference principle and a Gaussian reference distribution truncated by selection. The elementary example in Chapter 11 introduces the mechanism before the full polyhedral argument.

**Gao, L. L., Bien, J., and Witten, D. (2024).** *Selective inference for hierarchical clustering.* *Journal of the American Statistical Association*, 119(545), 332–342; published online in 2022.

**Read for:** the selected pair of clusters, the mean-contrast null, the selection event, and computation along a conditioned path. Use the [authors' technical guide](#) and [clusterpval tutorial](#) alongside Chapter 12.

**Chen, Y. T., and Witten, D. M. (2023).** *Selective inference for k-means clustering.* *Journal of Machine Learning Research*, 24(152), 1–41.

**Read for:** the role of the algorithm's intermediate assignments and initialisation in the conditioning event. Compare the procedure with an arbitrary call to `kmeans`. Use the [authors' tutorial](#) alongside Chapter 13.

**Neufeld, A., Dharamshi, A., Gao, L. L., and Witten, D. (2024).** *Data thinning for convolution-closed distributions.* *Journal of Machine Learning Research*, 25(57), 1–35.

**Optional:** study how the data model permits construction of independent pieces. Compare its assumptions with ordinary sample splitting, and identify the target retained by each approach.

## A one-page reading record

For each primary paper, record the observed data, latent quantities, target, identification assumptions, estimator or test, theorem statement, computational requirements, and a possible failure case. End with one experiment that would help you understand the result. This record can become the foundation of the [paper discussion](#) (page 20) or [research project](#) (page 21).

# 1. Statistical targets and observation mechanisms

## Core idea - The target comes before the algorithm

A dataset can support several questions, but each question needs its own loss, population, and evaluation unit. Decide whether the output is a class decision, a probability, or an inferential statement before judging the method. A small training error answers none of these questions on its own.

*Topics of Statistical Learning · Week 1*

## The question before the algorithm

A statistical learning problem starts with a target and an observation model. Before fitting a classifier, ask what will be predicted, for whom it will be predicted, which information is observed, and how errors will be evaluated. These questions become essential when a positive label is recorded selectively, two people supply a comparison instead of a score, or the data determine which hypothesis will be tested.

This chapter establishes notation used throughout the course. By the end, you should be able to distinguish population risk from training error, explain why an independent test sample is useful, identify the sampling unit, and describe information that an observed dataset does not contain.

## Population risk and the prediction target

Let  $(X, Y) \sim P$  describe a future observation from the target population. A prediction rule  $g$  maps features to an action or score, and a loss  $\ell(g(X), Y)$  describes its consequence. The **population risk** is

$$R_P(g) = E_P\{\ell(g(X), Y)\}.$$

Both the distribution and the loss are part of the problem. For binary classification, zero-one loss evaluates class decisions. Log loss and Brier loss evaluate probability predictions. For regression, squared error targets a conditional mean; absolute error

targets a conditional median. A method that is best for one loss need not be best for another.

For squared error, write  $m(x) = E(Y | X = x)$ . Conditioning on  $X$  gives

$$R_P(g) = E_P\{(g(X) - m(X))^2\} + E_P\{\text{Var}(Y | X)\}.$$

The cross term vanishes because  $E\{Y - m(X) | X\} = 0$ . The first term is prediction error relative to the optimal conditional mean. The second is irreducible under the available features and this loss. Adding a useful feature can change both the target function and the remaining conditional variance.

### **Worked example 1.1 — A class decision and a probability prediction**

Suppose  $P(X = 0) = 0.6$ ,  $P(X = 1) = 0.4$ , and

$$P(Y = 1 | X = 0) = 0.2, \quad P(Y = 1 | X = 1) = 0.8.$$

Under equal-cost zero-one loss, the optimal decisions are  $g(0) = 0$  and  $g(1) = 1$ . Their risk is  $0.6(0.2) + 0.4(0.2) = 0.20$ . Always predicting zero has risk  $P(Y = 1) = 0.44$ .

For Brier loss  $(q - Y)^2$ , the optimal probability scores are instead  $q(0) = 0.2$  and  $q(1) = 0.8$ . Their risk is  $E\{\eta(X)(1 - \eta(X))\} = 0.16$ , where  $\eta(x) = P(Y = 1 | X = x)$ . Using the hard class decisions as probability scores gives Brier risk 0.20. A good classification rule therefore need not provide the most informative probabilities.

## Learning a rule and evaluating it

Given independent training observations  $D_n = \{(X_i, Y_i)\}_{i=1}^n$ , the empirical risk is

$$\hat{R}_n(g) = \frac{1}{n} \sum_{i=1}^n \ell(g(X_i), Y_i).$$

An empirical risk minimizer chooses  $\hat{g} \in \arg \min_{g \in \mathcal{G}} \hat{R}_n(g)$ , possibly with a regularization penalty. A fixed rule's empirical risk is unbiased for its population risk under the stated sampling model. That statement cannot simply be transferred to a rule selected using those same observations.

For exact empirical risk minimization over a fixed class, assuming the expectations exist,

$$E\{\hat{R}_n(\hat{g})\} \leq \inf_{g \in \mathcal{G}} R_P(g) \leq E\{R_P(\hat{g})\}.$$

The first inequality follows by comparing the empirical minimizer with each fixed candidate before taking expectations. It describes average optimism; it does not assert that training error is smaller in every realization.

An independent test sample from  $P$  provides a simpler evaluation statement:

$$E\{\hat{R}_{\text{test}}(\hat{g}) | D_n\} = R_P(\hat{g}).$$

Conditional on training, the fitted rule is fixed and the test observations remain fresh. If the test results are repeatedly used to tune the method, this interpretation changes. A practical experiment separates fitting, tuning, and final assessment, and records which decisions use each sample.

## Which population and which sampling unit?

An observation may be a person, document, prompt, comparison, or time point. Rows need not be independent units. Several ratings of one prompt can share difficulty; several comparisons can share an annotator. Evaluation and resampling should respect the scientific sampling unit and the target of generalization.

A change in the target population also changes risk. A class imbalance is not merely a display choice: it determines how class-specific errors are weighted. Report whether the target is the observed population, a future population, or a deliberately balanced population.

### Worked example 1.2 — The same classifier in a different population

A classifier has sensitivity 0.8 and specificity 0.9. In a population with positive prevalence  $\pi = 0.2$ , its zero-one risk is

$$0.2(1 - 0.8) + 0.8(1 - 0.9) = 0.12.$$

If the prevalence becomes 0.6 while the class-conditional feature distributions remain the same, the same classifier has risk  $0.6(0.2) + 0.4(0.1) = 0.16$ . Its balanced error is  $\{0.2 + 0.1\}/2 = 0.15$  in either population.

The change does not require the classifier to have deteriorated within either class. It reflects a different target distribution. Under the stated common-class-conditional assumption, source observations could be weighted by  $0.6/0.2 = 3$  for positives and  $0.4/0.8 = 0.5$  for negatives to evaluate the new population risk. Without that assumption, these two weights are not sufficient.

## Identification, estimation, and guarantees

A target is **identified** when the observed-data distribution determines it uniquely within the assumed model. Estimation asks how to recover an identified quantity from finitely many observations. Numerical optimization then asks how well a computational procedure solves the chosen estimation problem. More data or a better optimizer cannot repair a target that the observation model leaves unidentified.

For example, an unlabelled feature distribution alone does not generally determine which observations belong to a scientifically named positive class. PU and UU learning add information through sampling assumptions, class proportions, or selected labels. We will state this information before deriving an empirical objective.

The course also uses several different guarantees:

| Question                                              | Example target or criterion                        |
|-------------------------------------------------------|----------------------------------------------------|
| How well will a rule predict?                         | Population risk or excess risk                     |
| How well does an estimated score recover a parameter? | Bias, variance, or interval coverage               |
| How costly were sequential decisions?                 | Regret relative to a stated comparator             |
| Is a selected hypothesis tested validly?              | Type I error under the appropriate selection event |

A low prediction error does not establish a valid selected-hypothesis test. Conversely, a correctly calibrated test need not imply useful prediction. We will keep the target, its assumptions, and the scope of its guarantee together.

## A decision can change when the cost changes

Suppose a false positive costs  $c_{FP}$  and a false negative costs  $c_{FN}$ , with correct decisions costing zero. At a feature value with positive probability  $\eta(x)$ , predicting positive has conditional cost  $c_{FP}\{1 - \eta(x)\}$ ; predicting negative has cost  $c_{FN}\eta(x)$ . The optimal threshold is therefore

$$\eta(x) > \frac{c_{FP}}{c_{FP} + c_{FN}}.$$

If missed positives cost four times as much as false alarms, the threshold is 0.2, not 0.5. A case with probability 0.3 is then classified positive even though negative is the more likely class. This is an illustrative decision calculation, not a new fitted-model result.

### Analysis check - align the score with the decision

Write down the target population, prediction unit, loss, and available information at prediction time. Evaluate the declared cost as well as any probability score. A method can improve a probability forecast without changing every class decision, or improve a cost-sensitive decision while reducing ordinary accuracy.

## Laboratory — Training error and future prediction risk

The standalone experiment fits polynomials of degrees 1, 2, and 8 to noisy quadratic data. It repeats the entire training procedure and evaluates conditional prediction risk on a fixed, fine midpoint grid. The evaluation uses the known simulation mean and adds the noise variance; it is an approximation to the integral over the specified uniform feature distribution. The true mean is available here only because this is a controlled simulation.

Inspect average training error, target risk, and Monte Carlo standard errors. The quadratic model matches the mean specification; the flexible model may fit noise. The experiment does not claim that one degree is universally optimal.

### Download lab

```
# Topics of Statistical Learning
# Lab 01: Training error and future prediction risk
# Base R only. Run with: Rscript 01-statistical-targets.R
set.seed(2026102101)

n_train <- 50L
n_rep <- 200L
degrees <- c(1L, 2L, 8L)
noise_sd <- 0.7
mean_function <- function(x) 1 + 2 * x + 0.5 * x^2
evaluation_x <- -1 + (seq_len(801L) - 0.5) * 2 / 801
evaluation_data <- data.frame(x = evaluation_x)
evaluation_mean <- mean_function(evaluation_x)

results <- array(NA_real_, dim = c(n_rep, length(degrees), 2L),
  dimnames = list(NULL, paste0("degree_", degrees),
    c("training_MSE", "target_risk")))
for (b in seq_len(n_rep)) {
  training <- data.frame(x = runif(n_train, -1, 1))
  training$y <- mean_function(training$x) + rnorm(n_train, sd = noise_sd)
  for (j in seq_along(degrees)) {
    degree <- degrees[j]
    fit <- lm(y ~ poly(x, degree = degree), data = training)
    train_prediction <- fitted(fit)
    test_prediction <- predict(fit, newdata = evaluation_data)
    results[b, j, "training_MSE"] <- mean((training$y - train_prediction)^2)
    # Midpoint approximation to  $E_X[(\hat{g}(X) - m(X))^2] + \sigma^2$ .
    results[b, j, "target_risk"] <-
      mean((test_prediction - evaluation_mean)^2) + noise_sd^2
  }
}

summary_table <- do.call(rbind, lapply(seq_along(degrees), function(j) {
  train <- results[, j, "training_MSE"]
  risk <- results[, j, "target_risk"]
  difference <- risk - train
  data.frame(degree = degrees[j], mean_training_MSE = mean(train),
    mean_target_risk = mean(risk), risk_MCSE = sd(risk) / sqrt(n_rep),
    mean_optimism = mean(difference),
    optimism_MCSE = sd(difference) / sqrt(n_rep))
}))

cat("Independent training replications:", n_rep, "\n")
cat("Training observations per replication:", n_train, "\n")
cat("Irreducible variance:", noise_sd^2, "\n")
cat("Target: X uniform on [-1, 1]; squared prediction error.\n\n")
print(summary_table, digits = 5, row.names = FALSE)
```

(continues on next page)

(continued from previous page)

```

cat("\nTarget risks use the known simulation mean and a midpoint integration grid.\n")
cat("MCSE describes uncertainty across repeated training datasets.\n")
cat("It is not a confidence interval for an individual fitted model's risk.\n")

figure_dir <- Sys.getenv("COURSE_FIGURE_DIR", unset = "")
if (nzchar(figure_dir)) {
  dir.create(figure_dir, recursive = TRUE, showWarnings = FALSE)
  png(file.path(figure_dir, "01-statistical-targets.png"), width = 1500,
      height = 700, res = 140, type = "cairo")
  par(mfrow = c(1, 2), mar = c(4.5, 4.5, 3.2, 1.2))
  means <- rbind(summary_table$mean_training_MSE, summary_table$mean_target_risk)
  positions <- barplot(means, beside = TRUE, col = c("#AFC3D4", "#1B4965"),
                      names.arg = paste("Degree", degrees), ylim = c(0, 1.1),
                      ylab = "Mean squared loss", main = "Training error and target risk")
  arrows(positions[2, ], summary_table$mean_target_risk - summary_table$risk_MCSE,
         positions[2, ], summary_table$mean_target_risk + summary_table$risk_MCSE,
         angle = 90, code = 3, length = 0.04)
  abline(h = noise_sd^2, lty = 2, col = "#B44A3D")
  legend("topleft", c("Training error", "Target risk (+/- 1 MCSE)", "Noise variance"),
        fill = c("#AFC3D4", "#1B4965", NA), border = NA,
        lty = c(NA, NA, 2), col = c(NA, NA, "#B44A3D"), bty = "n", cex = 0.8)
  boxplot(lapply(seq_along(degrees), function(j) results[, j, "target_risk"])),
        names = paste("Degree", degrees), log = "y", col = "#E5EDF3",
        ylab = "Target risk (log scale)", main = "Variation across training samples")
  abline(h = noise_sd^2, lty = 2, col = "#B44A3D")
  invisible(dev.off())
}

stopifnot(all(is.finite(results)),
          all(results[, , "target_risk"] >= noise_sd^2 - 1e-12))

```

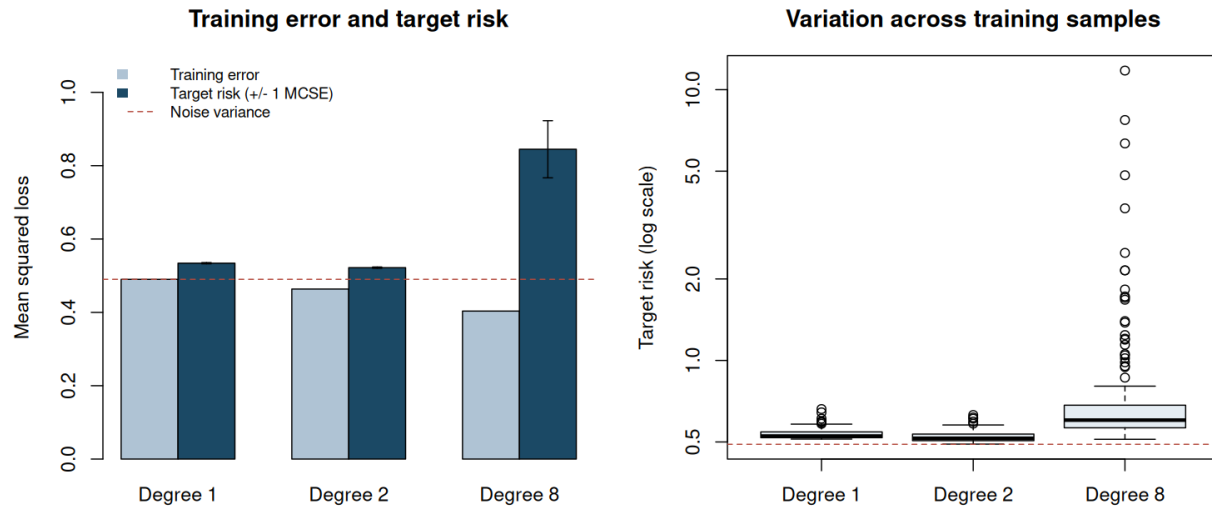
## Executed laboratory output

Independent training replications: 200  
 Training observations per replication: 50  
 Irreducible variance: 0.49  
 Target: X uniform on  $[-1, 1]$ ; squared prediction error.

| degree | mean_training_MSE | mean_target_risk | risk_MCSE | mean_optimism |
|--------|-------------------|------------------|-----------|---------------|
| 1      | 0.48989           | 0.53407          | 0.0016039 | 0.044178      |
| 2      | 0.46359           | 0.52189          | 0.0016984 | 0.058303      |
| 8      | 0.40349           | 0.84472          | 0.0778877 | 0.441230      |

optimism\_MCSE  
 0.0073334  
 0.0070737  
 0.0780512

Target risks use the known simulation mean and a midpoint integration grid.  
 MCSE describes uncertainty across repeated training datasets.  
 It is not a confidence interval for an individual fitted model's risk.



## Exercises

1. Suppose a false positive costs  $c_{FP}$  and a false negative costs  $c_{FN}$ , both positive. Derive the optimal decision threshold for  $\eta(x) = P(Y = 1 | X = x)$ .
2. Explain why the test-risk equality above can fail when several candidate models are compared using the test sample and the best-looking one is reported.
3. Modify the laboratory to increase the training sample size. Which parts of the squared-error decomposition should change? Which part should not?
4. Each of 100 prompts has 20 ratings. What additional information is needed before treating 2,000 rows as 2,000 independent observations for uncertainty assessment?

### **Worked solutions**

1. Predicting one has conditional expected loss  $c_{FP}(1 - \eta)$ , while predicting zero has loss  $c_{FN}\eta$ . Predict one when  $\eta > c_{FP}/(c_{FP} + c_{FN})$ ; either decision is optimal at equality.
2. The selected model now depends on test outcomes. Conditioning only on the original training data no longer makes the reported rule independent of the test sample. A separate final sample, or an evaluation procedure that accounts for this selection, is needed for the original interpretation.
3. More training observations can reduce error in estimating the mean function. They do not change the data-generating conditional variance. A misspecified linear model retains approximation error even with arbitrarily many observations; the experiment's irreducible variance remains  $0.7^2$ .
4. We need the dependence structure, how annotators are shared, and whether

the target is a new rating, prompt, annotator, or some combination. Prompt-level dependence can invalidate row-wise uncertainty estimates; crossed prompt-annotator dependence may require more than a one-way resampling scheme.

## Primary readings

- Shalev-Shwartz and Ben-David, *Understanding Machine Learning: From Theory to Algorithms*: selected sections on learning models, empirical risk, and generalization.
- Bartlett, Jordan, and McAuliffe (2006), “Convexity, Classification, and Risk Bounds”: an advanced connection between surrogate losses and classification risk.

**Next:** Chapter 2 makes the observation mechanism explicit when a recorded label is incomplete, ambiguous, or corrupted.

## 2. What weak supervision tells us

### **Core idea - The observation mechanism supplies the supervision**

The recorded label is a measurement of the target, not necessarily the target itself. Write the map from the latent class to the observed information. Identification asks whether that map can be inverted; stability asks how much uncertainty the inverse amplifies.

*Topics of Statistical Learning · Week 2*

## Two ways aggregate information identifies risk

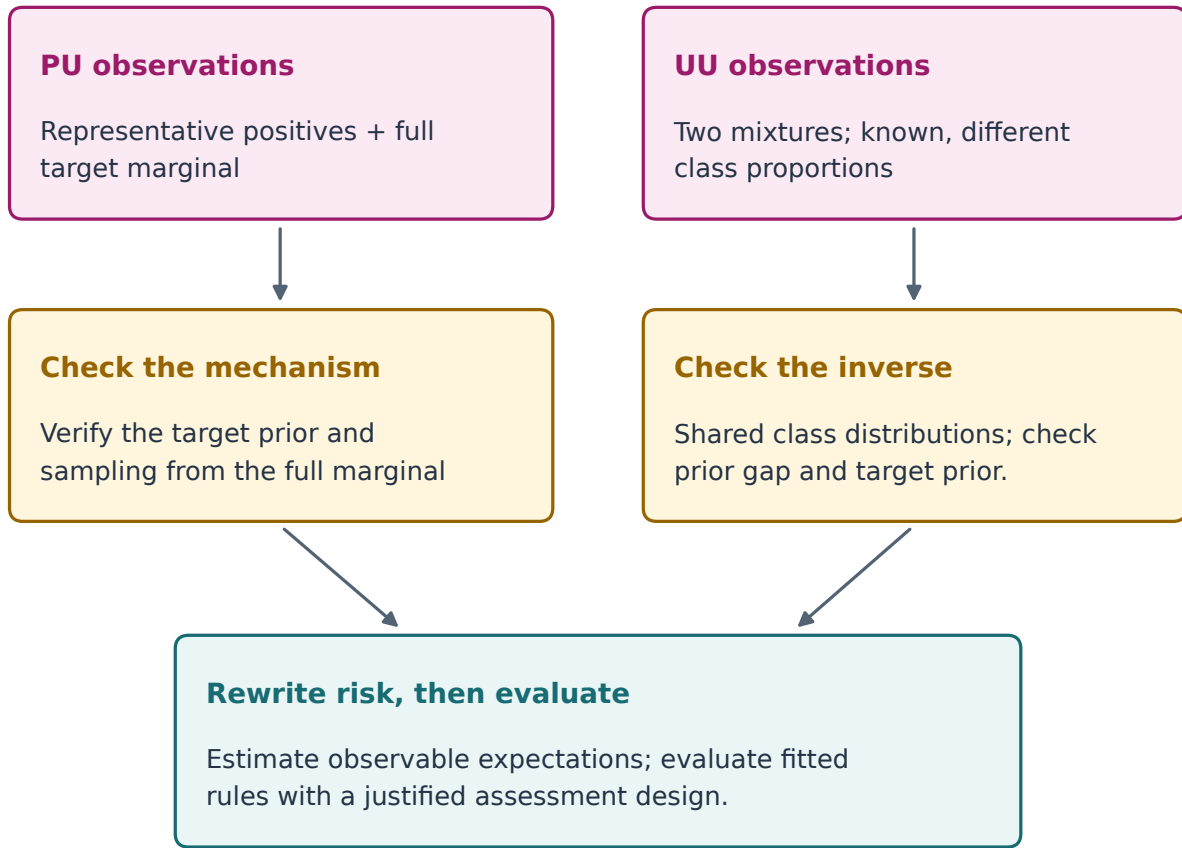


Fig. 2: PU uses representative positives and the full marginal. UU uses two mixtures with known different priors and shared class distributions. Both routes need a justified evaluation design.

## Record the observation, not an imagined complete label

Weak supervision describes several observation settings, not one interchangeable data type. A missing label, a wrong label, a candidate set, and a label attached to a group contain different information. A useful analysis writes down the relationship between the scientific target and the recorded information before choosing an algorithm.

Let  $Y$  be the target class and let  $O$  denote the recorded supervision. A mechanism  $P(O | X, Y)$  connects them. The observed-data distribution is

$$P(X, O) = \sum_y P(O | X, Y = y)P(X, Y = y).$$

Learning from  $(X, O)$  requires enough information about this mechanism and the la-

tent distribution to identify the intended target. Specifying a flexible neural network or a more powerful optimizer does not supply that missing information. This chapter develops class-conditional label noise as a worked mechanism and uses it to compare other forms of weak supervision.

## A map of observation settings

| Setting               | What is observed?                                | An essential modelling question                                                       |
|-----------------------|--------------------------------------------------|---------------------------------------------------------------------------------------|
| Positive-unlabelled   | Some confirmed positives and unlabelled features | How are positives selected, and what population does the unlabelled sample represent? |
| Noisy labels          | A possibly incorrect label $\tilde{Y}$           | Which classes can be confused, and does confusion depend on $X$ ?                     |
| Partial labels        | A candidate set $C$                              | Is $Y \in C$ guaranteed, and how is the set generated?                                |
| Complementary labels  | A class $\bar{Y}$ that is excluded               | How is the excluded class chosen?                                                     |
| Bag-level labels      | A label for a collection of observations         | How does the group label depend on instance labels?                                   |
| Multiple weak sources | Several labels or abstentions                    | Which sources share errors or information?                                            |

The terminology does not determine the assumptions. For example, a bag labelled positive might mean “at least one positive instance,” “a majority are positive,” or an aggregate outcome with a different interpretation. These lead to different likelihoods and targets. Likewise, semi-supervised learning with labelled examples from both classes is a different sampling problem from having only labelled positives.

## Class-conditional label noise

Let  $Y \in \{+1, -1\}$  and suppose the recorded label  $\tilde{Y}$  can be flipped. Define

$$\rho_+ = P(\tilde{Y} = -1 \mid Y = +1), \quad \rho_- = P(\tilde{Y} = +1 \mid Y = -1).$$

The **class-conditional** assumption means that these probabilities do not additionally depend on  $X$  once  $Y$  is given. With rows indexed by the true class and columns by the recorded class, the transition matrix is

$$T = \begin{pmatrix} 1 - \rho_+ & \rho_+ \\ \rho_- & 1 - \rho_- \end{pmatrix}.$$

Writing  $\eta(x) = P(Y = +1 \mid X = x)$  gives

$$\tilde{\eta}(x) = P(\tilde{Y} = +1 \mid X = x) = \rho_- + (1 - \rho_+ - \rho_-)\eta(x).$$

If the noise rates are known and  $d = 1 - \rho_+ - \rho_- \neq 0$ , the relation can be inverted. The usual informative-noise setting takes  $d > 0$ . When  $d = 0$ , the recorded-label probability is constant in  $\eta$ , so this observation cannot identify the class posterior. When  $|d|$  is small, inversion amplifies estimation errors. Unknown transition probabilities require additional assumptions or information; observing noisy labels alone does not automatically identify them.

### **i Worked example 2.1 — Noise can move the classification threshold**

Take  $\rho_+ = 0.2$  and  $\rho_- = 0.1$ . Then  $\tilde{\eta}(x) = 0.1 + 0.7\eta(x)$ . Thresholding the recorded-label probability at 0.5 is equivalent to thresholding the true posterior at  $4/7 \approx 0.5714$ .

For an observation with  $\eta(x) = 0.55$ , the true equal-cost Bayes decision is positive. Its recorded-label probability is only 0.485, so a well-fitted model for the noisy labels, used with the usual 0.5 threshold, makes the opposite decision. Accurate prediction of the observed label and accurate prediction of the target label are different tasks.

With known noise rates, inversion recovers  $\eta = (\tilde{\eta} - 0.1)/0.7$ . In finite samples, a fitted  $\tilde{\eta}$  can produce values outside  $[0, 1]$  after inversion; clipping is a modelling or estimation choice, not an exact unbiased transformation.

## Correcting the loss instead of the posterior

For a fixed feature value and prediction, collect the clean losses in  $L = (\ell_+, \ell_-)^\top$ , where  $\ell_\pm = \ell(g(X), \pm 1)$ . We seek recorded-label losses  $\tilde{L}$  satisfying  $T\tilde{L} = L$ . Solving this system gives

$$\tilde{\ell}_+ = \frac{(1 - \rho_-)\ell_+ - \rho_+\ell_-}{d}, \quad \tilde{\ell}_- = \frac{(1 - \rho_+)\ell_- - \rho_-\ell_+}{d}.$$

Under the stated mechanism,

$$E\{\tilde{\ell}(g(X), \tilde{Y}) \mid X, Y\} = \ell(g(X), Y).$$

Taking expectations therefore recovers the clean population risk. This is an expectation identity for a fixed prediction rule. It does not say that each corrected observation is a plausible individual loss, or that unrestricted minimization cannot overfit.

For the rates in Example 2.1, consider a rule predicting positive. Its clean losses are  $(0, 1)$ . The corrected losses are  $-2/7$  for an observed positive and  $8/7$  for an observed negative. Averaging over corruption gives zero when the true class is positive and one when it is negative. Negative contributions are the algebraic cost of undoing a mixture, a phenomenon that will reappear in PU learning.

## Candidate sets and complementary labels

A partial label supplies a set of possible classes. If every observation has candidate set  $\{A, B\}$ , that fact alone cannot distinguish  $A$  from  $B$ . A method must obtain information

from the features, variation in candidate sets, restrictions on their generation, or additional data.

A complementary label excludes one class. In binary classification, a correct excluded label determines the other class exactly. To illustrate genuinely weaker information, use at least three classes.

### **i Worked example 2.2 — A three-class complementary label**

Suppose  $Y \in \{A, B, C\}$  and  $\bar{Y}$  is drawn uniformly from the two incorrect classes. For arbitrary class-specific losses,

$$\bar{\ell}(g, \bar{Y}) = \sum_{k \in \{A, B, C\}} \ell(g, k) - 2\ell(g, \bar{Y})$$

has conditional expectation  $\ell(g, Y)$ . The expectation subtracts the sum of the two incorrect-class losses from the sum over all classes.

Under zero-one loss, a rule predicting  $A$  has corrected loss  $2\mathbf{1}\{\bar{Y} = A\}$ . If the true class is  $B$ , the excluded label is  $A$  or  $C$  with equal probability, so the expected corrected loss is one. If the true class is  $A$ , the excluded label is never  $A$ , so the expected corrected loss is zero.

Uniform complementary-label sampling is doing essential work. If a label provider systematically favours one incorrect class, this formula need not remain unbiased.

## **Several weak sources and the evaluation sample**

Multiple sources can supply useful supervision, but source count is not the same as independent information. Two rules based on the same keyword may repeat the same mistake. Abstention can also depend on difficulty or class. A probabilistic aggregation model must state enough assumptions to identify source reliability and its target labels.

Keep the evaluation problem explicit. A clean, independently collected test sample can assess prediction of  $Y$ . A noisy test label assesses prediction of  $\tilde{Y}$  unless a justified correction is applied. Comparing algorithms only against the same weak source used for training may reward reproduction of that source's bias.

## **Quantify what an inverse does to error**

The noisy-label posterior relation has slope  $d = 1 - \rho_+ - \rho_-$ . If the rates are treated as known, an estimation error  $\epsilon$  in the observed posterior becomes  $\epsilon/d$  in the recovered posterior. For example, with  $d = 0.2$ , an error of 0.02 becomes 0.10 before any clipping.

This calculation separates two questions: whether the posterior is identified at all, and whether the available sample estimates it precisely enough to be useful. The

same distinction appears in the UU denominator  $a - b$ . More observations can reduce estimation noise, but cannot identify an unrestricted model when the relevant denominator is exactly zero.

### Boundary - validation of the recorded label is not enough

A held-out noisy label can evaluate prediction of that noisy label. Evaluating the clean target requires trusted target labels or a justified correction with its own assumptions. Do not select a correction merely because it improves an uncorrected observed-label score.

## Laboratory — Correcting a noisy risk curve

The lab simulates a logistic class posterior and asymmetric class-conditional corruption with known rates. It evaluates a grid of threshold classifiers using noisy empirical risk and corrected empirical risk. A large independent labelled sample assesses the selected rules. The latent training labels are used only for an explicitly labelled oracle comparison.

The grid is deliberately simple: the aim is to inspect an observation mechanism rather than hide it inside an optimization routine. The script also checks the transition-matrix inversion and the complementary-label identity directly.

Download lab

```
# Topics of Statistical Learning
# Lab 02: Class-conditional noise and corrected empirical risk
# Base R only. Run with: Rscript 02-weak-supervision.R
set.seed(2026102102)

n_train <- 2000L
n_test <- 100000L
rho_positive <- 0.30
rho_negative <- 0.10
d <- 1 - rho_positive - rho_negative
stopifnot(d > 0)
transition <- rbind(c(1 - rho_positive, rho_positive),
                   c(rho_negative, 1 - rho_negative))
dimnames(transition) <- list(true = c("positive", "negative"),
                             recorded = c("positive", "negative"))

x <- rnorm(n_train)
y <- rbinom(n_train, 1L, plogis(1.2 * x))
flip_probability <- ifelse(y == 1L, rho_positive, rho_negative)
flipped <- runif(n_train) < flip_probability
y_recorded <- ifelse(flipped, 1L - y, y)

thresholds <- seq(-1.5, 1.5, length.out = 151L)
curves <- t(vapply(thresholds, function(threshold) {
  prediction <- as.integer(x > threshold)
```

(continues on next page)

(continued from previous page)

```

loss_positive <- 1 - prediction
loss_negative <- prediction
corrected_positive <- ((1 - rho_negative) * loss_positive -
  rho_positive * loss_negative) / d
corrected_negative <- ((1 - rho_positive) * loss_negative -
  rho_negative * loss_positive) / d
corrected <- ifelse(y_recorded == 1L, corrected_positive, corrected_negative)
c(noisy_risk = mean(prediction != y_recorded),
  corrected_risk = mean(corrected),
  oracle_clean_risk = mean(prediction != y))
}, numeric(3L)))

selected <- apply(curves, 2L, which.min)
x_test <- rnorm(n_test)
y_test <- rbinom(n_test, 1L, plogis(1.2 * x_test))
assessment <- do.call(rbind, lapply(seq_along(selected), function(j) {
  threshold <- thresholds[selected[j]]
  errors <- as.numeric((x_test > threshold) != y_test)
  data.frame(method = colnames(curves)[j], threshold = threshold,
    held_out_error = mean(errors),
    held_out_MCSE = sd(errors) / sqrt(n_test))
}))

cat("Known class-conditional transition matrix:\n")
print(transition)
cat("\nThresholds selected using each training criterion:\n")
print(assessment, digits = 5, row.names = FALSE)
cat("\nThe oracle clean criterion uses hidden training labels only as a benchmark.\n")
cat("Held-out labels are used for assessment, never threshold selection.\n")

# Identity check for the rule that always predicts positive.
clean_losses <- c(0, 1)
corrected_losses <- drop(solve(transition, clean_losses))
recovered <- drop(transition %*% corrected_losses)
cat("\nCorrected losses for a positive decision:\n")
print(data.frame(recorded_label = c("positive", "negative"),
  corrected_loss = corrected_losses, row.names = FALSE)
cat("Recovered expected losses conditional on the true class:\n")
print(recovered)

# Three-class example: uniform complementary labels and prediction of class A.
classes <- c("A", "B", "C")
complement_result <- vapply(classes, function(true_class) {
  possible_complements <- setdiff(classes, true_class)
  mean(2 * (possible_complements == "A"))
}, numeric(1L))
cat("\nComplementary-label corrected risk for a prediction of A:\n")
print(data.frame(true_class = classes,
  expected_corrected_loss = complement_result), row.names = FALSE)

figure_dir <- Sys.getenv("COURSE_FIGURE_DIR", unset = "")
if (nzchar(figure_dir)) {

```

(continues on next page)

(continued from previous page)

```

dir.create(file.path("02-weak-supervision.png"), width = 1500,
           height = 700, res = 140, type = "cairo")
par(mfrow = c(1, 2), mar = c(5.2, 4.5, 4.0, 1.2))
colours <- c("#B44A3D", "#1B4965", "#2D8063")
matplot(thresholds, curves, type = "l", lty = c(1, 1, 2), lwd = 2,
        col = colours, xlab = "Classification threshold", ylab = "Empirical risk",
        main = "Different recorded and target risks")
legend("top", c("Noisy labels", "Corrected target risk", "Clean-label oracle"),
      col = colours, lty = c(1, 1, 2), lwd = 2, bty = "n", cex = 0.8)
positions <- barplot(assessment$held_out_error, names.arg = c("Naive", "Corrected",
  ↪ "Oracle"),
                    col = colours, ylim = c(0, 0.45), ylab = "Clean-label error rate",
                    main = "Independent final assessment")
arrows(positions, assessment$held_out_error - 2 * assessment$held_out_MCSE,
       positions, assessment$held_out_error + 2 * assessment$held_out_MCSE,
       angle = 90, code = 3, length = 0.05)
mtext("Error bars show +/- 2 Monte Carlo SE", side = 3, line = 0.2, cex = 0.75)
invisible(dev.off())
}

stopifnot(max(abs(recovered - clean_losses)) < 1e-12,
          identical(as.numeric(complement_result), c(0, 1, 1)),
          all(is.finite(curves)))

```

## Executed laboratory output

Known class-conditional transition matrix:

|          | recorded |          |
|----------|----------|----------|
| true     | positive | negative |
| positive | 0.7      | 0.3      |
| negative | 0.1      | 0.9      |

Thresholds selected using each training criterion:

|  | method            | threshold | held_out_error | held_out_MCSE |
|--|-------------------|-----------|----------------|---------------|
|  | noisy_risk        | 0.94      | 0.37952        | 0.0015346     |
|  | corrected_risk    | -0.24     | 0.30695        | 0.0014585     |
|  | oracle_clean_risk | -0.14     | 0.30293        | 0.0014532     |

The oracle clean criterion uses hidden training labels only as a benchmark.  
Held-out labels are used for assessment, never threshold selection.

Corrected losses for a positive decision:

| recorded_label | corrected_loss |
|----------------|----------------|
| positive       | -0.500000      |
| negative       | 1.166667       |

Recovered expected losses conditional on the true class:

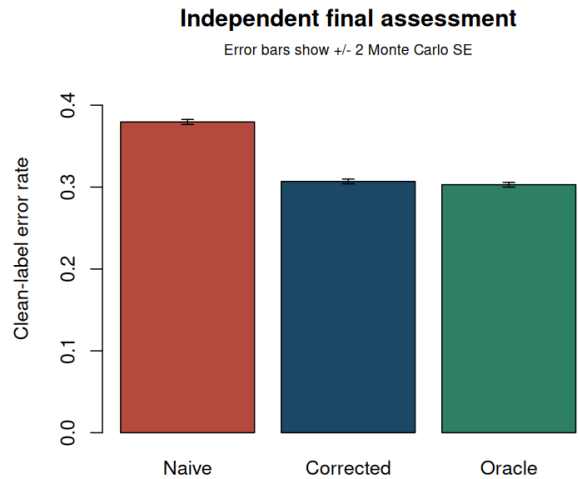
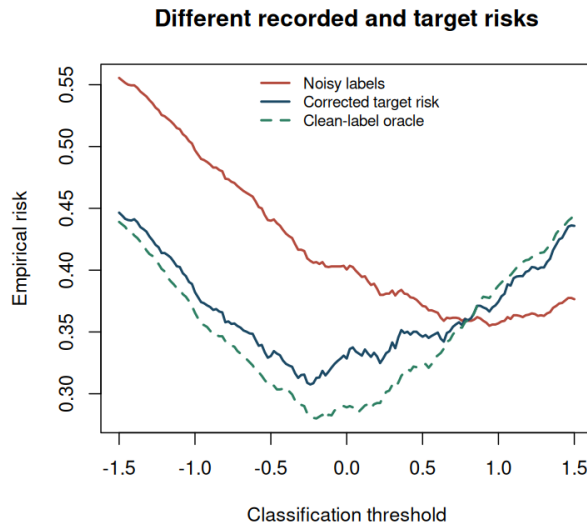
| positive | negative |
|----------|----------|
| 0        | 1        |

(continues on next page)

(continued from previous page)

Complementary-label corrected risk for a prediction of A:

| true_class | expected_corrected_loss |
|------------|-------------------------|
| A          | 0                       |
| B          | 1                       |
| C          | 1                       |



## Exercises

1. Derive  $\tilde{\eta}(x)$  by conditioning on  $Y$ . What happens if both noise rates equal  $1/2$ ?
2. Verify the two corrected losses by multiplying  $T\tilde{L}$ . Why can their variance increase as  $d$  approaches zero?
3. A three-class annotator always reports the first incorrect class alphabetically. Does the uniform complementary-label correction remain justified?
4. Modify the laboratory so the positive flip probability depends on  $X$ . Explain why using one global positive flip rate is not generally sufficient.

### **Worked solutions**

1. The law of total probability gives  $(1 - \rho_+)\eta + \rho_-(1 - \eta)$ . If both rates are  $1/2$ , the observed label is a fair coin independent of the true class under this mechanism.
2. Substitution cancels the cross-loss terms and returns  $(\ell_+, \ell_-)$ . The correction coefficients contain  $1/d$ ; fluctuations in empirical averages can therefore be amplified. Particular losses can have cancellations, so this is a stability warning rather than a claim of divergence in every possible special case.
3. No. The conditional probabilities of the excluded classes are no longer uniform.

The observation mechanism must be modelled with its actual transition probabilities, and its ability to identify the target must be checked.

4. The relevant identity is conditional on both  $X$  and  $Y$ . A class-average flip rate does not usually preserve it when losses and corruption probabilities vary together with features. Use known feature-dependent transitions, impose additional structure, or describe the resulting misspecification; a larger sample alone does not restore the original identity.

## Primary readings

- Natarajan et al. (2013), “Learning with Noisy Labels”: class-conditional corruption and loss correction.
- Ishida et al. (2019), “Complementary-Label Learning for Arbitrary Losses and Models”: risk rewriting for excluded labels.
- Ratner et al. (2016), “Data Programming: Creating Large Training Sets, Quickly”: learning from several weak labelling functions.

**Next:** Chapters 3 and 4 study the missing-label mechanism specific to positive-unlabelled learning.

# 3. Learning from positives and unlabelled observations

## Core idea - Subtract contamination, not observations

An unlabelled sample from the full feature marginal contains positive and negative cases. The PU identity removes the positive contribution to the negative-class loss **in expectation**. It does not identify which individual unlabelled rows are negative, and it depends on representative positives and the specified prior.

*Topics of Statistical Learning · Week 3*

## What is missing?

In positive-unlabelled learning, some observations are known to belong to a positive class, while an unlabelled collection contains both classes. Examples include confirmed relevant documents and a broader document collection, or confirmed examples of a phenomenon together with background observations. An unlabelled observation has an unknown class, not a known negative class.

The main task is to express a target classification risk using distributions from which data are actually available. The derivation is short, but its sampling assumptions are consequential. We first use an explicit two-sample model, then distinguish it from a single dataset in which only some positive labels are revealed.

## The two-sample model

Let  $Y \in \{+1, -1\}$ , let  $\pi = P(Y = +1) \in (0, 1)$  be the target class prior, and write

$$P_+ = P(X \mid Y = +1), \quad P_- = P(X \mid Y = -1).$$

The target feature marginal is

$$P_U = \pi P_+ + (1 - \pi) P_-.$$

The introductory sampling scheme provides independent observations  $X_1^+, \dots, X_{n_+}^+ \sim P_+$  and  $X_1^U, \dots, X_{n_U}^U \sim P_U$ . These are representative samples from the specified positive distribution and the full target feature marginal. For now,  $\pi$  is known.

The ratio  $n_+/(n_+ + n_U)$  is controlled by data collection and is not an estimator of the target prevalence in this two-sample design. We may deliberately collect many confirmed positives while the target population contains relatively few positives.

## Deriving the PU risk identity

For a score  $g$  and loss  $\ell$ , define  $\ell_{\pm}(X) = \ell(g(X), \pm 1)$ . The ordinary population risk is

$$R(g) = \pi E_+ \ell_+ + (1 - \pi) E_- \ell_-.$$

The unavailable negative expectation can be removed by applying the mixture relation to  $\ell_-$ :

$$E_U \ell_- = \pi E_+ \ell_- + (1 - \pi) E_- \ell_-.$$

Substituting gives

$$R(g) = \pi E_+ \ell_+ + E_U \ell_- - \pi E_+ \ell_-.$$

Each expectation on the right can be estimated from the observed samples. The subtraction removes the positive-class contribution that would otherwise be counted when unlabelled observations are temporarily evaluated as negative.

The identity applies to a loss with finite relevant expectations. It is not restricted to zero-one loss. Its empirical version is

$$\hat{R}_{\text{PU}}(g) = \frac{\pi}{n_+} \sum_i \ell(g(X_i^+), +1) + \frac{1}{n_U} \sum_j \ell(g(X_j^U), -1) - \frac{\pi}{n_+} \sum_i \ell(g(X_i^+), -1).$$

For a fixed  $g$  and the stated known-prior sampling model, this estimator is unbiased. A model fitted by minimizing it still needs generalization analysis or independent evaluation. The fitted model is not fixed with respect to its training sample.

### **Worked example 3.1 — Removing positive contamination**

Suppose  $\pi = 0.3$ , sensitivity is 0.8, and the false-positive rate is 0.1. Under zero-one loss,

$$E_+ \ell_+ = 0.2, \quad E_+ \ell_- = 0.8,$$

while  $E_U \ell_- = P\{g(X) = +1\} = 0.3(0.8) + 0.7(0.1) = 0.31$ . The PU identity yields

$$R(g) = 0.3(0.2) + 0.31 - 0.3(0.8) = 0.13.$$

The direct fully labelled calculation is  $0.3(0.2) + 0.7(0.1) = 0.13$ . The observed marginal predicted-positive rate 0.31 includes true positives as well as false positives. It is not the negative-class error contribution; subtracting 0.24 leaves the required contribution 0.07.

## A logistic objective that can be implemented directly

Take  $g_\beta(x) = \beta_0 + x^\top \beta_1$  and logistic losses

$$\ell(g, +1) = \log(1 + e^{-g}), \quad \ell(g, -1) = \log(1 + e^g).$$

Since  $\log(1 + e^{-g}) - \log(1 + e^g) = -g$ , the empirical PU risk simplifies to

$$\hat{R}_{\text{PU}}(\beta) = \frac{1}{n_U} \sum_j \log(1 + e^{g_\beta(X_j^U)}) - \frac{\pi}{n_+} \sum_i g_\beta(X_i^+).$$

For this linear score, the objective is convex. A positive quadratic penalty can stabilize estimation and ensure a finite minimizer; the penalty is a chosen regularizer rather than an unbiased component of the target risk. More flexible score classes introduce different computational and statistical issues. Chapter 4 examines the negative empirical risks that can arise from the subtraction.

Implement the softplus function stably as  $\max(z, 0) + \log(1 + e^{-|z|})$ , rather than forming a potentially overflowing  $e^z$ .

## A different design: labels revealed within one sample

Now suppose observations are sampled from the target population and  $S = 1$  indicates that a positive label is recorded. We assume  $P(S = 1 \mid Y = -1, X) = 0$ . Under **selected completely at random**, or SCAR,

$$P(S = 1 \mid Y = +1, X = x) = c$$

for a constant  $c > 0$ . Then  $P(S = 1 \mid X = x) = cP(Y = +1 \mid X = x)$ , and the labelled positives have distribution  $P_+$ . However, the observations with  $S = 0$  are not distributed as the full feature marginal.

### Worked example 3.2 — Unlabelled rows are not always the marginal sample

Let  $\pi = 0.3$  and  $c = 0.4$ . The probability of an observed positive label is 0.12. Among observations without a recorded label,

$$P(Y = +1 \mid S = 0) = \frac{\pi(1 - c)}{1 - \pi c} = \frac{0.18}{0.88} \approx 0.2045.$$

Using the classifier from Example 3.1, the predicted-positive rate in this subset is  $0.2045(0.8) + 0.7955(0.1) \approx 0.2432$ . If we incorrectly substitute that rate for  $E_U \ell_-$ , the PU expression becomes about 0.0632, rather than the target risk 0.13.

All feature vectors in the original population sample can represent the marginal distribution. The labelled-positive and full-feature averages then share observations, so their dependence matters for variance calculations. Alternatively, use a sampling model and risk expression designed specifically for the observed-label partition.

## What the assumptions do not supply

SCAR makes the recorded positives representative of the positive class. It does not by itself identify an unknown  $c$  or  $\pi$ . In the observed posterior  $P(S = 1 \mid X) = c\eta(X)$ , different pairs  $(c, \eta)$  can produce the same product without additional restrictions. Likewise, a positive distribution and its mixture with an unrestricted negative distribution need not uniquely determine an unknown mixture proportion.

Known prevalence, a separately collected validation sample, or additional identifiable structure can supply information. Any class-prior estimator introduced later must state its own assumptions. Treating a supplied prior as known is appropriate for an introductory simulation, provided the simulation also examines sensitivity to an incorrect value.

## Plan the positive and unlabelled sample sizes

For a fixed rule and known  $\pi$ , independent positive and marginal samples give

$$\text{Var}(\hat{R}_{\text{PU}}) = \frac{\pi^2}{n_+} \text{Var}_+(\ell_+ - \ell_-) + \frac{1}{n_U} \text{Var}_U(\ell_-).$$

The same positive observations appear in both positive-class averages, so their covariance is already included in  $\text{Var}_+(\ell_+ - \ell_-)$ . Treating these two averages as independent generally gives the wrong uncertainty calculation. Once the second term is small, collecting more unlabelled cases does little to reduce uncertainty dominated by the positive sample.

### Analysis check - distinguish the two sampling designs

Use this variance expression for independent positive and full-marginal samples and a fixed rule. In a single partially labelled population sample, averages may share observations. For a fitted rule, training, tuning, and uncertainty need additional analysis; the displayed calculation is not a fitted-model confidence interval.

## Laboratory — A linear PU classifier

The lab fits a regularized logistic PU objective to independent positive and marginal samples. It compares the resulting rule with a naive positive-versus-unlabelled fit, a fully labelled oracle benchmark, and the known Bayes score for the simulation. All evaluation uses a new labelled test sample. The final section verifies the SCAR subset calculation using a separate population sample.

Download lab

```
# Topics of Statistical Learning
# Lab 03: Logistic PU learning under an explicit two-sample design
```

(continues on next page)

(continued from previous page)

```

# Base R only. Run with: Rscript 03-pu-learning.R
set.seed(2026102103)

prior <- 0.30
n_positive <- 250L
n_unlabelled <- 1500L
n_test <- 50000L
lambda <- 0.01
softplus <- function(z) pmax(z, 0) + log1p(exp(-abs(z)))
design <- function(x) cbind(intercept = 1, x = x)
sample_population <- function(n) {
  y <- rbinom(n, 1L, prior)
  list(x = rnorm(n, mean = ifelse(y == 1L, 1, -1)), y = y)
}

x_positive <- rnorm(n_positive, mean = 1)
unlabelled <- sample_population(n_unlabelled)
z_positive <- design(x_positive)
z_unlabelled <- design(unlabelled$x)

pu_objective <- function(beta) {
  gu <- drop(z_unlabelled %*% beta)
  gp <- drop(z_positive %*% beta)
  mean(softplus(gu)) - prior * mean(gp) + lambda * sum(beta^2)
}
pu_gradient <- function(beta) {
  gu <- drop(z_unlabelled %*% beta)
  drop(crossprod(z_unlabelled, plogis(gu))) / n_unlabelled -
    prior * colMeans(z_positive) + 2 * lambda * beta
}
pu_fit <- optim(c(0, 0), pu_objective, pu_gradient, method = "BFGS",
               control = list(reltol = 1e-10, maxit = 1000L))

fit_labelled <- function(z, y) {
  objective <- function(beta) {
    g <- drop(z %*% beta)
    mean(softplus(g) - y * g) + lambda * sum(beta^2)
  }
  gradient <- function(beta) {
    drop(crossprod(z, plogis(drop(z %*% beta)) - y)) / nrow(z) +
      2 * lambda * beta
  }
  optim(c(0, 0), objective, gradient, method = "BFGS",
        control = list(reltol = 1e-10, maxit = 1000L))
}

# Naive benchmark: recorded positives = 1, all marginal U observations = 0.
naive_fit <- fit_labelled(rbind(z_positive, z_unlabelled),
                         c(rep(1, n_positive), rep(0, n_unlabelled)))
# Oracle benchmark only: uses latent labels that a PU learner cannot observe.
oracle_fit <- fit_labelled(z_unlabelled, unlabelled$y)
beta_bayes <- c(qlogis(prior), 2)

```

(continues on next page)

(continued from previous page)

```

test <- sample_population(n_test)
z_test <- design(test$x)
coefficients <- rbind(naive = naive_fit$par, PU = pu_fit$par,
                      labelled_oracle = oracle_fit$par, Bayes = beta_bayes)
assessment <- do.call(rbind, lapply(seq_len(nrow(coefficients)), function(j) {
  score <- drop(z_test %*% coefficients[j, ])
  probability <- plogis(score)
  errors <- as.numeric((score > 0) != test$y)
  data.frame(method = rownames(coefficients)[j],
             intercept = coefficients[j, 1], slope = coefficients[j, 2],
             test_error = mean(errors), test_error_MCSE = sd(errors) / sqrt(n_test),
             test_log_loss = mean(softplus(score) - test$y * score),
             test_Brier = mean((probability - test$y)^2))
}))
cat("Target prior:", prior, "; sample sizes P =", n_positive,
    ", U =", n_unlabelled, "\n")
cat("Positive fraction in the assembled training data:",
    n_positive / (n_positive + n_unlabelled), "(not the target prior)\n\n")
print(assessment, digits = 5, row.names = FALSE)

# Separate single-sample SCAR experiment.
population <- sample_population(50000L)
labelling_probability <- 0.4
s <- population$y == 1L & runif(length(population$y)) < labelling_probability
scar_table <- data.frame(
  quantity = c("Observed-label fraction", "Positive fraction among S=0",
              "Feature mean in full population", "Feature mean among S=0"),
  empirical = c(mean(s), mean(population$y[!s]), mean(population$x),
               mean(population$x[!s])),
  theoretical = c(prior * labelling_probability,
                 prior * (1 - labelling_probability) /
                   (1 - prior * labelling_probability),
                 2 * prior - 1,
                 2 * prior * (1 - labelling_probability) /
                   (1 - prior * labelling_probability) - 1))
cat("\nSeparate SCAR population sample:\n")
print(scar_table, digits = 5, row.names = FALSE)
cat("\nThe S=0 subset is not the full target feature marginal.\n")

figure_dir <- Sys.getenv("COURSE_FIGURE_DIR", unset = "")
if (nzchar(figure_dir)) {
  dir.create(figure_dir, recursive = TRUE, showWarnings = FALSE)
  png(file.path(figure_dir, "03-pu-learning.png"), width = 1500,
      height = 700, res = 140, type = "cairo")
  par(mfrow = c(1, 2), mar = c(4.5, 4.5, 3.2, 1.2))
  grid <- seq(-4, 4, length.out = 401L)
  positive_density <- dnorm(grid, 1)
  marginal_density <- prior * positive_density + (1 - prior) * dnorm(grid, -1)
  matplot(grid, cbind(positive_density, marginal_density), type = "l", lwd = 2,
          lty = c(1, 2), col = c("#2D8063", "#1B4965"), xlab = "Feature x",
          ylab = "Density", main = "Observed sampling distributions")

```

(continues on next page)

(continued from previous page)

```

legend("topright", c("Representative positives", "Full marginal U"),
      lty = c(1, 2), col = c("#2D8063", "#1B4965"), lwd = 2, bty = "n", cex = 0.8)
probabilities <- plogis(design(grid) %*% t(coefficients))
matplot(grid, probabilities, type = "l", lwd = 2, lty = c(1, 1, 2, 3),
        col = c("#B44A3D", "#1B4965", "#2D8063", "#333333"),
        xlab = "Feature x", ylab = "Predicted positive probability",
        main = "Independent models and Bayes reference")
abline(h = 0.5, lty = 3, col = "#BBBBBB")
legend("topleft", c("Naive", "PU", "Labelled oracle", "Bayes"),
      lty = c(1, 1, 2, 3), col = c("#B44A3D", "#1B4965", "#2D8063", "#333333"),
      lwd = 2, bty = "n", cex = 0.8)
invisible(dev.off())
}

stopifnot(pu_fit$convergence == 0L, naive_fit$convergence == 0L,
          oracle_fit$convergence == 0L, all(is.finite(coefficients)))

```

## Executed laboratory output

Target prior: 0.3 ; sample sizes  $P = 250$  ,  $U = 1500$

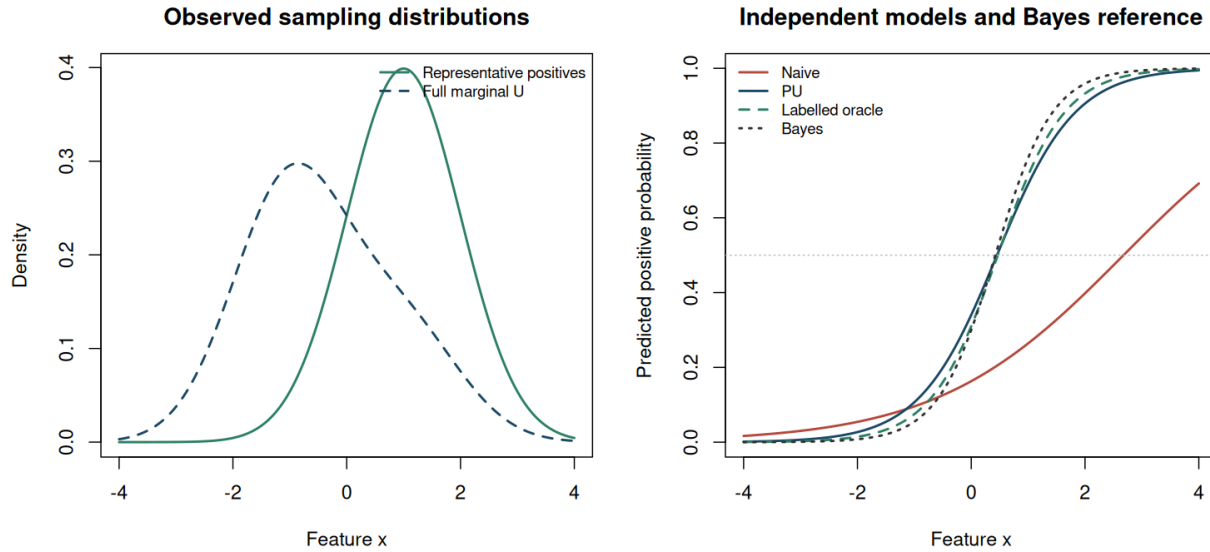
Positive fraction in the assembled training data: 0.1428571 (not the target prior)

|            | method          | intercept | slope   | test_error | test_error_MCSE | test_log_loss |
|------------|-----------------|-----------|---------|------------|-----------------|---------------|
|            | naive           | -1.63544  | 0.61106 | 0.29092    | 0.0020312       | 0.49859       |
|            | PU              | -0.66237  | 1.46182 | 0.13840    | 0.0015443       | 0.33206       |
|            | labelled_oracle | -0.80028  | 1.71235 | 0.13836    | 0.0015441       | 0.32360       |
|            | Bayes           | -0.84730  | 2.00000 | 0.13852    | 0.0015449       | 0.32073       |
| test_Brier |                 |           |         |            |                 |               |
|            |                 | 0.17327   |         |            |                 |               |
|            |                 | 0.10228   |         |            |                 |               |
|            |                 | 0.10066   |         |            |                 |               |
|            |                 | 0.10007   |         |            |                 |               |

Separate SCAR population sample:

|                                 | quantity | empirical | theoretical |
|---------------------------------|----------|-----------|-------------|
| Observed-label fraction         |          | 0.12218   | 0.12000     |
| Positive fraction among $S=0$   |          | 0.20510   | 0.20455     |
| Feature mean in full population |          | -0.39511  | -0.40000    |
| Feature mean among $S=0$        |          | -0.58891  | -0.59091    |

The  $S=0$  subset is not the full target feature marginal.



## Exercises

1. Derive the PU identity for squared probability loss as well as for zero-one loss. Which step changes?
2. In the two-sample design, why does increasing  $n_+$  while holding the target population fixed not increase  $\pi$ ?
3. Show the logistic simplification above and derive the gradient for a linear score with design vector  $z(x) = (1, x^T)^T$ .
4. In a single SCAR sample, derive  $P(Y = +1 \mid S = 0)$  and state when it equals  $\pi$ .

### **Worked solutions**

1. The mixture identity holds for any integrable function, so no algebraic step changes. Substitute the chosen loss for  $\ell_-$  and then for the full risk. The interpretation and optimization properties of the resulting objective do change with the loss.
2. The investigator controls the numbers sampled from  $P_+$  and  $P_U$ . The target prevalence belongs to the population represented by  $P_U$ , not to the artificially assembled training collection.
3. The difference of the two logistic losses is  $-g$ . With quadratic penalty  $\lambda \|\beta\|^2$ , the gradient is  $n_U^{-1} \sum_j \text{logit}^{-1}(z_j^T \beta) z_j - \pi n_+^{-1} \sum_i z_i + 2\lambda \beta$ .
4. Bayes' rule gives  $\pi(1 - c)/(1 - \pi c)$ . For  $0 < \pi < 1$ , equality with  $\pi$  requires  $c = 0$ , in which case no labels are observed. The degenerate all-negative or all-positive populations also require separate treatment. For a useful nondegenerate SCAR sample with  $c > 0$ , the unlabelled subset has a smaller positive proportion.

## Primary readings

- Elkan and Noto (2008), “Learning Classifiers from Only Positive and Unlabeled Data”: the observed-label mechanism and SCAR relationship.
- du Plessis, Niu, and Sugiyama (2015), “Convex Formulation for Learning from Positive and Unlabeled Data”: risk rewriting and convex formulations.
- Kiryo et al. (2017), “Positive-Unlabeled Learning with Non-Negative Risk Estimator”: the empirical-risk formulation and its practical limitations.

## 4. PU estimation, overfitting, and sensitivity

### Core idea - A correction does not repair the sampling model

Non-negative PU estimation controls one failure of empirical risk minimization. It does not supply a missing class prior or make selectively labelled positives representative. Separate optimization behaviour from the assumptions that make the risk identity correct.

*Topics of Statistical Learning · Week 4*

### An unbiased objective can still be difficult to minimize

The PU identity from Chapter 3 expresses risk as the sum of a positive-class contribution and a difference of two expectations. Population non-negativity does not prevent the empirical difference from becoming negative. A flexible score can exploit this difference by assigning very large positive scores to the observed positives without behaving similarly on future positives.

This chapter separates three questions: whether a risk estimator is unbiased for a fixed rule, whether minimizing it produces a useful fitted rule, and whether the sampling assumptions represent the application. These are related questions, but none is a substitute for the others.

### The non-negative risk correction

Write the empirical risk as  $\hat{A}(g) + \hat{B}(g)$ , where

$$\hat{A}(g) = \pi \hat{E}_+ \ell_+, \quad \hat{B}(g) = \hat{E}_U \ell_- - \pi \hat{E}_+ \ell_-.$$

For a non-negative loss, the population quantity  $B(g) = (1 - \pi)E_- \ell_-$  is non-negative. The **non-negative PU** estimator replaces its empirical version by

$$\hat{R}_{\text{nnPU}}(g) = \hat{A}(g) + \max\{0, \hat{B}(g)\}.$$

The correction applies to the estimated negative-class contribution, not simply to the total empirical risk. It prevents that contribution from rewarding the learner for becoming arbitrarily negative. The regularizer, score class, and numerical optimizer remain separate choices.

#### **Worked example 4.1 — Where the correction acts**

Use zero-one loss and a supplied prior  $\pi = 0.4$ . Suppose the positive sample has error rate 0.10, so the same observations evaluated as negative have loss 0.90. Suppose 30% of the marginal unlabelled sample is predicted positive. Then

$$\hat{A} = 0.4(0.10) = 0.04, \quad \hat{B} = 0.30 - 0.4(0.90) = -0.06.$$

The unbiased empirical expression is  $-0.02$ , whereas the non-negative expression is 0.04. Clipping the total expression at zero would instead give zero, a different estimator. None of these numbers reveals the unknown true risk of this particular fitted classifier without further assessment.

## **Bias, variance, and consistency are different properties**

For a fixed rule under the known-prior model,

$$\hat{R}_{\text{nnPU}} - \hat{R}_{\text{PU}} = (-\hat{B})_+ \geq 0.$$

Consequently, its expectation is at least the true risk whenever the original estimator is unbiased and the expectations exist. The correction generally introduces finite-sample upward bias. It can nevertheless improve mean squared error or the quality of the learned model. Such improvements depend on conditions; non-negative correction does not dominate every alternative in every finite sample.

For a fixed  $g$ , laws of large numbers for the relevant empirical averages and continuity of  $b \mapsto \max(0, b)$  imply consistency when  $B(g) \geq 0$ . A statement about a learned  $\hat{g}$  also needs control over the function class and optimization. Pointwise consistency for each fixed rule does not automatically justify minimization over an unrestricted set of rules.

Under independent positive and marginal samples, the variance of the unbiased fixed-rule estimator is

$$\text{Var}(\hat{R}_{\text{PU}}) = \frac{\pi^2}{n_+} \text{Var}_+(\ell_+ - \ell_-) + \frac{1}{n_U} \text{Var}_U(\ell_-).$$

The two positive-class averages use the same observations. Keeping their covariance through  $\ell_+ - \ell_-$  is essential. This formula changes if the samples overlap or the prior is estimated. For the non-negative estimator, the maximum creates an additional nonlinearity, so a normal approximation should not be assumed without checking its justification.

## Sensitivity to the supplied class prior

Suppose the target has true prior  $\pi$ , but the population correction uses  $\tilde{\pi}$ . For a fixed rule and unchanged positive and marginal distributions,

$$R_{\tilde{\pi}}^{\text{correction}}(g) - R(g) = (\tilde{\pi} - \pi)E_+(\ell_+ - \ell_-).$$

This is a misspecified correction for the same target distribution. It should not be confused with deliberately changing the target population's prevalence while also using the appropriate target distribution.

### **i** Worked example 4.2 — An incorrect prior can make the risk look smaller

A classifier has sensitivity and specificity both equal to 0.8, and the target prior is 0.3. Its true risk is 0.20. The relevant quantities are  $E_+\ell_+ = 0.2$ ,  $E_+\ell_- = 0.8$ , and  $E_U\ell_- = 0.38$ .

The correct expression is  $0.3(0.2) + 0.38 - 0.3(0.8) = 0.20$ . Substituting  $\tilde{\pi} = 0.45$  gives 0.11. The discrepancy is

$$(0.45 - 0.30)(0.20 - 0.80) = -0.09.$$

The negative-class contribution is still positive in this example, so non-negative correction does not fix the prior error. A numerical safeguard and a correct observation model solve different problems.

## Feature-dependent positive selection

In a single partially labelled population sample, let

$$e(x) = P(S = 1 \mid Y = +1, X = x), \quad P(S = 1 \mid Y = -1, X = x) = 0.$$

If  $e(x)$  varies, the observed positives need not represent  $P_+$ . When the propensity is known and positive wherever positive observations can occur, a useful identity is

$$R(g) = E \left[ \ell_-(X) + \frac{S}{e(X)} \{ \ell_+(X) - \ell_-(X) \} \right].$$

At any zero-propensity point with zero positive-class probability, define  $S/e(X) = 0$ ; positive-support positivity rules out zero propensity where the positive class can occur. To verify the identity, condition on  $(X, Y)$ . The expectation of  $S/e(X)$  is one for positives and zero for negatives. The baseline negative loss is therefore replaced by the positive loss exactly when the latent class is positive.

Small propensities can amplify variability. An unknown propensity is not automatically identified from PU observations:  $P(S = 1 \mid X) = e(X)P(Y = +1 \mid X)$  can admit multiple factorizations. The feature-dependent-selection literature supplies additional assumptions and estimation methods, not a general exemption from identification questions. Regions with zero positive labelling probability require further information or restrictions.

## Designing a useful evaluation

A simulation should distinguish the learner's information from the evaluator's information. The learner receives the positive and marginal samples and a stated prior. The evaluator can use latent classes on a new sample because the simulation generated them. A fully labelled training method is an oracle benchmark, not an available PU competitor.

Vary one consequential assumption at a time: positive sample size, prior accuracy, positive-selection bias, or score flexibility. Report prediction error and an appropriate score loss, not only the optimized training objective. Repeat the whole fitting procedure to assess variability across training datasets. With genuine PU validation data, use a justified risk estimator and its uncertainty rather than silently treating unknown labels as negatives.

## Separate three sources of failure in an experiment

A useful sensitivity study varies one component at a time. First hold the sampling model and prior correct while increasing model flexibility: this probes empirical overfitting. Next keep the rule class fixed and change the supplied prior: this probes prior sensitivity. Finally make positive selection depend on features: this challenges representativeness.

Keep the final labelled evaluation population fixed across those comparisons. Report prediction error, the estimated positive and negative risk components, and how often the non-negative correction activates. If changing a prior also changes the test population, the experiment confounds a method change with a target change.

### Boundary - a smaller objective can mean a different question

A low corrected training objective is not evidence that its supplied prior or selection model is right. Prior values and observation mechanisms require external information or explicit identifiable models. The correction addresses optimization behaviour within that model.

## Laboratory — Fixed-rule bias and fitted-rule behaviour

Part A repeatedly estimates the logistic risk of one fixed Bayes score. It reports bias, standard deviation, mean squared error, and how often the negative contribution is clipped. Part B fits flexible radial-basis scores with the unbiased and non-negative objectives, using the same parameter bounds, regularizer, and three deterministic starts. It reports optimizer status and independent test performance.

The finite parameter bounds and multiple starts are implementation choices. They do not certify a global minimum. The fixed-rule experiment and the fitted-rule comparison answer different questions and should be interpreted separately.

Download lab

```

# Topics of Statistical Learning
# Lab 04: nnPU bias, variability, and flexible-model fitting
# Base R only. Run with: Rscript 04-pu-practice.R
set.seed(2026102104)

prior <- 0.4
positive_mean <- 1.5
negative_mean <- -1.5
softplus <- function(z) pmax(z, 0) + log1p(exp(-abs(z)))
sample_population <- function(n) {
  y <- rbinom(n, 1L, prior)
  list(x = rnorm(n, ifelse(y == 1L, positive_mean, negative_mean)), y = y)
}
bayes_score <- function(x) qlogis(prior) + 3 * x
true_fixed_risk <- prior * integrate(function(x) {
  softplus(-bayes_score(x)) * dnorm(x, positive_mean)
}, -Inf, Inf)$value + (1 - prior) * integrate(function(x) {
  softplus(bayes_score(x)) * dnorm(x, negative_mean)
}, -Inf, Inf)$value

# Part A: the score is fixed before every sample is generated.
n_rep <- 1000L
fixed_estimates <- matrix(NA_real_, n_rep, 3L,
  dimnames = list(NULL, c("uPU", "nnPU", "negative_part")))
for (b in seq_len(n_rep)) {
  gp <- bayes_score(rnorm(18L, positive_mean))
  gu <- bayes_score(sample_population(60L)$x)
  positive_part <- prior * mean(softplus(-gp))
  negative_part <- mean(softplus(gu)) - prior * mean(softplus(gp))
  fixed_estimates[b, ] <- c(positive_part + negative_part,
    positive_part + max(0, negative_part), negative_part)
}
fixed_summary <- do.call(rbind, lapply(1:2, function(j) {
  value <- fixed_estimates[, j]
  data.frame(estimator = colnames(fixed_estimates)[j], mean = mean(value),
    bias = mean(value) - true_fixed_risk, SD = sd(value),
    bias_MCSE = sd(value) / sqrt(n_rep),
    MSE = mean((value - true_fixed_risk)^2))
}))
cat("PART A: fixed-rule logistic risk\n")
cat("Numerically integrated reference risk:", true_fixed_risk, "\n")
print(fixed_summary, digits = 5, row.names = FALSE)
cat("Fraction with negative estimated negative-class contribution:",
  mean(fixed_estimates[, "negative_part"] < 0), "\n")

# Part B: fit scores using only P and marginal U data.
x_positive <- rnorm(30L, positive_mean)
unlabelled <- sample_population(300L)
centres <- seq(-3.5, 4, length.out = 18L)
width <- 0.30
basis <- function(x) cbind(1, exp(-outer(x, centres, "-")^2 / (2 * width^2)))
zp <- basis(x_positive)

```

(continues on next page)

(continued from previous page)

```

zu <- basis(unlabelled$x)
ridge <- 1e-4

components <- function(beta) {
  gp <- drop(zp %*% beta)
  gu <- drop(zu %*% beta)
  c(positive = prior * mean(softplus(-gp)),
    negative = mean(softplus(gu)) - prior * mean(softplus(gp)))
}

fit_objective <- function(nonnegative) {
  objective <- function(beta) {
    parts <- components(beta)
    parts[1] + if (nonnegative) max(0, parts[2]) else parts[2]
  }
  penalized <- function(beta) objective(beta) + ridge * sum(beta^2)
  gradient <- function(beta) {
    pp <- plogis(drop(zp %*% beta))
    pu <- plogis(drop(zu %*% beta))
    grad_positive <- prior * drop(crossprod(zp, pp - 1)) / nrow(zp)
    grad_negative <- drop(crossprod(zu, pu)) / nrow(zu) -
      prior * drop(crossprod(zp, pp)) / nrow(zp)
    use_negative <- !nonnegative || components(beta)[2] > 0
    grad_positive + if (use_negative) grad_negative else rep(0, ncol(zp))
  }
  penalized_gradient <- function(beta) gradient(beta) + 2 * ridge * beta
  starts <- list(rep(0, ncol(zp)), rep(0.1, ncol(zp)), rep(-0.1, ncol(zp)))
  fits <- lapply(starts, function(start) {
    optim(start, penalized, penalized_gradient, method = "L-BFGS-B",
          lower = -12, upper = 12,
          control = list(maxit = 2000L, factr = 1e7, pgtol = 1e-7))
  })
  fits[[which.min(vapply(fits, function(fit) fit$value, numeric(1L)))]
}

fits <- list(uPU = fit_objective(FALSE), nnPU = fit_objective(TRUE))
test <- sample_population(30000L)
zt <- basis(test$x)
fitted_summary <- do.call(rbind, lapply(names(fits), function(method) {
  fit <- fits[[method]]
  parts <- components(fit$par)
  score <- drop(zt %*% fit$par)
  errors <- as.numeric((score > 0) != test$y)
  data.frame(method = method, training_positive_part = parts[1],
             training_negative_part = parts[2],
             test_log_loss = mean(softplus(score) - test$y * score),
             test_error = mean(errors), test_error_MCSE = sd(errors) /
↳ sqrt(length(errors)),
             optimizer_code = fit$convergence)
}))
cat("\nPART B: flexible fitted rules, assessed on independent labelled data\n")
print(fitted_summary, digits = 5, row.names = FALSE)
cat("\nOptimizer code 0 indicates numerical convergence, not global optimality.\n")

```

(continues on next page)

(continued from previous page)

```

cat("The score bounds, regularizer, and three starts are shared by both methods.\n")
cat("A single fitted-rule comparison is illustrative; repeat training for a method_
↪comparison.\n")

figure_dir <- Sys.getenv("COURSE_FIGURE_DIR", unset = "")
if (nzchar(figure_dir)) {
  dir.create(figure_dir, recursive = TRUE, showWarnings = FALSE)
  png(file.path(figure_dir, "04-pu-practice.png"), width = 1500,
      height = 700, res = 140, type = "cairo")
  par(mfrow = c(1, 2), mar = c(4.5, 4.5, 3.2, 1.2))
  boxplot(fixed_estimates[, "uPU"], fixed_estimates[, "nnPU"],
          names = c("Unbiased PU", "Non-negative PU"), col = c("#AFC3D4", "#BBDACD"),
          ylab = "Estimated logistic risk", main = "The same fixed score, repeated samples
↪")
  abline(h = true_fixed_risk, col = "#B44A3D", lty = 2, lwd = 2)
  legend("topleft", "True fixed-score risk", col = "#B44A3D", lty = 2, bty = "n", cex = 0.
↪8)
  grid <- seq(-4, 4, length.out = 401L)
  predictions <- cbind(plogis(drop(basis(grid) %*% fits$uPU$par)),
                      plogis(drop(basis(grid) %*% fits$nnPU$par)),
                      plogis(bayes_score(grid)))
  matplot(grid, predictions, type = "l", lwd = 2, lty = c(1, 1, 2),
          col = c("#1B4965", "#2D8063", "#B44A3D"), xlab = "Feature x",
          ylab = "Predicted positive probability", main = "Flexible scores fitted to one_
↪sample")
  legend("topleft", c("Unbiased PU", "Non-negative PU", "Bayes reference"),
          col = c("#1B4965", "#2D8063", "#B44A3D"), lty = c(1, 1, 2),
          lwd = 2, bty = "n", cex = 0.8)
  invisible(dev.off())
}

stopifnot(all(is.finite(fixed_estimates)),
          all(fixed_estimates[, "nnPU"] >= fixed_estimates[, "uPU"] - 1e-12),
          all(vapply(fits, function(fit) is.finite(fit$value), logical(1L))))

```

## Executed laboratory output

```

PART A: fixed-rule logistic risk
Numerically integrated reference risk: 0.1623882
  estimator   mean    bias    SD bias_MCSE    MSE
    uPU 0.18347 0.021085 0.44787 0.0141629 0.20083
    nnPU 0.30958 0.147196 0.30515 0.0096496 0.11469
Fraction with negative estimated negative-class contribution: 0.402

PART B: flexible fitted rules, assessed on independent labelled data
  method training_positive_part training_negative_part test_log_loss test_error
    uPU              0.0645723              -1.6771e+00      0.34373    0.131100
    nnPU              0.0045449              -1.3524e-10      0.31576    0.093567
test_error_MCSE optimizer_code
      0.0019486              0

```

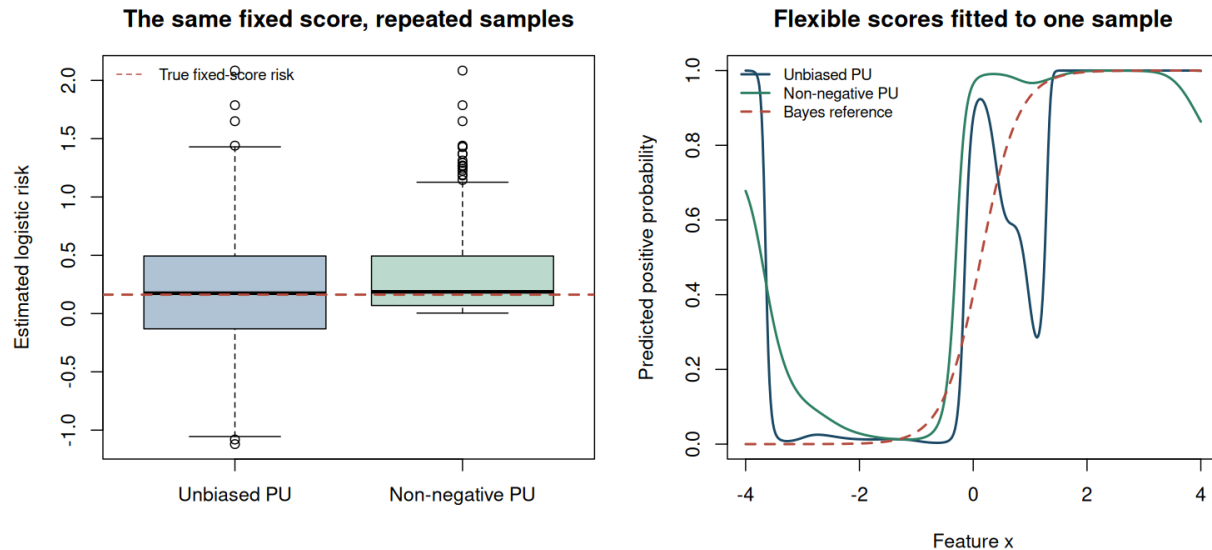
(continues on next page)

(continued from previous page)

0.0016814

0

Optimizer code 0 indicates numerical convergence, not global optimality.  
 The score bounds, regularizer, and three starts are shared by both methods.  
 A single fitted-rule comparison is illustrative; repeat training for a method comparison.



## Exercises

1. Prove the finite-sample upward-bias statement for a fixed rule. Does it imply worse mean squared error?
2. Derive the variance formula above. What goes wrong if the two positive-sample terms are incorrectly treated as independent?
3. Under logistic loss, simplify the prior-misspecification term. Which feature of the positive score distribution controls its magnitude?
4. Modify Part B to increase the positive sample size. Report both methods' independent test losses without claiming that one must always win.

### Worked solutions

1. The difference is  $(-\hat{B})_+$  pointwise. Taking expectations and using  $E\hat{R}_{PU} = R$  gives the result. Mean squared error equals variance plus squared bias, so an increase in bias can be offset by a reduction in variance. Dominance requires additional analysis.
2. Combine the positive terms into  $\pi\hat{E}_+(\ell_+ - \ell_-)$  and use independence across the two samples. Separating their variances without covariance generally gives the wrong answer because the same positive observations appear twice.

**3.** Logistic losses satisfy  $\ell_+ - \ell_- = -g(X)$ , so the discrepancy is  $-(\tilde{\pi} - \pi)E_+g(X)$ . A large average positive-class score can make the correction sensitive to the supplied prior. This calculation holds for a fixed rule; refitting under a different prior also changes the rule.

**4.** Larger  $n_+$  can reduce the variability of the subtraction, but the observed comparison also depends on the marginal sample, score class, regularization, and numerical optimization. Use repeated training samples and report the comparison with Monte Carlo uncertainty.

## Primary readings

- Kiryo et al. (2017), “Positive-Unlabeled Learning with Non-Negative Risk Estimator”: estimator, theory, and flexible-model experiments.
- Bekker, Robberechts, and Davis (2019), “Beyond the Selected Completely At Random Assumption for Learning from Positive and Unlabeled Data”: feature-dependent labelling and the assumptions needed to learn its mechanism.
- du Plessis, Niu, and Sugiyama (2015), “Convex Formulation for Learning from Positive and Unlabeled Data”: the underlying risk-correction formulation.

# 5. Learning from two unlabelled mixtures

## Core idea - Different mixtures provide an invertible contrast

The information comes from known, different mixing proportions and shared class distributions. A nonzero prior gap makes the population inverse available; a small gap can still make finite-sample estimates unusably noisy. The target prior is a separate choice from both sampling priors.

*Topics of Statistical Learning · Week 5*

## Where the supervision comes from

Unlabelled-unlabelled learning uses two collections without individual class labels. It is not ordinary clustering: the aim is to predict a specified positive or negative class, and aggregate information about the two collections supplies supervision. The basic model assumes that their class proportions are known and different, while their class-conditional feature distributions are shared.

The central calculation is a two-by-two linear inversion. It reveals both why learning is possible and why nearly equal mixture proportions produce unstable risk estimates. This chapter treats the prior information as known so the identification argument is transparent; estimating it requires additional assumptions and analysis.

## The two-mixture model and target population

Let  $P_+$  and  $P_-$  be the positive- and negative-class feature distributions. We observe independent samples from

$$Q_1 = aP_+ + (1 - a)P_-, \quad Q_2 = bP_+ + (1 - b)P_-,$$

where  $a, b \in [0, 1]$  are known and  $a \neq b$ . Usually both are strictly between zero and one, so both collections are genuinely unlabelled mixtures. The target population has a separately specified positive prior  $\pi$ , giving risk

$$R_\pi(g) = \pi E_+ \ell_+ + (1 - \pi) E_- \ell_-.$$

The target prior need not equal  $a$  or  $b$ . Setting  $\pi = 1/2$  deliberately evaluates balanced risk. It does not establish that either observed collection, or the application population, is balanced.

The assumption of common  $P_+$  and  $P_-$  is essential. Different feature distributions in the two collections must arise from their class proportions under this model. A change in measurement instruments or in the positive-class population can violate that relationship even when the proportions are known accurately.

## Inverting the mixtures

For any measurable set, the two mixture equations form a linear system with determinant  $a - b$ . Its inverse yields the distribution identities

$$P_+ = \frac{(1-b)Q_1 - (1-a)Q_2}{a-b}, \quad P_- = \frac{aQ_2 - bQ_1}{a-b}.$$

These are identities between probability measures under the assumed model, although their right sides contain signed coefficients. Their population values remain non-negative measures. Empirical versions need not have every property of a probability distribution in a finite sample.

For an integrable function  $h$ , the same equations hold with distributions replaced by expectations. Apply the first to  $\ell_+$  and the second to  $\ell_-$ :

$$\begin{aligned} R_\pi(g) &= \frac{\pi}{a-b} \{(1-b)E_{Q_1}\ell_+ - (1-a)E_{Q_2}\ell_+\} \\ &\quad + \frac{1-\pi}{a-b} \{aE_{Q_2}\ell_- - bE_{Q_1}\ell_-\}. \end{aligned}$$

Replacing these expectations with sample averages gives an unbiased estimator for a fixed rule under the known-prior model. As in PU learning, fitting a rule and selecting its tuning parameters require further control of generalization and optimization. Unbiasedness for a fixed rule is not a universal guarantee for every learning algorithm.

### **i** Worked example 5.1 — Recovering class-specific error from aggregate information

Let  $a = 0.8$ ,  $b = 0.2$ , and let a classifier predict positive on a feature region  $A$ . Suppose  $Q_1(A) = 0.69$  and  $Q_2(A) = 0.36$ . Inversion gives

$$P_+(A) = \frac{0.8(0.69) - 0.2(0.36)}{0.6} = 0.80,$$

$$P_-(A) = \frac{0.8(0.36) - 0.2(0.69)}{0.6} = 0.25.$$

The classifier therefore has false-negative rate 0.20 and false-positive rate 0.25. At target prior  $\pi = 0.4$ , its error is  $0.4(0.20) + 0.6(0.25) = 0.23$ . Its balanced error is 0.225.

Neither observed collection was labelled at the individual level. The different, known mixing proportions made the two class-specific probabilities recoverable.

Predicting which collection an observation came from is a related but different classification problem.

## Why small prior separation is a statistical problem

Let  $h_1$  and  $h_2$  collect the coefficients of the losses in the two expectations:

$$h_1(X) = \frac{\pi(1-b)\ell_+(X) - (1-\pi)b\ell_-(X)}{a-b},$$

$$h_2(X) = \frac{(1-\pi)a\ell_-(X) - \pi(1-a)\ell_+(X)}{a-b}.$$

For independent sample sizes  $n_1, n_2$  and a fixed rule,

$$\text{Var}(\hat{R}_\pi) = \frac{\text{Var}_{Q_1}(h_1)}{n_1} + \frac{\text{Var}_{Q_2}(h_2)}{n_2}.$$

The coefficients contain  $1/(a-b)$ . In typical nondegenerate examples, variance is amplified at the scale  $1/(a-b)^2$ . Particular losses and rules can create cancellations, so the exact variance is the appropriate quantity to inspect.

For balanced zero-one risk of a threshold classifier predicting positive when  $X > t$ , write  $F_k = P_{Q_k}(X \leq t)$ . The estimator simplifies to

$$\hat{R}_{1/2} = \frac{1}{2} + \frac{\hat{F}_1 - \hat{F}_2}{2(a-b)},$$

and its variance is

$$\frac{F_1(1-F_1)}{4n_1(a-b)^2} + \frac{F_2(1-F_2)}{4n_2(a-b)^2}.$$

**i Worked example 5.2 — An unbiased estimate outside the probability range**

Let  $P_+ = N(1, 1)$ ,  $P_- = N(-1, 1)$ ,  $t = 0$ , and  $\pi = 1/2$ . The true classification risk is  $\Phi(-1) \approx 0.1587$ . Take  $n_1 = n_2 = 500$  and fix  $a = 0.7$ .

When  $b = 0.3$ , the fixed-rule risk estimator has standard deviation about 0.038. When  $b = 0.69$ , the standard deviation is about 1.52. Both estimators remain unbiased under the model, but the second often falls below zero or above one.

This is not a floating-point error. The two collections contain little information for separating their latent class contributions. Clipping the result to  $[0, 1]$  changes the estimator and generally introduces bias. At  $a = b$ , the displayed inverse is undefined and the basic two-mixture model no longer identifies the class-specific distributions without additional restrictions.

## Identification failures and links to PU learning

If  $a = b$ , both observed distributions give the same mixture equation. An unrestricted pair  $(P_+, P_-)$  cannot generally be recovered from that single equation. If the proportions themselves are unknown, even their difference and class orientation need not be identifiable from the observed distributions alone. Extra assumptions can change this conclusion, but they must be stated explicitly.

If class-conditionals differ between collections, the algebra is solving the wrong system. A larger sample can then concentrate the risk estimate around an incorrect limit. Sensitivity experiments should examine this structural failure as well as finite-sample variability.

PU learning is a useful special case: set  $a = 1$  so  $Q_1 = P_+$  and set  $b = \pi$  so  $Q_2$  is the target feature marginal. Substitution reduces the UU expression to  $\pi E_+ \ell_+ + E_U \ell_- - \pi E_+ \ell_-$ . This unification concerns observation distributions and risk identities; it does not make a set of ordinary clustering labels into positive-class supervision.

## Allocate a budget according to variance contributions

Write the two variance contributions for a fixed rule as  $v_1/n_1 + v_2/n_2$ , where  $v_j = \text{Var}_{Q_j}(h_j)$  as defined above. With equal sampling costs, a fixed total  $N = n_1 + n_2$ , and positive  $v_1, v_2$ , differentiating with respect to  $n_1$  gives the continuous optimum

$$\frac{n_1}{n_2} = \sqrt{\frac{v_1}{v_2}}.$$

For illustrative values  $v_1 = 4v_2$ , the allocation ratio is two to one. This is a design calculation for specified variance contributions, not a claim that equal sample sizes are always inefficient. The contributions depend on the priors, loss, and rule; a pilot-based allocation must account for uncertainty in those inputs.

### Boundary - sample size cannot replace identification

A larger budget can reduce variance when  $a \neq b$ . At  $a = b$ , the two-mixture inverse is unavailable under the unrestricted shared-component model. Collect different information or state additional identifying restrictions before fitting.

## Laboratory — Prior separation and assumption failure

The lab holds one threshold classifier fixed and repeatedly estimates its balanced target risk. It compares four values of  $b$  while keeping  $a = 0.7$ , reports empirical and exact standard deviations, and records the fraction of estimates outside  $[0, 1]$ . A final experiment shifts the positive-class mean in the second collection and compares the misspecified correction's analytic limit with its simulation average.

Known latent classes are used only inside the generator. The risk estimator receives features and supplied proportions, not individual class labels. The analytic reference risk is available to the evaluator because the simulation model is specified.

## Download lab

```

# Topics of Statistical Learning
# Lab 05: Two-unlabelled-mixture risk estimation and instability
# Base R only. Run with: Rscript 05-uu-learning.R
set.seed(2026102105)

target_prior <- 0.5
a <- 0.7
b_values <- c(0.3, 0.5, 0.65, 0.69)
n_each <- 500L
n_rep <- 1000L
threshold <- 0.2
positive_mean <- 1
negative_mean <- -1

sample_mixture <- function(n, proportion, positive_location = positive_mean) {
  # Labels are internal to simulation and never passed to the risk estimator.
  latent_y <- rbinom(n, 1L, proportion)
  rnorm(n, mean = ifelse(latent_y == 1L, positive_location, negative_mean))
}

mixture_cdf <- function(proportion, positive_location = positive_mean) {
  proportion * pnorm(threshold, positive_location) +
    (1 - proportion) * pnorm(threshold, negative_mean)
}

risk_from_features <- function(x1, x2, a, b, target_prior) {
  if (abs(a - b) < 1e-12) stop("Distinct known mixture priors are required.")
  loss1_positive <- mean(x1 <= threshold)
  loss2_positive <- mean(x2 <= threshold)
  loss1_negative <- 1 - loss1_positive
  loss2_negative <- 1 - loss2_positive
  target_prior * ((1 - b) * loss1_positive - (1 - a) * loss2_positive) / (a - b) +
    (1 - target_prior) * (a * loss2_negative - b * loss1_negative) / (a - b)
}

true_risk <- target_prior * pnorm(threshold, positive_mean) +
  (1 - target_prior) * (1 - pnorm(threshold, negative_mean))
estimates <- matrix(NA_real_, n_rep, length(b_values))
summary_table <- do.call(rbind, lapply(seq_along(b_values), function(j) {
  b <- b_values[j]
  for (r in seq_len(n_rep)) {
    estimates[r, j] <- risk_from_features(sample_mixture(n_each, a),
                                          sample_mixture(n_each, b),
                                          a, b, target_prior)
  }
  value <- estimates[, j]
  f1 <- mixture_cdf(a)
  f2 <- mixture_cdf(b)
  coefficient1 <- (target_prior + b - 2 * target_prior * b) / (a - b)
  coefficient2 <- (target_prior + a - 2 * target_prior * a) / (a - b)
  exact_sd <- sqrt(coefficient1^2 * f1 * (1 - f1) / n_each +
                  coefficient2^2 * f2 * (1 - f2) / n_each)
  data.frame(a = a, b = b, prior_gap = a - b, mean_risk = mean(value),

```

(continues on next page)

(continued from previous page)

```

    bias = mean(value) - true_risk, empirical_SD = sd(value),
    exact_SD = exact_sd, mean_MCSE = sd(value) / sqrt(n_rep),
    fraction_outside_0_1 = mean(value < 0 | value > 1))
  ))

cat("Target prior:", target_prior, "; threshold:", threshold, "\n")
cat("Independent samples per replication:", n_each, "from each mixture\n")
cat("Replications:", n_rep, "; analytic target risk:", true_risk, "\n\n")
print(summary_table, digits = 5, row.names = FALSE)

# Misspecification: only the positive class in mixture 2 is shifted.
b_shift <- 0.3
shifted_positive_mean <- 1.75
f1 <- mixture_cdf(a)
f2_shift <- mixture_cdf(b_shift, shifted_positive_mean)
correction_limit <- target_prior * ((1 - b_shift) * f1 -
                                   (1 - a) * f2_shift) / (a - b_shift) +
  (1 - target_prior) * (a * (1 - f2_shift) -
                        b_shift * (1 - f1)) / (a - b_shift)
shift_estimates <- replicate(300L, {
  risk_from_features(sample_mixture(2000L, a),
                     sample_mixture(2000L, b_shift, shifted_positive_mean),
                     a, b_shift, target_prior)
})

cat("\nCommon-class-conditional assumption violated in mixture 2:\n")
print(data.frame(target_risk = true_risk, misspecified_limit = correction_limit,
                 simulated_mean = mean(shift_estimates),
                 mean_MCSE = sd(shift_estimates) / sqrt(length(shift_estimates)),
                 asymptotic_bias = correction_limit - true_risk),
      digits = 5, row.names = FALSE)

cat("\nThe target population still uses positive mean 1 and negative mean -1.\n")
cat("Larger samples reduce variability around the misspecified limit, not its bias.\n")

figure_dir <- Sys.getenv("COURSE_FIGURE_DIR", unset = "")
if (nzchar(figure_dir)) {
  dir.create(figure_dir, recursive = TRUE, showWarnings = FALSE)
  png(file.path(figure_dir, "05-uu-learning.png"), width = 1500,
      height = 700, res = 140, type = "cairo")
  par(mfrow = c(1, 2), mar = c(4.5, 4.5, 3.2, 1.2))
  plot(summary_table$prior_gap, summary_table$exact_SD, type = "l", log = "xy",
       lwd = 2, col = "#1B4965", xlab = "Mixture-prior separation |a - b| (log scale)",
       ylab = "Risk estimator SD (log scale)", main = "Small separation reduces precision")
  points(summary_table$prior_gap, summary_table$empirical_SD,
        pch = 19, col = "#2D8063", cex = 1.1)
  legend("topright", c("Exact fixed-rule SD", "Simulation SD"),
        col = c("#1B4965", "#2D8063"), lty = c(1, NA), pch = c(NA, 19),
        bty = "n", cex = 0.8)
  hist(shift_estimates, breaks = 24L, col = "#DDE9F1", border = "white",
       xlim = range(c(true_risk, shift_estimates)), xlab = "Estimated target risk",
       main = "Violation of shared class-conditionals")
  abline(v = true_risk, lwd = 2, lty = 2, col = "#B44A3D")
  abline(v = correction_limit, lwd = 2, col = "#1B4965")

```

(continues on next page)

(continued from previous page)

```

legend("topleft", c("True target risk", "Misspecified limit"),
      col = c("#B44A3D", "#1B4965"), lty = c(2, 1), lwd = 2, bty = "n", cex = 0.8)
invisible(dev.off())
}

stopifnot(all(is.finite(estimates)),
          all(is.finite(shift_estimates)),
          all(summary_table$exact_SD > 0))

```

## Executed laboratory output

Target prior: 0.5 ; threshold: 0.2  
 Independent samples per replication: 500 from each mixture  
 Replications: 1000 ; analytic target risk: 0.1634625

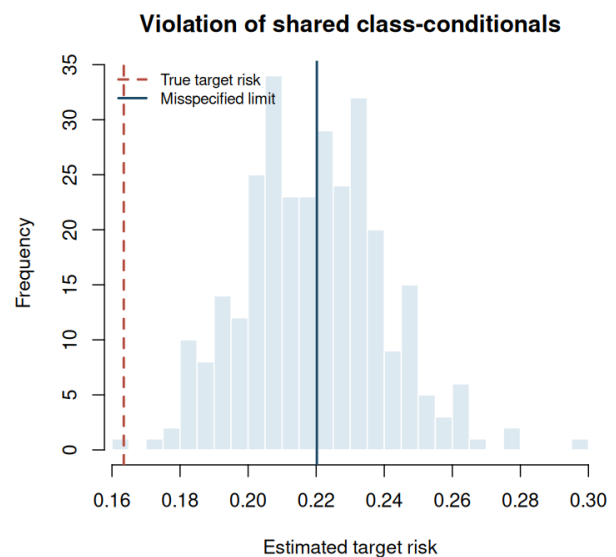
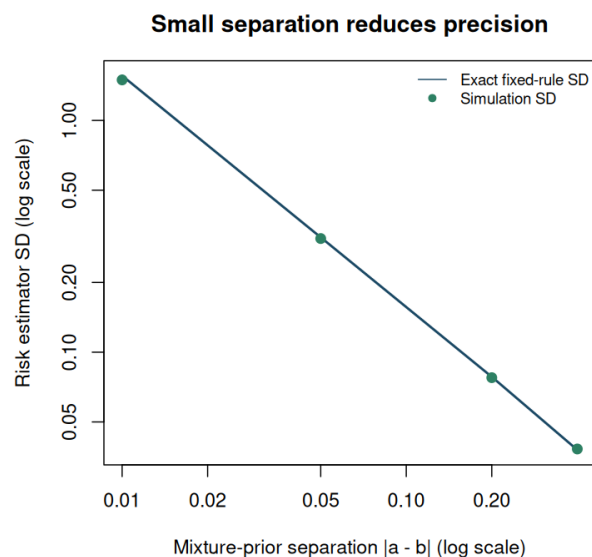
| a   | b    | prior_gap | mean_risk | bias        | empirical_SD | exact_SD | mean_MCSE |
|-----|------|-----------|-----------|-------------|--------------|----------|-----------|
| 0.7 | 0.30 | 0.40      | 0.16297   | -0.00048753 | 0.038230     | 0.037876 | 0.0012089 |
| 0.7 | 0.50 | 0.20      | 0.15915   | -0.00431253 | 0.077636     | 0.078280 | 0.0024551 |
| 0.7 | 0.65 | 0.05      | 0.17360   | 0.01013747  | 0.309276     | 0.312986 | 0.0097802 |
| 0.7 | 0.69 | 0.01      | 0.25130   | 0.08783747  | 1.495712     | 1.559242 | 0.0472986 |

fraction\_outside\_0\_1

|       |
|-------|
| 0.000 |
| 0.018 |
| 0.296 |
| 0.724 |

Common-class-conditional assumption violated in mixture 2:  
 target\_risk misspecified\_limit simulated\_mean mean\_MCSE asymptotic\_bias  
 0.16346 0.22019 0.21973 0.0011821 0.056732

The target population still uses positive mean 1 and negative mean -1.  
 Larger samples reduce variability around the misspecified limit, not its bias.



## Exercises

1. Verify both measure inversions by substitution. Why is the condition  $a \neq b$  necessary for this argument?
2. Substitute  $a = 1$  and  $b = \pi$  into the UU risk identity and recover the PU identity.
3. Derive the balanced zero-one estimator and its variance. Why must the two sample sizes appear separately?
4. If the separation  $|a - b|$  is approximately halved while other variance factors remain similar, how should both sample sizes change to maintain approximately the same precision?
5. What is the difference between fixing an unstable finite-sample estimate and fixing a violated common-class-conditional assumption?

### Worked solutions

1. The coefficient of  $P_+$  in the first numerator is  $a(1 - b) - b(1 - a) = a - b$ , and the coefficient of  $P_-$  is zero. The second identity is analogous. At equal priors the matrix is singular and this cancellation does not provide an inverse.
2. The first bracket becomes  $\pi E_+ \ell_+$ . The second becomes  $E_U \ell_- - \pi E_+ \ell_-$  because  $a - b = 1 - \pi$ . Their sum is the PU expression.
3. For a threshold rule,  $\ell_+ = \mathbf{1}(X \leq t)$  and  $\ell_- = 1 - \ell_+$ . At  $\pi = 1/2$ , collecting constants gives the stated estimator. Independent Bernoulli sample means have variances  $F_k(1 - F_k)/n_k$ ; the two independent samples may have different sizes and different event probabilities.
4. Increase both sizes by approximately a factor of four. This compensates for the squared inverse separation, provided the remaining variance terms are comparable. It does not repair exact equality of the priors.
5. More observations or appropriate regularization can improve a noisy but correctly specified estimator. Structural misspecification changes what the expression estimates. Addressing it requires an observation model, target, or data source that matches the application, not simply more repetitions of the same biased procedure.

## Primary readings

- Lu et al. (2019), “On the Minimal Supervision for Training Any Binary Classifier from Only Unlabeled Data”: the two-mixture risk formulation and its learning theory.
- Lu et al. (2020), “Mitigating Overfitting in Supervised Classification from Two Unlabeled Datasets: A Consistent Risk Correction Approach”: correction of empirical risks in flexible UU learning.

- Lu et al. (2021), “Binary Classification from Multiple Unlabeled Datasets via Surrogate Set Classification”: an optional extension beyond two collections.

**Connection to the next block:** Preference data replace an absolute class label with a comparison. We will again ask which target the observation mechanism identifies before fitting a model.

## 6. Pairwise preference learning

### Core idea - A comparison identifies a difference

Bradley-Terry strengths have no absolute origin. The comparison graph determines which differences can be identified, while the observed wins determine whether an unpenalized finite estimate exists. Connected observations, finite estimation, and precise estimation are three different requirements.

**Question.** What can we learn when a person can compare two alternatives but cannot supply an absolute response?

Download lab

A clinician may prefer one explanation to another, a reader may rank two retrieved passages, and a customer may choose between products. These observations are informative, but their meaning differs from a numerical measurement of quality. This chapter develops a statistical model for the comparisons, identifies its estimable quantities, and asks how the comparison schedule affects precision.

By the end, you should be able to derive a Bradley-Terry likelihood, diagnose two distinct failures of estimation, and connect preference learning to optimal experimental design.

## From a comparison design to a target

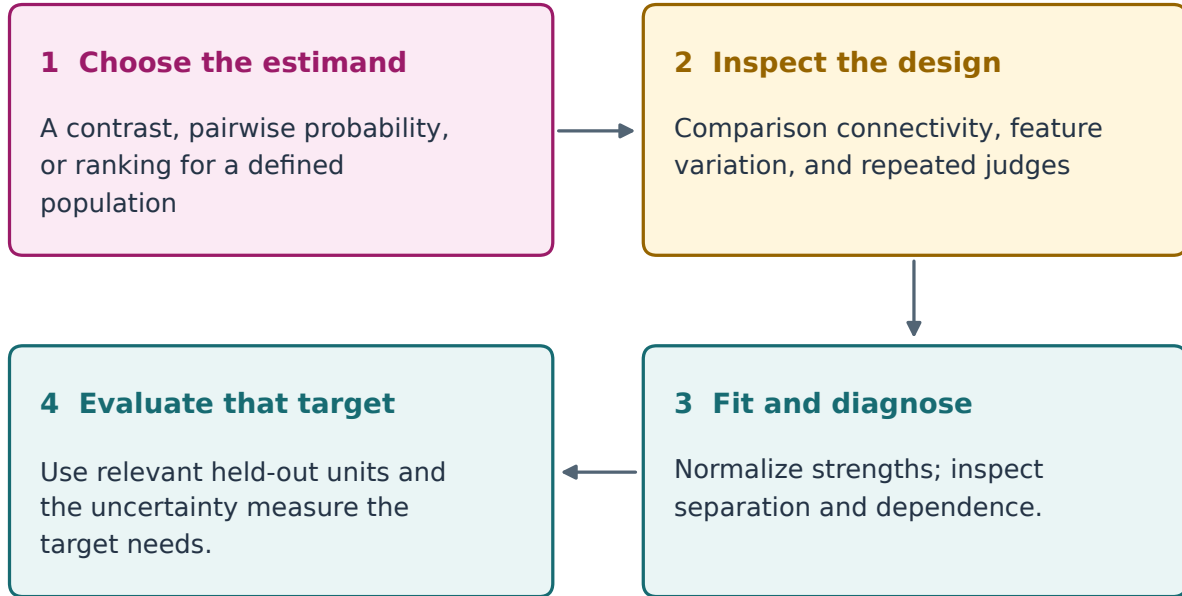


Fig. 3: A comparison workflow: choose an estimand, inspect graph and dependence, diagnose the fit, and evaluate that estimand. The checks distinguish identification, finite estimation, and precision.

### 6.1 · Specify the observation and target

There are  $K$  items. For a scheduled comparison of items  $i$  and  $j$ , let  $Y = 1$  mean that  $i$  is preferred. The basic model is

$$p_{ij} = \Pr(Y = 1 \mid i, j) = \frac{e^{\theta_i}}{e^{\theta_i} + e^{\theta_j}} = \sigma(\theta_i - \theta_j), \quad \sigma(u) = \frac{1}{1 + e^{-u}}. \quad (6.1)$$

Possible targets are a strength contrast  $\theta_i - \theta_j$ , a pairwise probability  $p_{ij}$ , or an ordering of the strengths. A confidence interval for a contrast does not automatically provide a simultaneous confidence statement about the entire ranking.

The introductory analysis assumes a fixed comparison schedule, constant strengths, binary outcomes without ties, and conditionally independent comparisons. These are modelling choices to check. Repeated judges, shared prompts, order effects, or changing preferences can violate the simple sampling model; Chapter 7 addresses some of these complications.

## 6.2 · Derive the likelihood

Suppose  $w_{ij}$  of  $n_{ij}$  independent comparisons favour item  $i$ . Write  $x_{ij} = e_i - e_j$ . Ignoring factors independent of the parameters, the log likelihood is

$$\ell(\theta) = \sum_{i < j} \left[ w_{ij} x_{ij}^T \theta - n_{ij} \log \{ 1 + \exp(x_{ij}^T \theta) \} \right].$$

This is a binomial logistic regression with a structured design matrix. The score and negative Hessian are

$$U(\theta) = \sum_{i < j} (w_{ij} - n_{ij} p_{ij}) x_{ij}, \quad I(\theta) = \sum_{i < j} n_{ij} p_{ij} (1 - p_{ij}) x_{ij} x_{ij}^T. \quad (6.2)$$

Adding the same constant to all strengths leaves every probability unchanged. Impose  $\theta_K = 0$  or  $\sum_i \theta_i = 0$  before fitting. This normalization fixes an arbitrary origin; it does not create information that the experiment failed to collect.

### **i** Example 6.1 · A preference probability and its uncertainty

**Setup.** In an illustrative fixed comparison, alternative A wins 12 times and B wins 8 times.

**Work it through.** The estimated log-strength difference is  $\hat{\delta} = \log(12/8) = 0.405$ . Its approximate standard error is  $\sqrt{1/12 + 1/8} = 0.456$ . A nominal 95% Wald interval is approximately  $[-0.489, 1.300]$ . Transforming the endpoints with the logistic function gives approximately  $[0.380, 0.786]$  for the win probability.

**What this shows.** A point estimate of 0.60 does not establish a reliable preference for A. The interval concerns repeated independent comparisons under the model; it is neither a probability that A is objectively correct nor a simultaneous ranking guarantee.

## 6.3 · Identification is different from a finite estimate

Construct an **undirected comparison graph** with an edge whenever  $n_{ij} > 0$ . If it is connected, all strength differences are identified in the model, subject to the common additive constant. If two components never meet, their relative location remains unidentified. No optimizer can recover an unsupported cross-component contrast from the likelihood alone.

Now construct a **directed win graph**, with an arrow from  $i$  to  $j$  when  $i$  has beaten  $j$ . A finite, unique unpenalized maximum-likelihood estimate after normalization requires this graph to be strongly connected: every item can reach every other along directed wins. An undefeated group can cause separation even when the undirected comparison graph is connected.

For example, observing only A beating B and B beating C gives a connected schedule but supports arbitrarily increasing strength differences. Penalization or a proper

prior can stabilize estimation, but the resulting answer depends on that additional assumption. Report it explicitly rather than treating a large finite optimizer output as a successful unpenalized fit.

## 6.4 · Why the comparison schedule matters

The information matrix in Equation (6.2) is a weighted graph Laplacian. Each edge contributes information about one difference. For a contrast  $a^\top \theta$  with  $a^\top \mathbf{1} = 0$ , a local large-sample variance approximation is  $a^\top I(\theta)^+ a$ , where  $I^+$  is the generalized inverse on the identifiable contrast space.

This gives a concrete design problem: allocate a fixed comparison budget over pairs to minimize variance for an important contrast, average variance over several contrasts, or another specified information criterion. Local information depends on the unknown strengths, so a locally optimal allocation uses nominal values or a pilot. A robust or sequential design must account for that uncertainty.

### Example 6.2 · Equal budget, unequal precision

**Setup.** Three equally strong items have  $p_{ij} = 1/2$ . We have 60 comparisons and want  $\theta_1 - \theta_3$ .

**Work it through.** Allocating 20 comparisons to each of the three pairs gives edge information  $20/4 = 5$  and contrast variance  $2/(3 \times 5) = 2/15$ . Allocating 30 each to pairs (1,2) and (2,3) gives two edge informations of 7.5; the endpoint difference adds their variances, yielding  $1/7.5 + 1/7.5 = 4/15$ .

**What this shows.** Both schedules identify the target, but the chain has twice the local variance. These numbers use the equal-strength model and a local information approximation. They do not prove that balanced comparisons are universally optimal for every target and parameter value.

## Ask what a new comparison would repair

Before allocating another comparison, identify the deficiency. A comparison connecting previously separate components can identify a previously unsupported contrast. Another comparison along an existing path can improve precision without changing identifiability. A newly observed reverse win can change separation behaviour, but its occurrence cannot be guaranteed by the schedule.

This distinction prevents treating all additional data as equivalent. For a target contrast, inspect which paths connect its two items and where information is weak. If a prior or penalty is required to obtain a finite estimate, describe that addition alongside the comparison evidence.

**Analysis check - three diagnostics, in order**

Check the undirected comparison graph, then the directed win graph, then uncertainty for the actual target contrast. A finite coefficient vector from an optimizer is not a substitute for any of these checks.

**6.5 · Execute the comparison experiment**

The base-R lab compares complete, chain, and disconnected schedules. It reports design ranks, local precision for an endpoint contrast, and a fixed-design simulation. Only connected schedules enter the estimation comparison. Flagged fits are counted; coverage summaries use the usable fits and therefore must be read alongside those counts.

Change one element at a time: the total budget, strength separation, or graph. Before running, predict which contrast should become harder to estimate. The likelihood implementation uses `glm.fit`; the full R source is included so the comparison matrix and normalization remain visible.

```
# Chapter 6: fixed comparison schedules and Bradley-Terry information.
# Base R only. Repeated runs with the same R version and seed reproduce the draws.
set.seed(2026102106)
options(width = 100)
K <- 6L
theta <- c(-0.9, -0.5, -0.1, 0.1, 0.5, 0.9)
all_pairs <- t(combn(seq_len(K), 2L))
chain_pairs <- cbind(seq_len(K - 1L), 2:K)
disconnected_pairs <- rbind(t(combn(1:3, 2)), t(combn(4:6, 2)))
comparison_matrix <- function(pairs) {
  X <- matrix(0, nrow(pairs), K)
  X[cbind(seq_len(nrow(pairs)), pairs[, 1])] <- 1
  X[cbind(seq_len(nrow(pairs)), pairs[, 2])] <- -1
  X
}
cat("Comparison-matrix ranks; a connected K-item design needs rank K-1:\n")
print(c(complete = qr(comparison_matrix(all_pairs))$rank,
        chain = qr(comparison_matrix(chain_pairs))$rank,
        disconnected = qr(comparison_matrix(disconnected_pairs))$rank))
budget <- 600L
a <- c(1, rep(0, K - 2L)) # theta_1 - theta_K after fixing theta_K=0.
local_se <- function(pairs) {
  X <- comparison_matrix(pairs)[, -K, drop = FALSE]
  n <- rep(budget / nrow(pairs), nrow(pairs))
  p <- plogis(comparison_matrix(pairs) %*% theta)
  information <- crossprod(X, X * as.vector(n * p * (1 - p)))
  sqrt(drop(crossprod(a, solve(information, a))))
}
cat("\nLocal standard error for theta_1 - theta_6 at the generating strengths:\n")
print(round(c(complete = local_se(all_pairs), chain = local_se(chain_pairs)), 4))
```

(continues on next page)

(continued from previous page)

```

run_design <- function(pairs, B = 150L) {
  X <- comparison_matrix(pairs)[, -K, drop = FALSE]
  n <- rep(as.integer(budget / nrow(pairs)), nrow(pairs))
  p <- as.vector(plogis(comparison_matrix(pairs) %*% theta))
  target <- theta[1] - theta[K]
  estimates <- ses <- rep(NA_real_, B)
  for (b in seq_len(B)) {
    wins <- rbinom(length(n), n, p)
    fit <- suppressWarnings(glm.fit(X, cbind(wins, n - wins),
                                   family = binomial(), intercept = FALSE))
    if (!fit$converged || any(!is.finite(fit$coefficients)) ||
        any(abs(fit$coefficients) > 20)) next
    phat <- fit$fitted.values
    information <- crossprod(X, X * as.vector(n * phat * (1 - phat)))
    inv <- tryCatch(solve(information), error = function(e) NULL)
    if (is.null(inv)) next
    estimates[b] <- fit$coefficients[1]
    ses[b] <- sqrt(inv[1, 1])
  }
  ok <- is.finite(estimates) & is.finite(ses)
  covered <- abs(estimates[ok] - target) <= qnorm(.975) * ses[ok]
  c(usable = sum(ok), flagged = sum(!ok),
    bias = mean(estimates[ok]) - target,
    rmse = sqrt(mean((estimates[ok] - target)^2)),
    mean_se = mean(ses[ok]), coverage = mean(covered),
    coverage_mcse = sqrt(mean(covered) * (1 - mean(covered)) / sum(ok)))
}
results <- rbind(complete = run_design(all_pairs), chain = run_design(chain_pairs))
cat("\nFixed-design simulation; coverage is estimated, not guaranteed:\n")
print(round(results, 4))
cat("\nThe disconnected design is not fitted: the endpoint contrast is unidentified.\n")
cat("Flagged fits are disclosed; reported simulation summaries use the usable fits.\n")

# A connected comparison graph can still have a separated win pattern.
win_graph <- matrix(FALSE, 3, 3)
win_graph[1, 2] <- win_graph[2, 3] <- TRUE
reach <- win_graph
diag(reach) <- TRUE
for (k in 1:3) reach <- reach | outer(reach[, k], reach[k, ], "&")
cat("\nA beats B, B beats C: undirected graph connected, directed wins strongly connected?
↔\n")
print(all(reach))

# Optional static figure; the course build supplies this directory.
figure_dir <- Sys.getenv("COURSE_FIGURE_DIR", "")
if (nzchar(figure_dir)) {
  dir.create(figure_dir, recursive = TRUE, showWarnings = FALSE)
  png(file.path(figure_dir, "06-pairwise-preferences.png"),
      width = 1100, height = 500, res = 110)
  par(mfrow = c(1, 2), mar = c(4.5, 4.5, 3.5, 1))
  values <- t(results[, c("rmse", "mean_se")])
  barplot(values, beside = TRUE, col = c("#007A87", "#A2006E"),

```

(continues on next page)

(continued from previous page)

```

ylim = c(0, 1.25 * max(values)), ylab = "Log-strength scale",
main = "Precision of the endpoint contrast",
legend.text = c("Empirical RMSE", "Mean estimated SE"),
args.legend = list(x = "topleft", bty = "n", cex = .8))
coverage <- results[, "coverage"]
mcse <- results[, "coverage_mcse"]
limits <- range(c(.95, pmax(0, coverage - 2 * mcse), pmin(1, coverage + 2 * mcse)))
plot(1:2, coverage, ylim = limits + c(-.01, .01), xlim = c(.6, 2.4),
     xaxt = "n", xlab = "Comparison schedule", ylab = "Estimated coverage",
     pch = 19, col = "#007A87", main = "Nominal 95% Wald intervals")
axis(1, at = 1:2, labels = rownames(results))
abline(h = .95, lty = 2, col = "#A2006E")
arrows(1:2, pmax(0, coverage - 2 * mcse), 1:2, pmin(1, coverage + 2 * mcse),
       angle = 90, code = 3, length = .05, col = "#007A87")
mtext("Bars show approximately two Monte Carlo SEs", side = 3, cex = .7)
invisible(dev.off())
}

```

Comparison-matrix ranks; a connected K-item design needs rank K-1:

|          |       |              |
|----------|-------|--------------|
| complete | chain | disconnected |
| 5        | 5     | 4            |

Local standard error for  $\theta_1 - \theta_6$  at the generating strengths:

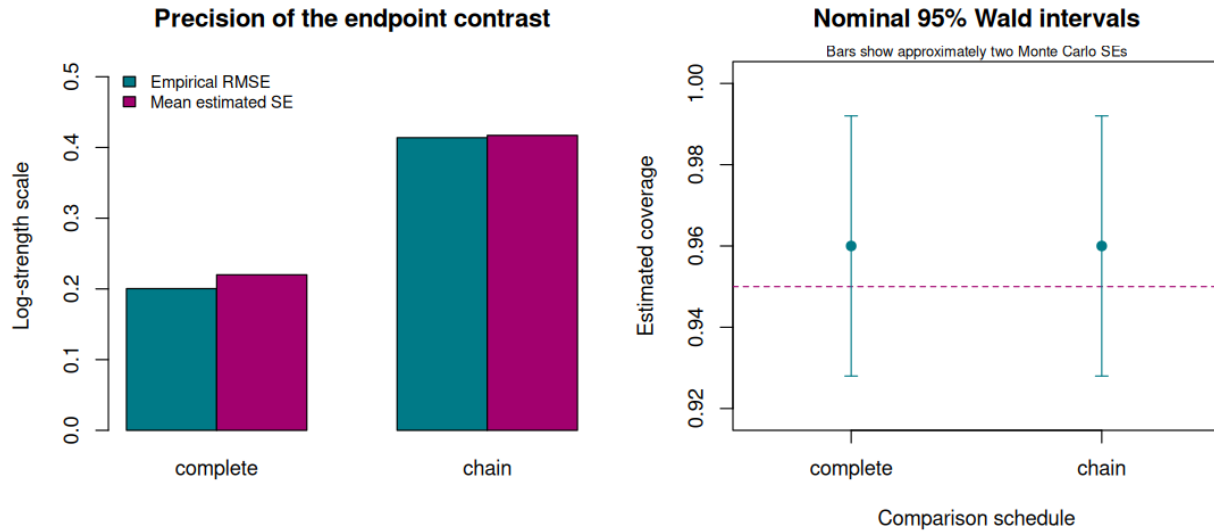
|          |        |
|----------|--------|
| complete | chain  |
| 0.2185   | 0.4152 |

Fixed-design simulation; coverage is estimated, not guaranteed:

|          | usable | flagged | bias    | rmse   | mean_se | coverage | coverage_mcse |
|----------|--------|---------|---------|--------|---------|----------|---------------|
| complete | 150    | 0       | -0.0206 | 0.2004 | 0.2200  | 0.96     | 0.016         |
| chain    | 150    | 0       | -0.0025 | 0.4139 | 0.4171  | 0.96     | 0.016         |

The disconnected design is not fitted: the endpoint contrast is unidentified.  
 Flagged fits are disclosed; reported simulation summaries use the usable fits.

A beats B, B beats C: undirected graph connected, directed wins strongly connected?  
 [1] FALSE



## 6.6 • Exercises

1. Show that replacing  $\theta$  by  $\theta + c\mathbf{1}$  leaves the likelihood unchanged. Explain why a disconnected graph introduces more than one arbitrary offset.
2. Derive the score and information matrix. Which term makes a nearly deterministic comparison less informative locally about a small change in its log odds?
3. Construct a connected comparison schedule with no finite unpenalized MLE. Explain what a ridge penalty changes and what it does not establish about the scientific target.
4. Modify the experiment to target  $\theta_2 - \theta_3$ . Compare the complete and chain designs at the same budget. Report estimation error, fit flags, and Monte Carlo uncertainty; do not choose seeds to obtain a preferred ranking.

### **i** Worked solutions

1. Every design row has entries summing to zero. Within a disconnected component, adding its own constant preserves all observed probabilities. Cross-component contrasts consequently change without changing the likelihood.
2. Differentiate  $w\eta - n \log(1 + e^\eta)$  to obtain  $w - n\sigma(\eta)$  and curvature  $n\sigma(\eta)\{1 - \sigma(\eta)\}$ . The latter becomes small as a probability approaches zero or one. Nearly certain wins provide little local information per comparison about the underlying log-strength contrast.
3. A beats B in every observed comparison and B beats C in every observed comparison. The win graph has no return path. A positive quadratic penalty selects finite strengths and an origin if applied appropriately, but this is a penalized model, not new evidence about the unconstrained contrast.

**4.** Replace the contrast vector and extract the matching fitted coefficient difference and covariance expression. The answer depends on the graph location and generating strengths. A legitimate result can favour either design for a particular target; comparisons should keep budget and simulation protocol fixed.

## Reading and transition

Read [Bradley and Terry \(1952\)](#) for the original model and [Turner and Firth \(2012\)](#) for its GLM representation and R implementation. [Yan \(2014\)](#) discusses failure of the strong-connection condition. Next, allow comparisons to depend on context and on who supplied the preference.

# 7. Contextual preferences, judges, and reward learning

## Core idea - A preference is conditional on a population and context

A comparison rate reflects the alternatives, the prompt, the judge population, and presentation. Specify whose preferences should generalize to which future tasks. Repeated rows from the same people or prompts do not automatically supply independent evidence.

**Question.** Whose preference is being learned, in which context, and with what uncertainty?

Download lab

A single strength per item can be inadequate when the same answer is helpful for one question and inappropriate for another. Contextual preference learning models comparisons conditional on information about the task and the alternatives. Its statistical challenge includes both the predictive model and the mechanism that generates the comparisons.

This chapter extends paired comparisons, handles repeated judges carefully, and derives a finite-action connection to direct preference optimization. It does not require language-model training.

## 7.1 · A contextual comparison model

Let  $x$  denote a context and  $a, b$  two alternatives. Define their feature difference

$$d(x, a, b) = \phi(x, a) - \phi(x, b).$$

A linear utility model gives

$$\Pr(a \succ b \mid x, a, b) = \sigma\{d(x, a, b)^\top \beta\}. \quad (7.1)$$

The target might be this conditional probability, a coefficient contrast, or the decision rule selecting the alternative with largest predicted utility. These targets require different evaluations. Pairwise prediction accuracy alone does not establish a correct global ranking or factual reliability.

For observed labels  $Y_r$  and differences  $d_r$ , the negative log likelihood is

$$\sum_r \left[ \log\{1 + \exp(d_r^\top \beta)\} - Y_r d_r^\top \beta \right].$$

This is ordinary logistic loss on feature differences. An item-independent utility intercept cancels, and covariates that never vary within a comparison provide no contrast information. Identification requires sufficient variation in the observed differences; separation remains possible.

## 7.2 • Presentation and sampling matter

Suppose  $h = 1$  when alternative  $a$  is displayed first and  $h = -1$  when it is displayed second. A model with an order effect is

$$\text{logit Pr}(a \succ b \mid x, a, b, h) = d(x, a, b)^\top \beta + \delta h.$$

Randomizing presentation helps separate an order effect from content features. It does not make all preference labels objective or remove systematic judge disagreement. Similarly, if comparisons are collected only for close alternatives, the training comparison distribution differs from a deployment distribution containing many easy pairs. State the evaluation population and ensure that the held-out sample represents it, or justify a weighting adjustment.

### Example 7.1 • Relevance and conciseness can disagree

**Setup.** A has one more unit of relevance and two fewer units of conciseness than B. The hypothetical coefficients are  $\beta = (1.2, 0.3)$ .

**Work it through.** The utility contrast is  $1.2 - 2(0.3) = 0.6$ , so the predicted preference probability is  $\sigma(0.6) = 0.646$ . If the order effect is  $\delta = 0.4$ , displaying A first changes the model probability to  $\sigma(1.0) = 0.731$ ; displaying it second gives  $\sigma(0.2) = 0.550$ .

**What this shows.** A reported preference rate can reflect both content and presentation. A preferred answer can still contain a factual error: the outcome here measures a judge's comparison, not an independently verified truth label.

## 7.3 • Repeated judges are not independent rows

If judge  $j$  evaluates several pairs, their responses can share a latent preference, expertise level, or order bias. One possible conditional model is

$$\text{logit Pr}(Y_{jr} = 1 \mid d_{jr}, h_{jr}, b_j) = d_{jr}^\top \beta + (\delta + b_j)h_{jr}.$$

Here  $b_j$  represents judge-specific order sensitivity. Integrating out  $b_j$  generally does not preserve the same logistic regression coefficients. A pooled logistic fit can instead

be understood as a working marginal model, with a population projection target determined by its expected loss. Distinguish this target from the conditional generating coefficient.

For independent judges with repeated observations, a cluster sandwich estimator uses summed score contributions  $U_j$ :

$$\hat{V} = \hat{H}^{-1} \left( \sum_j U_j U_j^\top \right) \hat{H}^{-1}.$$

Its justification needs enough independent clusters and suitable regularity. It is not an exact small-sample guarantee. A judge bootstrap resamples whole judges, preserving their within-judge observations. Resampling individual comparison rows would destroy that structure.

**i Example 7.2 · Eight hundred rows need not mean eight hundred independent measurements**

**Setup.** Forty judges each supply twenty responses. For a simple scalar mean, suppose responses within each judge have equal variance and common correlation  $\rho = 0.15$ .

**Work it through.** The variance inflation relative to independent rows is  $1 + (20 - 1)\rho = 3.85$ . The corresponding effective sample size for this particular mean calculation is  $800/3.85 \approx 208$ .

**What this shows.** The number of independent judges matters. This effective-sample-size formula is an illustrative exchangeable-mean calculation, not a universal replacement sample size for every logistic coefficient or crossed judge-prompt design.

When both prompts and judges repeat, dependence can be crossed. A one-way judge bootstrap may then be insufficient. The independent unit for validation and resampling must match the target population. For adaptive pair selection or stopping, inference must additionally be justified under that sequential sampling scheme; ordinary independent-row standard errors are not automatically valid.

## 7.4 · From preference likelihood to a policy

For a fixed context, let  $\pi_0(a) > 0$  be a reference distribution over finitely many actions. Given rewards  $r(a)$ , consider

$$\max_{\pi} \left\{ \sum_a \pi(a) r(a) - \tau \sum_a \pi(a) \log \frac{\pi(a)}{\pi_0(a)} \right\}, \quad \tau > 0.$$

A Lagrange multiplier for  $\sum_a \pi(a) = 1$  gives

$$\pi^*(a) = \frac{\pi_0(a) e^{r(a)/\tau}}{Z}, \quad r(a) = \tau \log \frac{\pi^*(a)}{\pi_0(a)} + \tau \log Z.$$

Under a Bradley–Terry reward comparison, the common  $\log Z$  cancels. The preference probability becomes a logistic function of the difference between two policy/reference log ratios. Replacing  $\pi^*$  with a fitted policy and minimizing the resulting negative log probability gives the central DPO objective.

The identity depends on the reward-comparison model, positive reference support, and the KL-regularized optimization setup. It does not establish that the labels are unbiased, that a restricted fitted policy reaches the optimum, or that preference alignment guarantees factual correctness. This can be an offline preference-learning problem: collecting a fixed comparison dataset does not itself make the analysis online learning.

## Design the holdout around the intended generalization

Predicting new comparisons from familiar judges is different from predicting preferences of new judges. Likewise, a new answer pair for a familiar prompt is different from an entirely new prompt. Choose the holdout unit to match the intended deployment: rows, judges, prompts, or a combination.

If both judges and prompts are reused, the dependence structure is crossed. A judge-only bootstrap does not automatically capture prompt-level dependence, and a prompt-only bootstrap does not automatically capture judge effects. The simple one-cluster analysis in the lab illustrates a particular sampling model; an application with crossed reuse needs a corresponding design and uncertainty analysis.

### Boundary - more rows need not mean more independent information

Report the number of independent judges and prompts as well as the number of comparisons. Keep model selection separate from final evaluation. State which judge and context population the reported preference probability represents.

## 7.5 • Execute the judge and policy experiment

The R lab simulates independent judges with repeated responses and heterogeneous order effects. It fits a working logistic model and compares independent-row, judge-sandwich, and judge-bootstrap standard errors. The conditional simulation coefficients are deliberately not treated as the exact targets of the marginal working model. The final calculation verifies the finite-action policy identity numerically.

```
# Chapter 7: uncertainty with repeated judges, using only base R.
set.seed(2026102107)
options(width = 100)
J <- 40L
m <- 20L
n <- J * m
judge <- rep(seq_len(J), each = m)
relevance <- rnorm(n)
```

(continues on next page)

(continued from previous page)

```

conciseness <- rnorm(n)
order <- sample(c(-1, 1), n, replace = TRUE)
judge_order_bias <- rnorm(J, 0, 1.1)
eta <- 1.1 * relevance + 0.35 * conciseness +
  (0.45 + judge_order_bias[judge]) * order
dat <- data.frame(y = rbinom(n, 1, plogis(eta)), relevance, conciseness, order, judge)
fit <- glm(y ~ 0 + relevance + conciseness + order, family = binomial(), data = dat)
X <- model.matrix(fit)
mu <- fitted(fit)
residual_score <- X * (dat$y - mu)
cluster_score <- rowsum(residual_score, dat$judge, reorder = FALSE)
bread <- solve(crossprod(X, X * as.vector(mu * (1 - mu))))
p <- ncol(X)
correction <- (J / (J - 1)) * ((n - 1) / (n - p))
cluster_cov <- correction * bread %*% crossprod(cluster_score) %*% bread

B <- 150L
boot <- matrix(NA_real_, B, length(coef(fit)))
for (b in seq_len(B)) {
  sampled_judges <- sample.int(J, J, replace = TRUE)
  idx <- unlist(lapply(sampled_judges, function(j) which(dat$judge == j)),
    use.names = FALSE)
  f <- suppressWarnings(glm(y ~ 0 + relevance + conciseness + order,
    family = binomial(), data = dat[idx, ]))
  if (f$converged && all(is.finite(coef(f)))) boot[b, ] <- coef(f)
}
ok <- complete.cases(boot)
out <- cbind(estimate = coef(fit), iid_se = sqrt(diag(vcov(fit))),
  judge_sandwich_se = sqrt(diag(cluster_cov)),
  judge_bootstrap_se = apply(boot[ok, , drop = FALSE], 2, sd))
cat("Working logistic model with repeated observations from each judge:\n")
print(round(out, 4))
cat("\nBootstrap usable/flagged replicates:", sum(ok), "/", sum(!ok), "\n")
cat("The working coefficients summarize a marginal logistic approximation.\n")
cat("They need not equal the conditional coefficients used to generate the data.\n")
cat("Judge-level resampling assumes independent judges; it is not a two-way bootstrap.\n")

# A finite-action illustration of the KL-regularized preference identity.
ref <- c(.5, .3, .2)
reward <- c(0, .4, 1)
tau <- .7
policy <- ref * exp(reward / tau)
policy <- policy / sum(policy)
pair <- c(3L, 1L)
reward_probability <- plogis(reward[pair[1]] - reward[pair[2]])
ratio_probability <- plogis(tau *
  (log(policy[pair[1]] / ref[pair[1]]) - log(policy[pair[2]] / ref[pair[2]])))
cat("\nFinite-action policy and reference probabilities:\n")
print(round(data.frame(action = 1:3, reference = ref, policy), 4))
cat("Preference probability from rewards / policy log ratios:\n")
print(round(c(reward = reward_probability, log_ratio = ratio_probability), 6))
stopifnot(abs(reward_probability - ratio_probability) < 1e-12)

```

(continues on next page)

(continued from previous page)

```

figure_dir <- Sys.getenv("COURSE_FIGURE_DIR", "")
if (nzchar(figure_dir)) {
  dir.create(figure_dir, recursive = TRUE, showWarnings = FALSE)
  png(file.path(figure_dir, "07-contextual-preferences.png"),
      width = 1000, height = 550, res = 110)
  par(mar = c(5, 4.5, 4, 1))
  se_values <- t(out[, c("iid_se", "judge_sandwich_se", "judge_bootstrap_se")])
  barplot(se_values, beside = TRUE, col = c("#B88A00", "#007A87", "#A2006E"),
          ylim = c(0, 1.25 * max(se_values)), ylab = "Estimated standard error",
          main = "Repeated judges change the uncertainty calculation",
          legend.text = c("Independent rows", "Judge sandwich", "Judge bootstrap"),
          args.legend = list(x = "topleft", bty = "n", cex = .85))
  invisible(dev.off())
}

```

Working logistic model with repeated observations from each judge:

|             | estimate | iid_se | judge_sandwich_se | judge_bootstrap_se |
|-------------|----------|--------|-------------------|--------------------|
| relevance   | 0.6684   | 0.0831 | 0.1047            | 0.1010             |
| conciseness | 0.3127   | 0.0776 | 0.0848            | 0.0786             |
| order       | 0.2026   | 0.0754 | 0.1846            | 0.1906             |

Bootstrap usable/flagged replicates: 150 / 0

The working coefficients summarize a marginal logistic approximation.

They need not equal the conditional coefficients used to generate the data.

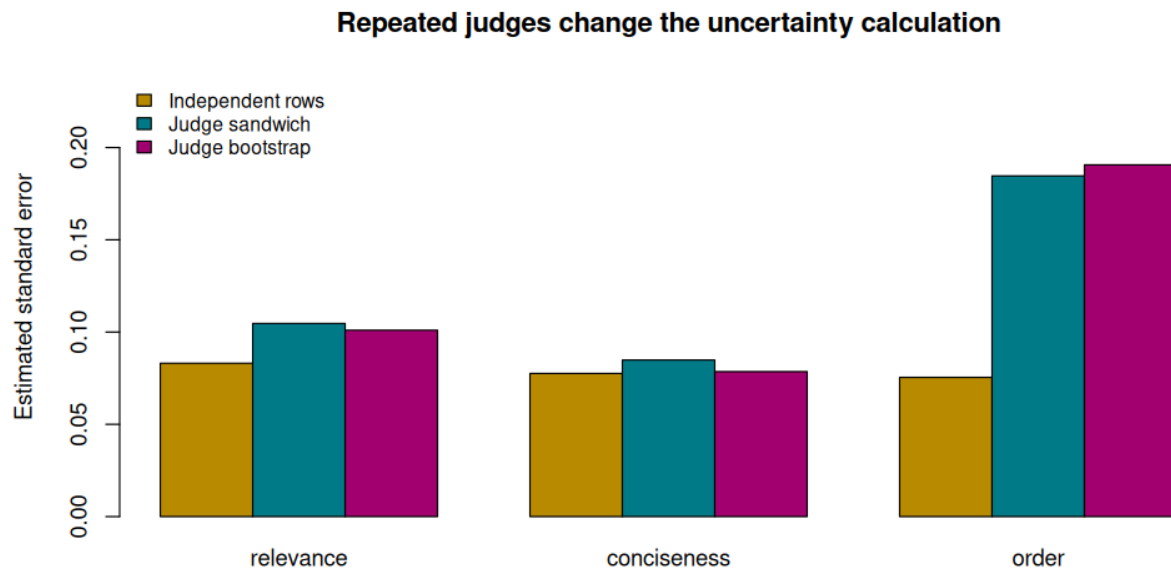
Judge-level resampling assumes independent judges; it is not a two-way bootstrap.

Finite-action policy and reference probabilities:

|   | action | reference | policy |
|---|--------|-----------|--------|
| 1 | 1      | 0.5       | 0.2680 |
| 2 | 2      | 0.3       | 0.2847 |
| 3 | 3      | 0.2       | 0.4473 |

Preference probability from rewards / policy log ratios:

|  | reward   | log_ratio |
|--|----------|-----------|
|  | 0.731059 | 0.731059  |



## 7.6 • Exercises

1. Explain why adding a constant feature to every alternative does not identify an absolute utility intercept.
2. Change the standard deviation of the judge-specific order effects to zero and then increase it. Which standard errors change most? Interpret the result without assuming that a working coefficient equals the conditional generating coefficient.
3. Derive the KL-regularized optimizer and explain why the normalizing constant cancels in a comparison at the same context.
4. Design a train/test split for a dataset containing repeated judges and repeated prompts. State whether your target is performance on new prompts, new judges, or both; identify which dependence remains.

### **Worked solutions**

1. A common feature subtracts to zero in  $\phi(x, a) - \phi(x, b)$ . The observations contain information about relative utility only. A normalization or prior cannot turn this into observed absolute utility.
2. Order-effect uncertainty is especially sensitive because the simulation places a shared random effect on that slope. Changes in other coefficients depend on the design. The three estimators answer an uncertainty question for the fitted working model; none corrects its interpretation into a conditional mixed-model coefficient automatically.

**3.** Differentiating with respect to  $\pi(a)$  gives  $r(a) - \tau\{\log(\pi(a)/\pi_0(a)) + 1\} + \lambda = 0$ . Exponentiate and normalize. The same context has the same  $Z$ , so subtracting rewards removes it. Comparing policies across contexts does not permit cancellation of unrelated normalizers in the same way.

**4.** A new-prompt target requires held-out prompts; a new-judge target requires held-out judges. To evaluate both, reserve both groups and ensure no shared observations leak through either index. For uncertainty about a crossed population, justify a suitable multiway method rather than automatically resampling rows.

## Reading and transition

Turner and Firth (2012) develops structured paired-comparison models and order effects. Rafailov et al. (2023) gives the DPO formulation. Next, change the learning protocol: decisions must be made before each new outcome is available.

## 8. Online learning and regret

### Core idea - Regret is defined by the protocol and comparator

An online decision must use only past information. Its guarantee compares accumulated loss with a specified benchmark under a specified feedback model. Changing the information revealed after a round or allowing the comparator to change decisions changes the problem.

**Question.** How should a learner act when each decision must precede the next observation?

Download lab

Batch learning chooses a rule after observing a training sample. Online learning specifies a sequence: act, observe feedback, update, and act again. The feedback might contain a full loss function or only a partial observation. This chapter begins with full information, where the mathematical distinction between prediction and adaptation is easiest to see.

The central target is cumulative loss relative to a declared comparator. Small regret is a precise guarantee, but its meaning depends on what that comparator is allowed to do.

## The information available at each round

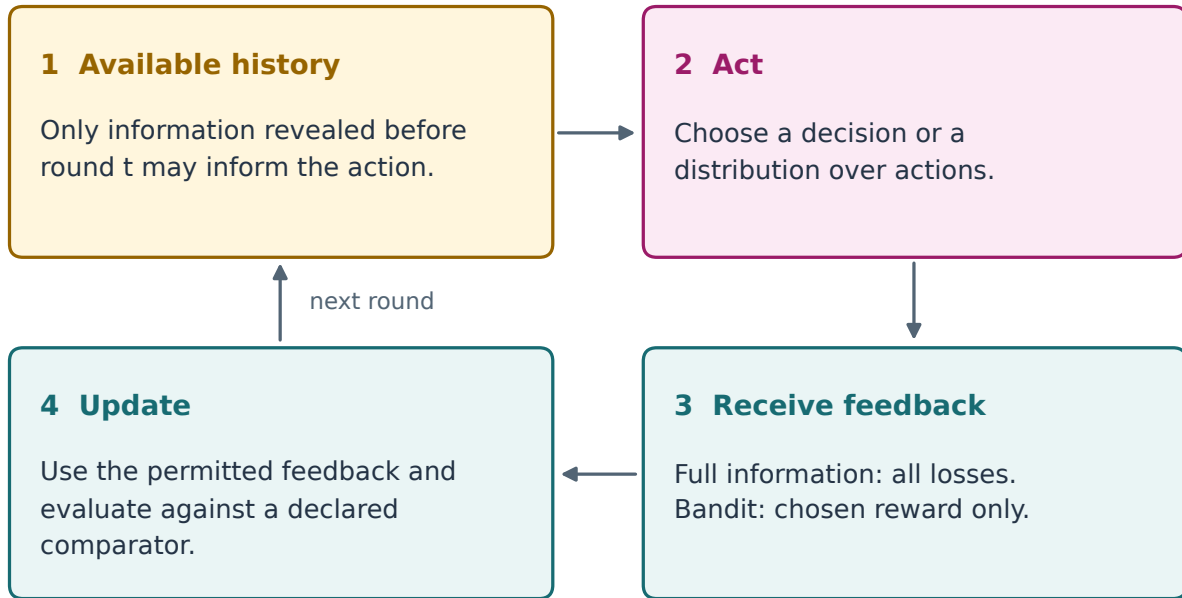


Fig. 4: The action precedes the feedback. The full-information and bandit settings differ in what is revealed after acting; a valid update must respect that boundary.

### 8.1 · Write the protocol before the algorithm

At round  $t$ , the learner chooses  $w_t \in \mathcal{W}$  using information available before the new loss. A loss function  $\ell_t$  is revealed, the learner incurs  $\ell_t(w_t)$ , and it updates its decision. In the full-information setting it can use the observed loss function or an appropriate gradient. In prediction with experts, it sees every expert's loss after acting.

For a fixed decision set, define

$$\text{Regret}_T = \sum_{t=1}^T \ell_t(w_t) - \min_{w \in \mathcal{W}} \sum_{t=1}^T \ell_t(w). \quad (8.1)$$

The comparator is the best **single fixed decision in hindsight**. This is not necessarily the best decision at each time. Regret can be negative, because the learner changes its decision while the comparator cannot. The useful goal is an upper bound growing more slowly than  $T$ .

The sequence need not be iid for the deterministic convex regret argument below. For randomized partial-feedback methods, specify what the environment can observe before choosing outcomes; an adversary that reacts to the current random action can change the guarantee.

## 8.2 · Learn from experts with Hedge

Suppose  $K$  experts supply advice and their losses satisfy  $L_{tk} \in [0, 1]$ . Hedge maintains weights

$$p_{tk} = \frac{\exp\{-\eta \sum_{s < t} L_{sk}\}}{\sum_j \exp\{-\eta \sum_{s < t} L_{sj}\}}.$$

The learner's mixture loss is  $\sum_k p_{tk} L_{tk}$ . It is the conditional expected loss when an expert is sampled using  $p_t$ . With a convex loss of a numerical prediction, combining predictions can sometimes have loss no larger than the mixture loss, by Jensen's inequality; that is an additional property of the prediction setup.

Full-information feedback is essential to this update: all expert losses are observed after the round. The exponential-weights analysis yields a bound of the form  $\log K/\eta + \eta T/8$  for bounded losses and the mixture-loss convention. Balancing the terms gives order  $\sqrt{T \log K}$  regret.

### **Example 8.1 · Good adaptation can still lose on the next round**

**Setup.** Two experts incur losses  $(0, 1)$  on the first round and  $(1, 0)$  on the second. Use uniform initial weights and  $\eta = \log 3$ .

**Work it through.** Round-one mixture loss is  $1/2$ . The updated weights are  $(3/4, 1/4)$ , so round-two mixture loss is  $3/4$ . Total loss is  $1.25$ ; either fixed expert loses exactly  $1$ , so regret is  $0.25$ .

**What this shows.** The weight update uses past evidence and cannot know that the next outcome reverses. A cumulative guarantee does not assert that every update improves the next prediction.

## 8.3 · Derive projected online gradient descent

Let  $\mathcal{W}$  be closed and convex with Euclidean diameter at most  $D$ . Suppose each  $\ell_t$  is convex and a subgradient  $g_t$  at  $w_t$  has norm at most  $G$ . Projected OGD is

$$w_{t+1} = \Pi_{\mathcal{W}}(w_t - \eta g_t).$$

For any fixed comparator  $u \in \mathcal{W}$ , convexity gives  $\ell_t(w_t) - \ell_t(u) \leq g_t^\top(w_t - u)$ . The projection inequality gives

$$\|w_{t+1} - u\|^2 \leq \|w_t - u\|^2 - 2\eta g_t^\top(w_t - u) + \eta^2 \|g_t\|^2.$$

Rearranging and summing cancels the intermediate squared distances. Consequently,

$$\text{Regret}_T \leq \frac{D^2}{2\eta} + \frac{\eta G^2 T}{2}. \quad (8.2)$$

For a known horizon,  $\eta = D/(G\sqrt{T})$  yields  $DG\sqrt{T}$ . The proof makes the assumptions visible: bounded domain, bounded subgradients, convex losses, and updates that occur after the current decision. Smoothness is not required for this bound.

**Example 8.2 · A guarantee with an explicit comparator**

**Setup.** Decisions lie in  $[-1, 1]$  and losses are linear,  $\ell_t(w) = g_t w$  with  $|g_t| \leq 1$ . Take  $T = 100$ .

**Work it through.** Here  $D = 2$ ,  $G = 1$ , and the constant step size is 0.2. Equation (8.2) gives regret at most 20. The best fixed decision has total loss  $-\sum_t g_t$ , while a separate best decision at every time would have loss  $-\sum_t |g_t|$ .

**What this shows.** The guarantee concerns the first comparator. Replacing it with the second changes the problem; the original proof does not establish small regret against an unrestricted sequence of decisions.

## 8.4 · Regret is not automatically statistical risk

Under iid observations  $Z_t$ , let  $\ell_t(w) = \ell(w, Z_t)$  and  $R(w) = E\ell(w, Z)$ . If  $w_t$  depends only on preceding observations, it is independent of the current iid  $Z_t$ . For convex loss, the average iterate  $\bar{w}_T = T^{-1} \sum_t w_t$  then satisfies an online-to-batch inequality of the form

$$E\{R(\bar{w}_T) - R(w^*)\} \leq \frac{E \text{Regret}_T}{T},$$

where  $w^*$  minimizes population risk and the expectations exist. Independence and convexity justify this conversion. An arbitrary dependent or drifting sequence does not permit the same argument merely because an online algorithm was used.

Likewise, a fixed-comparator regret bound does not establish calibrated confidence intervals, validity under arbitrary optional stopping, or successful tracking of unrestricted change. Dynamic or adaptive regret changes the comparator class and generally needs additional structure or a different guarantee.

## Zero static regret can coexist with a large tracking error

Consider an even number  $T$  of rounds with two experts whose losses alternate between  $(0, 1)$  and  $(1, 0)$ . A uniform mixture has loss  $T/2$ , and each single fixed expert also has loss  $T/2$ . Its static regret is exactly zero.

A comparator allowed to switch freely chooses the zero-loss expert each round and incurs zero loss. Relative to that comparator, the same learner loses  $T/2$ . Thus a static-regret guarantee does not promise to track every change in the best action. Dynamic comparisons require restrictions on the comparator or environment to obtain useful guarantees.

**Analysis check - complete the protocol**

Specify what is known before acting, what feedback arrives afterward, and which decisions the comparator may make. In the laboratory, distinguish a mixture's expected loss from the realized loss of a sampled expert; they are different random quantities even when their expectations agree.

**8.5 · Execute the full-information experiment**

The lab supplies all experts' losses after each decision. One expert has the smallest expected loss in the first half, and another in the second. It compares uniform mixing, ordinary Hedge, and a policy told the exact change boundary. The last is an oracle illustration, explicitly using information unavailable to a learner that must detect changes.

The table reports loss relative to both a fixed expert and a two-segment comparator. The OGD example separately checks the deterministic inequality on a realized sequence of linear losses. These examples distinguish a full-information calculation from the bandit experiment in the next chapter.

```
# Chapter 8: full-information Hedge and projected online gradient descent.
set.seed(2026102108)
options(width = 100)
T <- 800L
K <- 3L
losses <- matrix(0, T, K)
for (t in seq_len(T)) {
  error_probability <- if (t <= T / 2) c(.15, .3, .45) else c(.45, .3, .15)
  losses[t, ] <- rbinom(K, 1, error_probability)
}
hedge <- function(L, reset_at = integer()) {
  horizon <- nrow(L)
  eta <- sqrt(8 * log(ncol(L)) / horizon)
  score <- rep(0, ncol(L))
  incurred <- numeric(horizon)
  weights <- matrix(NA_real_, horizon, ncol(L))
  for (t in seq_len(horizon)) {
    if (t %in% reset_at) score[] <- 0
    z <- -eta * score
    w <- exp(z - max(z))
    w <- w / sum(w)
    weights[t, ] <- w
    incurred[t] <- sum(w * L[t, ])
    score <- score + L[t, ] # Update only after the round's loss is incurred.
  }
  list(loss = incurred, weights = weights)
}
ordinary <- hedge(losses)
restart <- hedge(losses, reset_at = T / 2 + 1L)
best_fixed <- min(colSums(losses))
```

(continues on next page)

(continued from previous page)

```

best_two_segments <- min(colSums(losses[1:(T / 2), ])) +
  min(colSums(losses[(T / 2 + 1):T, ]))
summarize <- function(v) c(total_loss = sum(v),
  fixed_regret = sum(v) - best_fixed,
  two_segment_regret = sum(v) - best_two_segments)
cat("Full-information losses; mixture loss is expected loss under sampled experts:\n")
print(round(rbind(uniform = summarize(rowMeans(losses)),
  hedge = summarize(ordinary$loss),
  known_boundary_restart = summarize(restart$loss)), 3))
cat("\nThe restart policy is given the true change boundary; it is an oracle illustration.\n")
cat("A negative fixed-comparator regret is possible because the learner can change weights.\n")
cat("Hedge weights before rounds 1, 400, 401 and 800:\n")
print(round(ordinary$weights[c(1, 400, 401, 800), ], 4))

# Linear convex losses  $\ell_t(w) = g_t * w$  on  $[-1, 1]$ ; diameter  $D=2$  and gradient bound  $G=1$ .
horizon <- 400L
g <- sample(c(-1, 1), horizon, replace = TRUE, prob = c(.55, .45))
D <- 2
G <- 1
eta <- D / (G * sqrt(horizon))
w <- 0
ogd_loss <- numeric(horizon)
for (t in seq_len(horizon)) {
  ogd_loss[t] <- g[t] * w
  w <- max(-1, min(1, w - eta * g[t]))
}
best_constant_loss <- -abs(sum(g))
regret <- sum(ogd_loss) - best_constant_loss
bound <- D^2 / (2 * eta) + eta * G^2 * horizon / 2
cat("\nProjected OGD, fixed-comparator regret and the deterministic upper bound:\n")
print(c(regret = regret, bound = bound, step_size = eta))
stopifnot(regret <= bound + 1e-10)

figure_dir <- Sys.getenv("COURSE_FIGURE_DIR", "")
if (nzchar(figure_dir)) {
  dir.create(figure_dir, recursive = TRUE, showWarnings = FALSE)
  png(file.path(figure_dir, "08-online-learning.png"),
    width = 1100, height = 850, res = 110)
  par(mfrow = c(2, 1), mar = c(4, 4.5, 3, 1))
  colours <- c("#007A87", "#A2006E", "#B88A00")
  matplot(seq_len(T), ordinary$weights, type = "l", lty = 1, lwd = 2,
    col = colours, xlab = "Round", ylab = "Weight before observing losses",
    main = "Ordinary Hedge retains evidence from the first regime")
  abline(v = T / 2, lty = 3)
  legend("topright", legend = paste("Expert", 1:K), col = colours,
    lty = 1, lwd = 2, bty = "n", cex = .8)
  comparator_loss <- losses[, which.min(colSums(losses))]
  loss_differences <- cbind(rowMeans(losses), ordinary$loss, restart$loss) - comparator_
  ↪ loss
  cumulative_difference <- apply(loss_differences, 2, cumsum)

```

(continues on next page)

(continued from previous page)

```

matplot(seq_len(T), cumulative_difference, type = "l", lty = 1, lwd = 2,
        col = colours, xlab = "Round", ylab = "Cumulative loss difference",
        main = "Comparator: the best fixed expert over the complete horizon")
abline(h = 0, v = T / 2, lty = 3)
legend("bottomleft", legend = c("Uniform", "Hedge", "Known-boundary restart"),
      col = colours, lty = 1, lwd = 2, bty = "n", cex = .8)
invisible(dev.off())
}

```

Full-information losses; mixture loss is expected loss under sampled experts:

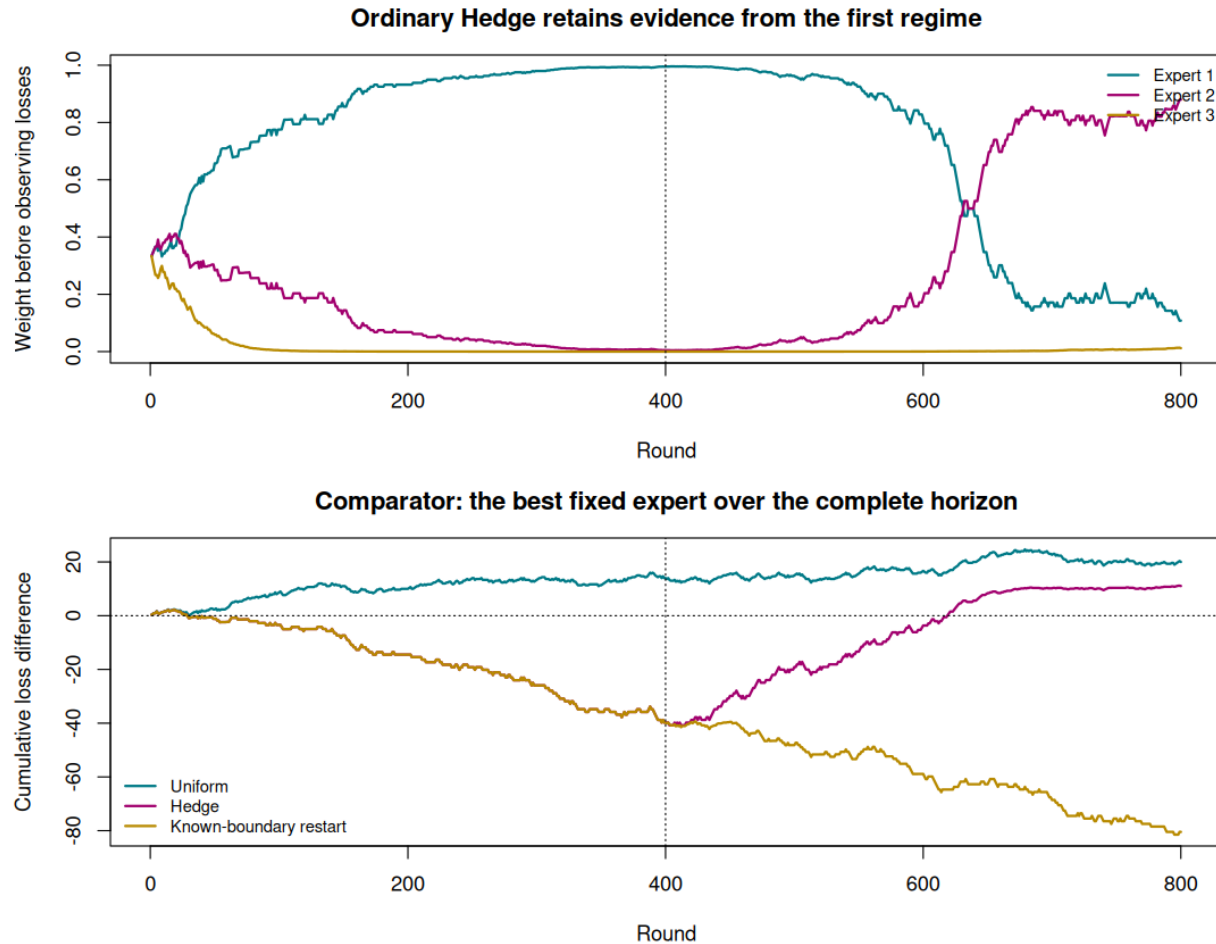
|                        | total_loss | fixed_regret | two_segment_regret |
|------------------------|------------|--------------|--------------------|
| uniform                | 240.000    | 20.000       | 123.000            |
| hedge                  | 231.052    | 11.052       | 114.052            |
| known_boundary_restart | 139.536    | -80.464      | 22.536             |

The restart policy is given the true change boundary; it is an oracle illustration.  
 A negative fixed-comparator regret is possible because the learner can change weights.  
 Hedge weights before rounds 1, 400, 401 and 800:

|      | [,1]   | [,2]   | [,3]   |
|------|--------|--------|--------|
| [1,] | 0.3333 | 0.3333 | 0.3333 |
| [2,] | 0.9953 | 0.0047 | 0.0000 |
| [3,] | 0.9953 | 0.0047 | 0.0000 |
| [4,] | 0.1081 | 0.8799 | 0.0120 |

Projected OGD, fixed-comparator regret and the deterministic upper bound:

| regret | bound | step_size |
|--------|-------|-----------|
| 23.2   | 40.0  | 0.1       |



## 8.6 • Exercises

1. Reproduce Example 8.1 with  $\eta = 0$ . Explain why the result does not show that learning is useless.
2. Complete the projection-and-telescoping proof. What breaks if the gradient has no finite bound on the decision set?
3. Run the experiment without changing the experts' loss probabilities halfway through. Report both comparators and explain when they coincide or differ in a finite sample.
4. A researcher sorts a static dataset into an arbitrary order and calls it a time series. What online-learning questions can still be studied, and which claims about temporal distribution shift are unsupported?

### **Worked solutions**

1. With zero learning rate, the weights stay uniform and total loss is 1, matching the best fixed expert in this particular two-round sequence. Other sequences re-

ward adaptation. A two-round counterexample to monotone improvement is not a universal comparison of algorithms.

2. Bound the first squared distance by  $D^2$ , discard the nonnegative final squared distance, and bound  $\sum_t \|g_t\|^2$  by  $TG^2$ . Without a gradient bound, that sum can be large or require a different data-dependent analysis.

3. Even under stationary expected losses, random outcomes can make the best realized expert differ across halves. The two-segment comparator is at least as flexible and has loss no larger than the best fixed expert. Report this distinction rather than interpreting every difference as evidence of population drift.

4. One can study sequential prediction under the chosen presentation protocol, computation, and regret relative to declared comparators. The imposed order does not supply evidence that the original population evolved over real time. A temporal claim requires a defensible timestamp or longitudinal data-generating interpretation.

## Reading and transition

Read Sections 1.1–1.3 and 3.1 of [Hazan, \*Introduction to Online Convex Optimization\*, second edition](#). The text also develops alternative comparator notions. Next, remove full feedback: choosing an action determines which reward is observed.

## 9. Bandits: learning from chosen actions

### Core idea - An action earns reward and determines what is learned

A bandit policy allocates both actions and information. The exploration bonus protects against prematurely abandoning an uncertain arm; it is not a reward forecast. Evaluate the reward earned during learning separately from the quality of the final recommendation.

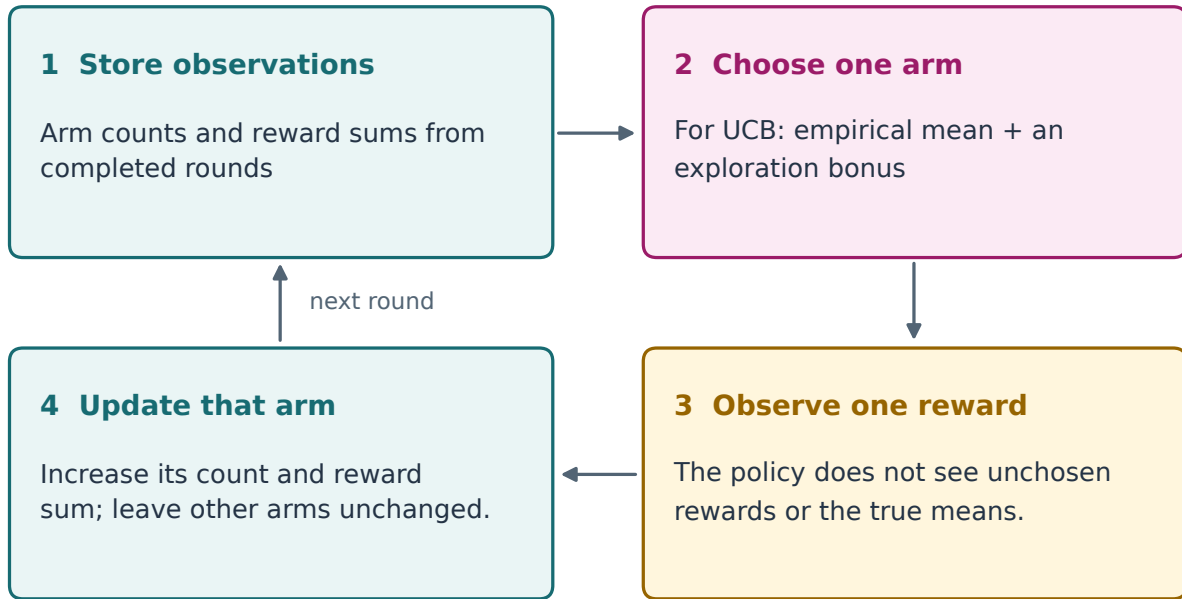
**Question.** How can we choose well while collecting the information needed to learn what is good?

Download lab

An online learner with full information sees the consequences of all candidate predictions after each round. A bandit learner sees only the reward of the action it chose. This creates an allocation problem: repeatedly choosing an apparently good action can prevent discovery of a better one.

We begin with stationary stochastic bandits. The assumptions make the target clear and allow a small experiment whose behaviour can be inspected without a large model or external service.

## The bandit information loop



Simulation evaluator only: true means give pseudo-regret.  
They are never passed into the policy's choice rule.

Fig. 5: Only the chosen reward enters the policy update. The simulation evaluator can use known means for pseudo-regret, but that information must not enter the policy.

### 9.1 · Define the stochastic model

There are  $K$  arms. Arm  $i$  supplies iid rewards with a fixed mean  $\mu_i$ ; for this chapter rewards lie in  $[0, 1]$ . At time  $t$ , the policy chooses  $A_t$  from past actions and observed rewards, then observes  $Y_t$  from that arm. No unchosen reward is revealed to the policy.

The independent-arm-stream formulation is convenient. More general conditional-mean assumptions are possible, but must be stated before importing a theorem. Here means do not drift, reward distributions do not depend on previous actions, and a reward is observed immediately.

Let  $\mu_* = \max_i \mu_i$  and  $\Delta_i = \mu_* - \mu_i$ . The realized allocation defines cumulative **pseudo-regret**

$$\bar{R}_T = \sum_{t=1}^T \Delta_{A_t} = \sum_{i=1}^K \Delta_i N_i(T), \quad (9.1)$$

where  $N_i(T)$  counts pulls of arm  $i$ . Its expectation measures the expected reward lost

through choosing suboptimal means. By contrast,  $T\mu_* - \sum_t Y_t$  also includes reward noise and can be negative. Distinguish these quantities when plotting a simulation.

## 9.2 · Why greedy allocation can fail

A greedy policy samples each arm initially and then repeatedly chooses the largest empirical mean. Early noise can make the best arm look poor. If that arm is never revisited, the error cannot be corrected.

An  $\varepsilon$ -greedy policy selects a random arm with probability  $\varepsilon$  and otherwise acts greedily. A fixed positive  $\varepsilon$  maintains exploration but keeps selecting suboptimal arms at a positive rate, so its asymptotic pseudo-regret is generally linear when gaps are positive. A finite-horizon experiment can still favour a particular tuning: asymptotic orders do not determine every finite-sample ranking.

Bandit learning is therefore connected to experimental design, but its objective includes the rewards obtained during learning. A design chosen only for final estimation precision may spend too much time on actions that are costly in terms of cumulative reward.

## 9.3 · Upper confidence bounds

After pulling each arm once, a common UCB rule for bounded rewards is

$$A_t \in \arg \max_i \left\{ \hat{\mu}_i(t-1) + \sqrt{\frac{2 \log t}{N_i(t-1)}} \right\}. \quad (9.2)$$

The empirical mean rewards an arm that looks promising. The bonus rewards uncertainty and decreases with the number of observations. The index can exceed 1: it is a selection score, not a fitted Bernoulli probability.

For a fixed sample size, Hoeffding's inequality bounds deviations of a bounded sample mean exponentially in its sample size. A bandit proof must additionally handle the adaptively chosen count and repeated opportunities for error. One cannot simply substitute a random count into an arbitrary fixed-sample interval and declare simultaneous or optional-stopping validity.

Under the stationary bounded-reward assumptions, UCB analysis bounds the expected number of pulls of each suboptimal arm by order  $\log T / \Delta_i^2$ , plus lower-order terms. Substituting into Equation (9.1) gives gap-dependent regret of order  $\sum_{i: \Delta_i > 0} \log T / \Delta_i$ . Small gaps make learning harder. This statement is about cumulative decisions, not an automatically valid final confidence interval.

### Example 9.1 · The lower empirical mean can be the sensible next choice

**Setup.** Before round 101, two arms have counts 80 and 20 and empirical means 0.60 and 0.50.

**Work it through.** Their UCB indices are approximately  $0.60 + \sqrt{2 \log 101/80} = 0.940$  and  $0.50 + \sqrt{2 \log 101/20} = 1.179$ . The second arm is selected.

**What this shows.** UCB is not claiming the second arm has an estimated success probability above one. It selects the less-explored arm because plausible improvements still matter to future decisions.

## 9.4 · Cumulative reward and final identification differ

In **best-arm identification**, the output is an arm, and the goal may be to make the probability of a wrong recommendation at most  $\delta$  using few observations. In cumulative-regret minimization, every reward during the experiment matters. The two goals are related but are not interchangeable.

### Example 9.2 · Account for both exploration and a wrong commitment

**Setup.** Two arms have means 0.65 and 0.55. Over a horizon of 1,000, an explore-then-commit policy pulls each arm 50 times and then chooses one arm permanently. Suppose, illustratively, that this pilot leads to the wrong choice with probability 0.10.

**Work it through.** Exploration costs  $50(0.10) = 5$  units of expected reward. A wrong commitment costs an additional  $900(0.10)$ , occurring with probability 0.10, for an expected cost of 9. Total expected pseudo-regret is 14.

**What this shows.** The error probability and its consequences both matter. The 0.10 error probability is a hypothetical input here, not a computed guarantee for this sample size. A real analysis must estimate or bound it under its reward model.

Inference after adaptive allocation is another distinct question. Selecting the largest estimated arm and reporting its ordinary fixed-sample interval can combine allocation effects, winner selection, and stopping. Later chapters study how data-dependent selection changes inferential targets. A low-regret theorem alone does not solve that problem.

## Read the allocation and the regret together

Suppose a stationary three-arm experiment runs for 1,000 rounds. The best arm is selected 720 times, an arm with gap 0.10 is selected 80 times, and an arm with gap 0.02 is selected 200 times. The pseudo-regret for this illustrative allocation is

$$\bar{R}_{1000} = 80(0.10) + 200(0.02) = 12.$$

The optimal-arm selection rate is 72%, but that percentage alone does not determine the cost. Moving the 200 suboptimal pulls to an arm with gap 0.20 raises pseudo-regret to 48 without changing the optimal-arm rate. Conversely, smaller gaps can reduce regret even while making the identity of the best arm harder to learn.

Report the horizon, gaps, allocation counts, cumulative pseudo-regret, and final recommendation. These quantities answer complementary questions.

### Boundary - keep the simulator separate from the policy

The simulator may know every mean and generate unchosen potential rewards to compare policies. The policy must receive only past actions and observed chosen rewards. Passing true means or unchosen outcomes into its action rule changes the feedback model and invalidates the bandit comparison.

## 9.5 · Execute a stationary bandit experiment

The lab compares uniform exploration, fixed- $\varepsilon$  greedy allocation, and UCB. Each simulation generates a table of potential rewards to couple the policies, but a policy receives only its chosen entry. The simulator uses the known means solely to calculate pseudo-regret.

Use the same horizon and initial sampling rule for every policy. The output reports mean pseudo-regret, its Monte Carlo standard error, average reward, and the proportion of optimal-arm selections. It also reports a paired difference. Unfavourable finite-horizon results are legitimate; interpret the horizon, reward gaps, and tuning rather than altering the seed.

```
# Chapter 9: stationary bounded rewards. Base R; no reward stream is shown to a policy.
set.seed(2026102109)
options(width = 100)
means <- c(.25, .40, .55, .65)
K <- length(means)
T <- 1200L
B <- 100L
policies <- c("uniform", "epsilon_0.1", "ucb")
one_run <- function(reward_table, policy) {
  counts <- reward_sum <- numeric(K)
  regret <- total_reward <- optimal_choices <- 0
  for (t in seq_len(nrow(reward_table))) {
    if (t <= K) {
      a <- t
    } else if (policy == "uniform") {
      a <- sample.int(K, 1)
    } else if (policy == "epsilon_0.1") {
      a <- if (runif(1) < .1) sample.int(K, 1) else which.max(reward_sum / counts)
    } else {
      index <- reward_sum / counts + sqrt(2 * log(t) / counts)
      a <- which.max(index)
    }
    y <- reward_table[t, a] # Only this chosen reward is supplied to the update.
    counts[a] <- counts[a] + 1
    reward_sum[a] <- reward_sum[a] + y
    regret <- regret + max(means) - means[a]
    total_reward <- total_reward + y
  }
}
```

(continues on next page)

(continued from previous page)

```

    optimal_choices <- optimal_choices + (a == which.max(means))
  }
  c(pseudo_regret = regret, reward_rate = total_reward / nrow(reward_table),
    optimal_fraction = optimal_choices / nrow(reward_table))
}
ans <- array(NA_real_, dim = c(B, length(policies), 3),
  dimnames = list(NULL, policies, c("pseudo_regret", "reward_rate", "optimal_
↪fraction")))
for (b in seq_len(B)) {
  # Common potential outcomes couple policies for paired comparisons.
  reward_table <- sapply(means, function(p) rbinom(T, 1, p))
  for (j in seq_along(policies)) ans[b, j, ] <- one_run(reward_table, policies[j])
}
out <- cbind(mean_regret = colMeans(ans[, , 1]),
  regret_mcse = apply(ans[, , 1], 2, sd) / sqrt(B),
  mean_reward = colMeans(ans[, , 2]),
  optimal_fraction = colMeans(ans[, , 3]))
cat("Stationary Bernoulli bandits; fixed horizon, all arms sampled once first:\n")
print(round(out, 4))
paired <- ans[, "ucb", "pseudo_regret"] - ans[, "epsilon_0.1", "pseudo_regret"]
cat("\nPaired UCB minus epsilon-greedy pseudo-regret (negative favours UCB):\n")
print(round(c(mean_difference = mean(paired), mcse = sd(paired) / sqrt(B)), 4))
cat("Finite-horizon rankings can depend on gaps, tuning and horizon.\n")
cat("The simulator knows means for scoring; the policies do not.\n")

figure_dir <- Sys.getenv("COURSE_FIGURE_DIR", "")
if (nzchar(figure_dir)) {
  dir.create(figure_dir, recursive = TRUE, showWarnings = FALSE)
  png(file.path(figure_dir, "09-bandits.png"), width = 1000, height = 600, res = 110)
  par(mar = c(5, 4.5, 4, 1))
  boxplot(as.data.frame(ans[, , "pseudo_regret"]),
    names = c("Uniform", "Epsilon = 0.1", "UCB"),
    col = c("#E9D28B", "#A6D4D7", "#DBAACB"),
    ylab = "Cumulative pseudo-regret", main = "Finite-horizon regret across_
↪simulations",
    sub = paste(B, "replications; horizon", T, "; stationary Bernoulli rewards"))
  points(seq_along(policies), out[, "mean_regret"], pch = 18, cex = 1.3)
  legend("topright", legend = "Monte Carlo mean", pch = 18, bty = "n")
  invisible(dev.off())
}

```

Stationary Bernoulli bandits; fixed horizon, all arms sampled once first:

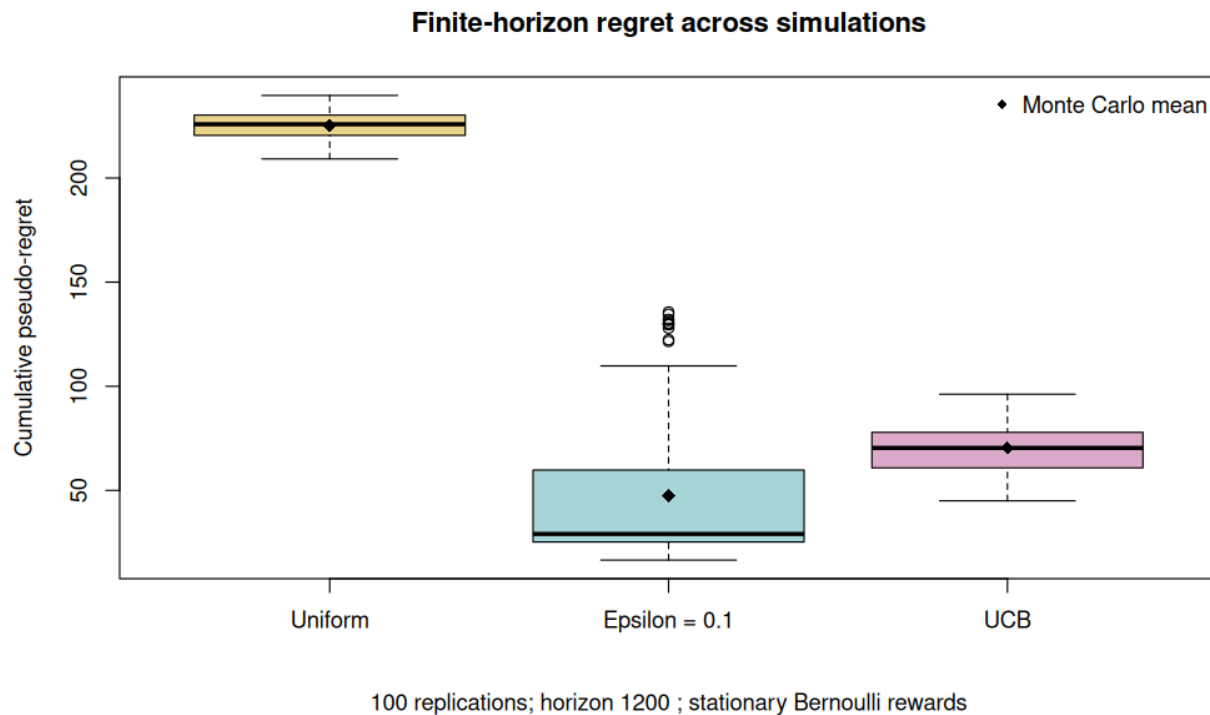
|             | mean_regret | regret_mcse | mean_reward | optimal_fraction |
|-------------|-------------|-------------|-------------|------------------|
| uniform     | 225.1350    | 0.6526      | 0.4613      | 0.2503           |
| epsilon_0.1 | 47.6945     | 3.5296      | 0.6103      | 0.7338           |
| ucb         | 70.3205     | 1.1183      | 0.5908      | 0.6587           |

Paired UCB minus epsilon-greedy pseudo-regret (negative favours UCB):

| mean_difference | mcse   |
|-----------------|--------|
| 22.6260         | 3.7281 |

Finite-horizon rankings can depend on gaps, tuning and horizon.

The simulator knows means for scoring; the policies do not.



## 9.6 • Exercises

1. Derive  $E\bar{R}_T = \sum_i \Delta_i E N_i(T)$ . Explain why reward noise is absent from the pseudo-regret conditional on the chosen arms.
2. Keep the total horizon fixed and move the two largest means closer together. Record regret and optimal-arm selection. Explain why these two metrics can change in different directions.
3. Replace fixed  $\varepsilon = 0.1$  with another declared exploration schedule. Compare it without tuning on the same replications used for the final report.
4. Change all means halfway through. Identify exactly which stationary assumptions fail and specify a more suitable comparator for evaluating tracking.

### **Worked solutions**

1. Group the terms  $\Delta_{A_t}$  by arm. Linearity of expectation yields the displayed identity without requiring that counts be independent of rewards. Pseudo-regret records mean opportunity cost; the action path remains random, but additional reward noise is excluded.
2. Smaller gaps make the identity of the best arm harder to learn, yet a wrong selection costs less. An algorithm can select the optimal arm less often while incurring less pseudo-regret. Report both quantities and the changed population means.

**3.** Use development simulations to select a schedule and independent simulation seeds or scenarios for its final assessment. Otherwise the best-looking tuning is selected on evaluation noise. A schedule's empirical success is not a proof of its regret rate.

**4.** The arm distributions and means are no longer fixed. A comparator that may switch at known segment boundaries is different from the best single arm over the whole horizon. A general tracking claim requires a restriction on change, an appropriate policy, and analysis for that setting.

## Reading and transition

Read Chapters 4, 6, and 7 of [Lattimore and Szepesvári, \*Bandit Algorithms\*](#). They define the stochastic protocol and develop explore-then-commit and UCB analyses. Next, replace numerical rewards with pairwise preference feedback.

# 10. Duelling bandits and adaptive comparisons

## Core idea - Name the winner before searching for it

Pairwise preferences can cycle. A Condorcet winner, a Borda winner, and a best latent strength need not be the same object, and some may not exist. The comparison policy and its performance measure must refer to the winner criterion actually assumed.

**Question.** Which comparison should we request next when feedback says only which alternative wins?

Download lab

Preference learning describes an observation type. Online learning describes a protocol. Duelling bandits combine them: choose a pair, observe a noisy comparison, and use that information to decide what to compare or deploy next. The bridge is useful for recommendation, evaluation, and adaptive experimental design, but it requires a precise definition of a good alternative.

This chapter specifies competing winner criteria, illustrates cycles, and develops a transparent baseline. Its experiment is deliberately small enough that every assumption and regret calculation can be inspected.

## 10.1 · The preference matrix

Let  $P_{ij}$  be the probability that item  $i$  wins against  $j$ . In a binary comparison model without ties,

$$P_{ij} + P_{ji} = 1, \quad P_{ii} = 1/2.$$

At time  $t$ , the learner selects  $(I_t, J_t)$  using past comparisons and observes a Bernoulli outcome with probability  $P_{I_t J_t}$ . The basic stochastic setting assumes a fixed matrix and conditionally independent comparison noise. Repeated judges, context changes, ties, or adversarial responses require a modified model.

Bradley-Terry is a structured submodel:  $P_{ij} = \sigma(\theta_i - \theta_j)$ . A unique largest strength then gives an item that beats every other with probability greater than one half. General

preference matrices need not have this structure.

## 10.2 · Declare what winning means

Three common targets are distinct:

| Criterion        | Definition                                            | What it emphasizes                                 |
|------------------|-------------------------------------------------------|----------------------------------------------------|
| Condorcet winner | $P_{wi} > 1/2$ for every $i \neq w$                   | Beats every opponent individually                  |
| Borda winner     | Maximizes $b_i = (K - 1)^{-1} \sum_{j \neq i} P_{ij}$ | Average win probability against a uniform opponent |
| Copeland winner  | Maximizes the number of opponents with $P_{ij} > 1/2$ | Number of pairwise majorities                      |

Borda scores depend on the opponent distribution. Uniform weighting is a target choice, not a law of preference. Copeland scores ignore the sizes of probability margins. A Condorcet winner may not exist, even though Borda or Copeland maximizers exist in a finite set.

### **i** Example 10.1 · A cycle with a Borda winner

**Setup.** A beats B with probability 0.75, B beats C with probability 0.65, and C beats A with probability 0.60.

**Work it through.** Every item loses to another, so there is no Condorcet winner. The Borda scores are  $b_A = (0.75 + 0.40)/2 = 0.575$ ,  $b_B = (0.25 + 0.65)/2 = 0.450$ , and  $b_C = (0.60 + 0.35)/2 = 0.475$ . A is the Borda winner. All three have one pairwise majority, so their Copeland scores tie.

**What this shows.** Saying only “find the best item” leaves the statistical target incomplete. A method justified under a Condorcet assumption cannot acquire that assumption by silently substituting the Borda winner.

## 10.3 · A declared regret convention

Suppose a Condorcet winner  $w$  exists. Define gaps  $\Delta_i = P_{wi} - 1/2$ , with  $\Delta_w = 0$ . One convention for strong cumulative pseudo-regret is

$$R_T^{\text{strong}} = \sum_{t=1}^T \frac{\Delta_{I_t} + \Delta_{J_t}}{2}. \quad (10.1)$$

The factor one half is a normalization; state it when comparing implementations. A weak-regret objective can instead use  $\min(\Delta_{I_t}, \Delta_{J_t})$ , which incurs zero whenever one selected item is the winner. These objectives can prefer different policies.

Self-comparisons also affect the attainable benchmark. Choosing  $(w, w)$  incurs zero under Equation (10.1) but provides no new preference information. If the application forbids self-comparisons and every other item has a positive gap, the best permitted pair has positive per-round cost under this convention. The action space and benchmark must agree.

**i Example 10.2 · A comparison can be informative and costly**

**Setup.** There is a Condorcet winner  $W$  with  $P_{WA} = 0.70$  and  $P_{WB} = 0.60$ .

**Work it through.** The gaps are  $\Delta_A = 0.20$  and  $\Delta_B = 0.10$ . A duel between  $A$  and  $B$  has strong pseudo-regret 0.15. A duel between  $W$  and  $A$  has strong pseudo-regret 0.10 but weak regret zero. A self-comparison of  $W$  has zero regret and no information about another item.

**What this shows.** Estimating all comparisons well and obtaining good choices while learning are different goals. A policy that quickly stops comparing may look efficient until it commits to the wrong item.

## 10.4 · Connect adaptive comparison to optimal design

For a fixed Bradley-Terry parameter, Chapter 6 derived information contributions

$$I_{ij}(\theta) = P_{ij}(1 - P_{ij})(e_i - e_j)(e_i - e_j)^T.$$

An estimation-oriented design can allocate comparisons to minimize uncertainty in a specified contrast or a collection of contrasts. An online policy must additionally weigh the opportunity cost of comparing inferior items. For instance, repeatedly comparing two nearly tied middle-ranked items may be useful for estimating their difference while contributing little to identifying the overall winner.

A pilot estimate can guide later allocations, but preserve sufficient graph connectivity for the desired contrasts. Greedy selection can neglect entire components. A local information criterion depends on unknown strengths and is not automatically optimal under misspecification or a different winner definition.

Adaptive sampling changes the inference argument. A correctly specified conditional likelihood can account for outcomes given predictable pair choices, but its standard errors and stopping behaviour still need an appropriate justification. A naive bootstrap of individual adaptively collected rows need not reproduce the sampling protocol. Low regret, precise final estimation, and valid inference are three separate claims.

## A model restriction predicts comparisons you did not observe

Suppose  $p_{AB} = p_{BC} = 0.7$ . An unconstrained preference matrix leaves  $p_{AC}$  unspecified. Under Bradley-Terry, however, log odds add along the path:

$$\text{logit}(p_{AC}) = 2 \log(0.7/0.3), \quad p_{AC} = \frac{49}{58} \approx 0.845.$$

If direct comparisons instead give a population value  $p_{AC} = 0.3$ , the three probabilities form a cycle and contradict that constant-strength model. The ability to infer an unobserved comparison came from transitivity imposed by the model, not from the two observed edges alone. In finite data, use uncertainty when assessing such a discrepancy.

### Boundary - a winner criterion is an assumption about the task

Do not declare a missing Condorcet winner to be an optimizer failure. Choose and report a criterion appropriate to the application, then evaluate the policy with the corresponding regret or identification loss. State any structural assumptions used to predict unobserved pairs.

## 10.5 · A transparent explore-then-commit baseline

The lab uses a fixed Bradley-Terry matrix with a unique Condorcet winner. During a balanced pilot, every pair is compared equally often. It estimates each item's average win frequency and selects the largest. Under Bradley-Terry, the highest strength is also the unique Borda maximizer, so this selection rule is aligned with the intended winner in this specific model.

After the pilot, the policy commits to self-comparing the selected item. A baseline continues balanced comparisons throughout. The code calculates strong pseudo-regret from the known simulated matrix and records whether the selected item is correct. Those probabilities are available to the evaluator, not to the learning rule.

This baseline is not asserted to be regret-optimal. With a fixed pilot and nonzero probability of a wrong commitment, extending the horizon indefinitely can produce linear expected regret. More sophisticated duelling algorithms continue collecting targeted evidence and require their own assumptions and analyses. The separate cyclic example demonstrates why the Condorcet regret target cannot simply be carried over to every matrix.

```
# Chapter 10: winner definitions, cycles, and a transparent duelling policy.
set.seed(2026102110)
options(width = 100)
theta <- c(-.8, -.3, 0, .4, .9)
K <- length(theta)
P <- plogis(outer(theta, theta, "-"))
```

(continues on next page)

(continued from previous page)

```

condorcet <- function(P) {
  which(vapply(seq_len(nrow(P)), function(i) all(P[i, -i] > .5), logical(1)))
}
borda <- function(P) (rowSums(P) - diag(P)) / (ncol(P) - 1)
winner <- condorcet(P)
stopifnot(length(winner) == 1L)
delta <- P[winner, ] - .5
pairs <- t(combn(seq_len(K), 2))
T <- 600L
explore <- 150L # Multiple of the number of pairs, so exploration is balanced.
B <- 150L
out <- matrix(NA_real_, B, 3, dimnames = list(NULL,
  c("balanced_regret", "explore_commit_regret", "selected_correctly")))
for (b in seq_len(B)) {
  wins <- matrix(0, K, K)
  trials <- matrix(0, K, K)
  balanced_regret <- explore_regret <- 0
  selected <- NA_integer_
  for (t in seq_len(T)) {
    pair <- pairs[(t - 1L) %% nrow(pairs) + 1L, ]
    i <- pair[1]; j <- pair[2]
    round_regret <- (delta[i] + delta[j]) / 2
    balanced_regret <- balanced_regret + round_regret
    if (t <= explore) {
      y <- rbinom(1, 1, P[i, j])
      wins[i, j] <- wins[i, j] + y
      wins[j, i] <- wins[j, i] + 1 - y
      trials[i, j] <- trials[i, j] + 1
      trials[j, i] <- trials[j, i] + 1
      explore_regret <- explore_regret + round_regret
      if (t == explore) selected <- which.max(rowSums(wins) / rowSums(trials))
    } else {
      # Commit to (selected, selected): no informative comparison is collected.
      explore_regret <- explore_regret + delta[selected]
    }
  }
  out[b, ] <- c(balanced_regret, explore_regret, selected == winner)
}
cat("Bradley-Terry preference matrix:\n")
print(round(P, 3))
cat("Condorcet winner:", winner, "\n")
cat("Borda scores:\n"); print(round(borda(P), 3))
cat("\nStrong pseudo-regret uses half the sum of the two winner gaps:\n")
print(round(rbind(mean = colMeans(out), mcse = apply(out, 2, sd) / sqrt(B)), 4))
cat("Explore-then-commit is a teaching baseline, not a claim of optimal regret.\n")

cycle <- matrix(.5, 3, 3)
cycle[1, 2] <- .75; cycle[2, 1] <- .25
cycle[2, 3] <- .65; cycle[3, 2] <- .35
cycle[3, 1] <- .60; cycle[1, 3] <- .40
cat("\nA cyclic matrix (row item beats column item):\n")
print(cycle)

```

(continues on next page)

(continued from previous page)

```

cat("Condorcet winners:", if (length(condorcet(cycle))) condorcet(cycle) else "none", "\n")
cat("Borda scores and Borda winner:\n")
print(borda(cycle)); print(which.max(borda(cycle)))
cat("A Condorcet-regret target is unavailable for this cyclic example.\n")

figure_dir <- Sys.getenv("COURSE_FIGURE_DIR", "")
if (nzchar(figure_dir)) {
  dir.create(figure_dir, recursive = TRUE, showWarnings = FALSE)
  png(file.path(figure_dir, "10-duelling-bandits.png"),
      width = 1100, height = 550, res = 110)
  par(mfrow = c(1, 2), mar = c(5, 4.5, 4, 1))
  boxplot(as.data.frame(out[, 1:2]), names = c("Balanced", "Explore then commit"),
          col = c("#A6D4D7", "#DBAACB"), ylab = "Strong cumulative pseudo-regret",
          main = "A finite-horizon policy comparison", cex.axis = .85)
  mtext(paste("Correct commitment:", round(mean(out[, 3]), 3)), side = 3, cex = .75)
  barplot(borda(cycle), names.arg = c("A", "B", "C"),
          col = c("#007A87", "#B9C7CA", "#B9C7CA"), ylim = c(0, .7),
          ylab = "Uniform-opponent win probability", main = "A cycle can still have a
↪ Borda winner")
  mtext("There is no Condorcet winner in this matrix", side = 3, cex = .75)
  invisible(dev.off())
}

```

Bradley-Terry preference matrix:

```

      [,1] [,2] [,3] [,4] [,5]
[1,] 0.500 0.378 0.310 0.231 0.154
[2,] 0.622 0.500 0.426 0.332 0.231
[3,] 0.690 0.574 0.500 0.401 0.289
[4,] 0.769 0.668 0.599 0.500 0.378
[5,] 0.846 0.769 0.711 0.622 0.500

```

Condorcet winner: 5

Borda scores:

```
[1] 0.268 0.403 0.489 0.603 0.737
```

Strong pseudo-regret uses half the sum of the two winner gaps:

|      | balanced_regret | explore_commit_regret | selected_correctly |
|------|-----------------|-----------------------|--------------------|
| mean | 113.6962        | 34.2002               | 0.9000             |
| mcse | 0.0000          | 1.4427                | 0.0246             |

Explore-then-commit is a teaching baseline, not a claim of optimal regret.

A cyclic matrix (row item beats column item):

```

      [,1] [,2] [,3]
[1,] 0.50 0.75 0.40
[2,] 0.25 0.50 0.65
[3,] 0.60 0.35 0.50

```

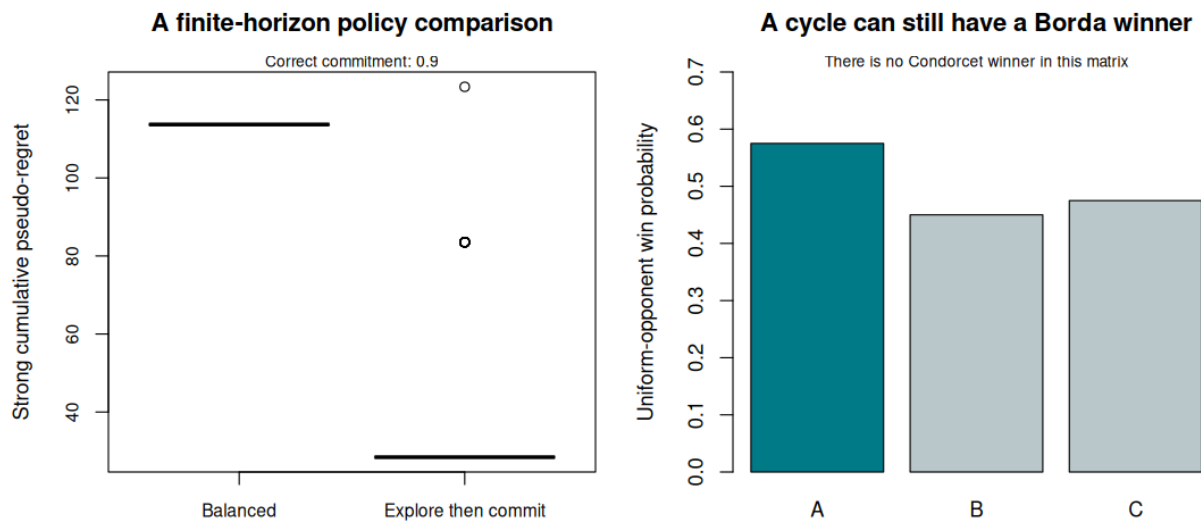
Condorcet winners: none

Borda scores and Borda winner:

```
[1] 0.575 0.450 0.475
```

```
[1] 1
```

A Condorcet-regret target is unavailable for this cyclic example.



## 10.6 • Exercises

1. Verify the three winner criteria in Example 10.1. Define a weighted opponent score  $b_i(q) = \sum_{j \neq i} q_j P_{ij} / (1 - q_i)$  for positive weights summing to one. Use  $q = (0.90, 0.05, 0.05)$  to show that its maximizer can differ from the uniform-opponent Borda winner.
2. Derive the strong and weak regret of the three duels in Example 10.2. Explain why comparing the winner against itself is a modelling convention that may not match every application.
3. Vary the pilot length while keeping the horizon fixed. Report regret, correct selection frequency, and Monte Carlo uncertainty. Explain why no single pilot length must be best across all gap configurations.
4. Design a study whose goal is a confidence interval for one preference contrast rather than cumulative reward. Specify the contrast, budget, comparison allocation, and the inference assumptions that an adaptive version would need.

### **Worked solutions**

1. Each item wins one majority and loses one, so Condorcet fails and Copeland ties. The stated Borda means select A. Under the specified weights, the scores become 0.575, 0.271, and 0.587, so C maximizes the new target. Each denominator conditions the opponent distribution on excluding the item itself. Changing the weighting changes what “average” means; it is an explicit change of target.
2. For A versus B, the strong regret is 0.15 and the weak regret is 0.10. For W versus A, they are 0.10 and 0. Self-comparing W makes both zero. An interface

requiring two distinct displayed items needs a benchmark respecting that restriction.

**3.** A longer pilot leaves fewer rounds for exploitation but can lower the probability of committing incorrectly. Smaller gaps can make the identification problem harder while reducing the cost of mistakes. Keep the pilot budget divisible by the number of pairs if retaining exactly balanced exploration.

**4.** For a fixed design, a contrast-specific information criterion supplies one defensible allocation objective. For an adaptive design, record pair-choice probabilities or the full selection rule, stopping decisions, and independent sampling units; justify the interval under that protocol. Reusing a fixed-design Wald formula is not itself a justification.

## Reading and transition

Yue, Broder, Kleinberg, and Joachims (2012) develops the K-armed duelling-bandit problem. The [Bandit Algorithms text](#) supplies the surrounding sequential-decision framework. The next block asks what inferential claims remain valid after the same data help select the question to be tested.

# 11. Selective inference: asking a question after seeing the data

## Core idea - Selection changes the reference experiment

When a statistic helped choose the question, its ordinary null distribution no longer describes the selected experiment. Conditional inference compares the observation with alternative data that would have produced the specified selection, after handling nuisance information.

## From a selected question to a conditional reference

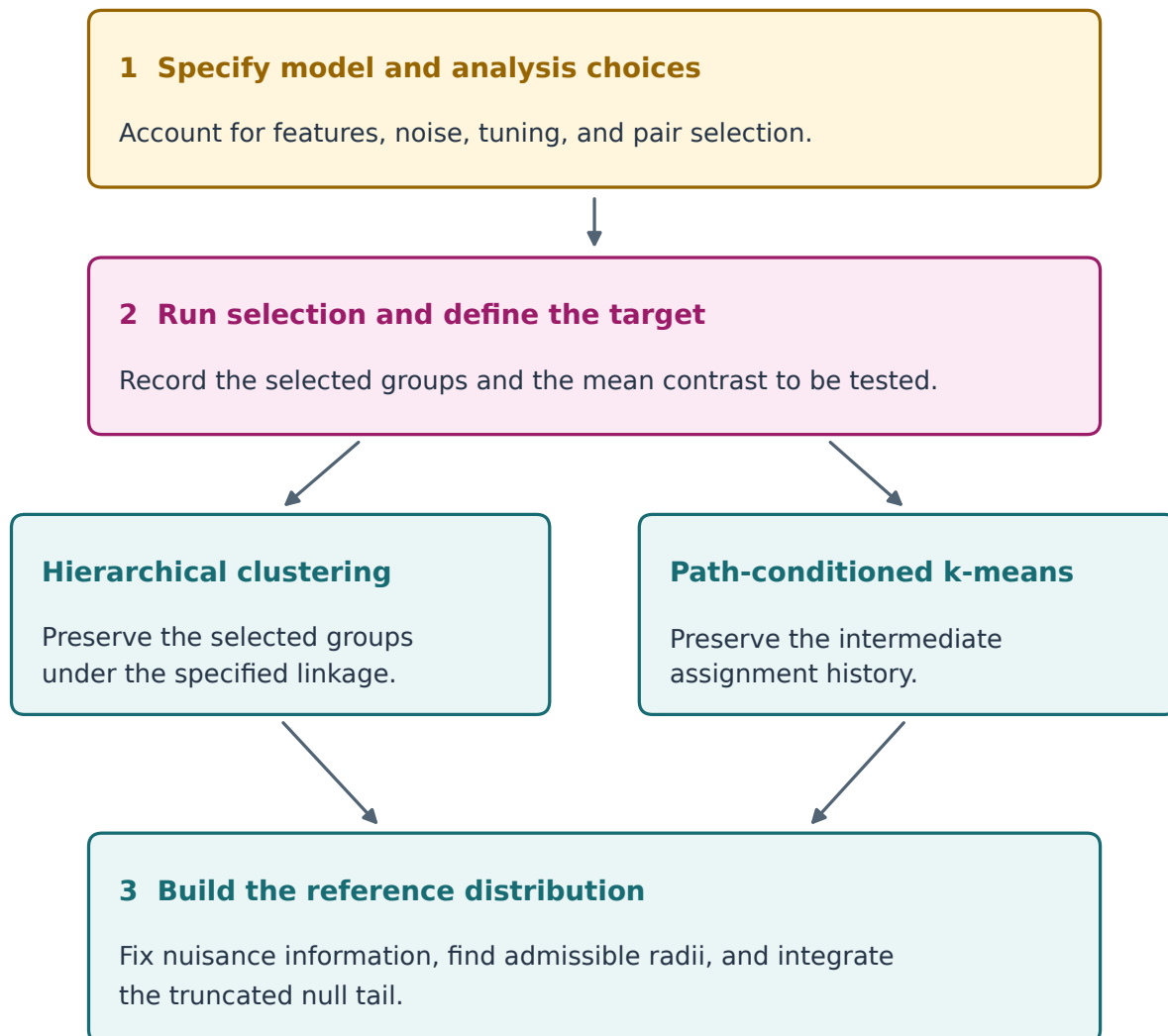


Fig. 6: Hierarchical and path-conditioned k-means procedures preserve different events. After nuisance conditioning, the admissible radii determine the truncated null distribution under the specified model.

## Learning objectives

By the end of this chapter, you should be able to:

1. distinguish a fixed inferential target from a target selected using observations;
2. write a conditional Type I error requirement for a selected hypothesis;
3. derive and compute an exact pivot for the largest absolute Gaussian observation;

4. explain when sample splitting creates the independence an analysis needs;
5. identify additional selection steps that a proposed guarantee has not covered.

The preceding chapters allowed the learner to choose a predictor, a comparison, or an action. We now allow the analyst to choose the *question*. This changes the reference distribution used for inference. A large estimate may be large partly because it won a search.

## 11.1 A selected parameter is still a parameter

Suppose that

$$Z_j \sim N(\theta_j, 1), \quad j = 1, \dots, m,$$

independently. Each  $\theta_j$  is a fixed unknown number. An analyst inspects all observations and reports

$$J = \operatorname{argmax}_j |Z_j|.$$

Ties have probability zero under this continuous model; the lab nevertheless uses a deterministic tie rule. The reported target is  $\theta_J$ . Its index is random, but the list of possible parameters was defined before observing the data. Saying which parameter was selected is part of specifying the inferential procedure.

For a fixed coordinate,  $2\{1 - \Phi(|Z_j|)\}$  is a valid two-sided p-value for  $H_{0j} : \theta_j = 0$ . Applying that formula to the winner does not retain the same calibration among reported hypotheses. Under the global null, the probability that the winning naïve p-value is below  $\alpha$  is

$$1 - (1 - \alpha)^m.$$

For  $m = 20$  and  $\alpha = 0.05$ , this is about 0.642. Increasing the number of opportunities to find an extreme observation changes the distribution of the reported statistic even though each original measurement remains well calibrated.

## 11.2 What probability should be controlled?

One selective requirement is, for each eligible  $j$  and every parameter vector satisfying  $\theta_j = 0$ ,

$$\Pr_{\theta} \{p_j \leq \alpha \mid J = j\} \leq \alpha.$$

Other coordinates need not have zero means. This is a statement about errors among repetitions in which the procedure asks question  $j$ . It differs from familywise error control across a collection of simultaneous rejections. A method can be designed for either goal, but the goal must be stated.

Conditioning more finely can simplify computation. We will also condition on the sign of the selected observation and on the unselected observations. If the resulting

p-value is uniform under each such conditional distribution, averaging over the additional information preserves the desired selective calibration. Conditioning discards information, so computational convenience can have a power cost.

Fithian, Sun, and Taylor formulate this connection between selection, conditional error, and information loss in [Optimal Inference After Model Selection](#).

## 11.3 Deriving an exact selected-Gaussian pivot

Fix a selected index  $j$ , the observed unselected vector  $z_{-j}$ , and the selected sign  $s \in \{-1, 1\}$ . Define

$$c = \max_{k \neq j} |z_k|, \quad T = sZ_j.$$

Before selection,  $T \sim N(s\theta_j, 1)$  and is independent of  $Z_{-j}$ . The selection event with sign  $s$  is simply  $T \geq c$ . Consequently,

$$f_{T|J=j, S=s, Z_{-j}}(t) = \frac{\phi(t - s\theta_j)}{1 - \Phi(c - s\theta_j)} \mathbf{1}(t \geq c).$$

Under  $H_{0j}$ , an upper-tail conditional p-value is

$$p_{\text{sel}} = \frac{1 - \Phi(|z_j|)}{1 - \Phi(c)}.$$

This upper tail tests extremeness in the direction selected by the data, while conditioning on that direction. It is not a claim that the direction was specified in advance. The probability integral transform gives a uniform p-value under the conditional null. The conditioning threshold  $c$  is therefore an essential part of the calculation.

For numerical stability, evaluate the ratio by subtracting log survival probabilities and exponentiating. Direct subtraction of a normal CDF from one loses precision for extreme observations.

### **Example 11.1 — A large observation with a strong competitor**

Let  $z = (2.4, -1.7, 0.5)$ . Coordinate 1 is selected, its sign is positive, and  $c = 1.7$ . The ordinary two-sided p-value is approximately 0.0164. The selective value is

$$\frac{1 - \Phi(2.4)}{1 - \Phi(1.7)} \approx 0.184.$$

The observation is unusual for an unselected standard normal, but much less unusual among positive observations known to exceed 1.7. This calculation does not establish that  $\theta_1 = 0$ . It explains why the original reference distribution overstates the evidence against that hypothesis.

**i Example 11.2 — A global-null test answers a different question**

Under  $\theta_1 = \dots = \theta_m = 0$ , the maximum  $M = \max_j |Z_j|$  has CDF

$$\Pr(M \leq t) = \{2\Phi(t) - 1\}^m, \quad t \geq 0.$$

Thus  $1 - \{2\Phi(M) - 1\}^m$  is an exact global-null p-value. It tests whether *all* coordinates have mean zero. The conditional p-value above instead targets the selected coordinate and permits other means to be nonzero. Substituting one for the other changes the null hypothesis.

## 11.4 From a one-dimensional truncation to polyhedral selection

For Gaussian data  $Y \sim N(\mu, \Sigma)$ , consider a linear contrast  $\eta^\top Y$  and a selection event  $AY \leq b$ . Define

$$T = \eta^\top Y, \quad v = \frac{\Sigma \eta}{\eta^\top \Sigma \eta}, \quad R = Y - vT.$$

Then  $\text{Cov}(R, T) = 0$ , so Gaussianity makes  $R$  and  $T$  independent. Once  $R = r$  is fixed, each selection inequality becomes a bound on the scalar  $T$ :

$$(Av)_i T \leq b_i - (Ar)_i.$$

Intersecting these bounds gives a truncation interval, possibly with zero-probability or vacuous cases. Conditional on  $R$  and selection,  $T$  follows its original Gaussian law restricted to that interval. This is the central geometric idea behind the polyhedral approach of [Lee, Sun, Sun, and Taylor \(2016\)](#).

The assumptions matter: this derivation uses Gaussianity, known covariance, and an affine description of the selection event. An arbitrary clustering or tuning pipeline need not produce one polyhedron. The next chapters use different selection geometry.

## 11.5 What splitting does and does not provide

If an independent discovery dataset chooses a fixed hypothesis and an untouched inference dataset supports a valid test of that hypothesis, ordinary inference can be conditionally valid given discovery. The price is reduced information at each stage.

Independence must concern the *inferential information*, not just a convenient division of row numbers. If held-out observations are assigned to clusters using the same features subsequently tested, their group membership still depends on the test observations. Similarly, selecting a threshold after viewing validation errors is another selection operation. A complete analysis records the question, the information that selected it, and the information used to answer it.

## Translate the conditioning event into a rejection threshold

In the selected-Gaussian example, condition on the other coordinates and on a positive selected sign. Let  $c$  be the largest absolute competing coordinate. Under the selected coordinate's zero-mean null, the remaining observation is a standard normal truncated to  $(c, \infty)$ . For  $z > c$ ,

$$p_{\text{sel}} = \frac{1 - \Phi(z)}{1 - \Phi(c)}.$$

A level- $\alpha$  rejection therefore requires

$$z \geq \Phi^{-1}(1 - \alpha\{1 - \Phi(c)\}).$$

At  $c = 2$  and  $\alpha = 0.05$ , this threshold is about 3.05. Being just larger than the competitor is not unusual within the selected experiment, even when the ordinary Gaussian tail is small. This is a calculation for the stated Gaussian selection model, not a universal adjustment for arbitrary searches.

### Analysis check - describe the repeated experiment

List the selected target, the selection event, the nuisance information conditioned on, and the remaining random statistic. Then state whether the error guarantee is conditional or unconditional. A p-value without that reference experiment is difficult to interpret.

## 11.6 Reproducible Laboratory

The script compares naïve and selective rejection rates under the global null and checks selected null coordinates when one other coordinate has a nonzero mean. It also reports Monte Carlo standard errors. An empirical rejection rate near 0.05 is useful validation of this implementation; the exact guarantee comes from the derivation, not the simulation.

Download lab

Listing 1: Exact inference for the largest absolute independent Gaussian observation

```
# Chapter 11: exact selected-Gaussian inference. Base R only.
# Run with: Rscript labs/11-selective-inference.R
# Optional plots: set COURSE_FIGURE_DIR to an existing or new directory.
set.seed(2026102111)
options(width = 100)

selected_tests <- function(z) {
  stopifnot(is.matrix(z), ncol(z) >= 2L)
  winner <- max.col(abs(z), ties.method = "first")
  t_obs <- abs(z[cbind(seq_len(nrow(z)), winner)])
```

(continues on next page)

(continued from previous page)

```

competitors <- abs(z)
competitors[cbind(seq_len(nrow(z)), winner)] <- -Inf
threshold <- apply(competitors, 1L, max)
log_tail <- pnorm(t_obs, lower.tail = FALSE, log.p = TRUE)
log_denom <- pnorm(threshold, lower.tail = FALSE, log.p = TRUE)
data.frame(
  winner = winner, statistic = t_obs, threshold = threshold,
  naive = 2 * exp(log_tail),
  selective = pmin(1, exp(log_tail - log_denom))
)
}

cat("Example 11.1\n")
print(selected_tests(matrix(c(2.4, -1.7, 0.5), nrow = 1)), digits = 5,
  row.names = FALSE)

repetitions <- 10000L
coordinates <- 20L
alpha <- 0.05
z_null <- matrix(rnorm(repetitions * coordinates),
  nrow = repetitions, ncol = coordinates)
null_result <- selected_tests(z_null)
z_signal <- matrix(rnorm(repetitions * coordinates),
  nrow = repetitions, ncol = coordinates)
z_signal[, 1L] <- z_signal[, 1L] + 2.5
signal_result <- selected_tests(z_signal)
selected_null <- signal_result$winner != 1L

summarize_p <- function(label, p_values) {
  rejected <- mean(p_values <= alpha)
  data.frame(
    procedure = label, repetitions = length(p_values),
    rejection_rate = rejected,
    monte_carlo_se = sqrt(rejected * (1 - rejected) / length(p_values))
  )
}

summary_table <- rbind(
  summarize_p("Global null: naive selected test", null_result$naive),
  summarize_p("Global null: exact selective test", null_result$selective),
  summarize_p("One signal: selected null coordinates",
    signal_result$selective[selected_null])
)

cat("\nSimulation calibration (alpha = 0.05)\n")
print(summary_table, digits = 4, row.names = FALSE)
cat("\nTheoretical global-null naive rejection rate:",
  format(1 - (1 - alpha)^coordinates, digits = 6), "\n")
cat("Known variance = 1; coordinates independent; no estimated nuisance parameters.\n")
stopifnot(all(is.finite(null_result$selective)),
  all(null_result$selective >= 0 & null_result$selective <= 1))

plot_dir <- Sys.getenv("COURSE_FIGURE_DIR", unset = "")

```

(continues on next page)

(continued from previous page)

```

if (nzchar(plot_dir)) {
  dir.create(plot_dir, recursive = TRUE, showWarnings = FALSE)
  png(file.path(plot_dir, "11-selective-inference.png"),
      width = 1400, height = 650, res = 130)
  old_par <- par(mfrow = c(1, 2), mar = c(4.2, 4.2, 3, 1))
  hist(null_result$naive, breaks = seq(0, 1, by = 0.05), freq = FALSE,
      col = "#2f6ca3", border = "white",
      main = "Naive p-values after selection", xlab = "p-value", ylim = c(0, 14))
  abline(h = 1, col = "#bd4b38", lwd = 2, lty = 2)
  hist(null_result$selective, breaks = seq(0, 1, by = 0.05), freq = FALSE,
      col = "#2a8a7c", border = "white",
      main = "Exact selective p-values", xlab = "p-value", ylim = c(0, 1.5))
  abline(h = 1, col = "#bd4b38", lwd = 2, lty = 2)
  par(old_par)
  invisible(dev.off())
}

```

## Computed output

### Example 11.1

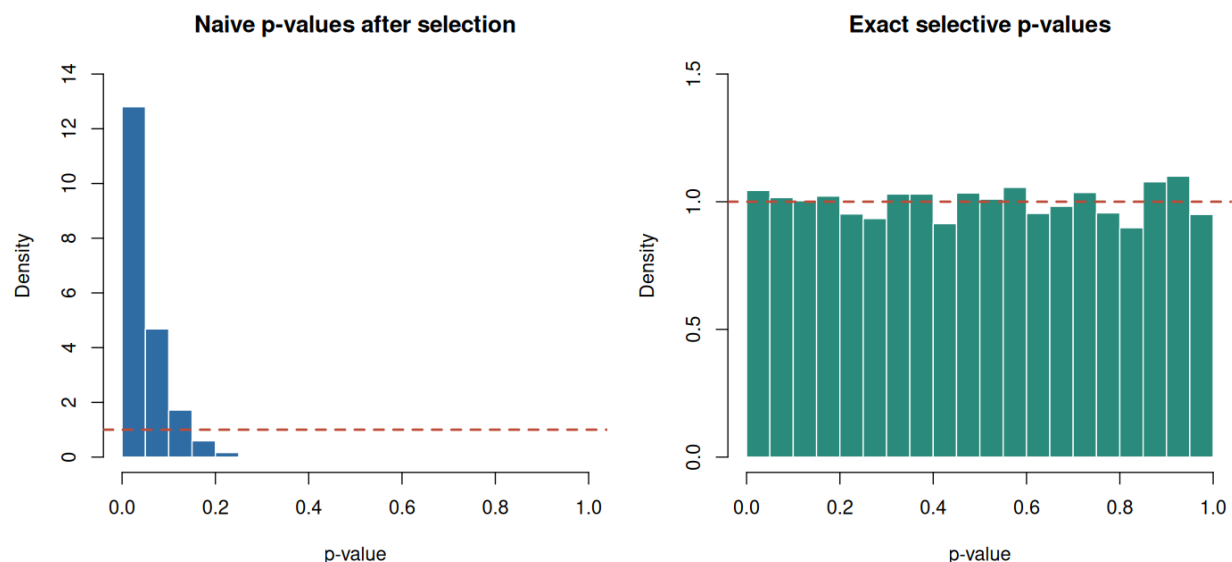
| winner | statistic | threshold | naive    | selective |
|--------|-----------|-----------|----------|-----------|
| 1      | 2.4       | 1.7       | 0.016395 | 0.18394   |

Simulation calibration ( $\alpha = 0.05$ )

|              | procedure                 | repetitions | rejection_rate | monte_carlo_se |
|--------------|---------------------------|-------------|----------------|----------------|
| Global null: | naive selected test       | 10000       | 0.64010        | 0.004800       |
| Global null: | exact selective test      | 10000       | 0.05220        | 0.002224       |
| One signal:  | selected null coordinates | 3683        | 0.05539        | 0.003769       |

Theoretical global-null naive rejection rate: 0.641514

Known variance = 1; coordinates independent; no estimated nuisance parameters.



## Exercises

**Exercise 11.1.** Derive  $1 - (1 - \alpha)^m$  for the naïve selected test under the global null. What happens as  $m$  grows?

**Exercise 11.2.** Prove that the conditional p-value is uniform when  $\theta_j = 0$ , conditional on  $Z_{-j}$ ,  $J = j$ , and the selected sign. Explain why the other means may be nonzero.

**Exercise 11.3.** Derive the lower and upper truncation bounds for  $AY \leq b$  after fixing  $R = r$ . Treat rows with  $(Av)_i = 0$ .

**Exercise 11.4.** A researcher selects the largest effect, reports its ordinary 95% interval, and applies a Bonferroni correction to a separate list of p-values. Has the interval acquired conditional 95% coverage? State the missing argument.

## Solutions

### **i** Solution to Exercise 11.1

Each ordinary two-sided p-value is independent uniform under the global null. The selected p-value is their minimum. Its probability of exceeding  $\alpha$  is  $(1 - \alpha)^m$ , so its rejection probability is the complement. For fixed  $\alpha > 0$ , this probability tends to one.

### **i** Solution to Exercise 11.2

Given the conditioning information,  $T$  has standard-normal density restricted to  $[c, \infty)$ . Its conditional survival function is  $\bar{\Phi}(t)/\bar{\Phi}(c)$ . Applying that survival function to a continuous random variable with this law gives a uniform random variable. Independence makes the conditional distribution of  $Z_j$  unaffected by the observed other coordinates before truncation; their unknown means therefore do not enter this pivot.

### **i** Solution to Exercise 11.3

Write  $a_i = (Av)_i$  and  $d_i = b_i - (Ar)_i$ . Positive  $a_i$  produce upper bounds  $d_i/a_i$ ; negative  $a_i$  produce lower bounds  $d_i/a_i$ . Their intersection is

$$\left[ \max_{i:a_i < 0} d_i/a_i, \min_{i:a_i > 0} d_i/a_i \right],$$

with infinite endpoints when a set of bounds is empty. For  $a_i = 0$ , require  $0 \leq d_i$ ; otherwise the conditional selection event is impossible.

**i Solution to Exercise 11.4**

No. Altering the p-values of another procedure does not alter the reported interval or prove its conditional coverage. A simultaneous confidence procedure could justify a different familywise coverage statement, while a selective interval requires inversion of tests or another argument under the stated conditioning event. The researcher must identify the desired coverage probability and construct an interval satisfying it.

**Reading and next step**

- **Core:** Fithian, W., Sun, D., and Taylor, J. *Optimal Inference After Model Selection*. [arXiv:1410.2597](#), 2014, revised 2017. Read the introduction and the definition of selective error.
- **Core excerpts:** Lee, J. D., Sun, D. L., Sun, Y., and Taylor, J. E. (2016). *Exact post-selection inference, with application to the lasso*. **Annals of Statistics** **44**, 907–927. [Published paper](#), especially the polyhedral argument.

Next we replace the selected coordinate with a pair of groups produced by hierarchical clustering. The same requirement—account for how the question was chosen—remains, but the statistic and selection event change.

# 12. Selective inference after hierarchical clustering

## Core idea - The selected groups define a mean contrast

The null concerns averages of the underlying means in the selected index sets. It is not a test that there are exactly two population types. The clustering rule affects which alternative datasets count in the reference distribution.

## Learning objectives

After this chapter, you should be able to define a selected-group mean contrast, explain why ordinary cluster comparisons can fail, derive the one-dimensional radial perturbation underlying a selective test, and distinguish exact known-variance inference from a plug-in analysis.

Clustering constructs labels from observations. If those labels are then treated as pre-existing scientific groups, an ordinary test ignores the construction step. A visually separated partition is especially easy to produce because separation is an objective of the clustering algorithm itself.

## 12.1 State the model before interpreting a cluster

Consider  $n$  independent observations of  $q$  features:

$$X_i \sim N_q(\mu_i, \sigma^2 I_q), \quad i = 1, \dots, n, \quad \sigma > 0 \text{ known.}$$

The means  $\mu_i$  are fixed unknown vectors and may differ across observations. This is a Gaussian mean model, not an assumed finite-mixture model. Let  $\mathcal{C}(X)$  denote the partition returned by a specified clustering algorithm with a fixed number of clusters  $K$ .

For two non-overlapping index sets  $C_1, C_2$  appearing in that partition, define

$$\bar{\mu}_C = \frac{1}{|C|} \sum_{i \in C} \mu_i, \quad H_0^{C_1, C_2} : \bar{\mu}_{C_1} = \bar{\mu}_{C_2}.$$

The null concerns the underlying means of the observations in these particular groups. It does not say that the entire population has one cluster, that  $K$  is correct, or that the groups represent two biological types. Those are different scientific questions requiring other models or evidence.

The framework and exact hierarchical-clustering procedure were developed by [Gao, Bien, and Witten \(2024\)](#). The article appeared online in 2022; the journal volume is dated 2024.

## 12.2 Why the ordinary reference distribution fails

For fixed groups, let

$$\nu_i = \frac{\mathbf{1}(i \in C_1)}{|C_1|} - \frac{\mathbf{1}(i \in C_2)}{|C_2|}, \quad D = X^\top \nu = \bar{X}_{C_1} - \bar{X}_{C_2}.$$

Then  $\|\nu\|^2 = |C_1|^{-1} + |C_2|^{-1}$  and, under the fixed-group null,

$$\frac{\|D\|^2}{\sigma^2 \|\nu\|^2} \sim \chi_q^2.$$

After clustering,  $\nu$  depends on  $X$ . Substituting the selected  $\nu$  into the fixed-group formula generally changes its null distribution. This is not solved by declaring the selected labels fixed *after* looking at the data: the repeated-sampling procedure still selects them.

### **i Example 12.1 — A significant difference created by the partition**

Suppose two estimated groups each contain ten observations,  $\sigma = 1$ , and their observed mean difference is  $(1.2, 0)^\top$ . A fixed-group calculation gives

$$W = \frac{1.2^2}{1/10 + 1/10} = 7.2, \quad \Pr(\chi_2^2 \geq 7.2) = e^{-3.6} \approx 0.0273.$$

This number does not account for the fact that the groups were chosen to separate the observations. Group sizes and means alone are insufficient to compute the selective p-value: the clustering algorithm and the geometry of the full dataset determine which alternative datasets would produce the same groups.

## 12.3 Reduce the problem to one varying distance

Write the observed contrast length as  $r = \|x^\top \nu\|$  and its direction as  $u = x^\top \nu / r$ . Under the continuous model,  $r = 0$  has probability zero. Define

$$P_\nu^\perp = I_n - \frac{\nu \nu^\top}{\|\nu\|^2}.$$

The data decompose into a component orthogonal to the contrast and a component along it:

$$X = P_\nu^\perp X + \frac{\nu}{\|\nu\|^2} (X^\top \nu)^\top.$$

Condition on  $P_\nu^\perp X = P_\nu^\perp x$  and on the direction of  $X^\top \nu$  being  $u$ . The remaining scalar is the contrast length. A dataset with candidate length  $\phi \geq 0$  is

$$x'(\phi) = P_\nu^\perp x + \frac{\nu}{\|\nu\|^2} \phi u^\top = x + \frac{\nu}{\|\nu\|^2} (\phi - r) u^\top.$$

Under the null, before imposing the clustering event, this length follows

$$\Phi_{\text{rad}} \sim \sigma \|\nu\| \chi_q.$$

The notation  $\Phi_{\text{rad}}$  denotes a random radius here, not the normal CDF. Gaussian independence separates the orthogonal component, and spherical symmetry under the null separates radius from direction.

### **Example 12.2 — What the perturbation actually moves**

If both groups have size ten,  $\|\nu\|^2 = 0.2$ . Increasing the contrast length by  $\Delta$  moves every observation in  $C_1$  by  $+\Delta u/2$  and every observation in  $C_2$  by  $-\Delta u/2$ . Observations outside these groups remain unchanged.

Within-group pairwise differences remain unchanged. The selected groups are pulled apart or pushed together while preserving the nuisance information on which we conditioned. With unequal sizes, the two shifts have unequal magnitudes so that the overall weighted mean is still preserved.

## 12.4 The truncation set remembers the algorithm

Define

$$\mathcal{S}(x) = \{\phi \geq 0 : C_1, C_2 \in \mathcal{C}(x'(\phi))\}.$$

A selective p-value is the conditional upper tail

$$p_{\text{sel}} = \frac{\int_{\mathcal{S}(x) \cap [r, \infty)} f_{\sigma \|\nu\| \chi_q}(t) dt}{\int_{\mathcal{S}(x)} f_{\sigma \|\nu\| \chi_q}(t) dt}.$$

The denominator conditions on observing those groups. The additional conditioning above removes nuisance parameters. Computing  $\mathcal{S}$  is the algorithmic challenge: as  $\phi$  varies, different merges can occur in the dendrogram.

For several hierarchical linkages with squared Euclidean distance, the authors provide an efficient exact characterization. The selected set can contain multiple intervals. A coarse grid of radii is useful for visualization, but it is not an exact replacement for that characterization; narrow intervals or tail intervals can be missed.

For teaching, use average linkage and squared Euclidean distance exactly as in the [authors' tutorial](#). Changing the metric while retaining a formula derived for another metric is not a harmless implementation choice.

## 12.5 What must be fixed or accounted for?

The first laboratory fixes  $K = 2$ , leaving a single pair. It also fixes the distance, linkage, features, and noise scale. If an analyst searches across values of  $K$ , chooses features because they make the separation look stronger, or selects a pair after inspecting its test statistic, the stated conditioning event may no longer describe the full procedure.

The package can estimate the noise scale when it is not supplied. That practical default does not convert an unknown-variance problem into the finite-sample known-variance theorem. Our simulation supplies  $\sigma = 1$ . An application should distinguish an exact guarantee under known covariance from an approximation or asymptotic argument involving estimated covariance.

Likewise, testing every pair introduces a multiple-testing problem. A valid selected-pair p-value alone does not assert familywise or false-discovery-rate control for an arbitrary collection of pairwise tests.

## 12.6 Why ordinary row splitting can still fail

Imagine clustering training observations, training a classifier to reproduce those labels, assigning held-out observations, and testing the held-out features between assigned groups. The assignment uses each held-out observation's own features. Those same features then determine both the group and the test statistic. The null for the selected held-out groups remains data-dependent.

Independent replicate measurements of the same units can provide a different route: discover groups from one measurement and compare their underlying means with an independent measurement, under a suitable joint model. Distribution-specific data thinning is another approach discussed in Chapter 14. Neither construction follows merely from calling some rows a test set.

## Equal average means do not require identical clusters

For fixed selected index sets of size two, consider underlying one-dimensional means  $(-1, 1)$  in one set and  $(-2, 2)$  in the other. Both averages are zero, so the selected-average null holds even though the means within and across the sets differ. This illustrative parameter configuration clarifies the null; it is not a claim about how often the clustering algorithm selects those sets.

Rejection concerns the chosen average contrast under the sampling model and selection adjustment. Non-rejection does not establish identical distributions, homogeneous clusters, or absence of scientific substructure. Specify additional targets when those are the substantive questions.

**Boundary - define all data-dependent choices**

The tested pair, feature set, preprocessing, distance, linkage, and cluster count must be fixed or handled by the inferential procedure. An exact calculation for one pipeline does not automatically cover a search over several pipelines.

## 12.7 Reproducible Laboratory

The required base-R experiment contrasts a fixed partition with average-linkage clustering under one Gaussian cloud. It demonstrates the calibration failure directly. When the authors' clusterpval package is installed, the same script additionally executes its exact known-variance procedure and records how many runs succeeded. It reports explicit status if that extension is unavailable; it never labels a base-R naïve test as selective inference.

Download lab

Listing 2: Fixed versus learned groups, with an optional authors' exact selective procedure

```
# Chapter 12: Gaussian cluster comparisons.
# Required experiment uses base R. The authors' exact method runs when installed.
set.seed(2026102112)
options(width = 145)

n <- 40L
q <- 2L
sigma <- 1
repetitions <- 500L
alpha <- 0.05

wald_p <- function(x, groups, known_sigma = 1) {
  stopifnot(length(unique(groups)) == 2L)
  g1 <- groups == sort(unique(groups))[1L]
  difference <- colMeans(x[g1, , drop = FALSE]) -
    colMeans(x[!g1, , drop = FALSE])
  variance <- known_sigma^2 * (1 / sum(g1) + 1 / sum(!g1))
  pchisq(sum(difference^2) / variance, df = ncol(x), lower.tail = FALSE)
}

samples <- lapply(seq_len(repetitions), function(i) matrix(rnorm(n * q), n, q))
fixed_groups <- rep(1:2, each = n / 2L)
fixed_p <- learned_p <- numeric(repetitions)
for (b in seq_len(repetitions)) {
  x <- samples[[b]]
  tree <- hclust(dist(x)^2, method = "average")
  learned_groups <- cutree(tree, k = 2L)
  fixed_p[b] <- wald_p(x, fixed_groups, sigma)
  learned_p[b] <- wald_p(x, learned_groups, sigma)
}
```

(continues on next page)

(continued from previous page)

```

summarize_p <- function(method, values) {
  rate <- mean(values <= alpha)
  mc_interval <- binom.test(sum(values <= alpha), length(values))$conf.int
  data.frame(method = method, runs = length(values), rejection_rate = rate,
             monte_carlo_se = sqrt(rate * (1 - rate) / length(values)),
             mc_lower = mc_interval[1L], mc_upper = mc_interval[2L])
}

cat("Global null: independent N_2(0, I) observations; fixed K = 2\n")
print(rbind(summarize_p("Fixed groups: ordinary Wald", fixed_p),
            summarize_p("Learned groups: naive Wald", learned_p)),
      row.names = FALSE, digits = 4)
cat("\nExample 12.1: naive p-value for squared standardized contrast 7.2 =",
    format(pchisq(7.2, df = 2, lower.tail = FALSE), digits = 6), "\n")
cat("mc_lower/mc_upper: exact 95% binomial intervals for simulation rejection_
↪probabilities.\n")

selective_plot <- NULL
if (requireNamespace("clusterpval", quietly = TRUE)) {
  selective_reps <- min(100L, repetitions)
  selective_p <- rep(NA_real_, selective_reps)
  errors <- character()
  for (b in seq_len(selective_reps)) {
    x <- samples[[b]]
    tree <- hclust(dist(x)^2, method = "average")
    fit <- tryCatch(
      clusterpval::test_hier_clusters_exact(
        x, link = "average", hcl = tree, K = 2L,
        k1 = 1L, k2 = 2L, sig = sigma
      ),
      error = function(e) e
    )
    if (inherits(fit, "error")) {
      errors <- c(errors, conditionMessage(fit))
    } else {
      selective_p[b] <- fit$pval
    }
  }
  successful <- is.finite(selective_p) & selective_p >= 0 & selective_p <= 1
  cat("\nAuthors' exact procedure EXECUTED; clusterpval version",
      as.character(utils::packageVersion("clusterpval")), "\n")
  cat("Successful finite p-values:", sum(successful), "of", selective_reps, "\n")
  if (all(successful)) {
    selective_plot <- selective_p
    print(summarize_p("Known-variance selective test", selective_p),
          row.names = FALSE, digits = 4)
  } else {
    cat("Calibration summary withheld because some selective runs failed.\n")
    if (length(errors)) print(unique(errors))
  }
  cat("The selective Monte Carlo check is small; its standard error matters.\n")
}

```

(continues on next page)

(continued from previous page)

```

} else {
  cat("\nOptional exact selective procedure NOT EXECUTED: clusterpval is not installed.\n")
  cat("Author installation instructions: https://lucylgao.github.io/clusterpval/\n")
  cat("The base-R results above are ordinary/naive tests, not selective p-values.\n")
}

plot_dir <- Sys.getenv("COURSE_FIGURE_DIR", unset = "")
if (nzchar(plot_dir)) {
  dir.create(plot_dir, recursive = TRUE, showWarnings = FALSE)
  png(file.path(plot_dir, "12-hierarchical-clustering.png"),
      width = 1400, height = 650, res = 130)
  old_par <- par(mfrow = c(1, 2), mar = c(4.2, 4.2, 3, 1))
  x <- samples[[1L]]
  groups <- cutree(hclust(dist(x)^2, method = "average"), k = 2L)
  plot(x, col = c("#2f6ca3", "#bd4b38")[groups], pch = 19,
      xlab = "Feature 1", ylab = "Feature 2", main = "One Gaussian cloud; two learned
↪ groups")
  plot(ecdf(fixed_p), col = "#2a8a7c", lwd = 2, do.points = FALSE,
      xlab = "p-value threshold", ylab = "Empirical rejection probability",
      main = "Calibration under the global null")
  lines(ecdf(learned_p), col = "#bd4b38", lwd = 2, do.points = FALSE)
  legend_labels <- c("Fixed groups", "Learned: naive")
  legend_colors <- c("#2a8a7c", "#bd4b38")
  if (!is.null(selective_plot)) {
    lines(ecdf(selective_plot), col = "#2f6ca3", lwd = 2, do.points = FALSE)
    legend_labels <- c(legend_labels, "Learned: selective")
    legend_colors <- c(legend_colors, "#2f6ca3")
  }
  abline(0, 1, lty = 2, col = "grey40")
  legend("bottomright", legend_labels,
      col = legend_colors, lty = 1, lwd = 2, bty = "n")
  par(old_par)
  invisible(dev.off())
}

```

## Computed output

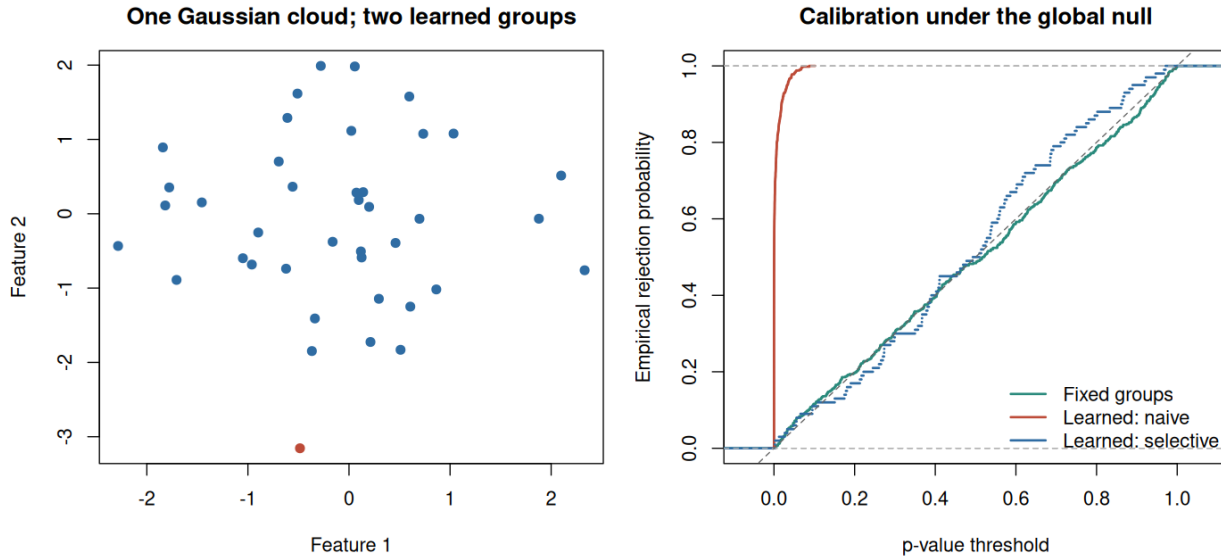
```

Global null: independent N_2(0, I) observations; fixed K = 2
              method runs rejection_rate monte_carlo_se mc_lower mc_upper
Fixed groups: ordinary Wald   500           0.068         0.011258  0.04755  0.09373
Learned groups: naive Wald   500           0.980         0.006261  0.96353  0.99037

Example 12.1: naive p-value for squared standardized contrast 7.2 = 0.0273237
mc_lower/mc_upper: exact 95% binomial intervals for simulation rejection probabilities.

Authors' exact procedure EXECUTED; clusterpval version 1.0.1
Successful finite p-values: 100 of 100
              method runs rejection_rate monte_carlo_se mc_lower mc_upper
Known-variance selective test  100           0.06         0.02375  0.02233  0.126
The selective Monte Carlo check is small; its standard error matters.

```



## Exercises

**Exercise 12.1.** Derive  $\text{Var}(D)$  for fixed groups and obtain the fixed-group chi-square statistic.

**Exercise 12.2.** For  $|C_1| = a$  and  $|C_2| = b$ , derive the two shifts induced by increasing  $\phi$  by  $\Delta$ .

**Exercise 12.3.** An analyst tries five linkages and reports the smallest selective p-value computed separately for each. Explain why the individual procedure's guarantee does not justify that report.

**Exercise 12.4.** In a power simulation, estimated groups sometimes mix two generating populations. Which mean contrast is being tested? Why should recovery probability and rejection probability both be reported?

## Solutions

### **i** Solution to Exercise 12.1

Independence gives  $\text{Var}(X^\top \nu) = \sum_i \nu_i^2 \sigma^2 I_q = \sigma^2(1/a + 1/b)I_q$ . Under the null the contrast mean is zero. Dividing by its common standard deviation gives  $q$  independent standard normals, whose squared sum is  $\chi_q^2$ . The derivation assumes  $\nu$  is fixed; selection is precisely what invalidates using it unchanged for learned groups.

### **i** Solution to Exercise 12.2

The multiplier is  $\nu_i/(1/a + 1/b)$ . It equals  $b/(a + b)$  in group 1 and  $-a/(a + b)$  in group 2. Thus the shifts are  $b\Delta u/(a + b)$  and  $-a\Delta u/(a + b)$ . Their weighted sum is zero, and their difference is  $\Delta u$ .

### **i Solution to Exercise 12.3**

Selecting the linkage by its p-value adds a data-dependent search. Each p-value was calibrated for its specified selection event, not for winning a competition across five analyses. The analyst must pre-specify the procedure or justify a method accounting for the additional search and the desired error criterion. Simply reporting the minimum retains neither the original conditional guarantee nor an automatic multiplicity guarantee.

### **i Solution to Exercise 12.4**

The target is the difference between averages of the underlying  $\mu_i$  over the *estimated* index sets. That can differ from the difference between the two generating population centres. Recovery probability describes whether the intended groups were obtained; conditional power given recovery describes detection after that event. A single unconditional rejection rate mixes these questions and can obscure a poor clustering procedure.

## Reading and next step

- **Core:** Gao, L. L., Bien, J., and Witten, D. (2024). *Selective Inference for Hierarchical Clustering*. **JASA** **119**(545), 332–342. [Open manuscript](#).
- **Implementation:** [clusterpval technical details](#) and [hierarchical tutorial](#).

The next chapter replaces a sequence of hierarchical merges with the iterative assignment history of Lloyd’s k-means algorithm.

# 13. Selective inference after k-means

## Core idea - The algorithmic path is part of the event

For the path-conditioned k-means method studied here, the final partition alone is insufficient. Initialization, intermediate assignments, and stopping choices determine the event being conditioned on. A fitting routine and its inferential routine must describe the same procedure.

## Learning objectives

This chapter develops three skills: translating an iterative learning algorithm into a selection event, distinguishing a selected mean contrast from clustering stability, and evaluating an implementation whose inferential guarantee depends on the algorithmic history.

We retain the known-variance Gaussian model and selected-group mean target from Chapter 12. The new difficulty is computational. A hierarchical procedure selects a sequence of merges. Lloyd's k-means procedure alternates assignments and centre updates. The event describing its output therefore has a different structure.

## 13.1 Lloyd's algorithm is part of the statistical procedure

Given initial centres  $c_1^{(0)}, \dots, c_K^{(0)}$ , Lloyd's algorithm alternates

$$g_i^{(t)} = \underset{k}{\operatorname{argmin}} \|x_i - c_k^{(t-1)}\|^2$$

and

$$c_k^{(t)} = \frac{1}{|\{i : g_i^{(t)} = k\}|} \sum_{i: g_i^{(t)} = k} x_i.$$

Tie handling, empty-cluster handling, initialization, the iteration limit, and stopping rules complete this description. Each assignment step and centre update cannot

increase the within-cluster sum of squares when the algorithm is well defined. Monotonic improvement does not guarantee a global optimum.

Different software can implement different algorithms under the name k-means. Base R's default is not Lloyd's algorithm. Even when Lloyd is requested, an inferential implementation may need information that ordinary fitting software discards. The [KmeansInference tutorial](#) explains why its fitting function retains the intermediate assignments.

**Example 13.1 — The final partition does not describe the full path**

Take one-dimensional observations  $(-3, -1, 1, 4)$  and initialize centres at  $-3$  and  $-1$ . The first assignment puts  $-3$  alone and the other three observations together. Updating gives centres  $-3$  and  $4/3$ . The next assignment gives groups  $\{-3, -1\}$  and  $\{1, 4\}$ , whose updated centres are  $-2$  and  $2.5$ .

Starting instead at  $-3$  and  $4$  reaches the same final groups immediately. These runs share a final partition but not an assignment history. A test that conditions on the full path must describe the path actually used.

## 13.2 Why the fitting criterion rewards the naïve test statistic

For  $K = 2$ , the sum-of-squares decomposition is particularly informative. Let  $n_1, n_2$  be cluster sizes,  $\bar{x}$  the overall mean, and  $D = \bar{x}_{C_1} - \bar{x}_{C_2}$ . Then

$$\text{TSS} = \text{WCSS} + \text{BSS}, \quad \text{BSS} = \frac{n_1 n_2}{n_1 + n_2} \|D\|^2.$$

The ordinary known-variance Wald statistic for fixed groups is

$$W = \frac{\|D\|^2}{\sigma^2(1/n_1 + 1/n_2)} = \frac{\text{BSS}}{\sigma^2}.$$

For a fixed dataset, minimizing within-cluster variation therefore increases precisely the between-group statistic that a naïve test treats as unselected. The problem is built into the optimization objective; it is not only a peculiarity of one software implementation.

**Example 13.2 — Optimization and significance share a statistic**

Suppose  $q = 2$ ,  $\sigma = 1$ , the total sum of squares is 80, and a two-cluster solution has within-cluster sum of squares 48. Then  $\text{BSS} = 32$  and  $W = 32$ . A fixed-group chi-square calculation would give  $e^{-16} \approx 1.13 \times 10^{-7}$ .

These illustrative values describe the calculation, not an executed simulation result. Searching for a small WCSS also searches for a large  $W$ . A selective analysis must account for that search before interpreting the small reference tail as evidence.

### 13.3 Conditioning on the assignment history

Chen and Witten (2023) develop an efficiently computable selective test by conditioning on the intermediate assignments of k-means. The target remains

$$H_0^{C_1, C_2} : \bar{\mu}_{C_1} = \bar{\mu}_{C_2}$$

for the selected groups, under the stated Gaussian model. Their method is specific to the algorithm and conditioning event; a hierarchical-clustering formula cannot simply be reused.

Use the same contrast vector  $\nu$ , observed direction  $u$ , and radius perturbation as before:

$$x'(\phi) = P_\nu^\perp x + \frac{\nu}{\|\nu\|^2} \phi u^\top.$$

Now require the algorithm to reproduce its observed assignment history as  $\phi$  varies. Along this path each observation is affine in  $\phi$ :

$$x'_i(\phi) = a_i + b_i \phi.$$

Once a previous assignment is fixed, each updated centre is an average of such observations and is also affine in  $\phi$ . The condition that observation  $i$  chooses centre  $k$  instead of centre  $\ell$  is

$$\|a_i + b_i \phi - c_k(\phi)\|^2 \leq \|a_i + b_i \phi - c_\ell(\phi)\|^2.$$

Both sides are quadratic in  $\phi$ . Intersecting the constraints across observations, candidate centres, and iterations yields an admissible set of radii. Integrating the appropriate scaled-chi density over this set gives the selective upper tail.

This derivation explains the computational opportunity without claiming that the original selection event is polyhedral in the full data matrix. Quadratic inequalities along a one-dimensional path can be manageable even when a general multidimensional description is complicated.

### 13.4 The choices outside the path also matter

The first demonstration fixes  $K$ , the features, the noise scale, and the initialization rule. The package's fitting and inference routines use the same seed and iteration limit. A selected run from many random restarts is a different procedure: the comparison between restarts is another selection step.

Similarly, choosing  $K$  from a silhouette plot, selecting coordinates after inspecting separation, or learning a transformation and treating it as known can move beyond the stated guarantee. This does not mean those actions are forbidden. It means that a theorem for one specified pipeline does not automatically cover an expanded pipeline.

Conditional validity also has a cost. Conditioning on more of the path can discard information. A procedure with a finer conditioning event may be easier to compute but have lower power than a feasible procedure conditioning less. There is no universal rule that additional conditioning is free.

## 13.5 Three different assessments of a clustering

| Question                                                           | Example quantity                                    | Interpretation                                                |
|--------------------------------------------------------------------|-----------------------------------------------------|---------------------------------------------------------------|
| Does the algorithm produce a similar partition under perturbation? | Pairwise co-membership agreement across runs        | Algorithmic or sampling stability                             |
| Did it recover a stipulated grouping?                              | Adjusted Rand index against known simulation labels | Recovery relative to that reference                           |
| Is a selected mean contrast inconsistent with its null?            | Selective p-value                                   | Evidence under a specified sampling model and selection event |

None of these quantities substitutes for the others. A stable partition of a continuous cloud need not reveal a mixture of populations. An unstable algorithm can still be paired with a correctly calibrated selected-hypothesis test. Recovery metrics require a reference grouping, while an inferential null must identify the relevant population parameter.

For a simulation under alternatives, report how often the intended groups are recovered and what rejection probability is obtained conditional on that recovery, or otherwise describe the actual selected contrasts. This prevents a favourable rejection rate from hiding a changing target.

### Keep a record of the choices that define the event

A reproducible fitting record needs more than the final cluster labels. Retain the input features and transformations, initialization rule, tie and empty-cluster rules, assignment history, stopping criterion, and any comparison among restarts. The path-conditioned test uses the relevant record to reconstruct which perturbed data would have followed that path.

If several restarts are tried and the smallest objective is chosen, that final comparison is another selection step. Saving all seeds makes the search reproducible but does not itself account for the search statistically. Use an inferential method that covers the actual procedure or restrict the analysis to its justified scope.

#### Analysis check - match fitting and inference

Check the algorithm, initialization, iteration limit, and returned path before passing a fit to an inferential routine. Report stability and selected-contrast evidence separately: repeated agreement on a partition is not a substitute for a calibrated test.

## 13.6 Reproducible Laboratory

The base-R component implements a transparent two-centre Lloyd trace and verifies that its objective does not increase. It compares initializations using a label-invariant co-membership measure and demonstrates naïve null rejection after learning groups.

If KmeansInference is installed, an additional block executes the authors' fitting-dependent selective procedure, with fixed arguments, and reports paired naïve and selective results for that same procedure. The script distinguishes this extension from the base-R diagnostic. Initialization agreement is not assumed between two separately written implementations.

Download lab

Listing 3: Lloyd's assignment path, naïve calibration, and the authors' selective method

```
# Chapter 13: transparent Lloyd algorithm and optional authors' inference.
set.seed(2026102113)
options(width = 145)

lloyd_trace <- function(x, initial_indices = c(1L, nrow(x)), max_iter = 30L) {
  centers <- x[initial_indices, , drop = FALSE]
  previous <- rep(0L, nrow(x))
  history <- list()
  objective <- numeric()
  for (iteration in seq_len(max_iter)) {
    distances <- sapply(seq_len(nrow(centers)), function(k)
      rowSums(sweep(x, 2L, centers[k, ], "-")^2))
    groups <- max.col(-distances, ties.method = "first")
    if (length(unique(groups)) != nrow(centers))
      stop("Empty cluster: this demonstration does not silently repair it.")
    centers <- t(vapply(seq_len(nrow(centers)), function(k)
      colMeans(x[groups == k, , drop = FALSE]), numeric(ncol(x))))
    history[[iteration]] <- groups
    objective[iteration] <- sum((x - centers[groups, , drop = FALSE])^2)
    if (identical(groups, previous)) break
    previous <- groups
  }
  list(groups = groups, centers = centers, history = history,
    objective = objective,
    converged = length(history) >= 2L &&
      identical(history[[length(history)]], history[[length(history) - 1L]]))
}

pair_agreement <- function(a, b) {
  keep <- upper.tri(matrix(FALSE, length(a), length(a)))
  mean((outer(a, a, "==") == outer(b, b, "=="))[keep])
}

wald_p <- function(x, groups, sigma = 1) {
  g <- groups == 1L
```

(continues on next page)

(continued from previous page)

```

d <- colMeans(x[g, , drop = FALSE]) - colMeans(x[!g, , drop = FALSE])
pchisq(sum(d^2) / (sigma^2 * (1 / sum(g) + 1 / sum(!g))),
      df = ncol(x), lower.tail = FALSE)
}

n <- 40L
q <- 2L
repetitions <- 400L
samples <- lapply(seq_len(repetitions), function(i) matrix(rnorm(n * q), n, q))
x <- samples[[2L]] # Fixed illustration; the calibration study uses every replicate.
first <- lloyd_trace(x, c(1L, n))
second <- lloyd_trace(x, c(10L, 30L))
cat("Lloyd trace for one Gaussian cloud\n")
print(data.frame(iteration = seq_along(first$objective),
                  within_sum_squares = first$objective),
      row.names = FALSE, digits = 5)
cat("Objective never increases:",
    all(diff(first$objective) <= 1e-8), "\n")
cat("Converged:", first$converged, "\n")
cat("Pairwise co-membership agreement across two fixed initializations:",
    format(pair_agreement(first$groups, second$groups), digits = 4), "\n")

naive_p <- vapply(samples, function(dat) {
  fit <- lloyd_trace(dat)
  wald_p(dat, fit$groups)
}, numeric(1))

summarize_p <- function(method, p) {
  rate <- mean(p <= 0.05)
  mc_interval <- binom.test(sum(p <= 0.05), length(p))$conf.int
  data.frame(method = method, runs = length(p), rejection_rate = rate,
             monte_carlo_se = sqrt(rate * (1 - rate) / length(p)),
             mc_lower = mc_interval[1L], mc_upper = mc_interval[2L])
}
cat("\nBase-R fixed-initialization Lloyd: global-null diagnostic\n")
print(summarize_p("Naive Wald after Lloyd clustering", naive_p),
      row.names = FALSE, digits = 4)
cat("This diagnostic is not a selective-inference procedure.\n")
cat("mc_lower/mc_upper are exact 95% binomial Monte Carlo intervals.\n")
cat("An observed rate of 1 has plug-in MCSE 0 but still has binomial uncertainty.\n")

if (requireNamespace("KmeansInference", quietly = TRUE)) {
  exact_reps <- 60L
  p_selective <- p_naive_author <- rep(NA_real_, exact_reps)
  errors <- character()
  for (b in seq_len(exact_reps)) {
    fit <- tryCatch(
      KmeansInference::kmeans_inference(
        samples[[b]], 2L, 1L, 2L, sig = 1,
        iter.max = 30L, seed = 2026L, verbose = FALSE
      ),
      error = function(e) e
    )
  }

```

(continues on next page)

(continued from previous page)

```

)
if (inherits(fit, "error")) {
  errors <- c(errors, conditionMessage(fit))
} else {
  tab <- summary(fit)
  p_selective[b] <- tab$p_selective[1L]
  p_naive_author[b] <- tab$p_naive[1L]
}
}
valid <- is.finite(p_selective) & p_selective >= 0 & p_selective <= 1 &
  is.finite(p_naive_author)
cat("\nAuthors' path-conditioned method EXECUTED; KmeansInference version",
    as.character(utils::packageVersion("KmeansInference")), "\n")
cat("Successful finite p-values:", sum(valid), "of", exact_reps, "\n")
if (all(valid)) {
  print(rbind(
    summarize_p("Authors' algorithm: naive", p_naive_author),
    summarize_p("Authors' algorithm: selective", p_selective)
  ), row.names = FALSE, digits = 4)
} else {
  cat("Calibration summary withheld because some runs failed.\n")
  if (length(errors)) print(unique(errors))
}
cat("The paired comparison uses the same authors' algorithm and seed.\n")
cat("Sixty repetitions are a demonstration, not a high-precision size study.\n")
} else {
  cat("\nOptional selective procedure NOT EXECUTED: KmeansInference is not installed.\n")
  cat("Author instructions: https://yiqunchen.github.io/KmeansInference/\n")
}

plot_dir <- Sys.getenv("COURSE_FIGURE_DIR", unset = "")
if (nzchar(plot_dir)) {
  dir.create(plot_dir, recursive = TRUE, showWarnings = FALSE)
  png(file.path(plot_dir, "13-kmeans-inference.png"),
      width = 1400, height = 650, res = 130)
  old_par <- par(mfrow = c(1, 2), mar = c(4.2, 4.2, 3, 1))
  plot(x, col = c("#2f6ca3", "#bd4b38")[first$groups], pch = 19,
      xlab = "Feature 1", ylab = "Feature 2",
      main = "Lloyd clusters in a Gaussian cloud")
  points(first$centers, pch = 4, cex = 2, lwd = 3)
  plot(seq_along(first$objective), first$objective, type = "b", pch = 19,
      col = "#2a8a7c", xlab = "Iteration", ylab = "Within-cluster sum of squares",
      main = "Observed optimization path")
  par(old_par)
  invisible(dev.off())
}

```

## Computed output

Lloyd trace for one Gaussian cloud

| iteration | within_sum_squares |
|-----------|--------------------|
| 1         | 66.109             |
| 2         | 65.540             |
| 3         | 63.685             |
| 4         | 61.906             |
| 5         | 61.432             |
| 6         | 61.432             |

Objective never increases: TRUE

Converged: TRUE

Pairwise co-membership agreement across two fixed initializations: 0.5192

Base-R fixed-initialization Lloyd: global-null diagnostic

|                                   | method | runs | rejection_rate | monte_carlo_se | mc_lower | mc_upper |
|-----------------------------------|--------|------|----------------|----------------|----------|----------|
| Naive Wald after Lloyd clustering | 400    | 1    | 0              | 0.9908         | 1        |          |

This diagnostic is not a selective-inference procedure.

mc\_lower/mc\_upper are exact 95% binomial Monte Carlo intervals.

An observed rate of 1 has plug-in MCSE 0 but still has binomial uncertainty.

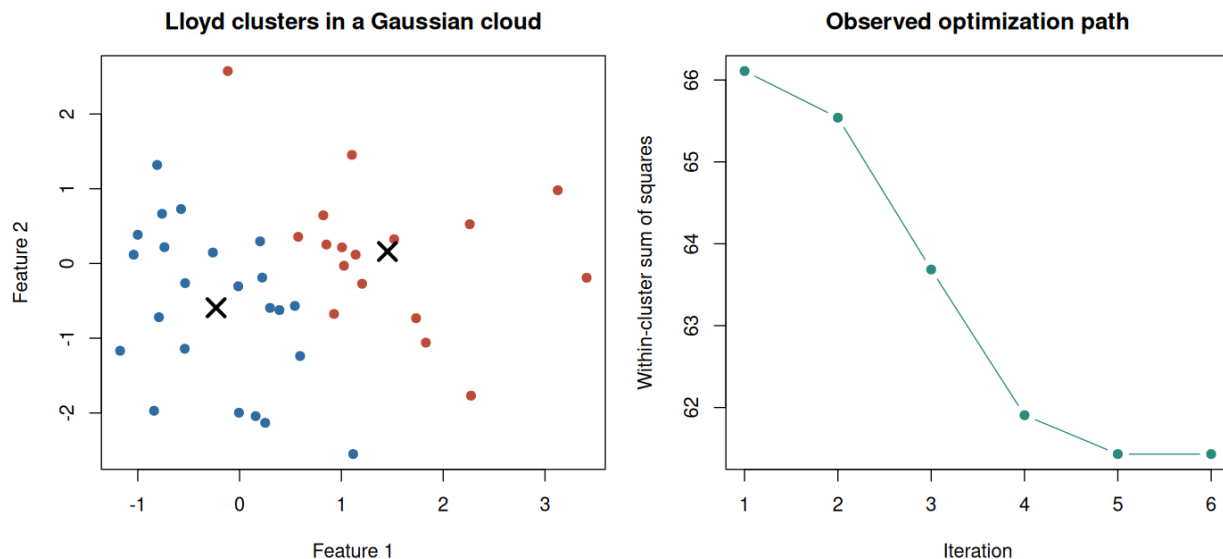
Authors' path-conditioned method EXECUTED; KmeansInference version 0.1.0

Successful finite p-values: 60 of 60

|                               | method | runs    | rejection_rate | monte_carlo_se | mc_lower | mc_upper |
|-------------------------------|--------|---------|----------------|----------------|----------|----------|
| Authors' algorithm: naive     | 60     | 1.00000 | 0.00000        | 0.940371       | 1.0000   |          |
| Authors' algorithm: selective | 60     | 0.03333 | 0.02317        | 0.004063       | 0.1153   |          |

The paired comparison uses the same authors' algorithm and seed.

Sixty repetitions are a demonstration, not a high-precision size study.



## Exercises

**Exercise 13.1.** Prove the two-cluster identity  $BSS = n_1 n_2 \|D\|^2 / (n_1 + n_2)$ .

**Exercise 13.2.** Expand one assignment inequality along  $x'(\phi)$  and show that it is a quadratic inequality. Why can its feasible set be disconnected?

**Exercise 13.3.** A software function returns a final partition but no iteration history. Is that enough to implement the path-conditioned procedure described here? What extra information is required?

**Exercise 13.4.** Construct a research report outline that presents stability, recovery, and selective testing without treating one as proof of another. Specify the null-model part of the experiment.

## Solutions

### **i** Solution to Exercise 13.1

Since  $\bar{x} = (n_1 \bar{x}_1 + n_2 \bar{x}_2) / n$ , we have  $\bar{x}_1 - \bar{x} = (n_2 / n) D$  and  $\bar{x}_2 - \bar{x} = -(n_1 / n) D$ . Therefore

$$n_1 \|\bar{x}_1 - \bar{x}\|^2 + n_2 \|\bar{x}_2 - \bar{x}\|^2 = \frac{n_1 n_2^2 + n_2 n_1^2}{n^2} \|D\|^2 = \frac{n_1 n_2}{n} \|D\|^2.$$

### **i** Solution to Exercise 13.2

Write the two difference vectors as  $r_k + s_k \phi$  and  $r_\ell + s_\ell \phi$ . Subtracting their squared norms gives

$$(\|s_k\|^2 - \|s_\ell\|^2) \phi^2 + 2(r_k^\top s_k - r_\ell^\top s_\ell) \phi + \|r_k\|^2 - \|r_\ell\|^2 \leq 0.$$

A downward-opening quadratic can be nonpositive outside its two roots, producing two rays. Intersections of multiple such conditions with  $\phi \geq 0$  can produce multiple intervals. A single truncation interval cannot be assumed.

### **i** Solution to Exercise 13.3

No. The procedure must recover the initial-centre rule, intermediate assignments, and relevant stopping and tie rules so that it can express the same selection event. Re-running a documented deterministic algorithm from preserved initialization information may recover the history; a bare final partition generally cannot identify it.

**i Solution to Exercise 13.4**

First assess null calibration with independent  $N_q(0, \sigma^2 I_q)$  rows and a known noise scale. Report selective rejection frequency with Monte Carlo uncertainty and all failures. Next introduce a documented mean structure, report recovery of its reference groups, and distinguish conditional power from unconditional rejection. Finally vary initialization or perturbations and show a separate co-membership stability measure. Keep the three headings and interpretations separate.

**Reading and next step**

- **Core:** Chen, Y. T., and Witten, D. M. (2023). *Selective inference for k-means clustering*. **JMLR** 24(152), 1–41. [Paper and code](#).
- **Implementation:** [KmeansInference](#) and its [software tutorials](#).

The final chapter turns the course’s methods into a reproducible research workflow and develops an independent-information comparison using Gaussian data thinning.

# 14. Research synthesis: assumptions, validation, and independent information

## Core idea - Match the evidence to the guarantee

Reproducible computation shows that an analysis can be repeated. Statistical validity also requires the correct target and assumptions. A useful extension changes an assumption deliberately and measures which guarantee survives, rather than treating one successful run as universal evidence.

## Separate discovery and inference information

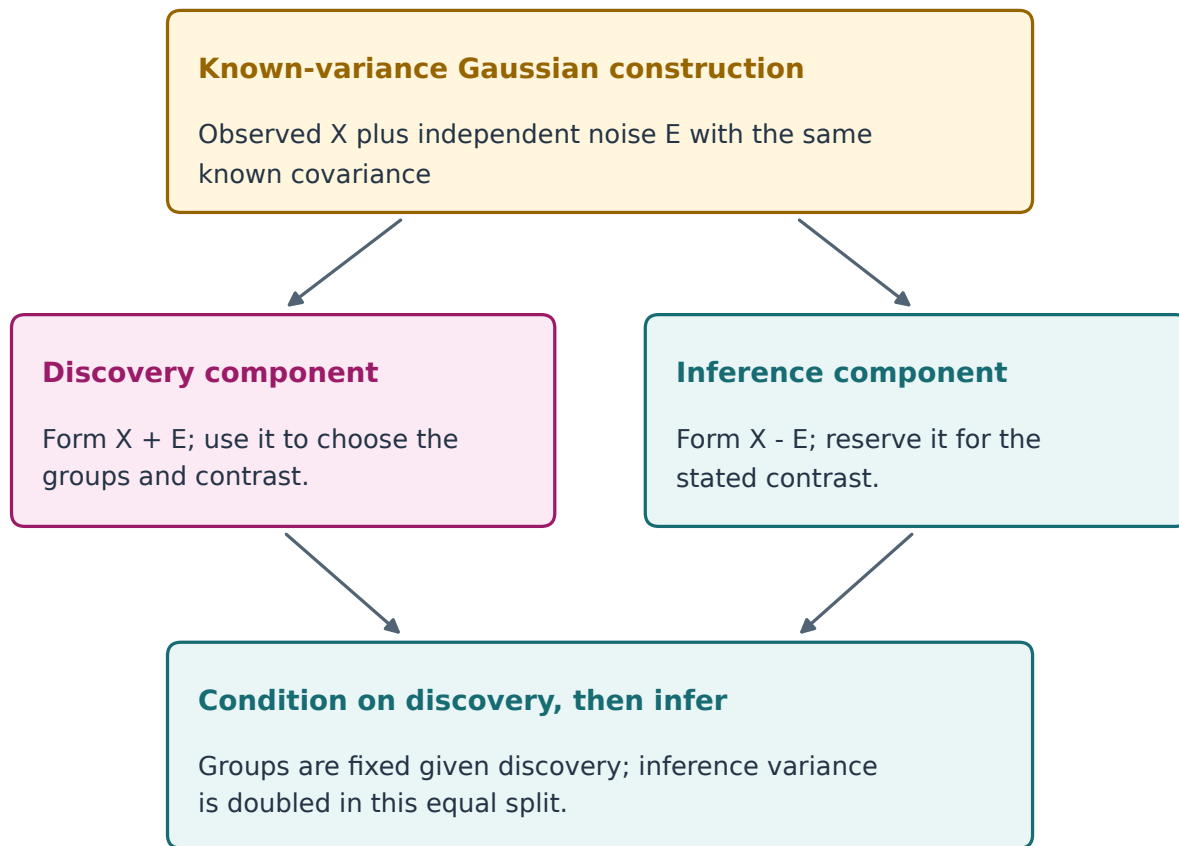


Fig. 7: For the equal split under the known-variance Gaussian model,  $X + E$  and  $X - E$  are independent under joint repeated sampling and each has twice the original variance. Independence is not conditional on the original  $X$ .

## Learning objectives

By the end of this chapter, you should be able to turn a course method into a focused research question, design a sensitivity study that targets its assumptions, separate several kinds of statistical guarantee, and implement Gaussian data thinning as an independent-information validation strategy.

A successful course project does not require a new theorem or training a large language model. It requires a clearly stated problem, a defensible method, a reproducible comparison, and an explanation of what the evidence establishes.

## 14.1 Start with the target and the observation mechanism

The course has considered several ways in which an ordinary supervised-learning problem changes:

| Setting                        | Available information                                   | Main question to resolve                                           |
|--------------------------------|---------------------------------------------------------|--------------------------------------------------------------------|
| Positive-unlabelled learning   | Some positive examples and an unlabelled sample         | Which labelling mechanism and class prior identify risk?           |
| Unlabelled-unlabelled learning | Two mixtures with different class proportions           | Are the shared components identifiable, and is inversion stable?   |
| Preference learning            | Comparisons, possibly repeated within prompts or people | What preference parameter is identified by this comparison design? |
| Online learning                | Sequentially revealed losses or rewards                 | Which comparator and feedback model define regret?                 |
| Selective inference            | A hypothesis chosen using observations                  | Which selection event and sampling model justify error control?    |

The same report template works across these settings: **target, observed data, assumptions, estimator or decision rule, guarantee, and failure mode**. Their guarantees remain different. Small excess prediction risk, small regret, a calibrated confidence interval, and a valid selected-hypothesis test cannot be ranked by a single shared metric.

## 14.2 A focused extension: independent information for the same units

In post-clustering inference, row splitting can leave the group assignment dependent on the feature being tested. An alternative is to split each observation into statistically independent information components while keeping the observation's identity.

Neufeld, Dharamshi, Gao, and Witten (2024) develop data thinning for convolution-closed distributions. We study the Gaussian case because its assumptions and derivation are transparent.

Let

$$X \sim N_n(\mu, \sigma^2 I_n), \quad E \sim N_n(0, \sigma^2 I_n), \quad X \perp E,$$

with known  $\sigma$ . For a discovery-information fraction  $\tau \in (0, 1)$ , form

$$A = \tau X + \sqrt{\tau(1-\tau)}E, \quad B = (1-\tau)X - \sqrt{\tau(1-\tau)}E.$$

These components sum to  $X$ . Their means are  $\tau\mu$  and  $(1-\tau)\mu$ , and their variances are  $\tau\sigma^2 I_n$  and  $(1-\tau)\sigma^2 I_n$ . Their covariance is

$$\tau(1-\tau)\sigma^2 I_n - \tau(1-\tau)\sigma^2 I_n = 0.$$

Because  $(A, B)$  is jointly Gaussian, zero covariance establishes independence.

Normalize the components to preserve the original means:

$$X^D = \frac{A}{\tau} = X + \sqrt{\frac{1-\tau}{\tau}} E, \quad X^I = \frac{B}{1-\tau} = X - \sqrt{\frac{\tau}{1-\tau}} E.$$

Then  $X^D$  and  $X^I$  are independent, both have mean  $\mu$ , and their variances are  $\sigma^2/\tau$  and  $\sigma^2/(1-\tau)$ . Larger  $\tau$  improves discovery precision while making inference noisier. Independent normal noise is added featurewise for the spherical multivariate model.

#### **Example 14.1 — An equal information split**

With  $\tau = 1/2$ , the normalized components are  $X^D = X + E$  and  $X^I = X - E$ . Both have variance  $2\sigma^2 I_n$ , their covariance is zero, and their average is  $X$ .

They are independent under the joint repeated-sampling model for  $(X, E)$ . They are **not** independent conditional on the original  $X$ , because their sum is then fixed. The validation argument conditions on discovery information, not on every variable generated during the construction.

### 14.3 Inference conditional on the discovered groups

Cluster the rows of  $X^D$  and obtain groups  $C_1, C_2$ . Conditional on all discovery information, these index sets are fixed and  $X^I$  retains its Gaussian law. For one feature, the target is

$$\delta_C = \frac{1}{|C_2|} \sum_{i \in C_2} \mu_i - \frac{1}{|C_1|} \sum_{i \in C_1} \mu_i,$$

and the estimator uses the corresponding means of  $X^I$ . Its conditional variance is

$$v_C = \frac{\sigma^2}{1-\tau} \left( \frac{1}{|C_1|} + \frac{1}{|C_2|} \right).$$

Therefore  $\hat{\delta}_C \pm z_{0.975} \sqrt{v_C}$  has conditional 95% coverage for this selected target under the stated model. The discovery algorithm can be complicated because its information is independent of the inference component. Any further decision based on  $X^I$  requires its own accounting.

The target still depends on the discovered groups. A good confidence procedure does not ensure those groups match a scientifically meaningful population partition. This is why the laboratory reports grouping agreement and interval coverage separately.

## 14.4 A sensitivity study must challenge an assumption

The independence construction uses the true noise scale. Suppose instead that the injected noise has variance  $s^2 I_n$  while the original observation variance remains  $\sigma^2 I_n$ . The normalized components have covariance

$$\text{Cov}(X^D, X^I) = (\sigma^2 - s^2) I_n.$$

The cancellation fails when  $s \neq \sigma$ . Substituting an estimate into both the randomization and the interval formula therefore requires further theory; the known-variance derivation does not automatically apply.

### Example 14.2 — Reproducibility is not a validity guarantee

Suppose the true standard deviation is one, but the procedure uses  $s = 0.7$ . Then  $\text{Cov}(X^D, X^I) = 0.51 I_n$ . The discovery and inference components remain dependent. Using the same random seed on every computer will reproduce that dependence exactly.

A useful project reruns the null experiment at  $s/\sigma \in \{0.7, 1, 1.3\}$  and reports the change in false rejection, including Monte Carlo uncertainty. It treats the mis-specified cases as stress tests, not as cases covered by the exact theorem.

This is a more informative extension than changing an arbitrary tuning parameter without explaining its role. In UU learning, a comparable study changes the gap between mixture priors; in preference learning, it changes comparison-graph connectivity or within-prompt dependence; in online learning, it changes stationarity or the feedback available to the learner.

## 14.5 Design a small but interpretable experiment

Use one main question and a short sequence of experiments:

1. **Reproduction:** implement or reproduce a central result under its stated assumptions.
2. **Controlled change:** vary one assumption or resource constraint and keep the rest fixed.
3. **Matched comparison:** use the same simulated datasets or random-number streams when comparing methods.
4. **Uncertainty:** report Monte Carlo standard errors or uncertainty across independent repetitions.
5. **Interpretation:** explain which part of the theoretical argument survives the change.

For a rejection rate  $\hat{p}$  from  $B$  independent repetitions, the approximate Monte Carlo standard error is  $\{\hat{p}(1 - \hat{p})/B\}^{1/2}$ . With  $B = 400$  and a true rate near 0.05, this is about

0.011. A difference of one percentage point cannot be interpreted confidently from that experiment alone.

When comparing two procedures on the same replicates, use the replicate-wise difference rather than treating two estimated rates as independent. Record computational failures separately. Removing failed runs without disclosure changes the evaluated procedure and can hide instability.

## 14.6 What the final project should contain

The proposal should identify the target, a small set of assumptions, one baseline, and one planned modification. The reproduction checkpoint should show that the baseline works in a setting where its behaviour is understood. The final report should contain the mathematical setup, methods, experiment design, primary results, limitations, and instructions to regenerate the results.

Reusing an assignment implementation is appropriate when the report identifies what is reused and adds a new question or comparison. For example, extend the PU assignment by deliberately violating constant positive-labelling probability, or extend the clustering assignment by varying the discovery-information fraction. Comparing increasingly complicated methods without a fixed target is not a substitute for a research question.

Maintain a source record for data, a fixed seed strategy, software versions, the exact analysis script, and a description of the independent sampling unit. Repeated ratings from one prompt or annotator must not be counted as independent observations merely because they occupy separate rows.

## Information reserved for inference has a measurable cost

Conditional on a given pair of discovered groups, the standard error of the normalized inference contrast is inflated by  $1/\sqrt{1-\tau}$  relative to using an unselected full observation with the same fixed groups. For  $\tau = 0.2, 0.5, 0.8$ , the factors are approximately 1.12, 1.41, 2.24.

More discovery information can improve the grouping, while less remaining inference information widens a conditional interval. Actual groups can change with  $\tau$ , so these factors are not an unconditional power comparison between different discovered partitions. Choose the split through a declared design, not by repeatedly examining the eventual inferential result.

### Boundary - protect the reserved information

Once the inference component has influenced feature choice, group selection, or tuning, the original independence argument no longer describes the full workflow. Fix those choices using discovery information, and use inference information only for the stated assessment or account for the additional selection.

## 14.7 Reproducible Laboratory

The base-R script studies Gaussian thinning at three discovery fractions. It reports null rejection, selected-target interval coverage, interval length, estimation error, and a separate co-membership agreement measure under a simulated mean structure. It then challenges the known-noise assumption.

The intervals target the means of the discovered index sets. The script computes those targets from the simulated  $\mu_i$  rather than substituting the original generating group centres. All numerical results are produced by the downloadable script.

Download lab

Listing 4: Gaussian information splitting, selected-target coverage, and scale sensitivity

```
# Chapter 14: Gaussian data thinning and sensitivity. Base R only.
set.seed(2026102114)
options(width = 145)

n <- 60L
q <- 2L
B <- 400L
sigma <- 1
taus <- c(0.2, 0.5, 0.8)
reference_groups <- rep(1:2, each = n / 2L)
noise <- lapply(seq_len(B), function(b) matrix(rnorm(n * q), n, q))
randomization <- lapply(seq_len(B), function(b) matrix(rnorm(n * q), n, q))

pair_agreement <- function(a, b) {
  keep <- upper.tri(matrix(FALSE, length(a), length(a)))
  mean((outer(a, a, "==") == outer(b, b, "=="))[keep])
}

one_run <- function(mu, z, e, tau, assumed_sigma = 1) {
  x <- mu + sigma * z
  discovery <- x + sqrt((1 - tau) / tau) * assumed_sigma * e
  inference <- x - sqrt(tau / (1 - tau)) * assumed_sigma * e
  groups <- cutree(hclust(dist(discovery), method = "ward.D2"), k = 2L)
  # Fix label orientation using discovery information only.
  if (mean(discovery[groups == 1L, 1L]) >
      mean(discovery[groups == 2L, 1L])) groups <- 3L - groups
  left <- groups == 1L
  estimate <- mean(inference[!left, 1L]) - mean(inference[left, 1L])
  target <- mean(mu[!left, 1L]) - mean(mu[left, 1L])
  standard_error <- assumed_sigma / sqrt(1 - tau) *
    sqrt(1 / sum(left) + 1 / sum(!left))
  half_width <- qnorm(0.975) * standard_error
  c(covered = abs(estimate - target) <= half_width,
    length = 2 * half_width, squared_error = (estimate - target)^2,
    reject_zero = abs(estimate) > half_width,
    agreement = pair_agreement(groups, reference_groups))
}
```

(continues on next page)

(continued from previous page)

```

}

mu_null <- matrix(0, n, q)
mu_separated <- mu_null
mu_separated[, 1L] <- rep(c(-1.25, 1.25), each = n / 2L)
summaries <- list()
counter <- 1L
for (scenario in c("Global null", "Separated means")) {
  mu <- if (scenario == "Global null") mu_null else mu_separated
  for (tau in taus) {
    values <- vapply(seq_len(B), function(b)
      one_run(mu, noise[[b]], randomization[[b]], tau), numeric(5L))
    coverage <- mean(values["covered", ])
    summaries[[counter]] <- data.frame(
      scenario = scenario, discovery_fraction = tau,
      coverage = coverage,
      coverage_mcse = sqrt(coverage * (1 - coverage) / B),
      mean_interval_length = mean(values["length", ]),
      rmse_for_selected_target = sqrt(mean(values["squared_error", ])),
      null_rejection = if (scenario == "Global null")
        mean(values["reject_zero", ]) else NA_real_,
      grouping_agreement = if (scenario == "Separated means")
        mean(values["agreement", ]) else NA_real_
    )
    counter <- counter + 1L
  }
}

cat("Known sigma = 1; Ward discovery clustering; inference feature = 1\n")
cat("Coverage target: the actual selected groups' average underlying means.\n")
display_table <- do.call(rbind, summaries)
names(display_table) <- c("scenario", "tau", "coverage", "coverage_mcse",
  "mean_CI_length", "selected_RMSE",
  "null_rejection", "group_agreement")
print(display_table, row.names = FALSE, digits = 4)
cat("NA entries mark quantities deliberately not interpreted in that scenario.\n")

scales <- c(0.7, 1, 1.3)
sensitivity <- lapply(scales, function(assumed_sigma) {
  values <- vapply(seq_len(B), function(b)
    one_run(mu_null, noise[[b]], randomization[[b]],
      tau = 0.5, assumed_sigma = assumed_sigma), numeric(5L))
  rate <- mean(values["reject_zero", ])
  data.frame(
    assumed_sigma = assumed_sigma,
    component_covariance = sigma^2 - assumed_sigma^2,
    null_rejection = rate,
    monte_carlo_se = sqrt(rate * (1 - rate) / B)
  )
})
cat("\nScale sensitivity at discovery fraction 0.5; true sigma remains 1\n")
print(do.call(rbind, sensitivity), row.names = FALSE, digits = 4)

```

(continues on next page)

(continued from previous page)

```

cat("Only assumed_sigma = 1 is covered by the independence derivation.\n")

plot_dir <- Sys.getenv("COURSE_FIGURE_DIR", unset = "")
if (nzchar(plot_dir)) {
  dir.create(plot_dir, recursive = TRUE, showWarnings = FALSE)
  tab <- do.call(rbind, summaries)
  selected <- tab[tab$scenario == "Separated means", ]
  png(file.path(plot_dir, "14-research-synthesis.png"),
      width = 1400, height = 650, res = 130)
  old_par <- par(mfrow = c(1, 2), mar = c(4.2, 4.2, 3, 1))
  plot(selected$discovery_fraction, selected$grouping_agreement,
      type = "b", pch = 19, col = "#2a8a7c", ylim = c(0.5, 1),
      xlab = "Discovery information fraction", ylab = "Co-membership agreement",
      main = "Discovery improves with more information")
  plot(selected$discovery_fraction, selected$mean_interval_length,
      type = "b", pch = 19, col = "#bd4b38",
      xlab = "Discovery information fraction", ylab = "Mean 95% interval length",
      main = "Inference loses precision")
  par(old_par)
  invisible(dev.off())
}

```

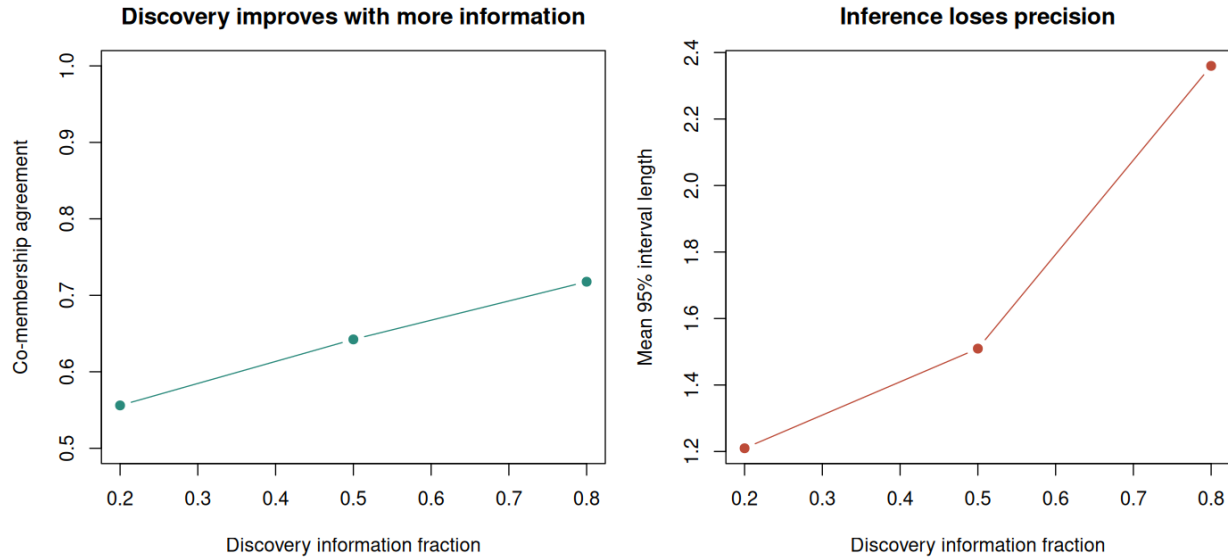
## Computed output

```

Known sigma = 1; Ward discovery clustering; inference feature = 1
Coverage target: the actual selected groups' average underlying means.
      scenario tau coverage coverage_mcse mean_CI_length selected_RMSE null_rejection_
↪group_agreement
  Global null 0.2   0.9675      0.008866          1.203      0.2899      0.0325  ↪
↪      NA
  Global null 0.5   0.9400      0.011874          1.536      0.3844      0.0600  ↪
↪      NA
  Global null 0.8   0.9475      0.011152          2.429      0.6389      0.0525  ↪
↪      NA
  Separated means 0.2   0.9625      0.009499          1.210      0.2992      NA      ↪
↪      0.5560
  Separated means 0.5   0.9700      0.008529          1.510      0.3592      NA      ↪
↪      0.6423
  Separated means 0.8   0.9600      0.009798          2.360      0.5772      NA      ↪
↪      0.7178
NA entries mark quantities deliberately not interpreted in that scenario.

Scale sensitivity at discovery fraction 0.5; true sigma remains 1
  assumed_sigma component_covariance null_rejection monte_carlo_se
      0.7              0.51          0.4575          0.02491
      1.0              0.00          0.0600          0.01187
      1.3             -0.69          0.1425          0.01748
Only assumed_sigma = 1 is covered by the independence derivation.

```



## Exercises

**Exercise 14.1.** Derive the means, variances, and covariance of  $A$  and  $B$ . Which step requires Gaussianity to conclude independence?

**Exercise 14.2.** With fixed discovered groups, derive the variance of  $\hat{\delta}_C$ . What happens to inference precision as  $\tau$  approaches one?

**Exercise 14.3.** Explain why coverage must be evaluated against  $\delta_C$  rather than the difference between the two generating population centres when recovery is imperfect.

**Exercise 14.4.** Choose one earlier course module and write a five-sentence project specification: target, observation mechanism, baseline, assumption to challenge, and main evaluation measure.

## Solutions

### **i** Solution to Exercise 14.1

Linearity gives the stated means. Independence of  $X$  and  $E$  gives  $\text{Var}(A) = \tau^2 \sigma^2 I + \tau(1-\tau)\sigma^2 I = \tau\sigma^2 I$ , with the analogous expression for  $B$ . The cross-covariance terms cancel. Zero covariance alone is insufficient in general; joint Gaussianity converts it into independence.

### **i** Solution to Exercise 14.2

Each inference observation has variance  $\sigma^2/(1 - \tau)$  and the two group averages involve disjoint independent rows. Adding their variances gives  $v_C$ . As  $\tau$  approaches one, this variance diverges for fixed group sizes. Better discovery can therefore be accompanied by much wider intervals.

### **i Solution to Exercise 14.3**

The test and interval use the estimated index sets. If they mix generating populations, the corresponding averages of  $\mu_i$  are mixtures of the generating centres, and their difference can be much smaller. Scoring the interval against another target confounds inference error with clustering recovery and does not evaluate the stated coverage guarantee.

### **i Solution to Exercise 14.4**

One possible specification is: “Estimate binary-classification risk from two unlabelled mixtures. Observe independent mixture samples with known, distinct priors and shared class-conditionals. Reproduce the unbiased UU risk estimator. Shrink the prior gap while holding sample sizes and component distributions fixed. Evaluate empirical bias, variance, and downstream excess risk, reporting uncertainty across repetitions.” A complete project would then specify the loss, target prior, model class, and finite-sample implementation.

## Reading and course conclusion

- Neufeld, A., Dharamshi, A., Gao, L. L., and Witten, D. (2024). *Data Thinning for Convolution-Closed Distributions*. **JMLR** 25(57), 1–35. [Paper](#) and [author code](#).

The recurring statistical task is to identify what the observations reveal under stated assumptions, then construct a learning or inferential procedure that respects that information. The final presentation should make that chain visible from the research question to the reported result.