UNIVERSAL SYLLABUS · MACHINE LEARNING

Machine Learning

Rule sets, first-order rules and EBL

The notes for the lesson “Rule sets, first-order rules and EBL”: chapters 6.3 to 6.5, with their figures, worked examples, exam traps, quizzes and exam question bank, in one file.

3 chapters | Chapters 6.3 to 6.5

About these notes

These are the notes for the University free n easy video lesson **Rule sets, first-order rules and EBL** on Machine Learning: chapters 6.3 to 6.5, with every figure, worked example, exam trap and quiz, and the long-answer exam questions on these chapters.

6.3 Learning rule sets: sequential covering algorithms

Sequential covering learns a classifier as a collection of if–then rules. It finds one rule that covers useful positive examples, removes the positives already handled and repeats. Each rule describes a small region; their combination describes a broader concept.

Only a few universities teach this advanced topic. It extends the classification ideas in an earlier lesson, especially decision trees, but searches for one rule at a time instead of growing an entire tree.

Rules and coverage

A rule has the form IF conditions THEN class. Its conditions usually form a conjunction, such as overdue = yes AND balance > ₹500. An example is covered when it satisfies every condition, whether its true label is positive or negative. A positive covered example is a correct positive prediction; a negative covered example is an error. Coverage and correctness are therefore different quantities.

Rules can form an unordered disjunction: predict positive if any rule fires, otherwise predict negative. An ordered decision list instead uses the first matching rule and ends with a default prediction. These semantics affect conflicts, so always state them. Here we learn an unordered positive rule set with a negative default. Multiple rules covering the same positive do not create multiple correct examples.

Separate and conquer

1. Keep the currently uncovered positives and the training negatives.
2. Start a candidate rule with an empty body, which covers every example.
3. Add conditions that improve the chosen rule-quality measure. Each added condition specializes the rule and shrinks its covered set.
4. Stop growing under the stated rule criterion, then prune using held-out data or another justified estimate.
5. Accept a useful rule and remove the positive examples it covers. Repeat until no useful rule remains or all positives are covered.

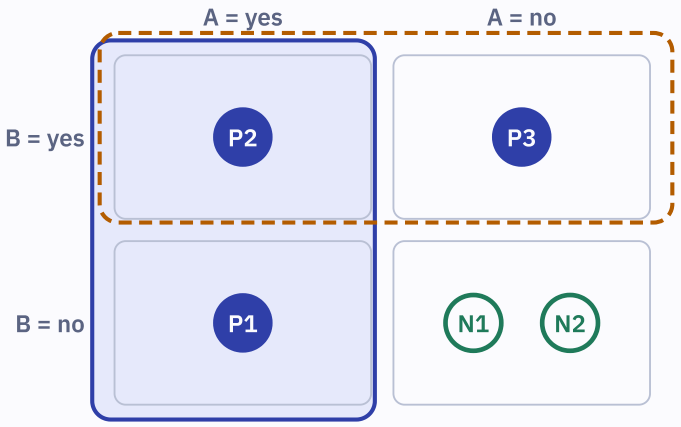
This is also called separate-and-conquer learning. It differs from divide-and-conquer tree learning, where a split creates branches that partition the data at a node. Overlapping rules need not partition examples. A quality measure such as precision $p/(p + n)$ uses covered-positive count p and covered-negative count n. Maximizing precision alone can select a tiny rule that memorizes one example; minimum coverage or validation can discourage that result.

Retaining negatives matters. If a negative covered by a bad rule simply disappears, later search can lose evidence about errors. Different covering algorithms have different removal and pruning conventions; report the chosen convention. On noisy data, demanding zero training errors can create long, brittle rules. A stopping rule, minimum improvement threshold and default class make the final classifier well defined even when some positives remain uncovered.

Inspecting a learned set

Use a separate validation set to choose complexity, then a test set for final evaluation, as in an earlier lesson. Measure the whole rule set rather than averaging individual-rule precisions: overlaps make that average misleading.

Sequential covering on 3 positives and 2 negatives



empty rule: 3 of 5 positive
precision 0.6

rule 1: IF A = yes
covers P1, P2; precision 2/2
remove P1, P2 from the pool

rule 2: IF B = yes
covers P3 (and P2 again);
no negative

positive if A = yes OR B = yes
TP 3, FP 0, FN 0, TN 2

P2 sits in both rules: 2 + 2 = 4 coverings, but only 3 distinct positives.

Figure 6.3.1 Rule 1, A = yes (solid box), covers P1 and P2 with no negative. With those removed, rule 2, B = yes (dashed box), covers the last positive P3. The two rules overlap on P2, and both negatives stay uncovered.

WORKED EXAMPLE

Three positives have attributes P1 = (A yes, B no), P2 = (A yes, B yes), P3 = (A no, B yes). Two negatives N1 and N2 both have A no and B no. Learn rules for the positive class.

The empty body covers p = 3 and n = 2, giving precision 3/5 = 0.6. Candidate A = yes covers P1 and P2 only: precision 2/2 = 1. Candidate B = yes covers P2 and P3 only, also precision 1. Break the tie by choosing A first. Remove P1 and P2 from the uncovered-positive pool.

P3 remains. The rule B = yes covers that remaining positive and no negative, so add it. The final classifier is positive if A = yes OR B = yes, otherwise negative. On the original five examples, TP = 3, FP = 0, FN = 0 and TN = 2. Precision and recall are both 1 on this training set. Summing rule coverages gives 4, which is wrong as a count of distinct positives because P2 belongs to both sets. No test accuracy follows from these five training outcomes.

EXAM TRAPS

- Adding an AND condition specializes one rule; adding an OR rule broadens the rule set.
- Do not remove every negative merely because a rule covers it.
- State whether rules are ordered and what happens when no rule fires.

CHECK YOURSELF

- REMEMBER **What is sequential covering?** 4 marks

  Learn one rule, remove positives it handles and repeat on the remaining task.

- UNDERSTAND **Does coverage imply a correct prediction?** 4 marks

  No. A rule may cover negatives as well as positives; their labels determine correctness.

- APPLY **A rule covers 8 positives and 2 negatives. Find its precision.** 4 marks

  Precision = 8/(8 + 2) = 0.8. Recall additionally needs the total positive count.

- APPLY **Two rules cover positive sets {1, 2, 3} and {3, 4}. How many distinct positives are covered?** 4 marks

  The union is {1, 2, 3, 4}, so four, rather than five.

- ANALYZE **Why prune a perfect training rule?** 4 marks

  It may memorize noise or an accidental combination. Validation can reveal that a shorter rule generalizes better.

BEYOND THE SYLLABUS

This is not for the exam. Rule learners remain useful when decisions need inspection by domain experts. Search next for RIPPER, rule lists and subgroup discovery. Interpretability also requires manageable rule counts and meaningful features; if–then syntax alone does not guarantee it.

6.4 Learning first-order rules

First-order rule learning represents relationships among objects instead of describing each example only through a fixed attribute vector. Shared variables connect facts, allowing a learned rule to express patterns such as two people having a common parent.

Only a few universities teach this advanced topic. Read Chapter 6.3 for covering; here the search language becomes relational.

Predicates, variables and witnesses

A predicate names a relation. For example, parent(asha, bala) says that Asha is a parent of Bala. Lower-case names here denote constants, while X, Y and Z denote variables. A positive Horn rule has one conclusion, or head, and a conjunction of premises, or body. The rule grandparent(X, Z) ← parent(X, Y) AND parent(Y, Z) says that a two-step parent chain establishes the target relationship.

The shared Y is essential. Both premises must hold under one consistent substitution of constants for variables. A body variable not in the head acts as an existential witness when finding answers: there must be some matching intermediate person. The clause as a logical implication is universally quantified over its variables. These two views agree: any successful witness supplies a body instance whose head follows.

A propositional rule might use fixed features such as “has_parent_named_bala”. A relational rule can use a newly encountered person's name without inventing a separate feature for every constant. This expressiveness also enlarges the search space. Type restrictions, connected literals and limits on body length can keep candidate rules meaningful and computation manageable.

Learning one clause and then a set

1. Choose the target predicate and supply labelled positive and negative target tuples plus background relations.
2. Start with a general target clause and enumerate legal body literals.
3. Evaluate specializations by how they retain positive bindings while excluding negative bindings.
4. Add a useful literal, update the bindings and continue until the stopping criterion is met.
5. Add the accepted clause to the set, remove covered positive target tuples and repeat for other patterns.

FOIL, First Order Inductive Learner, is a classic specialization approach. A commonly used literal score is FOIL-gain = $t[\log_2(p_1/(p_1 + n_1)) - \log_2(p_0/(p_0 + n_0))]$. Counts p_0 and n_0 refer to positive and negative bindings before the literal; p_1 and n_1 refer to extended bindings after it. The factor t counts positive bindings from before specialization that retain an extension. Bindings can multiply when a new variable is introduced, so these counts need not equal counts of distinct target examples.

Inductive logic programming, or ILP, studies learning logical descriptions using examples and background knowledge. FOIL is one approach within this broader area. Practical systems must specify whether missing facts mean false under a closed-world assumption or merely unknown.

A learner cannot silently treat incomplete records as explicit negative evidence without changing the learning problem.

Testing entailment

A candidate covers a target tuple when the background facts together with the clause establish it. Testing a conjunction resembles joining relational tables on shared variables. Check every body literal under the same substitution. Learning a set of clauses uses OR across clauses and AND inside a body.

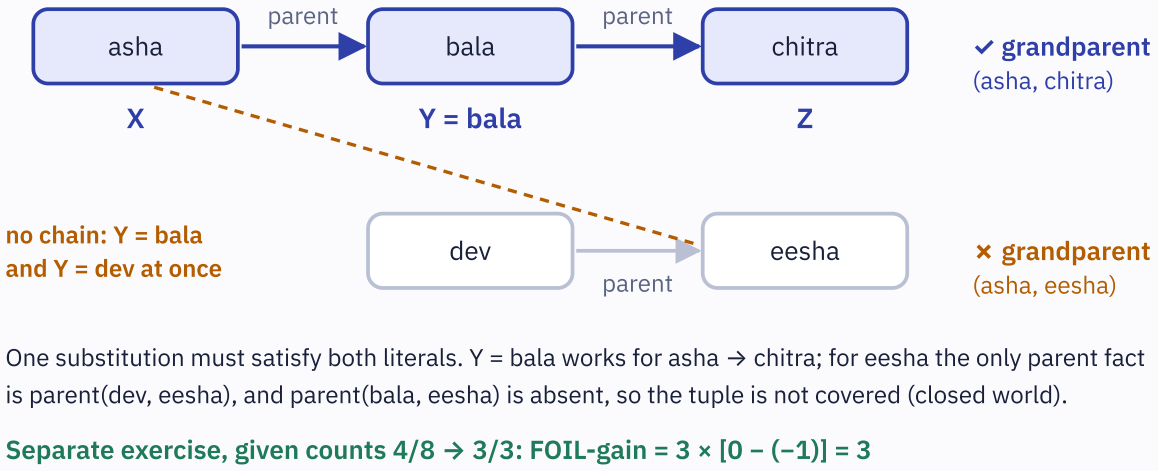


Figure 6.4.1 With Y = bala the two parent facts chain asha to chitra, so grandparent(asha, chitra) is covered. No single Y links asha to eesha: parent(dev, eesha) would need Y = dev, which breaks the first literal.

WORKED EXAMPLE

Background facts are parent(asha, bala), parent(bala, chitra) and parent(dev, eesha). Test grandparent(asha, chitra) and grandparent(asha, eesha) with the clause above.

For the first tuple, bind X = asha and Z = chitra. The first body fact gives Y = bala; parent(bala, chitra) is present, so both literals hold. The positive tuple is covered. For the second, the first literal again requires Y = bala, but parent(bala, eesha) is absent. Under the stated closed-world assumption, the tuple is not covered. The fact parent(dev, eesha) cannot help because it would require Y = dev, inconsistent with the first literal.

For a separate scoring exercise, suppose $p_o = 4$, $n_o = 4$, $p_1 = 3$, $n_1 = 0$ and $t = 3$. FOIL-gain = $3[\log_2(1) - \log_2(1/2)] = 3[0 - (-1)] = 3$. This measures improvement for the candidate literal under these binding counts; it is not a test-set accuracy or a probability.

EXAM TRAPS

- Keep repeated variables consistent; two unrelated facts do not form a chain.
- Do not identify binding counts with distinct example counts after introducing a variable.
- State how absent facts are interpreted; unknown and false differ.

CHECK YOURSELF

– REMEMBER **What is a predicate?** 4 marks

A named relation over arguments, such as parent(person, child).

– UNDERSTAND **Why share a variable between two literals?** 4 marks

It requires the corresponding arguments to denote the same object and therefore links the facts.

– APPLY **Given parent(a, b) and parent(b, c), infer the target tuple.** 4 marks

Substitute X = a, Y = b, Z = c. The grandparent clause entails grandparent(a, c).

– APPLY **Before precision is 1/4, after precision 1/2, and t = 2. Find FOIL-gain.** 4 marks

$2[\log_2(1/2) - \log_2(1/4)] = 2[-1 - (-2)] = 2$.

– ANALYZE **Why can closed-world learning mislabel a family relation?** 4 marks

A missing parent fact may be unrecorded rather than false. The assumed negative can therefore contradict the real relationship.

BEYOND THE SYLLABUS

This is not for the exam. Relational learning connects to knowledge graphs and program synthesis. Search next for inductive logic programming and differentiable reasoning. These methods must balance expressive rules with reliable search and the quality of background facts.

6.5 Explanation-based and inductive-analytical learning

Analytical learning uses prior knowledge to explain why an example satisfies a target concept. Explanation-based learning generalizes the proof into a useful rule. Inductive-analytical learning also uses observed examples to guide or correct what prior knowledge suggests.

Only a few universities teach this advanced topic. The first-order clauses of Chapter 6.4 provide a language for expressing the background theory and the explanation.

From one example to an explanation

Explanation-based learning, or EBL, takes a training example, a target concept, a domain theory and an operationality criterion. The domain theory contains facts and inference rules. The operationality criterion specifies which predicates are acceptable in the learned rule, often because they can be measured cheaply. The learner proves that the example satisfies the target, then keeps the proof's relevant requirements while replacing unnecessary constants by variables.

Consider a parcel accepted for a delivery robot. A useful explanation might depend on its measured mass and the robot's capacity, not its colour or tracking number. EBL asks which conditions made the proof work. It does not simply erase every constant: a safety threshold justified by the theory must remain. Nor is any condition true of the example automatically necessary for the proof.

1. Prove the target for the training instance using the domain theory.
2. Trace which facts and rule premises support that proof.

3. Generalize the proof while preserving shared variables and required constraints.
4. Express the resulting sufficient condition in operational predicates.
5. Check that the learned rule covers the example and is justified by the theory.

Under a correct theory, the learned condition is sufficient for the target. It need not be necessary: another valid explanation may establish the same target. EBL can improve problem-solving speed by compiling a long inference into a short test. Pure deductive EBL does not establish new consequences beyond what the theory already entails. Learning from one example is possible because the theory supplies strong knowledge; it is not a general statistical guarantee from one observation.

Combining induction and analysis

A domain theory can be incomplete, too specific or wrong. Induction learns regularities from labelled examples, while analysis uses theory to constrain the search. Inductive-analytical methods combine these sources instead of assuming either is sufficient. For example, a theoretical rule can seed a candidate description, then training data can suggest extra literals or alternative clauses. Validation determines whether the resulting change generalizes.

FOCL, First Order Combined Learner, combines explanation-based guidance with a FOIL-style first-order rule search. Prior knowledge can suggest useful operational conditions; empirical candidate evaluation still matters. An incorrect theory should not be treated as an unquestionable source of labels. Conflicts can indicate faulty premises, noisy examples or a mismatch between the theory's scope and the actual task.

What to evaluate

Evaluate both predictive behaviour and computational cost. Use separate validation data, as in an earlier lesson, and record which theory assumptions were used.

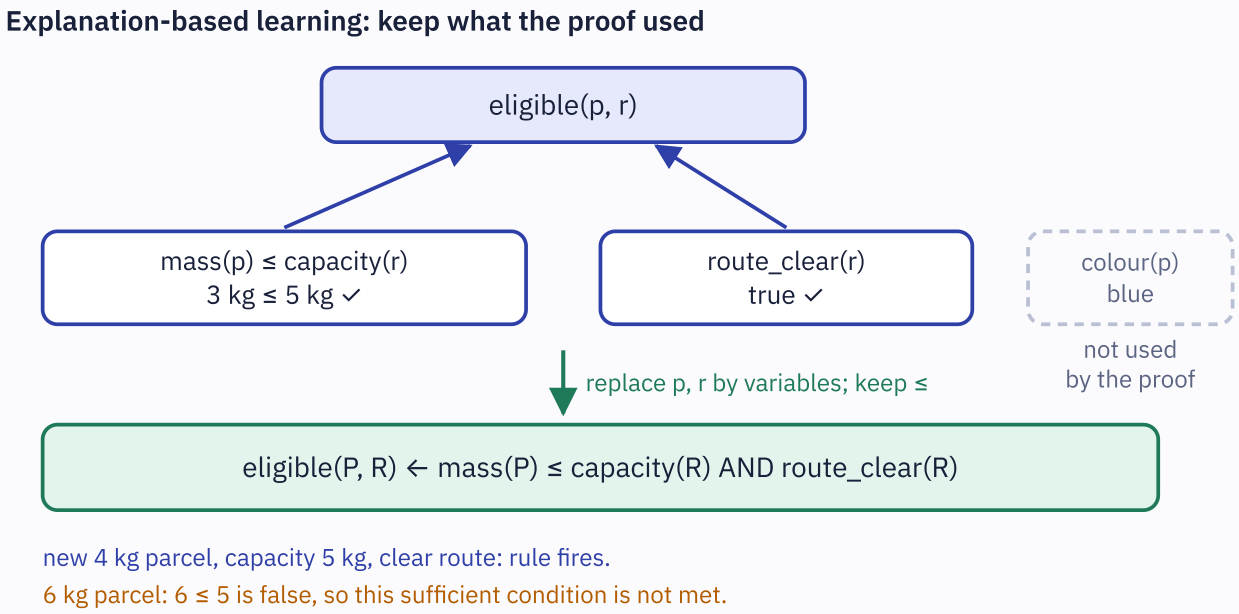


Figure 6.5.1 The proof that parcel p is eligible used only the mass check (3 kg ≤ 5 kg) and the clear route. Turning p and r into variables gives a reusable rule; the parcel's colour never entered the proof, so it is left out.

WORKED EXAMPLE

A theory states eligible(P, R) if mass(P) ≤ capacity(R) AND route_clear(R). Parcel p weighs 3 kg; robot r has capacity 5 kg and a clear route. Parcel colour is blue. Derive a reusable operational rule.

First, 3 kg ≤ 5 kg is true and route_clear(r) is true. Both premises prove eligible(p, r). Replace p and r by variables P and R while retaining the mass-capacity comparison. The generalized rule is eligible(P, R) ← mass(P) ≤ capacity(R) AND route_clear(R). Colour is excluded because the proof never used it. The numerical values 3 and 5 are replaced by measured quantities, but the ≤ relation remains essential.

A new parcel of 4 kg with capacity 5 kg and a clear route satisfies the rule. A 6 kg parcel does not, since 6 ≤ 5 is false. This failed test does not prove ineligibility under every possible theory; it shows that this sufficient condition is not met. If operational data include a rejected 4 kg parcel because of excessive width, the theory may lack a width constraint. A hybrid learner can propose that condition, then test it on additional labelled data.

EXAM TRAPS

- An explanation is a logical proof, not a verbal justification invented after prediction.
- A sufficient learned rule need not cover every valid target example.
- Do not confuse theory-guided generalization with statistical evidence from one example.

CHECK YOURSELF

– REMEMBER **List the four EBL inputs.** 4 marks

Training example, target concept, domain theory and operationality criterion.

– UNDERSTAND **Why omit the blue colour?** 4 marks

It was irrelevant to the proof. Keeping it would unnecessarily restrict the generalized condition.

– APPLY **Mass is 7 kg, capacity 8 kg, route_clear is true. Does the rule fire?** 4 marks

Yes. The comparison $7 \leq 8$ and the route predicate are both true, so eligibility follows under this theory.

– APPLY **What happens for mass 4 kg, capacity 5 kg and blocked route?** 4 marks

The route premise is false, so this rule cannot establish eligibility despite sufficient capacity.

– ANALYZE **Why use induction when a theory already exists?** 4 marks

The theory may be incomplete or inaccurate. Examples can expose failures and guide refinements, which still require independent evaluation.

BEYOND THE SYLLABUS

This is not for the exam. Neuro-symbolic systems explore combinations of learned representations and explicit reasoning. Search next for theory revision and knowledge-guided learning.

Key terms of this lesson

TERM	MEANING OR FORMULA	CHAPTER
Rule precision	p/(p + n), using covered positives p and negatives n	6.3
Relational rule	Shared variables link predicates under one substitution	6.4
FOIL-gain	$t[\log_2 \text{precision_after} - \log_2 \text{precision_before}]$	6.4
EBL	Generalize a proof into an operational sufficient condition	6.5

Exam question bank

Long-answer exam questions on these chapters, each with an outline of the points an examiner looks for. Answer one in full on paper before you read its outline.

Remember and understand

- **Explain sequential covering and its stopping choices. (10 marks)**

  Describe rule growth, quality evaluation, pruning, positive removal, repeated search and the default class. State ordered or unordered semantics and how noise affects stopping.

- **Describe first-order rules and explain consistent substitution. (10 marks)**

  Define predicates, constants, variables, head and body. Use a two-link relationship, demonstrate a shared witness and distinguish absent from false background facts.

Apply

- **Learn a covering rule set for the five examples in Chapter 6.3. (10 marks)**

  Compare initial precision 0.6 with each single-condition rule precision 1. Choose A first, remove P1 and P2, then choose B. Count the union as three distinct positives; state the negative default.

- **Prove two target tuples using parent facts and compute FOIL-gain. (12 marks)**

  Show X, Y and Z substitutions from Chapter 6.4. The asha–chitra tuple follows; the asha–eesha tuple does not under closed-world semantics. With the supplied binding counts, gain is 3.

Analyze and evaluate

- **Compare exploration in reinforcement learning with coverage in supervised rule learning. (12 marks)**

  Span Chapters 6.1–6.3. RL actions affect future observations and rewards; a fixed labelled rule dataset does not offer that interaction. Compare evaluation and generalization without treating coverage as exploration.

- **Explain how sequential covering extends to a set of relational clauses. (12 marks)**

  Span Chapters 6.3–6.4. Preserve the outer covering loop, replace attribute conditions by predicates, track bindings and joins, and distinguish OR across clauses from AND within one clause.

- **Derive an EBL rule and discuss a counterexample caused by an incomplete theory. (16 marks)**

  Use the parcel proof. Preserve the mass-capacity comparison and clear-route premise, omit colour, then analyze missing width information. Distinguish a sufficient proof condition from complete characterization.

- **Compare FOIL, EBL and FOCL as sources of generalization. (16 marks)**

  Span Chapters 6.4–6.5. FOIL searches clauses using labelled evidence; EBL compiles theory-supported explanations; FOCL combines theoretical guidance with inductive search. Discuss binding consistency, theory errors and independent validation.

Glossary (terms used in this lesson)

State	Information used to predict the next transition given an action.
Coverage	The examples satisfying a rule body.
Sequential covering	Learning rules one at a time while removing covered positives.
Predicate	A named relation over objects.
Substitution	Consistent assignment of terms to variables.

Horn rule	An implication with a conjunction of premises and one positive head.
Closed-world assumption	Treating facts not established by the knowledge base as false.
Domain theory	Background facts and rules used for reasoning.
Operationality	Acceptability of predicates for the learned description.
EBL	Explanation-based learning through generalized proofs.
FOCL	First Order Combined Learner, combining inductive and analytical guidance.

Sources and further reading

- Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*, second edition: MDPs, temporal-difference learning and tabular control.
- Tom M. Mitchell, *Machine Learning*: rule learning, analytical learning and combining induction with prior knowledge. No unverified page references are used.
- Dive into Deep Learning: Reinforcement Learning, open chapter introduction. Opened as the public source text; covers sequential decisions, MDPs and Q-learning.
- scikit-learn: Decision Trees, open documentation source. Opened for the contrast with tree-based decision rules, interpretability and overfitting; it is not a reference implementation of sequential covering.

Syllabus: the University free n easy universal syllabus, built from top universities' syllabi.

Built for the University free n easy YouTube channel from the universal syllabus of Machine Learning (Oct 2026). Written with top universities' syllabi and standard textbooks; check a formula against your prescribed book before an exam, and report mistakes on the channel.