MAT 3373
Methods of Machine Learning

Chapter 1
Machine Learning 101

P. Boily (uOttawa)

Fall – 2023

Outline

Outline

Main Reference:

- *Data Understanding, Data Analysis, and Data Science*, chapter 19.

1 – Machine Learning 101

Data scientists are often introduced to their field *via* machine learning concepts, algorithms, and applications.

In this chapter, we will discuss some **preliminary non-technical notions**, as well as some of the **issues and challenges** encountered in various **learning tasks**.

We also provide a first example of an unsupervised machine learning task: **association rules mining**.

We will discuss the classical tasks of machine learning (**value estimation**, **classification**, and **clustering**) in subsequent chapters.

1.1 – Introduction

A challenge of working in **data science** (DS), **machine learning** (ML), and **artificial intelligence** (AI): nearly **all quantitative work** can be described with some combination of the terms DS/ML/AI!

Difficult to differentiate the discipline from other quantitative fields $\Longrightarrow$ **studying and learning it properly** is often harder than it needs to be.

> "Data is not information, information is not knowledge, knowledge is not understanding, understanding is not wisdom." (C. Stoll)

Very important!

Robinson suggests an inclusive **hierarchical** structure:

- 1st stage – **DS** provides "**insights**" *via* **visualization** and **(manual) inferential analysis**;
- 2nd stage – **ML** yields "**predictions**" (or "advice"), while **reducing** the operator's **analytical**, **inferential**, and **decisional**;
- 3rd stage – **AI** removes the **need for oversight**, allowing for **automatic "actions"** to be taken by a **completely unattended system**.

We are fairly competent at tasks related to stages 1 and 2; for stage 3, not so much as of now ... although **ChatGPT?**

Goals of AI: great from an **academic perspective**, but in practice, stakeholders should **not give up their agency in the decision-making process**.

Suggestion: further split **AI** into "**general AI**" and "**augmented intelligence**" (**ML** "on steroids").

My definition of the **DS/ML/AI approach:**

> quantitative processes (working intersection of **statistics**, **engineering**, **computer science**, **domain expertise**, and "**hacking**") that can help users **learn actionable insights** about their situation without completely **abdicating their decision-making responsibility**.

1.2 – Types of Learning and ML Tasks

Humans learn by **taking in their environment** and:

- **answering questions about it**;
- **testing hypotheses**;
- **creating concepts** and **categories**;
- **making predictions**, and
- **classifying and grouping its various objects and attributes**.

"We learn from failure, not from success!" (B. Stoker, *Dracula*)

Main DS/ML/AI concept: teach machines to **extract insight from data**, properly and efficiently, without **biases** and **pre-conceived notions**.

Or, **can we design algorithms that can learn?** This is not the same thing as: **should we design such algorithms.**

The simplest DS/ML/AI method is **exploring the data** to:

- **provide a summary** – mean, variance, etc.;
- **make multi-dimensional structure evident** – data visualization, and
- **look for consistency**, considering what is in there and what is missing.

More sophisticated approaches: **supervised** and/or **unsupervised** learning.

1.2.1 – Types of Learning

Supervised learning (SL): "learning with a **teacher**."

Typical tasks: classification, **regression**, **rankings**, **recommendations**.

Supervised algorithms use **labeled training data** to build a predictive model; performance is evaluated using **test data** with unused **labels**.

Example: students try to answer exam questions based on what they learned from worked-out examples provided by teacher. The teacher provides the correct answers and marks the exam questions using the key.

There are fixed **targets** against which to train the model (age categories, plant species), which are **known** prior to the analysis.

Unsupervised learning (UL): "**self-**learning by finding **similarities**."

Typical tasks: clustering, **association rules**, **anomaly detection**.

Unsupervised algorithms use **unlabeled data** to find **natural** patterns in the data; accuracy **cannot be evaluated** to the same degree.

Example: students try to create a study guide of similar questions; the teacher is not involved in the process. Different students might end up with different study guides, without anyone being wrong.

The **target**, if it **exists at all** is **unknown/unknowable**; we are simply looking for **natural groups** in the data.

Some techniques fit into both camps; but there are other approaches.

Semi-supervised learning (SSL): some observations have **labels**, but most **do not**. This may occur when acquiring data is costly.

Example: teacher provides worked-out examples and a list of unsolved problems to try out; the students try to find similar groups of unsolved problems and compare them with the solved problems to find close matches.

Reinforcement learning (RL): an agent attempts to collect as much (short-term) **reward** as possible while minimizing (long-term) **regret**.

Example: embarking on a Ph.D. with an advisor, with all of the highs and the lows, sometimes getting closer, sometimes getting futher (and **maybe** a diploma at the end of the process?).

We focus mostly on **SL** and **UL**.

1.2.2 – ML Tasks

Outside of academia, DS/ML/AI methods are only really interesting when they help **ask and answer useful questions**. Compare, for instance:

- **Analytics** – "How many clicks did this link get?"
- **Data Science** – "Based on the previous history of clicks on links of this publisher's site, how many people from Manitoba will read this specific page in the next three hours?"
- **Quantitative Methods** – "We have reasons to believe that the number of hits will be strongly correlated with the temperature in Winnipeg. Based on the weather forecast, can we predict how many people will access the specific page over the next week?"

DS and **ML** models are usually **predictive** (not **explanatory**): they show connections, and exploit correlations to make predictions, but **they don't reveal why such connections exist**.

Quantitative methods assume a certain level of causal understanding based on various **first principles**. That distinction is not always understood.

Common DS/ML tasks:

- **classification** and **probability estimation** – which undergraduates are likely to succeed at the graduate level?
- **value estimation** – how much is a given client going to spend at a restaurant?
- **similarity matching** – which prospective clients are most similar to a company's established best clients?
- **clustering** – do signals from a sensor form natural groups?
- **association rules discovery** – what books are commonly purchased together at an online retailer?
- **profiling** and **behaviour description** – what is the typical cell phone usage of a certain customer segment?
- **link prediction** – J. and K. have 20 friends in common: perhaps they'd be great friends with one another?

1.2.3 – Example: Mushrooms Dataset

Data: *Mushroom Dataset*, UCI Machine Learning Repository

Consider *Amanita muscaria* (shown below).

Amanita muscaria (fly agaric), in the wild. Does it look dangerous to you?

Problem: is *Amanita muscaria* **edible** or **poisonous**? What do you think?

Easy solution: **eat it, wait, and see.** If you do not **die** or **get sick** upon ingestion, it is **edible**; otherwise, it is **poisonous**.

This test in unappealing – apart from the **obvious risk of death**, we might not learn much from the experiment.

It is possible that *Amanita muscaria* is actually edible in general, but:

- this **specific specimen** was poisonous due to a **mutation**, say, or
- the ingester had a **pre-existing condition** which combined with the fungus to **cause discomfort**.

A **predictive model** using a vast collection of training mushroom data (with **features** and **class labels**) could help discover:

- what poisonous mushrooms have in common;
- what properties edible mushrooms share.

Note: this is not the same as understanding **why** a mushroom is poisonous or edible – the data alone cannot provide an answer to that question.

Assume *Amanita muscaria* has the following features:

- **habitat**: **woods**
- **gill size**: **narrow**
- **spores**: **white**
- **odor**: **none**
- **cap color**: **yellow**

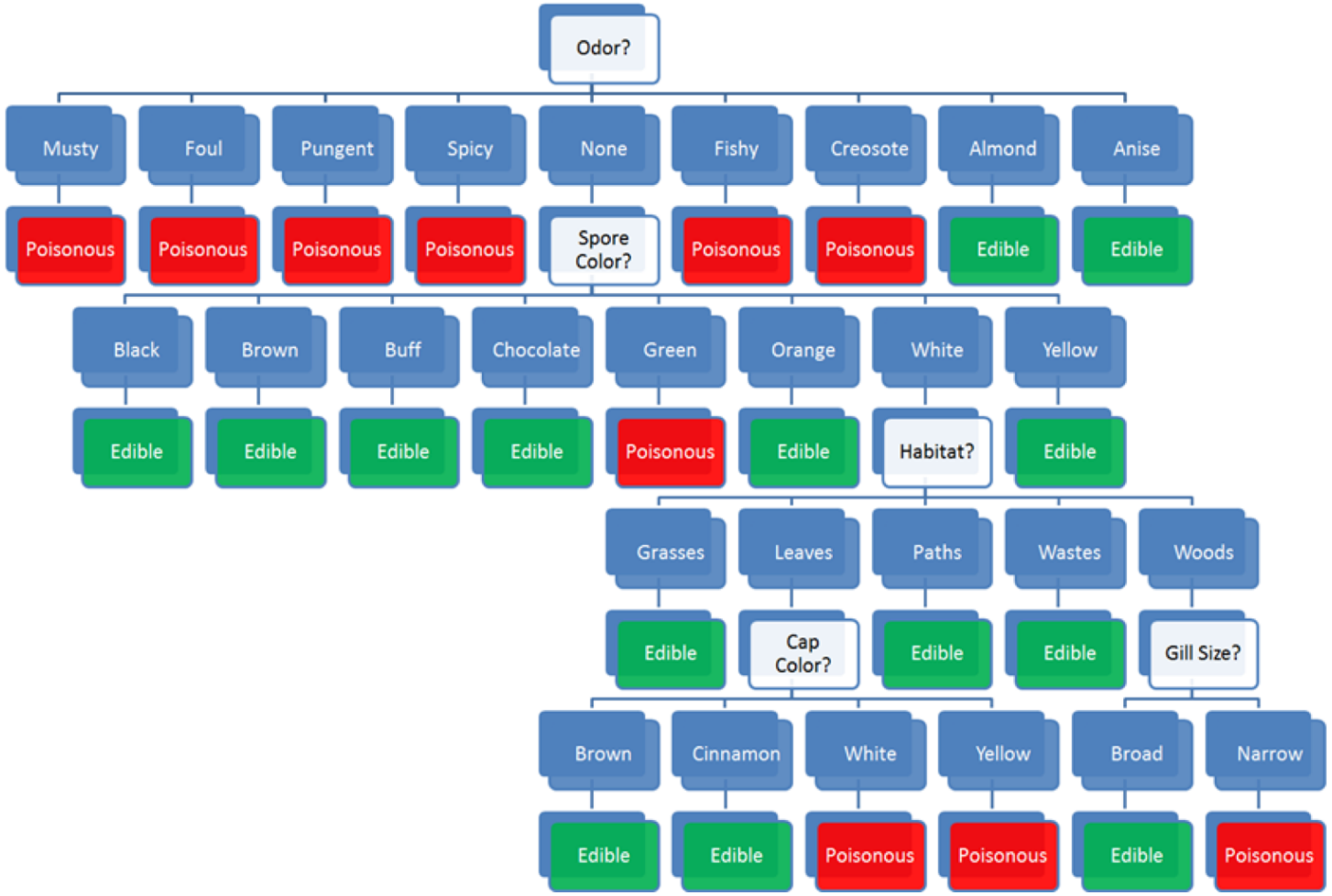
We do not know **a priori** whether it is poisonous or edible. Is the available information sufficient to answer the question?

A mycologist could perhaps deduce the answer from these features alone, but she would be using her **experiences with fungi** to make a prediction (not looking at the features in a *vacuum*).

We could use **past data**, with correct **edible** or **poisonous labels** and **identical predictors** to build various **(supervised) classification** models to try to answer the question.

A simple form of such model, a **classification tree**, is shown on the next slide.

The model prediction for $Amanita\ muscaria$ follows a **decision path**.



Classification tree for the mushroom classification problem

Decision path:

1. some mushroom **odors** are associated with poisonous mushrooms, some with edible mushrooms;

2. there are mushrooms with **no specific odor** in either category;

3. we need to incorporate additional features into the decision path for proper classification;

4. among mushrooms with **no specific odor**, some **spore colours** are associated with edible mushrooms, some with poisonous mushrooms;

5. there are mushrooms with **white spores** in either category;

Decision path:

6. the combination **no odor** and **white spores** does not provide enough information – we need to incorporate additional features into the decision path for proper classification;

7. among mushrooms of **no odor** with **white spores**, some **habitats** are associated with edible mushrooms;

8. there are mushrooms in either category that are found in **woods**;

9. the combination **no odor**, **white spores**, and **found in the woods** does not provide enough information – we need to incorporate additional features into the decision path for proper classification;

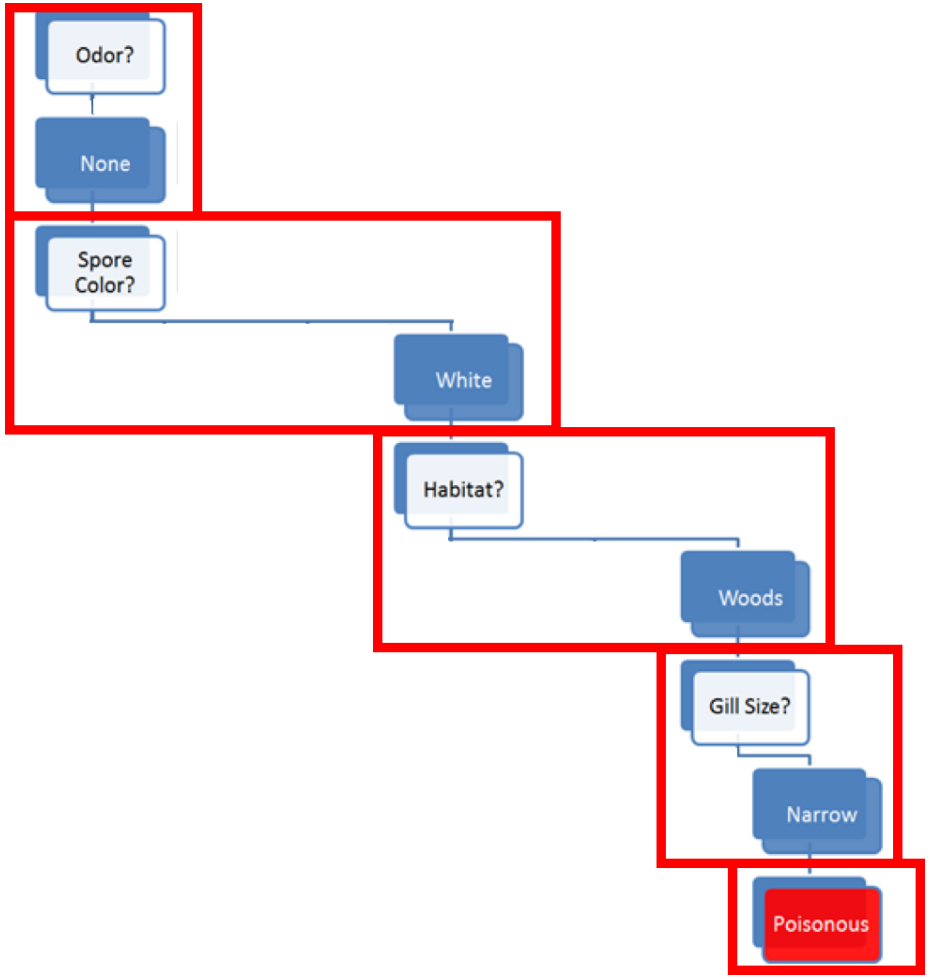
Decision path:

10. among **white-spored forest** mushroom with **no odor**, a **broad gill size** is associated with edible mushrooms, whereas a **narrow gill size** is associated with poisonous mushrooms;

11. thus, the decision path predicts that $Amanita\ muscaria$ is **poisonous**.

The **cap color** does not affect the decision path (it would have had $Amanita\ muscaria$'s habitat been **leaves**).

The classification tree model **does not explain why** this combination of features is associated with poisonous mushrooms – the decision path is not **causal**.



Decision path for *Amanita muscaria*

Questions:

- Would you have trusted an **edible** prediction?
- How are the features measured?
- What is the true cost of making a mistake?
- Is the data on which the model is built representative?
- What data is required to build trustworthy models?
- What do we need to know about the model in order to **trust it**?

This mushroom classification problem has all of the **hallmarks of a ML problem**.

Keep it in mind as a **representative** of the discipline in the rest of the course.

1.3 – ML Issues and Challenges

The data science landscape is littered with issues and challenges. We discuss some of them briefly (we will revisit some of them in future chapters).

> "We all say we like data, but we don't. We like getting insight out of data. That's not quite the same as liking data itself. In fact, I dare say that I don't quite care for data, and it sounds like I'm not alone." (Q.E. McCallum, *Bad Data Handbook*)

Before embarking on our technical ML adventure, we will touch on **bad data**, **underfitting/overfitting**, the **transferability of results**, and various **myths** and **common mistakes**.

1.3.1 – Bad Data

Data issues:

- it is not always **representative** of the situation that we want to model;
- it is not always **consistently collected**;
- it may be formatted for **human consumption**, not **machine readability**;
- it may contain **lies** and/or **mistakes**;
- it might not **reflect reality**, and
- there might be sources of **bias** and **errors** (**imputation bias**, **proxy reporting**, etc.).

Seeking **perfection** in the data can hurt DS/ML efforts – **different** quality requirements exist for **academic**, **professional**, **economic**, **government**, **military**, **service**, **commercial** data: "**close enough is good enough**" (**completeness**, **coherence**, **correctness**, **accountability**).

Main challenge: defining what is "close enough" in **applications**.

Pitfalls: (even when most data issues have been "solved")

- analyzing data without **understanding the context**;
- using **one and only one tool** (by choice or by fiat) – neither the "**cloud**", nor **Big Data**, nor **Deep Learning**, nor **Artificial Intelligence** will solve all of an organization's problems;
- analyzing data just for **the sake of analysis**,
- having **unrealistic expectations** of data analysis/DS/ML/AI $\Longrightarrow$ to produce **actionable data insights**, we must first recognize the methods' **domains of application** and their **limitations**.

1.3.2 – Underfitting/Overfitting

In traditional statistics, $p-$**values** and **goodness-of-fit** statistics are used to **validate the model**.

Such statistics cannot always be computed for predictive data science models: a model is **good** if it **performs "well" on unseen/new data**.

In practice, training sets and ML methods are used to search for **rules** and **models** that are **generalizable to new data**.

Problem: knowledge gained from ML may not **generalize properly**.

Ironically, this may occur if the rules or models **fit the training data too well** – the results are **too specific** to the **training data**.

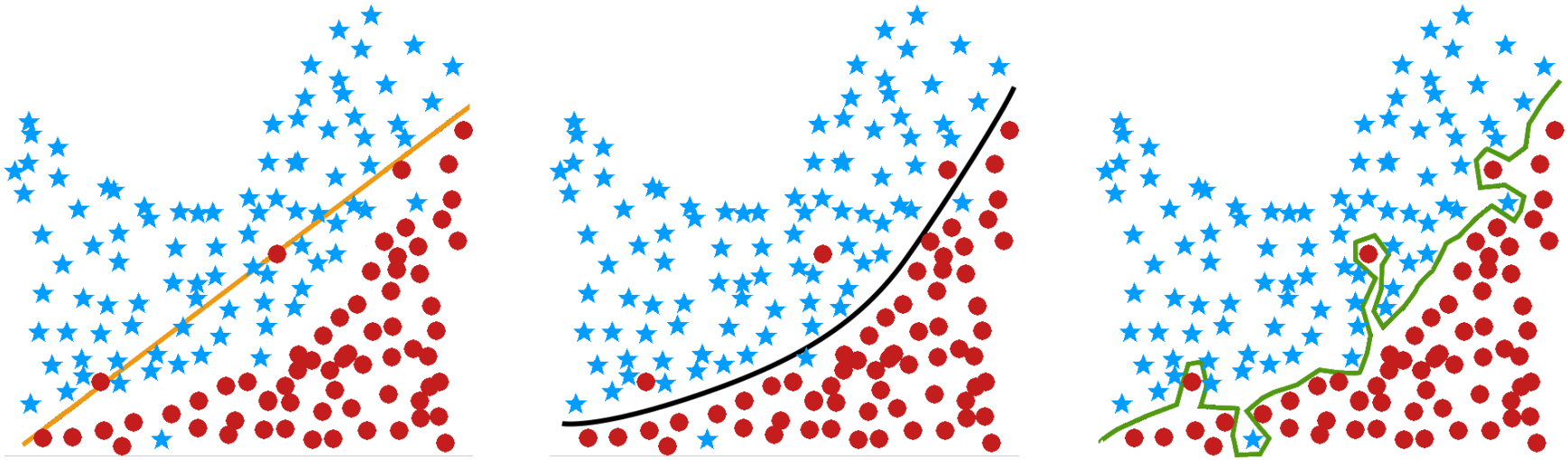


Illustration of **underfitting** (left) and **overfitting** (right) for a classification task – an **optimal classifier** (middle) might reach a **compromise** between accuracy and simplicity

Example: consider the following rules regarding human hair colours.

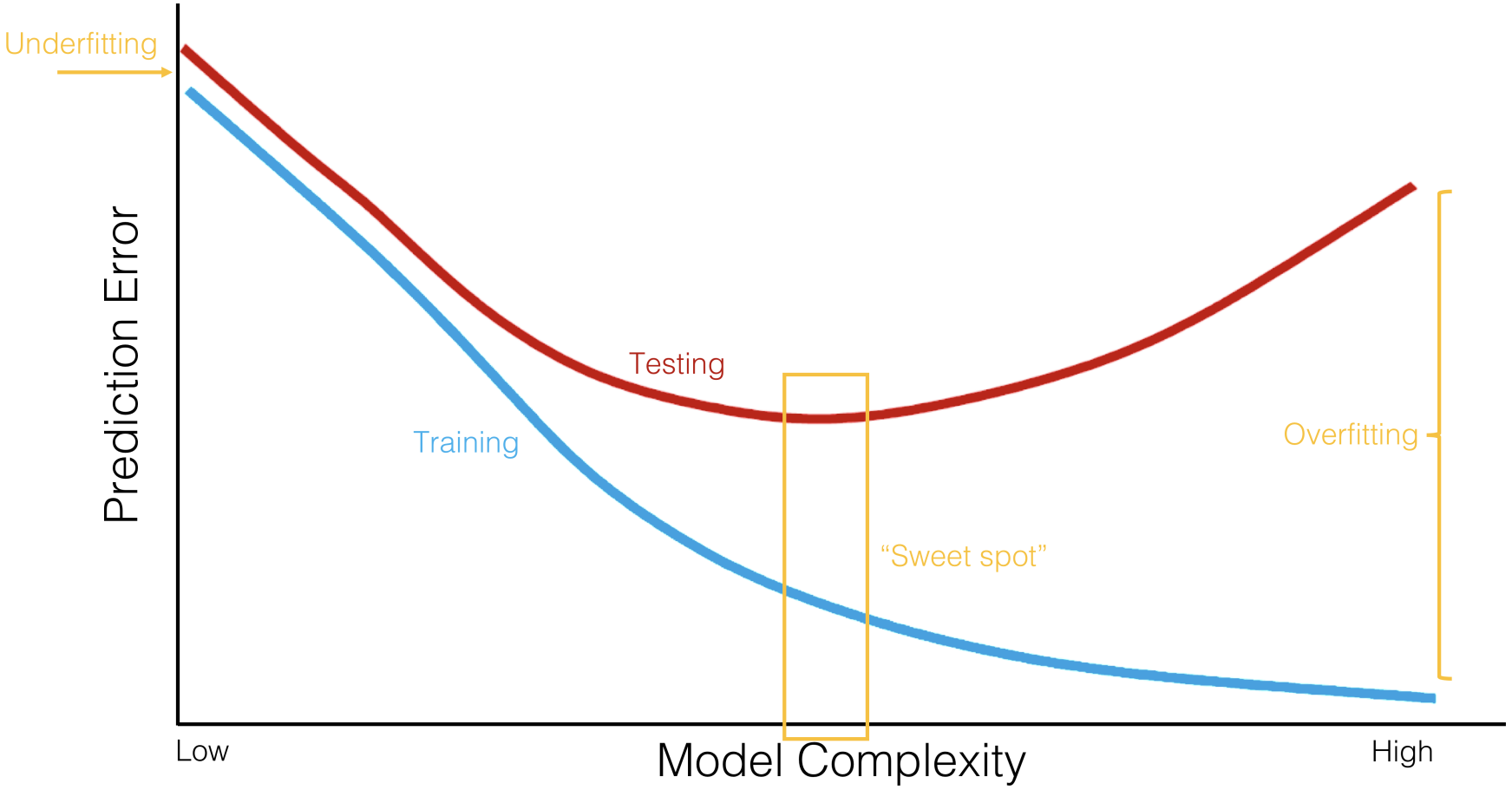
- **vague rule** – some people have black hair, some have brown hair, some blond, and some red (obviously true, but **too general** to be useful);

- **reasonable rule** – in populations of European descent, approximately 45% have black hair, 45% brown hair, 7% blond and 3% red;

- **overly specific rule** – in every 10,000 individuals of European descent, we predict there will be 46.32% with black hair, 47.27% with brown hair, 6.51% with blond hair, and 0.00% with red hair.

With the **overly specific rule**, we predict that there are no redheads in populations of European descent, which is false – this rule is **too specific** to the particular training subset that was used to produce it.

Perhaps the problem is that the data is **not representative** – using a training set with redheads would yield a "better" rule. But "**over-reporting/overconfidence**" (significant digits) is also part of the problem.



Underfitting and overfitting as a function of model complexity (we will revisit this)

Underfitting can be overcome by using **more complex** models (or models that use a **larger proportion** of a dataset's variables).

Overfitting can be overcome in several ways:

- **using multiple training sets** with overlap being allowed – this has the effect of reducing the odds of finding **spurious patterns** based on **quirks of the training data**;
- **using larger training sets** may also remove signal which is **too specific** to small training sets: a $70\%/30\%$ split is often suggested;
- using **simpler** models (or models that use a **smaller** proportion of a dataset's variables).

We will re-visit this.

When using **multiple training sets**, the **size** of the dataset may also affect the suggested strategy.

When faced with:

- **small datasets** (depends on numerous factors: CPU power and number of tasks) – use 100-200 repetitions of a **bootstrap procedure**;
- **average-sized datasets** (ditto, **less than a few thousand observations**, in genereal), use a few repetitions of **10-fold cross-validation**;
- **large datasets**, use a few repetitions of a **holdout split** $(70\%/30\%?)$.

No matter which strategy is eventually selected, the ML approach requires ALL models to be evaluated on **unseen data**. More to come.

1.3.3 – Appropriateness and Transferability

DS models will continue to be used **heavily** in the near future.

There are **pros** and **cons** to their use on **ethical/other non-technical grounds**, but their applicability is also driven by **technical considerations**.

DS/ML/AI methods are **not** appropriate if:

- **legacy** datasets **must** be used instead of **ideal** datasets;
- the dataset has attributes that accurately predict a value of interest, but these attributes **are not available** when a prediction is required, and
- class membership or numerical outcome is going to be predicted using **UL**.

Models make assumptions about what is **relevant** to their workings, but we tend to only gather data **assumed to be relevant** to a particular situation.

If the data is used in other contexts $\Longrightarrow$ no way to **validate the results**.

Not just an esoteric consideration: **over-generalizations** and **inaccurate predictions** can lead to **harmful results**.

Examples:

- Clustering loan default data might lead to a cluster contains many defaulters – if new instances get added to this cluster, should they be classed as loan defaulters?
- The total time spent on a website may be predictive of a visitor's purchases, but the prediction must be made before the total time spent on the website is known.
- Can we use mortgage-default models to predict if a borrower will default on a car loan?

1.3.4 – Myths and Mistakes

We end with a few **DS/ML myths**:

1. DS/ML is only about **algorithms**;
2. DS/ML focuses only on **predictive accuracy**;
3. DS/ML requires **data warehousing**;
4. DS/ML requires **large quantities of data**, and
5. DS/ML only requires **technical expertise**.

Common data analysis/DS/ML mistakes:

1. selecting the **wrong problem**;
2. getting by without **metadata understanding**;
3. not **planning** the data analysis **process**;
4. insufficient **business/domain knowledge**;
5. using **incompatible** data analysis tools;
6. using **too-specific** tools;
7. constantly favouring **aggregates** over **individual outcomes**;
8. running **out of time**;
9. measuring results **differently than stakeholders**, and
10. naïvely believing **what one is told about the data**.

Analysts must address these issues with stakeholders **ASAP**; safer to assume that everyone is on **different pages** – **prod** and **ask**, **early** and **often**.

1.4 – Association Rules Mining

Association rules discovery (ARD) is a type of **unsupervised learning** that finds **connections** among the attributes (variables) and levels (values), and combinations thereof, of a dataset's observations.

For instance, we might analyze a (hypothetical) dataset on the physical activities and purchasing habits of North Americans and discover that

- runners who are also triathletes (the **premise**) tend to drive Subarus, drink microbrews, and use smart phones (the **conclusion**), or
- individuals who have purchased home gym equipment are unlikely to be using it 1 year later, say.

The presence of a **correlation** between the **premise** and the **conclusion** does not necessarily imply the existence of a **causal relationship** between them.

It is difficult to "demonstrate" **causation** *via* data analysis; in practice, decision-makers pragmatically (and often **erroneously**) focus on the second half of "Correlation isn't causation. But it's a big hint." (E. Tufte)

When it is raining, then it there are clouds in the sky (**causal**) vs.
When there are clouds in the sky, it is raining (**not causal**).

Example: being a triathlete does not **cause** one to drive a Subaru, but Subaru Canada thought that the connection was strong enough to offer to reimburse the registration fee at an IRONMAN 70.3 competition in 2018!

1.4.1 – Overview

ARD is also known as **market basket analysis** after the original application: supermarkets record the contents of **baskets** at check-out to determine which items are **frequently purchased together**.

Example: bread and milk are **often** purchased together, but this is of **little** interest given the **high** frequency of baskets containing milk **or** bread.

If 70% of baskets contain milk and 90% contain bread, we expect:

$$\textbf{at least } 90\% \times 70\% = \mathbf{63}\%$$

of baskets contain milk **and** bread (assuming **total independence**).

If we then observe that 72% of baskets contain **both items** (a $1.15-$fold increase on the expected proportion), we conclude that there is at best a **weak correlation** between the purchase of milk and the purchase of bread.

Example: sausages and buns are not purchased **as frequently as milk** and **bread**, but they are purchased **as a pair** more often than expected given the frequency of baskets containing sausages **or** buns.

If 10% of baskets contain sausages, and 5% contain buns, we expect:

$$\textbf{at least } 10\% \times 5\% = \mathbf{0.5\%}$$

of baskets contain sausages **and** buns (assuming **total independence**).

If we then observe that 4% of baskets contain **both items** (an $8-$fold increase on the expected proportion), we conclude that there is a **strong correlation** between the purchase of sausages and the purchase of buns.

Application: supermarkets can use this information

- advertise a sale on sausages while **quietly** raising the price of buns;
- this may bring in **higher customer numbers** into the store;
- this may increase the **sale volume for both items** while keeping the combined price of the two items **constant**.

The marketing team is banking on customers **not shopping around** to get the best deal on hot dogs **and** buns, which may not be a valid assumption.

Applications

- finding **related concepts** in text documents – looking for combination of words that represent a joint concept;
- detecting **plagiarism** – looking for specific sentences that appear in multiple documents, or for documents that share specific sentences;
- searching for diseases frequently associated with a set of **biomarkers**;
- **altering** circumstances to take advantage of correlations or to **modify** the likelihood of certain outcomes – **suspected causal effects**;
- imputing missing data, text autofill and autocorrect, etc.

Correlation and Causation

Association rules can automate **hypothesis discovery**, but we must remain **correlation-savvy**.

If attributes A and B are correlated in a dataset, there are 4 possibilities:

- A and B are correlated **entirely by chance** in this particular dataset;
- A is a **re-labeling** of B (or *vice-versa*);
- A **causes** B (or *vice-versa*), or
- some combination of attributes $C_1, \ldots, C_n$ (which may not be available in the dataset) **cause both** A **and** B.

Real-life examples:

- Walmart has found that sales of strawberry Pop-Tarts increase about seven-fold in the days preceding the arrival of a hurricane;
- Xerox employees engaged in front-line service and sales-based positions who use Chrome and Firefox browsers perform better on employment assessment metrics and tend to stay with the company longer, and
- University of Cambridge researchers found that liking "Curly Fries" on Facebook is predictive of high intelligence.

It can be tempting to try to **explain** these results.

Possible explanations:

- when faced with a coming disaster, people stock up on comfort or nonperishable foods;
- the fact that an employee takes the time to install another browser shows that they are an informed individual and that they care about their productivity, or
- an intelligent person liked this Facebook page first, and her friends saw it, and liked it too, and since intelligent people have intelligent friends (?), the likes spread among people who are intelligent.

While these explanations **might** be the right ones, there is **nothing in the data** that supports them.

ARD **finds** interesting rules, but it **does not explain them**.

Cannot be over-emphasized: **correlation does not imply causation**.

Analysts might not have much **control over the matter**, but they should do what they can so that the following **do not see the light of day**.

Misleading headlines:

- "Pop-Tarts" get hurricane victims back on their feet;
- Using Chrome of Firefox improves employee performance, or
- Eating curly fries makes you more intelligent.

Definitions

A **rule** $X \to Y$ is a statement of the form "if X (**premise**), then Y (**conclusion**)" built from **logical combinations** of **attribute levels**.

A rule **does not need to be true for all observations** in the dataset – there may be instances where only the **premise** is satisfied.

Some of the "best" rules may be those which are only accurate 10% of the time, as opposed to rules which are accurate 5% of the time, say.

As always, **it depends on the context**.

To determine a **rule's strength**, we compute various **rule metrics**:

- the **support** measures the **frequency** at which a rule occurs in a dataset – **low support** values indicate rules that **rarely** occur;
- the **confidence** measures the **reliability** of the rule (how often the conclusion occurs in the data given that the premise has occurred) – rules with **high confidence** are "**truer**", in some sense;
- the **interest** measures the **difference** between its **confidence** and the relative **frequency of its conclusion** – rules with **higher absolute** interest are... **more** interesting (?);
- the **lift** measures the **increase in the frequency of the conclusion** which can be **explained by the premises** – with **high** lifts ($\gg 1$), the conclusion occurs **more** frequently than it would if it were independent of the premise;
- the **conviction**, the **all-confidence**, the **leverage**, the **collective strength**, etc.

In a dataset with N observations, let $\mathrm{Freq}(A) \in \{0, 1, \ldots, N\}$ represent the **count** of the dataset's observations for which property A holds.

Rule metrics:

$$\text{Support}(X \to Y) = \frac{\mathbf{Freq}(X \cap Y)}{N} \in [0, 1]$$

$$\text{Confidence}(X \to Y) = \frac{\mathbf{Freq}(X \cap Y)}{\mathbf{Freq}(X)} \in [0, 1]$$

$$\text{Interest}(X \to Y) = \mathbf{Confidence}(X \to Y) - \frac{\mathbf{Freq}(Y)}{N} \in [-1, 1]$$

$$\text{Lift}(X \to Y) = \frac{N^2 \cdot \mathbf{Support}(X \to Y)}{\mathbf{Freq}(X) \cdot \mathbf{Freq}(Y)} \in (0, N^2)$$

$$\text{Conviction}(X \to Y) = \frac{1 - \mathbf{Freq(Y)}/N}{1 - \mathbf{Confidence}(X \to Y)} \geq 0$$

Example: British Music Dataset

Consider a music dataset containing data for $N = 15,356$ British music lovers and a **candidate rule** RM:

"If an individual is born before 1976 (X), then they like *Help!* (Y)".

Let's assume further that

- $\text{Freq}(X) = \mathbf{3888}$ individuals were born before 1976;
- $\text{Freq}(Y) = \mathbf{9092}$ individuals like *Help!*, and
- $\text{Freq}(X \cap Y) = \mathbf{2720}$ individuals were born before 1976 and like *Help!*.

We can easily compute the 5 metrics for RM:

$$\text{Support}(\text{RM}) = \frac{2720}{15,356} \approx \mathbf{18\%}$$

$$\text{Confidence}(\text{RM}) = \frac{2720}{3888} \approx \mathbf{70\%}$$

$$\text{Interest}(\text{RM}) = \frac{2720}{3888} - \frac{9092}{15,356} \approx \mathbf{0.11}$$

$$\text{Lift}(\text{RM}) = \frac{15,356^2 \cdot 0.18}{3888 \cdot 9092} \approx \mathbf{1.2}$$

$$\text{Conviction}(\text{RM}) = \frac{1 - 9092/15,356}{1 - 2720/3888} \approx \mathbf{1.36}$$

These values are easy to interpret: RM occurs in $\mathbf{18\%}$ of the dataset's instances, and it holds true in $\mathbf{70\%}$ of the instances where the individual was born prior to 1976.

Is RM a **meaningful rule** about the dataset? Are being older and liking *Help!* **linked properties**?

If being younger and not liking *Help!* are **not also linked**, RM is not as meaningful **as it would appear at first glance**.

RM's **lift** is $\mathbf{1.2}$, which can be re-written as

$$1.2 \approx \frac{\mathbf{0.70}}{\mathbf{0.56}},$$

i.e. $\mathbf{56\%}$ of younger individuals also like the song.

Help!'s approval rates are **different** for the 2 age categories, but perhaps not as significantly as we would deduce using only the rule's **confidence** and **support**, which is reflected by its "**low**" **interest**, at 0.11.

The rule's **conviction** is 1.36, which means that the rule would be incorrect 36% more often if X and Y were **completely independent**.

Conclusion: the rule RM is **not entirely useless**, but meaningfulness depends on **context** and on **other rules**.

Recommendation: conduct a **preliminary exploration** of the AR space (using **domain expertise** when appropriate) in order to determine reasonable **threshold ranges** for the situation; candidate rules are then **discarded/retained** depending on these thresholds.

This requires the ability to **easily** generate potentially meaningful rules.

1.4.2 – Generating Rules

Challenge: generating a set of candidate rules which will be **retained**, without wasting time generating rules which will be **discarded**.

An **itemset** (**instance set**) is a list of **attributes** and **values**. We create **rules** from the **itemset** by adding "**IF ... THEN**" blocks to the instances.

As an example, from the instance set

$$\{\text{membership} = \text{True}, \text{age} = \text{Youth}, \text{purchasing} = \text{Typical}\},$$

we can create the following $3-$item rule, say:

$$\textbf{IF } (\text{membership} = \text{True } \textbf{AND } \text{age} = \text{Youth})$$
$$\textbf{THEN } \text{purchasing} = \text{Typical};$$

Suppose we care to generate k-item rules from an $n-$item set.

Each item can only be found **either** in the **premise** or in the **conclusion**. There are thus 2^k possible rules for a given choice of k items.

The rule **IF ... THEN** $\varnothing$ cannot be interpreted; we drop it from the list. There are thus $2^k - 1$ **interpretable** rules for a given $k-$itemset.

Additionally, there are $\binom{n}{k}$ ways to select k items from the $n-$itemset.

Thus, we can generate $\binom{n}{k}(2^k - 1)$ $k-$item rules from an $n-$item set, or

$$\textbf{total \# rules} = \sum_{k=1}^{n} \binom{n}{k}(2^k - 1) = 3^n - 2^n.$$

Note: rules of the form $\varnothing \to X$ (or **IF** $\varnothing$ **THEN** X) are denoted by X.

For the $3-$itemset example, there are:

- $\binom{3}{3}(2^3 - 1) = \mathbf{7}$ rules with 3 items;
- $\binom{3}{2}(2^2 - 1) = \mathbf{9}$ rules with 2 items;
- $\binom{3}{1}(2^1 - 1) = \mathbf{3}$ rules with 1 items,

Total: $\mathbf{19}$ rules built from the $3-$itemset (DUDADS, 19.3, *Generating Rules*)

Take-away: the number of rules **increases exponentially** when the number of features **increases linearly**.

This **combinatorial explosion** is a problem – it instantly disqualifies the **brute force** approach for any dataset with a realistic number of attributes.

How can we generate a small number of **promising** candidate rules?

1.4.3 – The A Priori Algorithm

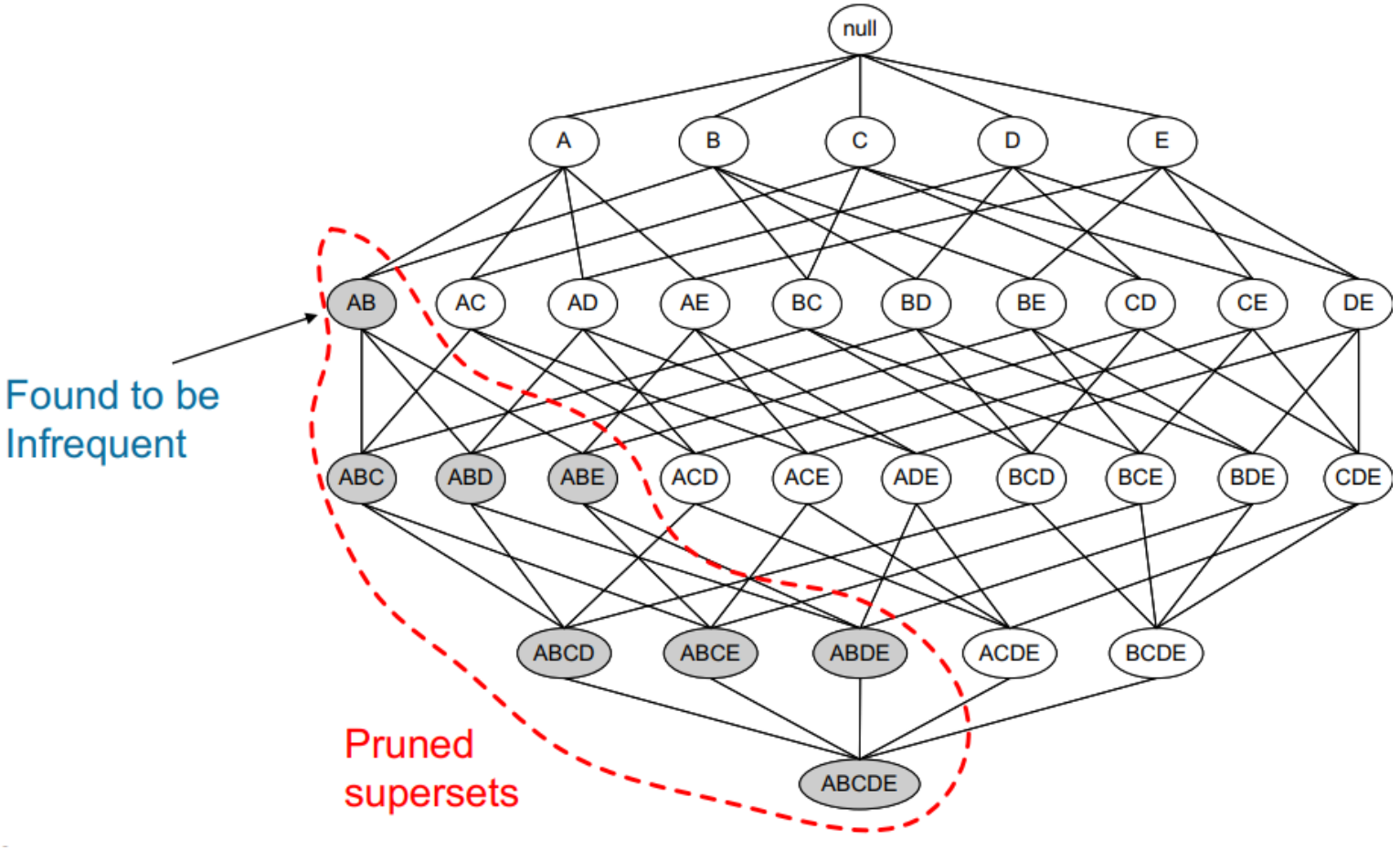
The **a priori** algorithm is an early attempt to answer that question.

Initially developed to work for **transaction data** (**goods** as **columns**, **customer purchases** as **rows**); every reasonable dataset can be transformed into such a dataset using **dummy variables**.

The algorithm looks for **frequent itemsets** from which to build candidate rules, instead of building rules from **all possible itemsets**.

Frequent **individual items** are extended into larger **item supersets**, themselves retained only if they occur **frequently enough**.

All non-empty subsets of a **frequent itemset** must also be **frequent**; all supersets of an **infrequent itemset** must also be **infrequent**.



Infrequent itemset in the $a\ priori$ network of a dataset with 5 items; no rule from the grey itemsets

ML technical jargon: a priori uses a **bottom-up approach** and the **downward closure property of support**.

The **memory savings** arise from the fact that the algorithm prunes candidates with **infrequent sub-patterns** and removes them from consideration for any future itemset: if a $1-$itemset is not considered to be **frequent enough**, any $2-$itemset containing it is also **infrequent**.

(See NHL Playoffs example, DUDADS, 19.3, $\mathit{The\ A\ Priori\ Algorithm}$)

This process requires a support threshold **input** $\Longrightarrow$ no guaranteed way to pick a "good" value:

- it has to be set sufficiently **high** to minimize the number of frequent itemsets **being considered**, but
- **not so high** that it removes too many candidates from the **output list**.

The optimal threshold values are **dataset-specific**.

A priori terminates when **no further** itemsets extensions are retained, which **must** occur in datasets with a finite # of categorical levels.

- **Strengths:** easy to implement and to parallelize
- **Limitations:** slow, requires frequent data set scans, not ideal for finding rules for infrequent and rare itemsets

In practice, we use **more efficient** algorithms:

- **Max-Miner** tries to identify frequent itemsets without enumerating them – it performs jumps in itemset space instead of using a bottom-up approach;
- **Eclat** is faster and uses depth-first search, but requires extensive memory storage.

A priori and Eclat are both implemented in the `R` package `arules`.

1.4.4 – Validation and Comments

How **reliable** are association rules? What is the likelihood that they occur **entirely by chance**? How **relevant** are they? Can they be generalized **outside the dataset**, or to **new data streaming in**?

These questions are **notoriously difficult to answer** for ARD: **statistically sound** ARD can help reduce the risk of finding **spurious** associations to a **user-specified significance level**.

ARD and a priori comments:

- Since **frequent** rules correspond to instances that occur repeatedly in the dataset, algorithms that generate itemsets often try to **maximize coverage**.

- When **rare events** are more meaningful (such as detection of a **rare disease** or a **threat**), we need algorithms that can generate rare **itemsets**. **This is not a trivial problem**.

- Continuous data has to be binned into **categorical** data in order to generate AR. There are many ways to accomplish this, so the same dataset can give rise to **completely different** AR. This can create **credibility** issues with clients and stakeholders.

- Other popular algorithms include: AIS, SETM, aprioriTid, aprioriHybrid, PCY, Multistage, Multihash, etc.

- Additional evaluation metrics can be found in the `arules` documentation.

1.4.5 – Example: Titanic Dataset

The *Titanic* dataset consists of 4 categorical attributes for each of the 2201 people aboard the Titanic when it sank in 1912:

- **class** (1st class, 2nd class, 3rd class, crewmember)
- **age** (adult, child)
- **sex** (male, female)
- **survival** (yes, no)

The natural question of interest for this dataset is:

"**How does survival relate to the other attributes?**"

This is not **UL**: the interesting rules' **structure** is fixed to conclusions of the form $\mathrm{survival} = \mathrm{Yes}$ or $\mathrm{survival} = \mathrm{No}$ (we could evaluate the performance on a test set).

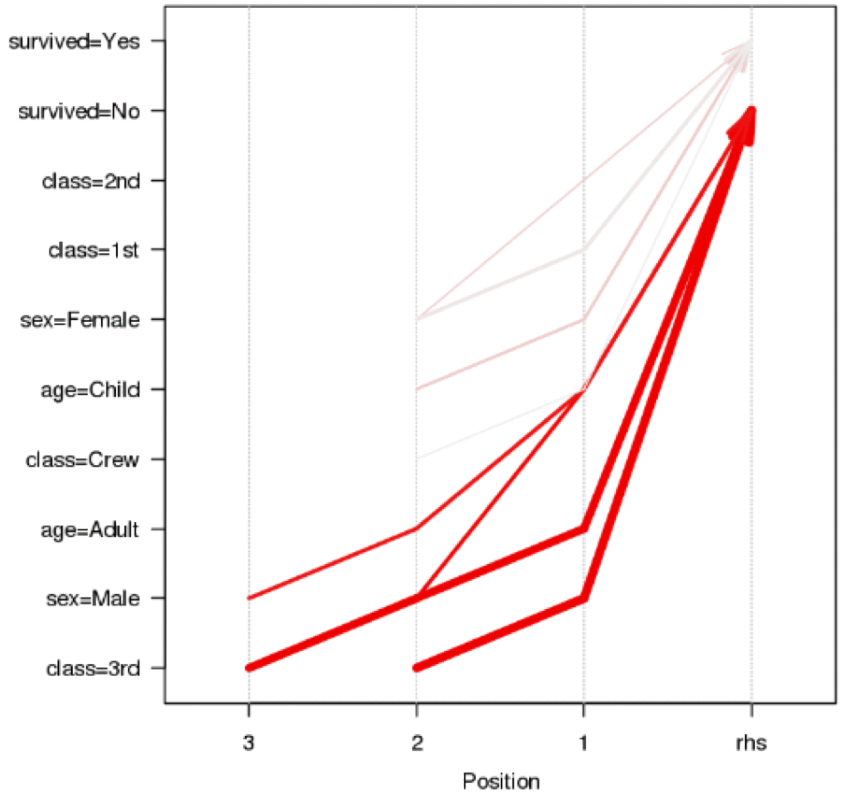
In this example, we treat the problem as a **descriptive task**, not a **predictive task**: the situation on the Titanic has **little bearing** on survival patterns in modern times.

Problem: use fixed-structure association rules to **describe** and **explore** survival conditions on the Titanic. Who survived? Who did not?

Solution: we use the `arules` implementation of the a priori algorithm in `R` to generate and prune candidate rules, eventually leading to **8 rules**.

But remember: **correlation does not imply causation!**

Rule	Supp	Conf	Lift
IF class = 2nd AND age = Child THEN survived = Yes	0.01	1	3.10
IF class = 1st AND sex = Female THEN survived = Yes	0.06	0.97	3.01
IF class = 2nd AND sex = Female THEN survived = Yes	0.04	0.88	2.72
IF class = Crew AND sex = Female THEN survived = Yes	0.00	0.87	2.70
IF class = 2nd AND sex = Male AND age = Adult THEN survived = No	0.07	0.92	1.35
IF class = 2nd AND sex = Male THEN survived = No	0.07	0.86	1.27
IF class = 3rd AND sex = Male AND age = Adult THEN survived = No	0.18	0.84	1.24
IF class = 3rd AND sex = Male THEN survived = No	0.19	0.83	1.22



Visualization of 8 Titanic association rules with parallel coordinates

(See DUDADS, 19.7, $ARM: Titanic$ for details)

References

T.Hastie, R.Tibshirani, and J.Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. Springer, 2008.

G.James, D.Witten, T.Hastie, and R.Tibshirani, An Introduction to Statistical Learning: With Applications in R. Springer, 2014.

C.C.Aggarwal and C.K.Reddy, Eds., Data Clustering: Algorithms and Applications. CRC Press, 2014.

C.C.Aggarwal, Data Mining: The Textbook. Springer, 2015.

D.Barber, Bayesian Reasoning and Machine Learning. Cambridge Press, 2012.

D.Robinson, "What's the difference between data science, machine learning, and artificial intelligence?". Variance Explained, Jan 2018.

D.Woods, "Bitly's Hilary Mason on 'what is a data scientist?'". Forbes, Mar 2012.

F.Provost and T.Fawcett, Data Science for Business. O'Reilly, 2015.

D.Dua and E.Karra Taniskidou, "UCI Machine Learning Repository." Irvine, CA: University of California, School of Information; Computer Science, 2017.

A.B.Jensen et al., "Temporal disease trajectories condensed from population-wide registry data covering 6.2 million patients," Nature Communications, vol.5, 2014, doi: 10.1038/ncomms5022.

S.E.Brossette, A.P.Sprague, J.M.Hardin, K.B.Waites, W.T.Jones, and S.A.Moser, "Association Rules and Data Mining in Hospital Infection Control and Public Health Surveillance," Journal of the American Medical Informatics Association, vol.5, no.4, pp.373–381, Jul.1998, doi: 10.1136/jamia.1998.0050373.

Subaru Canada, "Athlete rebate."

E.Siegel, Predictive analytics: The power to predict who will click, buy, lie or die. Predictive Analytics World, 2016.

E.Garcia, C.Romero, S.Ventura, and T.Calders, "Drawbacks and solutions of applying association rule mining in learning management systems," 2007.

Wikipedia, "Association rule learning." 2020.

E.R.Omiecinski, "Alternative interest measures for mining associations in databases," IEEE Transactions on Knowledge and Data Engineering, vol.15, no.1, pp.57–69, 2003, doi: 10.1109/TKDE.2003.1161582.

G.Piatetsky-Shapiro, "Discovery, analysis, and presentation of strong rules," 1991.

C.C.Aggarwal and P.S.Yu, "A new framework for itemset generation," in Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on principles of database systems, 1998, pp.18–24.doi: 10.1145/275487.275490.

P.-N.Tan, V.Kumar, and J.Srivastava, "Selecting the right objective measure for association analysis," Inf.Syst., vol.29, no.4, pp.293–313, Jun.2004, doi: 10.1016/S0306-4379(03)00072-3.

M.Hahsler and K.Hornik, "New probabilistic interest measures for association rules," CoRR, vol.abs/0803.0966, 2008.

T.Chou, "Apriori: Association Rule Mining In-Depth Explanation and Python Implementation," Towards Data Science, Oct 2020.

J.Leskovec, A.Rajamaran, and J.D.Ullman, Mining of Massive Datasets. Cambridge Press, 2014.

M.Risdal, "Exploring survival on the titanic," Kaggle.com, 2016.