Contents

Chapter 1

Decision Trees

Let's start with training set:

Definition 1.0.1: Training set

is defined training set a *set of examples*, where: $\langle x^{(i)}, y^{(i)} \rangle$ where:

- i is the instance of the example
- $x^{(i)} \in X$ is the set of *input*
- $y^{(i)} \in Y$ is the set of *output*

the problem of machine learning is to find a function $h : X \to Y$ that approximates the real function $f : X \to Y$. We have two types of problems:

- **Classification**: Y is a discrete set of values (e.g. $\{0, 1\}$)
- **Regression**: Y is a continuous set of values (e.g. $\mathbb{R}$)

1.1 Hypothesis space

In machine learning the *hypothesis space* H is defined as the set of all possible functions that can be used to approximate the real function $f : X \to Y$. Formally:

Definition 1.1.1: Hypothesis space

A hypothesis space H is defined as the set: $H = \{h | h : X \to Y\}$ where:

- h is a function (hypothesis) that maps input X to output Y
- X the input space (features, domain of data).
- Y the output space (labels, range of data).
- $|H|$ is the size of the hypothesis space (number of possible hypotheses)

this let us to define the model:

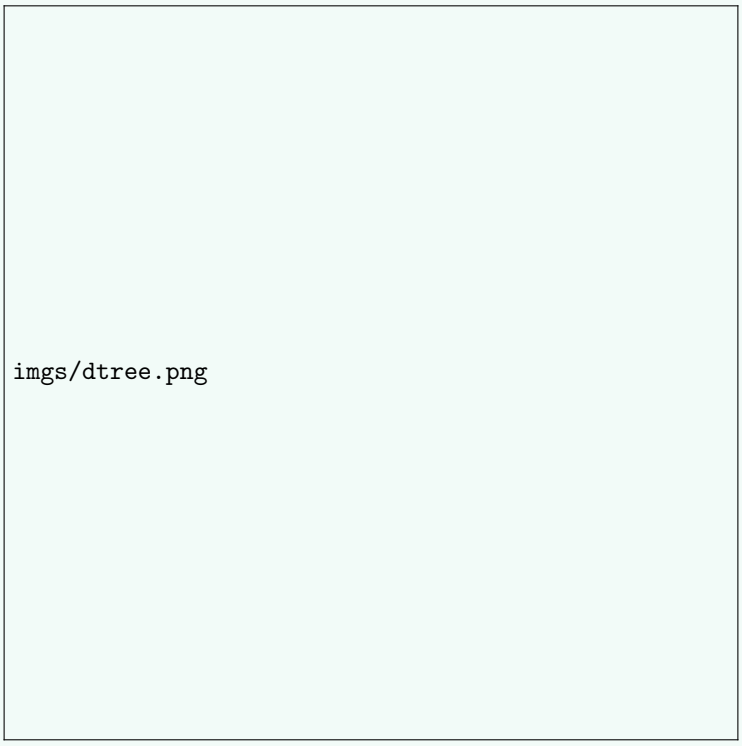
Definition 1.1.2: Model

A model is a way to compute a function $h \in H$ from the training set.

Example 1.1.1 (Decision tree)

A good day to play tennis? Our function F is:

$$\mathbb{F} : Outlook \times Humidity \times Wind \times Temp \rightarrow PlayTennis?$$

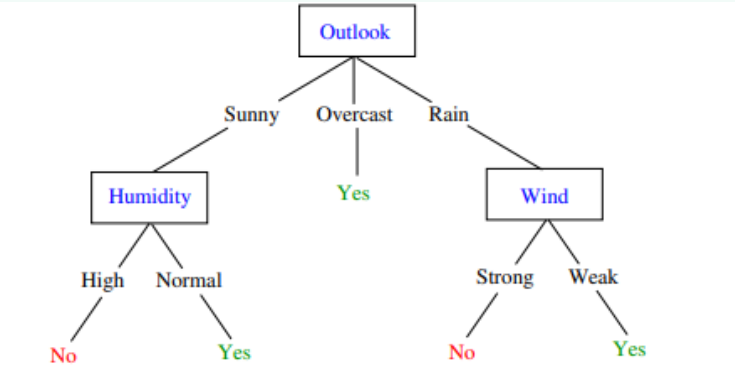


where:

- $Outlook \in \{Sunny, Overcast, Rain\}$
- $Humidity \in \{High, Normal\}$
- $Wind \in \{Weak, Strong\}$
- $PlayTennis? \in \{Yes, No\}$

Every node tests an attribute. Each branch corresponds to one of the possible values for that attribute. Each leaf node assigns a classification (Yes or No), in other words predicts the answer Y.
The prblem configurationg is the following:

- X is the set of all possible $x \in X$ that corresponds to a vector of attributes $(Outlook, Humidity, Wind, Temp)$
- Target function $f : X \rightarrow Y$ is the function that maps the attributes to the target variable $PlayTennis?$ (booleans)
- Hypothesis space $H = \{h | h : X \rightarrow Y\}$ is the set of all possible decision trees that can be constructed using the attributes in X to predict the target variable Y



1.1.1 Top-down inductive construction

Let $X = X_1 \times X_2 \cdots \times X_n$ where $X_i = \{\text{Truee}, \text{False}\}$
Can we represent, for instance, $Y = X_2 \wedge X_5$? or $Y = X_2 \wedge X_5 \vee (\neg X_3) \wedge X_4 \wedge X_1$?
and:

- do we have a decision tree for each h in the space hypothesis?
- if the tree exists, is it unique?
- if it is not unique, do we have a preference?

Theorem 1.1.1 Basta - Bonzo

Main loop:

- **Pick the "best" attribute** X_i: At the current node, choose which featue/attribute will bwst split the training data.

  Best means: the attribute that gives the most information gain
- **Create a child node for each possible value of** X_i: for instance if attribute is "weather" with values "sunny", "rainy", "overcast", create three child nodes.
- **check if all examples in the chils node are pure**: if all examples belong to the same class (e.g., all "yes" or all "no"), make that node a leaf node with that class label. If not repeat the process recursively for each child node.

1.1.2 Entropy

Definition 1.1.3: Entropy

The entropy $H(S)$ of a set of examples S is defined as:

$$H(S) = -\sum_{i=1}^{n} P(X = i) \log_2 P(X = i)$$

where:

- $P(X = i)$ is the proportion of examples in S that belong to class i
- n is the number of classes (the number of possibile values of X)

In other words, Entropy meausers the *degree of uncertainty* of the information. It is maximal when X is uniformly distributed (all classes have the same probability) and minimal (zero) when all examples belong to the same class (pure set)

Missing image: imgs/entropy.png

Information Theory (Shannon 1948)

The entropy is the average amount of information produced by a stochastic source of data. The *information* is associated to the *probability* of each datum (the surprise element):

- An event with probability 1 (certain event) provides no information (no surprise): $I(1) = 0$.
- An event with probability 0 (impossible event) provides infinite information (really surprising): $I(0) = \infty$.
- Given two independent events A and B, the information provided by both events is the sum of the information provided by each event:
$$I(A \cap B) = I(A) + I(B)$$

So is natoral defining
$$I(p) = -\log_2(p)$$

Code Theory (Shannon-Fano 1949, Huffman 1952)

The entropy is also related to the avarage number of bits required to transmit outcomes produces by a stochastic source process x.
Let suppose to have n events with same probability $p_i = \frac{1}{n}$. Home many bits do we need to encode these events? The answer is $\log_2(n)$ bits. For instance, if we have 4 events, we need 2 bits to encode them:.
In this case:
$$H(X) = -\sum_{i=1}^{n} P(X = i) \log_2 P(X = i) = -\sum_{i=1}^{n} \frac{1}{n} \log_2 \frac{1}{n} = \log_2(n)$$

1.1.3 Information Gain

In a decision tree, the goal is to maximize the information gain during the execution of the algorithm. In other words, the final split should result in the minimum possible impurity. Here are the main formulas:

Theorem 1.1.2 Entropy of X

$$H(X) = -\sum_{i=1}^{n} P(X = i) \log_2 P(X = i)$$

Theorem 1.1.3 Conditional Entropy of X given a specific $Y = v$

$$H(X \mid Y = v) = -\sum_{i=1}^{n} P(X = i \mid Y = v) \log_2 P(X = i \mid Y = v)$$

This measures the entropy of X restricted to the subgroup where $Y = v$.

Theorem 1.1.4 Conditional Entropy of X given Y

$$H(X \mid Y) = \sum_{v=1}^{m} P(Y = v)\, H(X \mid Y = v)$$

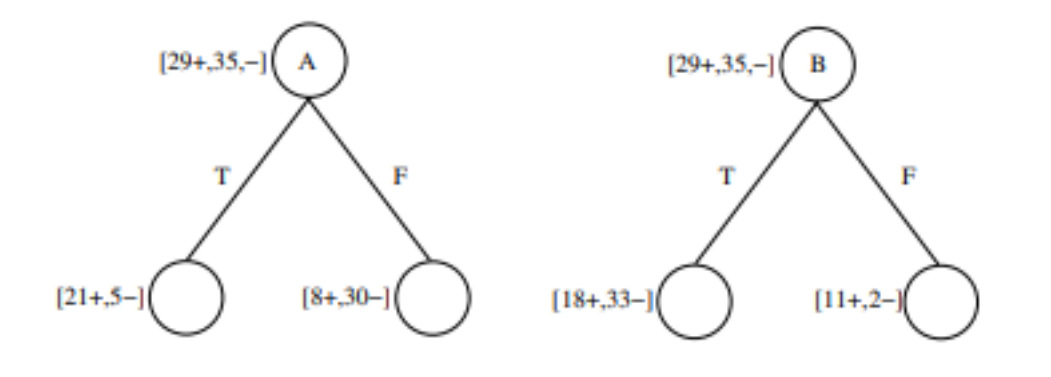
This is the generalization of **??**, used to evaluate the utility of an attribute. It measures the average impurity that remains in X after splitting the data using all possible values of Y.

Theorem 1.1.5 Information Gain between X and Y

Here we are! $I(X,Y) = H(X) - H(X\|Y) = H(Y) - H(Y\|X)$

Example 1.1.2 (Information gain)

Let us measure the entropy reduction of the target variable Y due to some attribute X, that is the information gain $I(Y,X)$ between Y and X



$$H(Y) = -\frac{29}{64}\log_2\left(\tfrac{29}{64}\right) - \frac{35}{64}\log_2\left(\tfrac{35}{64}\right) = 0.994$$

$$H(Y \mid A = T) = -\frac{21}{26}\log_2\left(\tfrac{21}{26}\right) - \frac{5}{26}\log_2\left(\tfrac{5}{26}\right) = 0.706$$

$$H(Y \mid A = F) = -\frac{8}{38}\log_2\left(\tfrac{8}{38}\right) - \frac{30}{38}\log_2\left(\tfrac{30}{38}\right) = 0.742$$

$$H(Y \mid A) = 0.706 \cdot \tfrac{26}{64} + 0.742 \cdot \tfrac{38}{64} = 0.726$$

$$I(Y,A) = H(Y) - H(Y \mid A) = 0.994 - 0.726 = 0.288$$

$$H(Y \mid B) = 0.872$$

$$I(Y,B) = 0.122$$

Chapter 2

Overfitting

Let us consider the error of the hypothesis h

- on the training set, $error_{train}(h)$
- on the full data set $\mathcal{D}$, $error_{\mathcal{D}}(h)$

Definition 2.0.1: Overfitting

It's said taht h *overfits* the training set if there exixsts another hypotheses h' such that:

$$\text{error}_{\text{train}(h)} < \text{error}_{\text{train}(h')}$$

but

$$\text{error}_{\mathcal{D}(h)} > \text{error}_{\mathcal{D}(h')}$$

These models (h and h') represent two different situations. The first corresponds to a model that fits the training dataset very closely, including its uncertainty and noise. The second is simpler: it captures only the general trend of the training data and avoids fitting the noise. As a consequence, the error with respect to the true data distribution $\mathcal{D}$ is larger for the first model than for the second. The second one is better! Let's generalised.
But *We do not know* $\mathcal{D}$

2.1 Avoiding the over fitting

2.1.1 Detecting the Overfitting: validation set

For Detecting the Overfitting it's usefull divinding the dats aviables in two disjoint sets:

- **Training set**: set of datas that the model *use for learning*. The dtree is built by this datas
- **validation set**: This set is not shown during the training- It's used as "test" for evaluating the accuracy of the model

2.1.2 Early stopping

This is a proactive strategy. Instead of let the tree grows untill his major complexity, it's sopped first the possibility of Overfitting. The growing of a branch is stopped if these two conditions is verified:

- **The improvement is too small**: if a possible divsion of datas produces a gain of information below a certain threshold, it means that it's not usefull to continue
- **There are not enaugh datas**: if a node contains a number of examples too musch low, any decision taken would be statistically unreliable and probably based on noise. The tree stops to avoid creating rules based on coincidences.

2.1.3 Post - Pruning

This strategy is **reactive**. The decision tree is let grow completely on the training set, which may lead to overfitting, and then the useless or harmful branches are pruned.

Definition 2.1.1: Reduce-Error Post-Pruning

The *reduce-error post-pruning* technique works as follows:

- build the tree completely
- evaluate each branch using a validation set
- prune the branch whose removal improves accuracy the most
- repeat until no further pruning improves the accuracy

Chapter 3

Probabilistic approach

3.1 core idea

we have two main points of views:

- **traditional view**: we wanna to approximate a function $f : X \to X$
- **Probabilist view**: we wanna compute probablilities: $p : P(Y \mid X)$

3.1.1 Probs basics

Random variables

A random variables X represents an oyt come about which we're ncertain

Example 3.1.1 (Random variables)

- X =`true` if a randomly drawn stdent is male
- X = first name of the student
- X =`true` if a randomly drawn stdent have the same birthday

Formal def:

Definition 3.1.1: Probs variables

the set Ω of the possible outcomes is called the sample space. It is said random variable a measurable function over Ω:

- Discrete: $\Omega \to \{m, f\}$
- Continuos: $\Omega \to \mathbb{R}$

Definition 3.1.2: Probs def

it is defined $P(X)$ is the fraction of times X is true in repeated runs of the same experiment.

Note:

The definition requires that all samples

Pay attention:

Wrong Concept 3.1: bad examples

Sample space, let Ω be a space made the possibile sum:

$$\Omega = \{2, 3, 4, \ldots, 12\}$$

Problem: not all sums are equally likely! It should be:

$$P(sum = 2) = 1/11$$
$$P(sum = 7) = 1/11$$

but in reality:

- Sum = 2: can only happen one way: $(1, 1)$
- Sum = 7: can happen six ways:$(1, 6), (2, 5), (3, 4), (4, 3), (5, 2), (6, 1)$

so

$$P(sum = 2) \neq P(sum = 7)$$

A correct approach is

Claim 3.1.1 correct approach

Be $\Omega = (1, 1), (1, 2), (1, 3), ..., (6, 5), (6, 6)$, where $|\Omega| = 36$outcomes
each pair has equally probability $= \frac{1}{36}$
Now here is a correctly computing:

$$P(sum = 2) = \frac{|(1,1)|}{36} = \frac{1}{36}$$
$$P(sum = 7) = \frac{|(1,6),(2,5),(3,4),(4,3),(5,2),(6,1)|}{36} = \frac{6}{36}$$

The Axioms of Probability Theory

These are the fundamental rules that make probability a "reasonable theory of uncertainty":

Axioms of probability theory

(1) Non-negativity: $0 \leqslant P(A) \leqslant 1$ for all events A. (3.1)

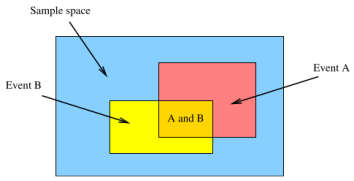
(2) Normalization: $P(\Omega) = 1$. (3.2)

(3) Countable additivity: If $A_1, A_2, \ldots$ are disjoint, then $P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i)$. (3.3)

Then:

Corollary 3.1.1 consequences of the axioms

- Monotonicity: If $A \subseteq B$, then $P(A) \leq P(B)$
- Union rule (for two events): $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
- $P(True) = 1$
- $P(False) = 0$



Derivied theorems

Corollary 3.1.2 Complement Rule

$$P(\neg A) = 1 - P(A)$$

Dm:

$$P(A \cup \neg A) = P(A) + P(\neg A) - P(A \cap \neg A)$$

But:

$$P(A \cup \neg A) = P(\text{True}) = 1 \quad \text{and} \quad P(A \cap \neg A) = P(\text{False}) = 0$$

Therefore:

$$1 = P(A) + P(\neg A) - 0 \quad \Longrightarrow \quad P(\neg A) = 1 - P(A)$$

Corollary 3.1.3 Partition Rule

$$P(A) = P(A \cap B) + P(A \cap \neg B)$$

Proof:

$$\begin{aligned} A &= A \cap (B \cup \neg B) && \text{[since } B \cup \neg B \text{ is always True]} \\ &= (A \cap B) \cup (A \cap \neg B) && \text{[distributive law]} \end{aligned}$$

Hence,

$$\begin{aligned} P(A) &= P((A \cap B) \cup (A \cap \neg B)) \\ &= P(A \cap B) + P(A \cap \neg B) - P((A \cap B) \cap (A \cap \neg B)) \\ &= P(A \cap B) + P(A \cap \neg B) - P(\text{False}) \\ &= P(A \cap B) + P(A \cap \neg B) \end{aligned}$$

Multivalued Discrete Random Variables

Definition 3.1.3: k-value Discrete Random Variables

A random variable A is *k-valued discrete* if it takes exactly one value from

$$\{v_1, v_2, \ldots, v_k\}.$$

Proposition 3.1.1 Key proprieties

1. **Mutual exclusivity:** For $i \neq j$,
$$P(A = v_i \cap A = v_j) = 0$$
2. **Exhaustiveness:**
$$P(A = v_1 \cup A = v_2 \cup \cdots \cup A = v_k) = 1$$

Conditional Probability

Definition 3.1.4: Conditional probs

The Conditional probs of the event A *given* the event B is defined as the quantity

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

Corollary 3.1.4 Cahin roule

$$P(A \cap B) = P(B)P(A \mid B) = P(A)P(B \mid A)$$

Independent Events

Definition 3.1.5: Independent Events

Events A and B are independent when:

$$P(A \mid B) = P(B)$$

(Meaning: B provides no information about A.)

Corollary 3.1.5 consequences

- $P(A \cap b) = P(A)P(B)$ (from chail roule)
- $P(B|A) = P(B)$ (symmetry)

Bayes' Rule: The Heart of Probabilistic ML (ok chat... really?)

Theorem 3.1.1 Bayes's roule

Now we have Bayes roule

$$P(A \mid B) = \frac{P(A)P(B|A)}{P(B)}$$

Proof: It's true by the chain roule that: $P(A \cap B) = P(B)P(A \mid B)$. It's true also the reverse case $P(A \cap B) = P(B)P(A \mid B)$.
Since both expressions equal $P(A \mid B)$, they must equal each other:

$$P(A)P(B \mid A) = P(B)P(A \mid B)$$

that it's equal to

$$P(A \mid B) = \frac{[P(A)P(B \mid A)]}{P(B)}$$

Example 3.1.2 (The trousers problem)

Setup:

- 60% of students are boys, 40% are girls
- girls wear in the same number skirt and trousers
- boys only wear trousers

If we see a student wearing trousers, what is the probability that is a girl?

Solution: The probs a priori that a strudent is a girl is

$$P(G) = \frac{2}{5}$$

the probability that a student wears trousers is

$$P(T) = \frac{1}{5} + \frac{3}{5} = \frac{4}{5}$$

the probability that a student wear trousers, given that the student is a girl, is

$$P(T \mid G) = 1/2$$

So

$$P(G \mid T) = \frac{p(G)p(T \mid G)}{P(T)} = \frac{2/5 \cdot 1/2}{4/5} = 1/4$$

Machine Learning Form

Machine Learning Form For discrete Y with values $\{y_1, y_2, \ldots, y_m\}$ and X with values $\{x_1, x_2, \ldots, x_n\}$:

$$P(Y = y_i \mid X = x_j) = \frac{P(Y = y_i) \cdot P(X = x_j \mid Y = y_i)}{P(X = x_j)}$$

Expanding the denominator:

$$\begin{aligned} P(X = x_j) &= \sum_i P(X = x_j, Y = y_i) \quad \text{[sum over all } Y \text{ values]} \\ &= \sum_i P(Y = y_i) \cdot P(X = x_j \mid Y = y_i) \quad \text{[chain rule]} \end{aligned}$$

Complete Bayes' Rule:

$$P(Y = y_i \mid X = x_j) = \frac{P(Y = y_i) \cdot P(X = x_j \mid Y = y_i)}{\sum_i P(Y = y_i) \cdot P(X = x_j \mid Y = y_i)}$$

Terminology:

$$\underbrace{P(Y \mid X)}_{\text{posterior}} = \frac{\overbrace{P(X \mid Y)}^{\text{likelihood}} \cdot \overbrace{P(Y)}^{\text{prior}}}{\underbrace{P(X)}_{\text{marginal}}}$$

- **Posterior** $P(Y \mid X)$: What we want – probability of Y given observed X
- **Likelihood** $P(X \mid Y)$: How likely is X if Y is true?
- **Prior** $P(Y)$: What we believed before seeing X
- **Marginal** $P(X)$: Overall probability of observing X (normalization constant)

Alternative form:

$$\text{Posterior} = \frac{\text{Likelihood} \cdot \text{Prior}}{\text{Marginal Likelihood}}$$

where:

$$\text{Marginal} = \sum_Y P(X \mid Y) \cdot P(Y)$$

The term "marginal" means we've **marginalized** (integrated/summed) over Y.

3.2 The Joint Distribution

Definition 3.2.1: Joint Distribution

Let $X_1, X_2, \ldots, X_n$ be discrete random variables. The *joint probability distribution* (or *joint distribution*) of these variables is the function:

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n)$$

which assigns to every possible combination of values $(x_1, x_2, \ldots, x_n)$ the probability that the random variables simultaneously take those values.
Formally, for discrete variables, the joint distribution satisfies:

- $0 \leqslant P(x_1, x_2, \ldots, x_n) \leqslant 1$ for all $(x_1, \ldots, x_n)$
- $\sum_{x_1} \sum_{x_2} \cdots \sum_{x_n} P(x_1, x_2, \ldots, x_n) = 1$

Let's see an example

Example 3.2.1 (Joint distribution)

- build a table with all possible combinations of values of random variables (features)
- compute the probability for any different combination of values

This table is the "Joint distribution"!
Having that we may compute the probability of any event expressible as a logical combination of the features, with this formula

$$P(E) = \sum_{row \in E} (row)$$

in words for calculating an event we must add each row that is contained by the event. Let's provide an example (of an example)
Let us compute the probability $P(M, poor)$

gender	w. hours	wealth	prob.
F	≤ 40	poor	0.25
F	≤ 40	rich	0.03
F	> 40	poor	0.04
F	> 40	rich	0.01
M	≤ 40	poor	0.33
M	≤ 40	rich	0.10
M	> 40	poor	0.13
M	> 40	rich	0.11

we have: $P(M, poor) = 0.33 + 0.13 = 0.46$

3.2.1 Inference with the Joint distribution

Here are with the inference:

Definition 3.2.2: Contintional probability

Let E_1 and E_2 be two events defined as logical conditions over subsets of the random variables (e.g., $E_1 : X_i = a, X_j = b$; $E_2 : Y = y$)
Then, *conditional probability* of E_1 given E_2 is:

$$P(E_1 \mid E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum_{row \in (E_1 \wedge E_2)} P(row)}{\sum_{row \in (E_2)} P(row)}$$

for instance:

Example 3.2.2 (Conditional probability)

Let's compute $P(M|poor) = \frac{P(M \wedge poor)}{P(poor)}$. We know that $P(M, poor) = 0.46$. Let us compute $P(poor)$:

gender	w. hours.	wealth	prob.
F	≤ 40	poor	0.25
F	≤ 40	rich	0.03
F	> 40	poor	0.04
F	> 40	rich	0.01
M	≤ 40	poor	0.33
M	≤ 40	rich	0.10
M	> 40	poor	0.13
M	> 40	rich	0.11

Easy! $P(poor) = .75 \wedge P(M|poor) = 0.46/0.75 = 0.61$

3.2.2 Complexity issues

Let us build the joint table relative to

$$P(Y = wealth | X_1 = gender, X_2 = orelav.)$$

X_1=gender	X_2=ore lav.	$P(rich \mid X_1, X_2)$	$P(poor \mid X_1, X_2)$
F	≤ 40	.09	.91
F	> 40	.21	.79
M	≤ 40	.23	.77
M	> 40	.38	.62

To fill the table we need to compute $4 = 2^2$ parameters
If we have n random variable $X = X_1 \times X_2, \ldots, X_n$ where each X_i is boolean, we need to compute 2^n parameters.
These parameters are *probabilities*: to get reasonable value we would need a huge amount of data.
In particular the The Joint Distribution Requires *Exponential parameters*

Example 3.2.3 (features and params)

- With just 10 binary features, you need $2^{11} - 1 = 2047$ parameters
- With 20 features: over 1 million parameters
- With 100 features: 2^{101} a number larger than the estimated atoms in the observable universe.

This is computationally and statistically infeasible.

USing Bayes

for reducing complexity, we can rewrite the formula with the Bayes' rule:

$$P(Y = y_i \mid X = x_j) = \frac{P(Y = y_i) \cdot P(X = x_j \mid Y = y_i)}{\sum_i P(Y = y_i) \cdot P(X = x_j \mid Y = y_i)}$$

generalising:

$$P(Y \mid X_1, X_2, \ldots, X_n) = \frac{P(Y) \cdot P(X_1, X_2, \ldots, X_n \mid Y)}{P(X_1, X_2, \ldots, X_n)}$$

But... there is a problem, it's required to know

$$P(X_1, X_2, \ldots, X_n \mid Y)$$

that is the joint distribution of the features given Y, that requires, another time, 2^n params

3.2.3 Naive Bayes

For atteniung the complexity, it's possible assume an indipendencies conditional hypotesis, called "Naïve Bayes":

$$P(X_1, X_2, \ldots, X_n|Y) = \prod_i P(X_i|Y)$$

So given Y, X_i and X_j are independent from each other. In other therms:

$$P(X_i|X_j, Y) = P(X_i|Y)$$

Note:

This means: onece we know Y, the feature $X_i \forall i$ are independents between each others

Example 3.2.4 (example 1)

A box contains two coins: a regular coin and a fake two-headed coin ($P(H) = 1$).Choose a coin at random, toss it twice and consider the following events:

- A = First coin toss is H
- B = Second coin toss is H
- C = First coin is regular

Example 3.2.5 (example 2)

For individuals, height and vocabulary are not independent, but they are if age is given.

Giga formula with naive bayes

Theorem 3.2.1 Bayes rule

$$P(Y = y_i \mid X_1, \ldots, X_n) = \frac{P(Y = y_i) \cdot P(X_1, \ldots, X_n \mid Y = y_i)}{P(X_1, \ldots, X_n)}$$

Proof: Left to mesco as exercice

Theorem 3.2.2 Naïve Bayes

$$P(Y = y_i \mid X_1, \ldots, X_n) = \frac{P(Y = y_i) \cdot \prod_j P(X_j \mid Y = y_i)}{P(X_1, \ldots, X_n)}$$

Proof: Left to Bonzo as exercice

Theorem 3.2.3 Classification of a new sample $x^{\text{new}} = \langle x_1, \ldots, x_n \rangle$

Given a new instance represented by the feature vector $x^{\text{new}} = (x_1, x_2, \ldots, x_n)$, the predicted class is obtained as:

$$Y^{\text{new}} = \arg\max_{y_i} P(Y = y_i) \cdot \prod_j P(X_j = x_j \mid Y = y_i)$$

Proof: Left to Bastiality as an exercise.

Note:

Theorem **??** expresses the decision rule of the Naïve Bayes classifier. Given a new vector of features $x^{\text{new}} = (x_1, x_2, \ldots, x_n)$, we estimate the most probable class y_i by maximizing the posterior probability $P(Y = y_i \mid X_1 = x_1, \ldots, X_n = x_n)$, which—under the conditional independence assumption—reduces to the product of the prior $P(Y = y_i)$ and the individual likelihoods $P(X_j = x_j \mid Y = y_i)$.

3.3 Learning algorithm

Given discrete Random Variables X_i, Y, there are two phases

- **Training**: in this phases the maching learn from the data of training set, estimating two types of probs:
  - **Prior** (prob of the classes). For any possible value y_k of Y , estimate

    $$\pi_k = P(Y = y_k)$$

    example: if 9 out of 14 matches are "Play = Yes", then $\pi_{yes} = \frac{9}{14}, \quad \pi_{no} = \frac{5}{14}$
  - **Likelihoods**:(conditional probabilities of features). for any possible value x_{ij} of X_i estimate:

    $$\theta_{ijk} = P(X_i = x_{ij} \mid Y = y_k)$$

    It's the probability that a certain feature X_i assumes the value X_{ij}, given y_k.
    example: $P(Outlook = Sunny \mid Play = Yes) = \frac{2}{9}$
- **Classification of** $a^{new} = \langle a_1, \ldots a_n \rangle$ (a vector with n observed values (one for each feature)). We want to establish which class it belong to

  decision-making formula:

  $$\begin{aligned} Y^{\text{new}} &= \arg\max_{y_k} P(Y = y_k) \cdot \prod_i P(X_i = a_i \mid Y = y_k) \\ &= \arg\max_k \pi_k \prod_j \theta_{ijk} \end{aligned}$$

  where:
  - $P(Y = y_k)$: prior
  - $P(X_i = a_i \mid Y = y_k)$: likelihood for each features
  - the prod $\prod_i$ is given by Naive assumption

Example 3.3.1 (a good day to play tennis?)

we wanna build a model that, given certain weather conditions, predict whether it is a good day to play tennis or not
Our class variable is:

$$Y = Play \in \{Yes, No\}$$

and the features observed are:

$$X_1 = Outlook \quad X_2 = Temp \quad X_3 = Humidity \quad X_4 = Wind$$

Here we have the dataset:

Table 3.1: Dataset for the *Play Tennis* classification problem

Outlook	Temp	Humidity	Wind	Play
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

TODO: TABELLA FATTA FARE DA UN LLM NON È VENUTA BENISSIMO

Calculating the prior

From the dataset we can compute the prior probabilities of the class variable Y:

$$P(Y = \text{Yes}) = \frac{9}{14}, \qquad P(Y = \text{No}) = \frac{5}{14}.$$

These represent the empirical frequencies of the two possible outcomes of Y.

Calculating the likelihoods

For each feature X_i and each class $Y = y_k$, we estimate the conditional probabilities

$$P(X_i = x_{ij} \mid Y = y_k),$$

that is, the probability of observing a certain feature value x_{ij} given that the class is y_k.
For example:

$$P(\text{Outlook} = \text{Sunny} \mid Y = \text{Yes}) = \frac{2}{9}, \qquad P(\text{Outlook} = \text{Sunny} \mid Y = \text{No}) = \frac{3}{5}.$$

These values are computed as the relative frequencies in the dataset.

Classification of a new instance

Suppose we want to classify the new day

$$x^{\text{new}} = (\text{Outlook} = \text{Sunny},\ \text{Temp} = \text{Cool},\ \text{Humidity} = \text{High},\ \text{Wind} = \text{Strong}).$$

We apply the Naïve Bayes decision rule:

$$Y^{\text{new}} = \arg\max_{y_i} P(Y = y_i) \cdot \prod_j P(X_j = x_j \mid Y = y_i).$$

For $Y = \text{Yes}$:

$$P(\text{Yes}) \cdot P(\text{Sunny}|\text{Yes}) \cdot P(\text{Cool}|\text{Yes}) \cdot P(\text{High}|\text{Yes}) \cdot P(\text{Strong}|\text{Yes}) = \frac{9}{14} \cdot \frac{2}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \approx 0.0053$$

For $Y = \text{No}$:

$$P(\text{No}) \cdot P(\text{Sunny}|\text{No}) \cdot P(\text{Cool}|\text{No}) \cdot P(\text{High}|\text{No}) \cdot P(\text{Strong}|\text{No}) = \frac{5}{14} \cdot \frac{3}{5} \cdot \frac{1}{5} \cdot \frac{4}{5} \cdot \frac{3}{5} \approx 0.0205$$

Decision:
Since

$$P(Y = \text{No} \mid x^{\text{new}}) > P(Y = \text{Yes} \mid x^{\text{new}}),$$

the predicted class is

$$Y^{\text{new}} = \text{No}.$$

Therefore, according to the Naïve Bayes model, it is **not a good day to play tennis**.

3.4 Generative techniques

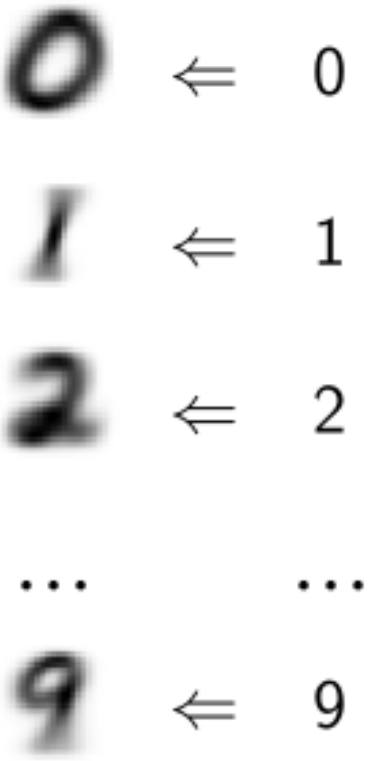
When we want to classify data (determine which category Y something belongs to given its features X), there is a fundamentally philosophical approaches, the **Generative Approach** that "learn how each class generates data", here is a sketch:

- Ask: "How does each class produce its characteristic data?"
- Model: $P(X|Y)$ (probability of features given category)
- Then use Bayes' Rule to reverse it and get $P(Y|X)$

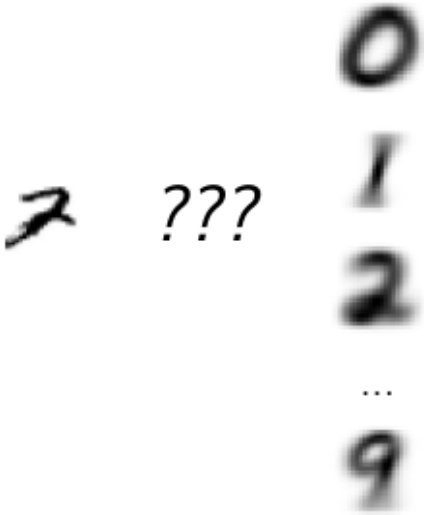
The term "Generative" come from the fact we're modeling the data generation process. We're essentially saying: "if I knew the category Y, I could generate/simulate typical data X from that category"

3.4.1 Big example: the visual intuition

We want to calssify images into categories $\{0, 1, \ldots, 9\}$ The generative approach says: "for each digit, learn waht imgs form that category typically look like. Then given a new img, see which category would most naturally produce such an img"



Okay, now we want to classify a new img:



Which of these distributions would most likely have generated this image?
Mathematically, for each category k:

$$\begin{aligned}\text{Score for category k} &= P(Y = k) \cdot P(X = \text{image} | Y = k) \\ &= \text{Prior} \cdot \text{Likelihood}\end{aligned}$$

You pick the category with the highest score. You're asking which generative mode (which category's distribution) best explains the observed data

Note:

The "score" is just the numerator of byes rule:

$$P(Y = k|X = \text{image}) = [P(Y = k) \cdot P(X = \text{image}|Y = k)]/P(X = \text{image})$$

Joint Distribution vs. Naïve Bayes

Ideally we'd want to model the Complete joint distribution:

$$P(X_1, X_2, \ldots, X_n|Y = y_i)$$

For MNIST, this would be the distribution of all 784 pixels (28×28 image) for images of each digit. This distribution would capture all the correlations between pixels - how pixel 1 relates to pixel 2, how groups of pixels form edges and curves, etc. Howver, modeling the full joint distribution for 784 dimensions is impossibly complex. We'd need:

- 2^{784} parameters just for binary pixels (more than atoms in the universe!)
- An astronomical amount of training data

So we play the card "Naïve Bayes" with 2500 atk and 1000 def:

$$P(X_1, \ldots, X_n \mid y = y_i) \approx \prod_j P(X_j \mid Y = y_i)$$

The assumption (effect) is: Given the category Y, all pixels are *independent*. This means we model each pixel separately:

$$P(X_1|Y = 0), P(X_2|Y = 0), ..., P(X_{784}|Y = 0)$$
$$P(X_1|Y = 1), P(X_2|Y = 1), ..., P(X_{784}|Y = 1)$$

This is computationally feasible, but we've lost all information about how pixels relate to each other!

3.4.2 Caution 1: The Zero Probability Problem

From a previous example if we have $P(Play = No|Outlook = Overcast)$ the result is $\theta_Overcast, No = 0$, This happened because in our training data, we never observed a "No" (don't play tennis) when the outlook was overcast

Note:

Remember the classification formula:

$$\text{Score} = \pi_k \cdot \prod_i \theta_{ijk}$$

If any single $\theta_{ijk} = 0$, the entire product becomes zero. So A single feature value you've never seen in training can completely eliminate a category from consideration, even if all other features strongly support it!

3.4.3 Caution 2: The Independence Assumption

Naïve Bayes assumes events are independent from each other (given Y). What if this is not the case?

The XOR Problem: A Fatal Limitation

Consider random binary images where pixels are either 0 or 1. We classify based on two pixels: p_1 and p_2.
Classification rule:

- **Category A:** if $p_1 = p_2$ (both same) — Images: $\{(0, 0), (1, 1)\}$
- **Category B:** if $p_1 \neq p_2$ (different) — Images: $\{(0, 1), (1, 0)\}$

This is an **XOR (exclusive OR)** relationship — a simple logical rule.
What Naïve Bayes learns:
For Category A (training: $(0,0)$ and $(1,1)$):

$$P(p_1 = 1 \mid A) = \frac{1}{2} \quad \text{[one out of two has } p_1 = 1\text{]}$$
$$P(p_2 = 1 \mid A) = \frac{1}{2} \quad \text{[one out of two has } p_2 = 1\text{]}$$

For Category B (training: $(0,1)$ and $(1,0)$):

$$P(p_1 = 1 \mid B) = \frac{1}{2} \quad \text{[one out of two has } p_1 = 1\text{]}$$
$$P(p_2 = 1 \mid B) = \frac{1}{2} \quad \text{[one out of two has } p_2 = 1\text{]}$$

Result: All probabilities are identical! For any test image (a, b):

$$\text{Score}_A = P(A) \cdot P(p_1 = a \mid A) \cdot P(p_2 = b \mid A) = 0.5 \cdot 0.5 \cdot 0.5 = 0.125$$
$$\text{Score}_B = P(B) \cdot P(p_1 = a \mid B) \cdot P(p_2 = b \mid B) = 0.5 \cdot 0.5 \cdot 0.5 = 0.125$$

Naïve Bayes cannot distinguish the categories! It achieves only 50% accuracy (random guessing) despite the trivially simple classification rule.
Why? The features p_1 and p_2 are **not independent** given the category — they're perfectly correlated:

- In Category A: if $p_1 = 0$ then $p_2 = 0$ (with certainty)
- In Category B: if $p_1 = 0$ then $p_2 = 1$ (with certainty)
- $P(p_2 = 1 \mid p_1 = 1, A) = 1 \neq P(p_2 = 1 \mid A) = 0.5$ — violates independence!

General lesson: Naïve Bayes cannot learn relationships between features. It only learns how common each individual feature value is within each class, not how features combine, interact, or correlate.