Set Theory, Counting, and Bayes' Theorem for Machine Learning

Basics of Set Theory: Union, Intersection, and Complement

Set theory is the mathematical foundation for probability. A **set** is a collection of distinct objects (called elements).

Notation

A set is written with curly braces:

```
A = {1, 2, 3, 4, 5}
B = {red, blue, green}
C = {x | x is even and x < 10} = {2, 4, 6, 8}
```

Element notation:

- $x \in A$ means "x is in set A"
- $x \notin A$ means "x is not in set A"

Cardinality (size of set):

- $|A| = 5$ (set A has 5 elements)

Union: Combining Sets (A ∪ B)

The **union** of two sets contains all elements in either set (or both).

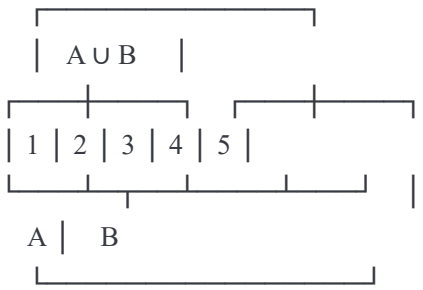
```
A ∪ B = {x | x ∈ A or x ∈ B}
```

Example:

```
A = {1, 2, 3}
B = {3, 4, 5}
A ∪ B = {1, 2, 3, 4, 5}
```

Note: 3 appears only once (sets have no duplicates).

Venn Diagram:



Probability interpretation:

$$P(A \cup B) = P(A \text{ or } B) = P(A) + P(B) - P(A \cap B)$$

Why subtract $P(A \cap B)$? Because elements in both sets are counted twice.

Intersection: Common Elements ($A \cap B$)

The **intersection** of two sets contains only elements in both sets.

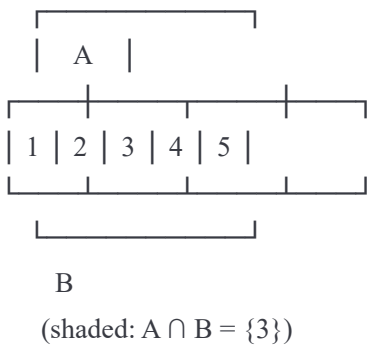
$$A \cap B = \{x \mid x \in A \text{ and } x \in B\}$$

Example:

$A = \{1, 2, 3\}$
$B = \{3, 4, 5\}$
$A \cap B = \{3\}$

Only 3 is in both sets.

Venn Diagram:



(shaded: $A \cap B = \{3\}$)

Probability interpretation:

$$P(A \cap B) = P(A \text{ and } B) = P(A) \times P(B|A)$$

For independent events: $P(A \cap B) = P(A) \times P(B)$

Special case—Disjoint Sets ($A \cap B = \emptyset$):

- No common elements
- $P(A \cap B) = 0$

Complement: Everything Else (A^c)

The **complement** of a set contains all elements NOT in that set.

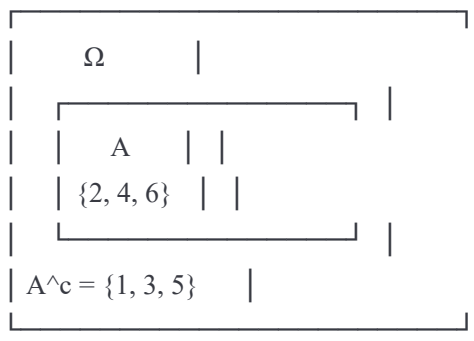
$$A^c = \{x \mid x \notin A\}$$

(Complement is relative to the universal set Ω—all possible elements)

Example:

Ω = {1, 2, 3, 4, 5, 6} (all outcomes of rolling a die)
A = {2, 4, 6} (even numbers)
A^c = {1, 3, 5} (odd numbers)

Venn Diagram:



Probability interpretation:

$$P(A^c) = 1 - P(A)$$

This is the **complement rule**—fundamental in probability.

De Morgan's Laws

These laws describe relationships between unions, intersections, and complements:

$(A \cup B)^c = A^c \cap B^c$ (not (A or B) = not A and not B)
$(A \cap B)^c = A^c \cup B^c$ (not (A and B) = not A or not B)

Intuition:

- "I don't like red or blue" = "I don't like red AND I don't like blue"
- "I don't like coffee and tea" = "I don't like coffee OR I don't like tea"

Applications in ML

1. Feature selection:

- Set A = features relevant to task
- Set B = features computationally efficient
- $A \cap B$ = features we actually use (both useful and efficient)

2. Data partitioning:

- Training set $\cup$ Validation set $\cup$ Test set = entire dataset
- These are disjoint ($A \cap B = \emptyset$)

3. Classification:

- Set A = positive class
- Set B = negative class
- A^c = negative class (complement of positive)

4. Event detection:

- A = email is spam
- B = email matches pattern
- $A \cap B$ = spam with pattern (both conditions)

Counting Techniques: Permutations and Combinations

When we can't list all outcomes, we count them. Two fundamental techniques: **permutations** and **combinations**.

Factorial Notation

Factorial n! is the product of all positive integers up to n:

$$n! = n \times (n-1) \times (n-2) \times \ldots \times 2 \times 1$$

Examples:

$0! = 1$ (by definition)
$1! = 1$
$3! = 3 \times 2 \times 1 = 6$
$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$
$10! = 3{,}628{,}800$

Growth: Factorials grow explosively (important for computational complexity).

Permutations: Order Matters

A **permutation** is an arrangement of objects where **order matters**.

Formula for permutations of n objects taken r at a time:

$$P(n,r) = n! / (n-r)!$$

Or equivalently:

$$P(n,r) = n \times (n-1) \times (n-2) \times \ldots \times (n-r+1)$$

Intuition: For the first position, choose from n objects. For the second, choose from n-1 remaining. Continue r times.

Example: Arranging 3 runners from 5 competitors

How many ways to award 1st, 2nd, 3rd place?

$$P(5,3) = 5! / (5-3)! = 5! / 2! = (5 \times 4 \times 3 \times 2 \times 1) / (2 \times 1) = 120 / 2 = 60$$

Or directly: 5 choices for 1st place × 4 for 2nd × 3 for 3rd = 60 ways.

Example: Password with 4 distinct digits

Choose 4 distinct digits from 0-9, order matters:

$$P(10,4) = 10! / 6! = 10 \times 9 \times 8 \times 7 = 5{,}040$$

Combinations: Order Doesn't Matter

A **combination** is a selection of objects where **order doesn't matter**.

Formula for combinations of n objects taken r at a time:

$$C(n,r) = n! / (r! \times (n-r)!)$$

Or equivalently:

$$C(n,r) = P(n,r) / r! = [n \times (n-1) \times \ldots \times (n-r+1)] / r!$$

The division by r! removes overcounting due to order.

Example: Selecting 3 runners from 5 for a relay team (order doesn't matter)

$$C(5,3) = 5! / (3! \times 2!) = (5 \times 4 \times 3 \times 2 \times 1) / ((3 \times 2 \times 1) \times (2 \times 1))$$
$$= 120 / (6 \times 2) = 120 / 12 = 10$$

Compare to permutations: P(5,3) = 60. The ratio is 60/10 = 6 = 3!, which is the number of ways to arrange 3 people.

Example: Selecting 2 items from 4 in a menu

$$C(4,2) = 4! / (2! \times 2!) = (4 \times 3) / (2 \times 1) = 6$$

The 6 combinations are: {1,2}, {1,3}, {1,4}, {2,3}, {2,4}, {3,4}.

Permutations vs Combinations Summary

Aspect	Permutations	Combinations
Order matters?	YES	NO
Formula	n! / (n-r)!	n! / (r!(n-r)!)
Example	Arranging books on shelf	Selecting books for a bag
Result	More outcomes	Fewer outcomes
Relation	P(n,r) = C(n,r) × r!	C(n,r) = P(n,r) / r!

Applications in ML

1. Feature selection:

- Select k features from d total features
- Number of possible combinations: C(d,k)
- If d=100 and k=10: C(100,10) ≈ 17 trillion combinations
- This is why exhaustive search is impractical

2. Hyperparameter tuning:

- Grid search over parameter combinations
- If 3 parameters with 5, 4, 6 values each: 5 × 4 × 6 = 120 combinations

3. Cross-validation splits:

- k-fold cross-validation partitions data into k groups
- Number of ways to select validation fold: C(n, n/k)

4. Ensemble diversity:

- Number of ways to sample with replacement: allows diversity in ensemble models

5. Probability calculations:

- Many probability problems require counting outcomes
- Example: "Probability of getting 3 heads in 5 flips" uses C(5,3)

Conditional Probability and Bayes' Theorem

Conditional Probability: The Foundation

Conditional probability is the probability of an event given that another event has already occurred.

Mathematical definition:

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

Read as: "Probability of A given B"

Intuition: If we know B happened, what's the probability that A also happened?

We restrict our focus to outcomes where B is true, then ask what fraction also have A.

Example: Drawing cards

Standard deck: 52 cards, 13 spades

Event A: Drawing a spade
Event B: Drawing a face card (J, Q, K)

$P(A \mid B) = P(A \cap B) / P(B)$
$= P(\text{drawing a spade AND face card}) / P(\text{drawing face card})$
$= (3/52) / (12/52)$
$= 3/12$
$= 1/4$

Without condition: $P(\text{spade}) = 13/52 = 1/4$
With condition: $P(\text{spade} \mid \text{face card}) = 1/4$ (same!)

This shows spades and face cards are **independent**—knowing one doesn't change probability of the other.

Independence

Events A and B are **independent** if:

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

Equivalently:

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

Knowing B occurred doesn't change the probability of A.

Example: Independent

Event A: Coin 1 shows heads
Event B: Coin 2 shows heads
$P(A \mid B) = P(A) = 0.5$ (independent)

Example: Dependent

Event A: It rained today
Event B: Grass is wet
$P(A \mid B) > P(A)$ (knowing grass is wet increases probability of rain)

Bayes' Theorem: The Foundation of ML

Bayes' Theorem relates conditional probabilities in opposite directions:

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

Or more generally:

$$P(A \mid B) = P(B \mid A) \times P(A) / [P(B \mid A) \times P(A) + P(B \mid A^c) \times P(A^c)]$$

The denominator is the **law of total probability**.

Components of Bayes' Theorem

Bayes' Theorem has four key components:

1. Posterior $P(A \mid B)$: What we want

- Probability of hypothesis A given observed data B
- "Updated belief after seeing evidence"

2. Likelihood $P(B \mid A)$: What the data shows

- Probability of observing data B if hypothesis A is true
- "How well does hypothesis explain data?"

3. Prior $P(A)$: Initial belief

- Probability of hypothesis before seeing data
- "Background knowledge or assumption"

4. Evidence $P(B)$: Normalizing constant

- Total probability of observing data (under all hypotheses)
- Ensures probabilities sum to 1

The formula:

$$\text{Posterior} = \text{Likelihood} \times \text{Prior} / \text{Evidence}$$

Simple Example: Disease Testing

You take a test for a rare disease:

- Disease prevalence: P(disease) = 0.01 (1% of population)
- Test accuracy: P(positive | disease) = 0.99 (99% of sick people test positive)
- False positive rate: P(positive | no disease) = 0.05 (5% of healthy people test positive)

Question: You tested positive. What's the probability you actually have the disease?

Using Bayes' Theorem:

$$P(\text{disease} \mid \text{positive}) = P(\text{positive} \mid \text{disease}) \times P(\text{disease}) / P(\text{positive})$$

Step 1: Calculate P(positive) (law of total probability)

$$\begin{aligned} P(\text{positive}) &= P(\text{positive} \mid \text{disease}) \times P(\text{disease}) + P(\text{positive} \mid \text{no disease}) \times P(\text{no disease}) \\ &= 0.99 \times 0.01 + 0.05 \times 0.99 \\ &= 0.0099 + 0.0495 \\ &= 0.0594 \end{aligned}$$

Step 2: Apply Bayes' Theorem

$$\begin{aligned} P(\text{disease} \mid \text{positive}) &= (0.99 \times 0.01) / 0.0594 \\ &= 0.0099 / 0.0594 \\ &\approx 0.167 \text{ or } 16.7\% \end{aligned}$$

Shocking result: Even with a 99% accurate test, a positive result only means 16.7% chance of disease!

Why? The disease is very rare (base rate = 1%). Even though the test is 99% accurate, false positives from the 99% healthy population outnumber true positives from the 1% sick population.

This is the **base rate fallacy**—ignoring how rare the condition is.

Intuitive Explanation

Imagine 10,000 people:

100 have disease
- 99 test positive (99% sensitivity)
- 1 tests negative

9,900 don't have disease
- 495 test positive (5% false positive rate)
- 9,405 test negative

Total positive tests: 99 + 495 = 594
Of those, actually sick: 99
Probability: 99/594 ≈ 16.7%

Most positive tests are false positives!

Extending to Multiple Hypotheses

When comparing multiple hypotheses H_1, H_2, ..., H_n:

$$P(H_i \mid data) = P(data \mid H_i) \times P(H_i) / \Sigma_j P(data \mid H_j) \times P(H_j)$$

The denominators sums over all hypotheses.

Example: Spam classification

H_1: Email is spam
H_2: Email is ham (not spam)

$$P(spam \mid email) = P(email \mid spam) \times P(spam) / [P(email \mid spam) \times P(spam) + P(email \mid ham) \times P(ham)]$$

If email contains word "free":

- Likelihood: P("free" | spam) = 0.8 (80% of spam contain "free")
- Likelihood: P("free" | ham) = 0.1 (10% of legitimate email contain "free")
- Prior: P(spam) = 0.3 (30% of all emails are spam)

$$P(spam \mid \text{"free"}) = (0.8 \times 0.3) / (0.8 \times 0.3 + 0.1 \times 0.7)$$
$$= 0.24 / (0.24 + 0.07)$$
$$= 0.24 / 0.31$$
$$\approx 0.77 \text{ or } 77\%$$

Seeing "free" increases belief that it's spam from 30% to 77%.

Applications of Conditional Probability in ML

1. Classification Models

Every classification model computes:

$$P(\text{class} \mid \text{features}) = P(\text{features} \mid \text{class}) \times P(\text{class}) / P(\text{features})$$

This is directly Bayes' Theorem!

Example: Logistic Regression

$$P(y=1 \mid x) = \text{sigmoid}(w \cdot x + b)$$

The sigmoid function converts linear combination to probability that matches Bayesian reasoning.

Naive Bayes Classifier: Assumes features are conditionally independent given the class:

$$P(\text{class} \mid x_1, x_2, ..., x_n) \propto P(\text{class}) \times P(x_1|\text{class}) \times P(x_2|\text{class}) \times ... \times P(x_n|\text{class})$$

Why naive? Real features are not independent, but the assumption simplifies computation and works surprisingly well.

2. Spam Detection

Training data:

- 70% legitimate emails
- 30% spam emails

Observed features in spam:

- Contains "click here": 60% of spam
- Contains "click here": 5% of legitimate

New email has "click here":

$$\begin{aligned} P(\text{spam} \mid \text{"click here"}) &= P(\text{"click here"} \mid \text{spam}) \times P(\text{spam}) / P(\text{"click here"}) \\ &= 0.6 \times 0.3 / [0.6 \times 0.3 + 0.05 \times 0.7] \\ &= 0.18 / [0.18 + 0.035] \\ &= 0.18 / 0.215 \\ &\approx 0.84 \text{ or } 84\% \end{aligned}$$

Email with "click here" is 84% likely to be spam.

3. Medical Diagnosis

Doctors combine symptoms (evidence) with prior beliefs:

```
P(disease | symptoms) = P(symptoms | disease) × P(disease) / P(symptoms)
```

Example: Patient has fever (symptom)

```
P(flu | fever) = P(fever | flu) × P(flu) / P(fever)
               = 0.9 × 0.05 / [0.9 × 0.05 + 0.3 × 0.95]
               = 0.045 / [0.045 + 0.285]
               = 0.045 / 0.33
               ≈ 0.136 or 13.6%
```

Even though fever is common in flu (90%), it's much more common in non-flu (30% of healthy people have fever). So fever alone doesn't strongly indicate flu.

But if fever is combined with other symptoms:

```
P(flu | fever AND cough AND sore throat) >> 13.6%
```

Multiple pieces of evidence (symptoms) strongly update belief.

4. Recommendation Systems

Personalized recommendations use conditional probability:

```
P(user likes item | user history) = ?
```

Collaborative filtering:

- Find similar users
- Recommend items they liked
- Uses conditional probability: P(rating | similar user history)

Content-based:

- Check if item matches user preferences
- P(user likes | item features)

5. Natural Language Processing

Language models compute:

```
P(word | previous words) = ?
```

Example: Predicting next word

```
Given: "The weather is ___"
P(sunny | weather is) = 0.4
P(rainy | weather is) = 0.3
P(cloudy | weather is) = 0.2
P(snowing | weather is) = 0.1

Predict: "sunny" (highest probability)
```

6. A/B Testing and Decision Making

When comparing variants A and B:

```
P(A is better | observed data) = ?
```

Use Bayesian approach:

- Prior: P(A is better) = 0.5 (initially equal)
- Likelihood: P(data | A is better) based on observed results
- Posterior: P(A is better | data) = updated belief

As more data comes in, posterior probability increases or decreases.

7. Anomaly Detection

Flag unusual patterns:

```
P(normal | observation) = ?
```

If P(normal) is very low, flag as anomaly.

Example: Credit card fraud

```
P(normal transaction | high amount, unusual location, fast sequence) = low
→ Flag as potential fraud
```

8. Bayesian Optimization (Hyperparameter Tuning)

Iteratively test hyperparameters:

```
P(good performance | hyperparameters) = ?
```

Use Bayesian optimization to balance:

- Exploring unknown regions (might find better values)
- Exploiting promising regions (refine good values)

Based on conditional probability of performance given hyperparameters.

9. Prior and Posterior Updates

Bayesian methods update beliefs as data arrives:

Before seeing data (prior):

```
P(model is correct) = 0.5 (uncertain)
```

After seeing supporting data (posterior):

```
P(model is correct | data) = 0.9 (more confident)
```

With more confirming data:

```
P(model is correct | more data) = 0.99 (very confident)
```

This is how learning works: updating beliefs with evidence.

Bayes' Theorem in Real Practice: A Complete Example

Scenario: Email Spam Filter

You're building a spam filter using Bayesian methods.

Training data analysis:

Total emails: 1000
Spam: 300 (P(spam) = 0.3)
Legitimate: 700 (P(legitimate) = 0.7)

Feature: Contains "Buy Now"
In spam: 200/300 = 0.67
In legitimate: 20/700 = 0.03

P(spam) = 0.3
P("Buy Now" | spam) = 0.67
P("Buy Now" | legitimate) = 0.03

New email arrives with "Buy Now":

P(spam | "Buy Now") = P("Buy Now" | spam) × P(spam) / P("Buy Now")

P("Buy Now") = P("Buy Now" | spam) × P(spam) + P("Buy Now" | legitimate) × P(legitimate)
= 0.67 × 0.3 + 0.03 × 0.7
= 0.201 + 0.021
= 0.222

P(spam | "Buy Now") = (0.67 × 0.3) / 0.222
= 0.201 / 0.222
≈ 0.91 or 91%

Action: Classify as spam with 91% confidence.

With multiple features:

Email contains: "Buy Now", "Free shipping", "Click here"

P(spam | all three) = P(all three | spam) × P(spam) / P(all three)

Assuming independence:

= P("Buy Now" | spam) × P("Free" | spam) × P("Click" | spam) × P(spam) / P(all three)

Each feature increases confidence in spam classification.

Summary: From Sets to Bayesian Reasoning

Set theory provides the mathematical language for defining events and relationships between them.

Counting techniques (permutations, combinations) calculate how many ways outcomes can occur—essential for probability.

Conditional probability captures how evidence changes beliefs: P(A | B) asks "what's the probability of A if B occurred?"

Bayes' Theorem provides the formula: $P(A \mid B) = P(B \mid A) \times P(A) / P(B)$

In machine learning, Bayes' Theorem is fundamental to:

- Classification (computing P(class | features))
- Inference (updating beliefs with data)
- Decision-making (choosing actions that maximize expected utility)
- Optimization (finding best parameters)

Understanding these concepts—from sets to Bayes—gives you the theoretical foundation to understand why ML algorithms work and how to apply them correctly. Bayesian thinking is how we learn from data: start with prior beliefs, observe evidence, update to posterior beliefs.