Multi-Instance Learning

What Is Multi-Instance Learning?

- Each example = a **bag of instances** instead of a single instance.
- Only the **bag** has a label — individual instances do **not**.
- First introduced in drug activity prediction.
- Useful when instance-level labels are unknown or ambiguous.

Why Multi-Instance Learning?

- Used when:
  - Objects have multiple components/segments.
  - Only overall label is known.
  - Some components may be irrelevant.
- Applications:
  - Drug discovery
  - Image classification (regions)
  - Document classification (paragraphs/sentences)
  - Medical imaging (multi-slice scans)

Classical MIL Assumption

- A bag is **positive** if **at least one** instance is positive.
- A bag is **negative** if **all** instances are negative.

Bag Representation

- Example:

```
Bag 1: {x1, x2, x3} → Positive
Bag 2: {x4, x5}     → Negative
```

- Instances vary in number per bag.
- Bag-level label given; instance labels unknown.

Two Main Approaches

1. **Aggregate the Input** $\rightarrow$ summarize each bag into one instance.
2. **Aggregate the Output** $\rightarrow$ classify each instance, then combine.

Approach 1: Aggregating the Input

- Convert a bag to one summarized instance using:
  - Minimum
  - Maximum
  - Mean
  - Mode
  - Other statistics
- Example:

```
Bag of 20 instances → summarized using min/max per attr
```

Why Aggregate Input?

- Converts MIL into standard supervised learning.
- Allows use of algorithms like SVMs and logistic regression.
- Shown to perform surprisingly well on early MIL datasets.

Limitations of Input Aggregation

- Summary statistics may lose key instance-level info.
- Not effective when only **one special instance** determines the bag label.

Approach 2: Aggregating the Output

- Steps:
  1. Assign bag's label to each instance during training.
  2. Train classifier on all instances.
  3. At test time: classify each instance $\rightarrow$ aggregate predictions.

Output Aggregation Methods

- Majority voting.
- “At least one positive” rule.
- Averaging probabilities.
- Weighted voting.

Example of Output Aggregation

- Training:

```
Bag A (Positive): x1(+), x2(+), x3(+)
Bag B (Negative): x4(-), x5(-)
```

- Prediction:
  - Classify each instance in a test bag.
  - Combine via vote or probability.

Handling Unequal Bag Sizes

- To avoid larger bags dominating:
    - Give each instance weight $= 1$ / (# of instances in its bag)
- Ensures each **bag** contributes equally.

Pros & Cons of Output Aggregation

- **Pros:**
  - Preserves instance-level details.
  - Leverages standard ML models.
  - Flexible and extensible.
- **Cons:**
  - Requires careful combination rules.
  - Bag prediction sensitive to noisy instances.

Why Standard Learning Fails

- Instances in a bag may vary widely.
- Only a few instances may determine the label.
- Standard learners assume **instance labels are known**, which is false.

Drug Activity Example

- Molecule = bag.
- Conformations = instances.
- Molecule active $\rightarrow$ at least one conformation binds well.
- Only bag-level label known $\rightarrow$ perfect use case for MIL.

MIL in Modern Settings

- MIL now used in:
  - Weakly-supervised deep learning.
  - MIL pooling layers.
  - Attention-based MIL neural networks.
  - Multi-instance SVMs.

Strengths of MIL

- Handles ambiguity in instance labels.
- Works for complex, structured objects.
- Often compatible with standard learners.

Limitations of MIL

- MIL assumption may not hold universally.
- Aggregations may oversimplify.
- High computational cost for large bags.
- Hard to interpret instance contributions.

Summary

- MIL treats each example as a **bag of instances**.
- Two solutions:
  - Aggregate input (summaries).
  - Aggregate output (combine predictions).
- Useful for images, text, molecules, medical scans.