Applied ML

Hierarchical Classification & Open Problems in Applied ML

09-07-2024 @ LMU

Outline

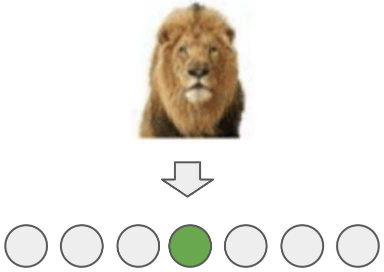
- Hierarchical Classification
  - Metrics
  - Approaches

- Some open problems in applied ML
  - Learning on partially labelled trees
  - Classifying sets

A short primer: Multi-class vs. multi-label classification

Multi-class

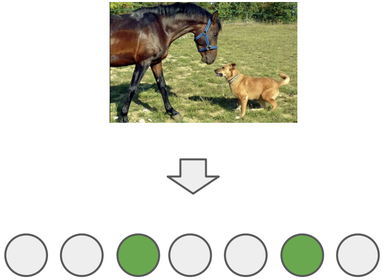
Classify an instance as **one of** a fixed set of classes.



- Categorical losses
- Classification metrics

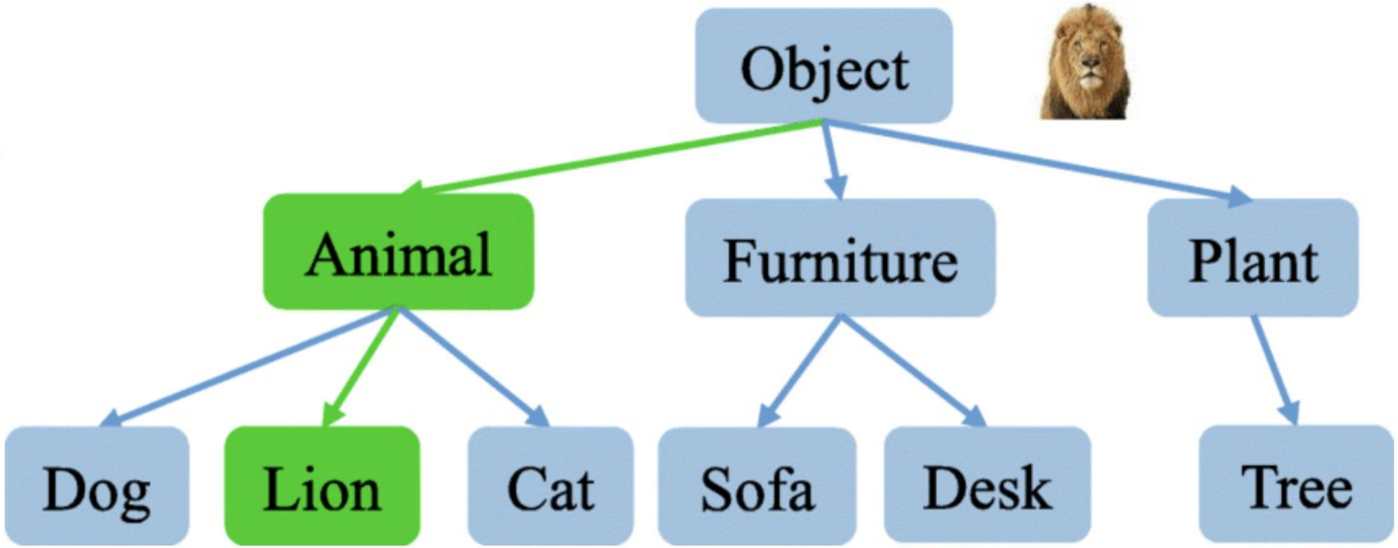
Multi-label

Classify an instance as **any of** a fixed set of classes



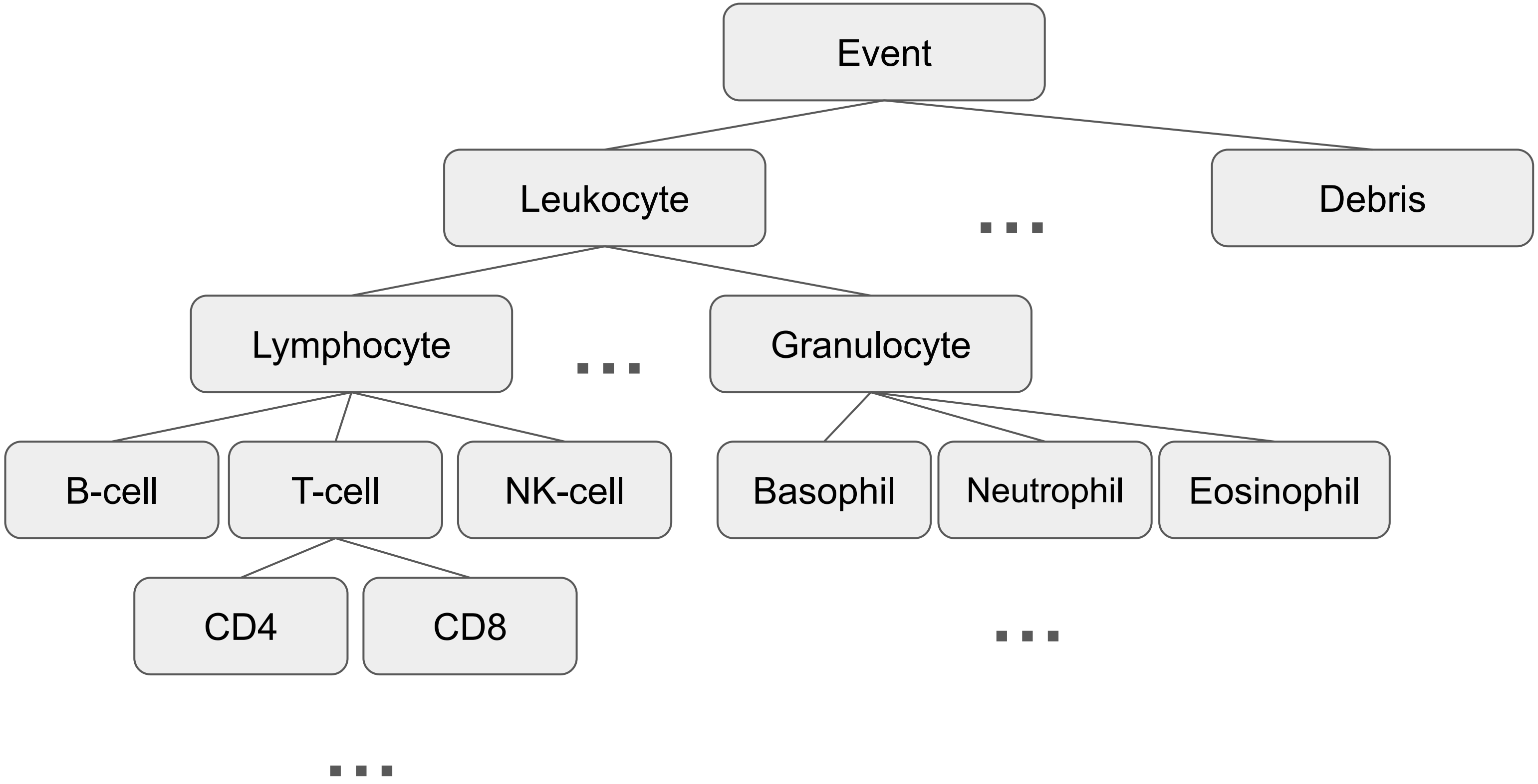
- Binary losses per class
- Multi-label metrics

Hierarchical classification



Zheng et al., 2021 Cost-sensitive hierarchical classification via multi-scale information entropy for data with an imbalanced distribution

Hierarchical Classification in Biology



Hierarchical Classification

- Labels follow a taxonomy:
  - ‘Coarse’ labels in higher levels of the tree
  - ‘Fine-grained’ labels at the leaf nodes

- Predictions can be done at any level of the tree.

- Real problems often have 1000s of labels and few examples per label.

- Errors are not equally important:
  - *Confusing a ‘dog’ and a ‘wolf’ can be less bad than confusing ‘cat’ and ‘car’.*

Notation

Our goal is to learn a classifier **f(x)**: **X** -> **Y** that classifies individual samples x ∈ **X** into labels y ∈ **Y**.
In a hierarchical setting, denote with **Y** the set of all nodes.
In addition, we denote with **Y*** ⊂ **Y** the set of all leaf nodes.

- Each node y can have **parent** pa(y) and **child** nodes ch(y).
- Denote with **A**(y) the set of all **ancestors** of y and y
  {y, pa(y), pa(pa(y)), ...}

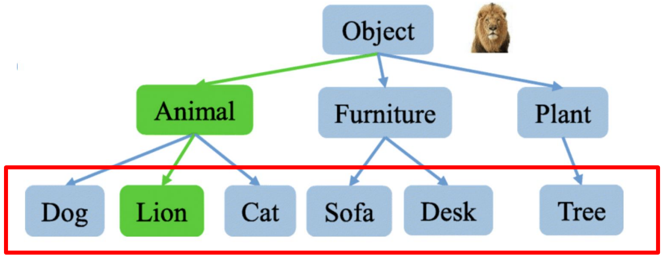
We are now looking for an inducer ***I***, that produces **f** given a set of training data pairs **(x,y)**.

Metrics for hierarchical classification

Metrics for Hierarchical Classification

Use **multi-class** performance metrics and classify leaf nodes **Y*.**

- Allows using widely understood metrics: Accuracy, F1, AUC, …
- Only classify leaf nodes
- Disregards tree structure



Use **multi-label** performance metrics on flattened tree

- ‘flatten’ the entire tree, encode as {‘animal’, ‘lion’}

Hornung, 2023: Evaluating machine learning models in non-standard settings: An overview and new findings https://arxiv.org/pdf/2310.15108

Silla, C.N., Freitas, A.A. A survey of hierarchical classification across different application domains. *Data Min Knowl Disc* **22**, 31–72 (2011). https://doi.org/10.1007/s10618-010-0175-9

Cost-sensitive learning

Cost-sensitive metrics

- Define costs C_{ij} for miss-classifying y_i as y_j.
- Could encode loss for predicting parent node
  instead of leaf node.
- Requires good judgement as to the real 'cost' of miss-classifications.

$\hat{y}$ \ y	1	2	3	4	5
1	0	1	1	1	1
2	1	0	1	1	1
3	3	1	0	2	1
4	1	1	1	0	1
5	1	2	1	1	0

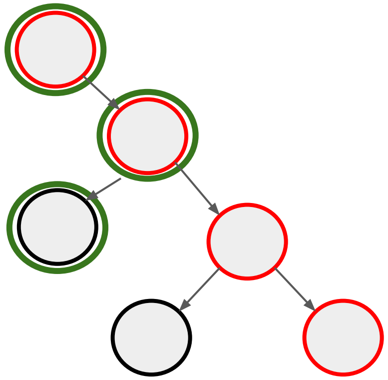
Metrics for Hierarchical Classification

Hierarchical metrics encode the tree structure:

- Measure overlap between predicted and actual path.

  Hierarchical **Recall, Precision, F1 (micro)**

  re: mean($|A(y) \cap A(\hat{y})| / |A(y)|$)
  pr: mean($|A(y) \cap A(\hat{y})| / |A(\hat{y})|$)
  *F1: 2 * re * pr / (re + pr)*



recall = 2/3
prec = 2/4
f1 = ~0.57

Hornung, 2023: Evaluating machine learning models in non-standard settings: An overview and new findings https://arxiv.org/pdf/2310.15108

Silla, C.N., Freitas, A.A. A survey of hierarchical classification across different application domains. *Data Min Knowl Disc* **22**, 31–72 (2011). https://doi.org/10.1007/s10618-010-0175-9

Hierarchical classification approaches

Flat classification

- Ignore tree structure

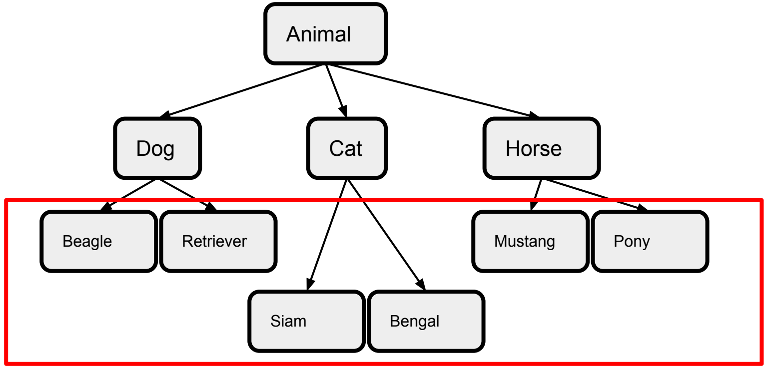
- Predict only leaf nodes

This converts the problem to a standard multiclass-classification problem:

$$f(x): \mathbf{X} \to \mathbf{Y}^*$$

Problems:

- Are all errors equally bad?
- What happens when we do not have labels until every leaf?



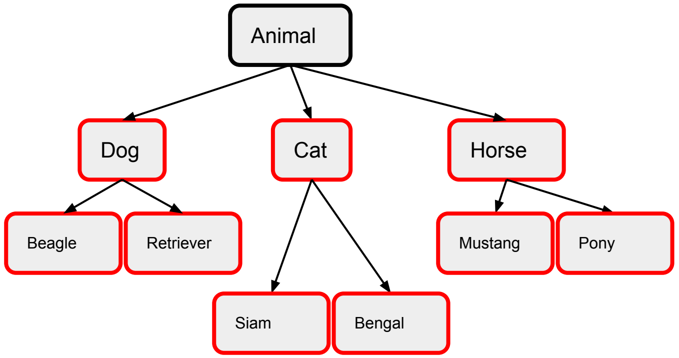
C.N. Silla & A.A. Freitas, [A survey of hierarchical classification across different application domains] (2011),

Classifier Cascades I

Split into several ‘local’ tasks at each level, train a classifier at each node.

Example:
Fit a binary classifier at each node.
If $P(y_k) > thr_k$: Assign y_k.

Problem:
- Smaller dataset with increasing depth.
- Complex models and prediction logic.
- Need to set ‘k’ thresholds
- Can yield predictions that violate tree structure



C.N. Silla & A.A. Freitas, A survey of hierarchical classification across different application domains (2011)

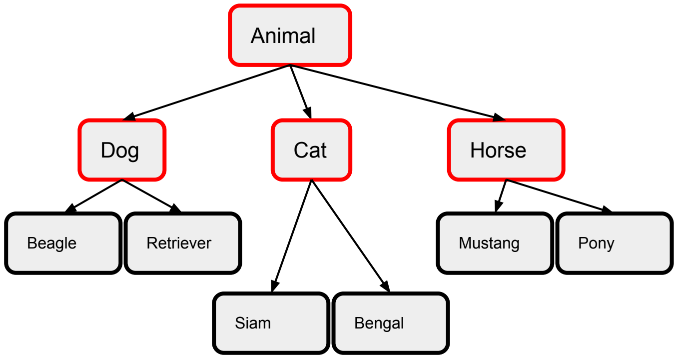
Classifier Cascades II

Split into several 'local' tasks.
Train a classifier at each 'level'.

- Train a multiclass classifier at each level of the tree.
- Train a multiclass model for each parent node.

Problem:

- Can produce predictions inconsistent with tree structure.
- Per level approach is not well defined if the tree is 'uneven'.

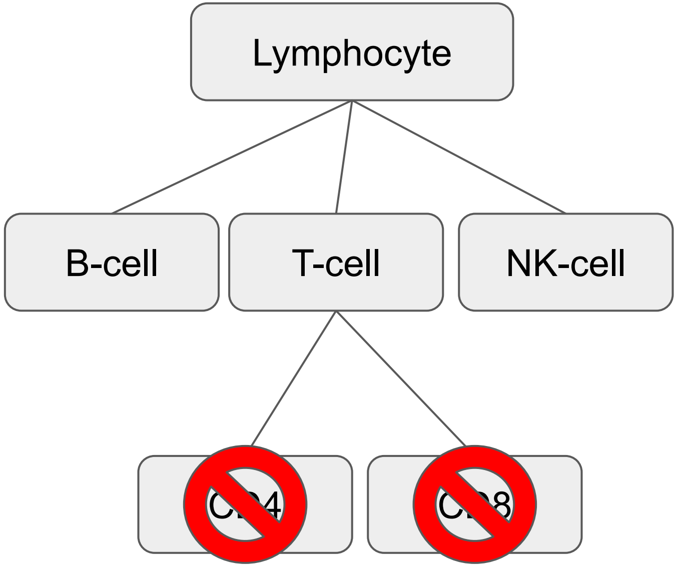


C.N. Silla & A.A. Freitas, [A survey of hierarchical classification across different application domains]() (2011)

Learning to abstain

If the model is not sure between elements in the leaf nodes, stop predicting and return parent node.

if P(y) < *thr* : *assign* pa(y)



Problem:

- How to calibrate the decision wrt. the global model?

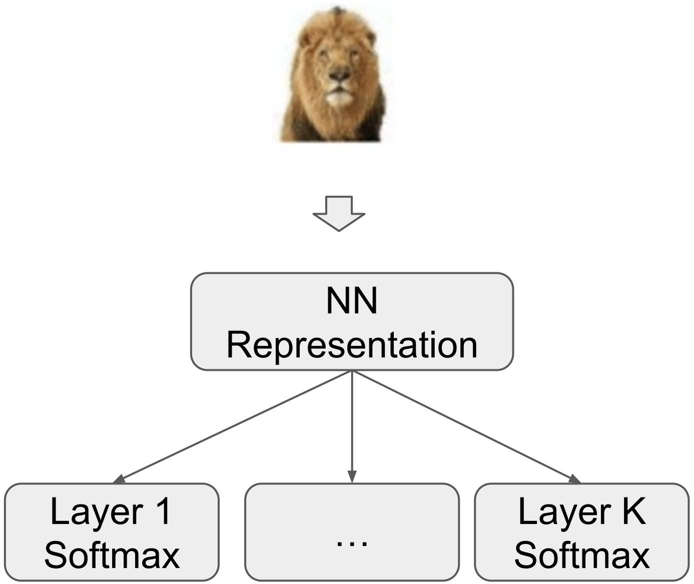
Deep Learning

Hierarchical neural networks tackle the problem via a softmax prediction at each layer of the tree.

- Extract a representation via NN (FFN, CNN, RNN, …)
  Predict softmax for each layer in the tree
- Enforce consistency via added 'consistency loss' in each layer *k*:

  $$L(\theta) = \sum_k \text{Cross Entropy}_k + \sum_k \text{Consistency}_k$$

- Consistency_k: $I(\hat{y}_{k-1} = pa(\hat{y}_k)) * I(y_k = \hat{y}_k)$



c.f. Gao et al., Deep Hierarchical Classification for Category Prediction in E-commerce System (2020

Software for Hierarchical Learning

- R
  - **hierclass**: Algorithms and metrics for hierarchical classification based on mlr3. https://github.com/RomanHornung/hierclass
  - **tabnet**: The tabnet package seems to include options for hierarchical classification.

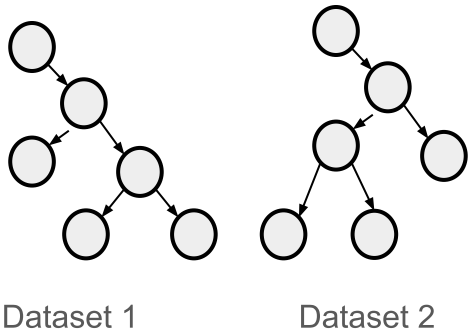
- Python
  - **hiclass:** Scikit-learn compatible wrapper for hierarchical classification https://github.com/scikit-learn-contrib/hiclass
  - **sklearn-hierarchical-classification**: Another scikit-learn wrapper https://github.com/globality-corp/sklearn-hierarchical-classification

Open Problems in Applied ML

Open Problem:
Classification on partially labelled trees

- Consider settings where we have ground truth labels at different specificity.
- Lack of specificity might e.g. stem from availability of required measurements.
- Losses need to reflect what's possible to predict.

  There might be a MSc thesis in that direction!



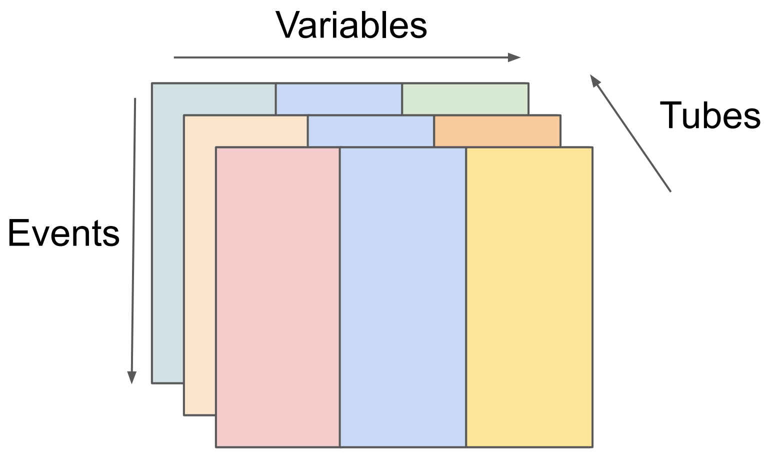
Open Problem: Classifying sets

Example: Flow cytometry:

- Patient blood/bone marrow is split into 3-10 tubes.

- For each tube, we record ~5e4 measurements (cells) of ~10 variables

Goal: Predict a label for each event



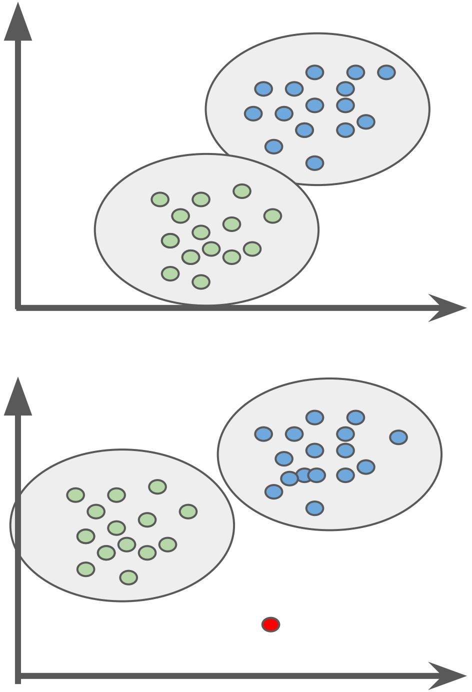
Patient sample

Example: Flow cytometry contd.

- 'Context' is important: Label is understood in context / based on geometry relative to other cells.

- Labels are only assigned if sufficient similar cells exist (= cells form a population)

- Instead of classifying a single sample, we need to classify an entire dataset.



Takeaways

- A common sentiment in applied ML is ‘xgboost’ is all you need
  - This holds only in limited contexts
  - In practices often ‘oversimplify’ problems so they fit the tools they know.

~~XGBoost~~
~~Attention~~ Is All You Need

- Measuring the correct thing is one of the most important skills in ML: Metrics need to encode the real structure of the problem so they provide us with relevant information.