Multi-label classification

Diane Lingrand
diane.lingrand@univ-cotedazur.fr

UNIVERSITÉ
CÔTE D'AZUR

Master Data Science M1

2024

Outline

Introduction

- Multi-class classification
  - classes are mutually exclusive
    - eg : a picture of a cat
  - one data belongs to only one class
- Multi-label classification
  - one data may belongs to several classes
    - eg : a picture of a cat, in an apartment, playing with a dog
- Hierarchical multi-label classification
  - one data belongs to several classes that belongs to a hierarchy : tree or DAG
    - eg : a picture of a cat which is a feline, which is a mammal, which is an animal ...

Outline

Multi-label classification

- transformation of the classification problem
  - into binary classification
    - binary relevance
  - into multi-class classification
    - classification chains
    - label powerset
- adaptation of the algorithm
  - ML-kNN for kNN
  - Clare for trees
  - Adaboost.MH and Adaboost.MR for adaboost
  - BP-MLL : Back-Propagation for Multi-Label Learning (neural network)

A simple example

for the 3 next algorithms :

data	**labels**
x_0	l_0
x_1	l_1
x_2	l_0, l_1
x_3	l_2
x_4	l_0, l_2
x_5	l_0, l_1

Binary Relevance

- ensemble of single-label binary classifiers
  - union of predicted classes
  - don't work well when labels are dependent
- similar to 'ovr' when multiple classifiers answers 'yes'
- implementation in `BinaryRelevance` `skmultilearn.problem_transform`
- Example :
  - 3 classes l_0, l_1 and l_2
  - 3 classifiers : c_i answering 1 if label l_i, else 0
  - for data x_2, if c_0 and c_1 answer 1, then x_2 has 2 labels : l_0 and l_1

data	**labels**	c_0	c_1	c_2
x_0	l_0	1	0	0
x_1	l_1	0	1	0
x_2	l_0, l_1	1	1	0
x_3	l_2	0	0	1
x_4	l_0, l_2	1	0	1
x_5	l_0, l_1	1	1	0

Classification Chains

[Read etal, ECML PKDD 2009]

- chains of binary classifiers c_i, $0 \leq i \leq n$ where n is the number of classes
- c_i uses the predictions of all c_j, $j < i$ by adding priors to features
  - can take labels correlation into account
- order of chain is important
  - try all orders ? ensemble of different chains with different orders ?
- implementation
  - in `ClassifierChain` from `sklearn.multioutput`
  - in `ClassifierChain` from `skmultilearn.problem_transform`
- Example :
  - c_0 is trained with $\mathbf{x} = [x_0, x_1, x_2, x_3, x_4, x_5]$ and $\mathbf{y} = [1, 0, 1, 0, 1, 1]$
  - c_1 is trained with $\mathbf{x} = [(x_0, 1), (x_1, 0), (x_2, 1), (x_3, 0), (x_4, 1), (x_5, 1)]$ and $\mathbf{y} = [0, 1, 1, 0, 0, 1]$
  - c_2 is trained with $\mathbf{x} = [(x_0, 1, 0), (x_1, 0, 1), (x_2, 1, 1), (x_3, 0, 0), (x_4, 1, 0), (x_5, 1, 1)]$ and $\mathbf{y} = [0, 0, 0, 1, 1, 0]$

Label Powerset

- transform the problem into a multi-class classification problem
- cannot consider labels correlation
- a class in the new problem is a combination of classes from the original set of classes :
  - in the worst case : $2^{|c|}$ classifiers
- implementation in `LabelPowerset` from `skmultilearn.problem_transform`
- Example :

data	multi labels	powerset labels
x_0	l_0	λ_0
x_1	l_1	λ_1
x_2	l_0, l_1	λ_2
x_3	l_2	λ_3
x_4	l_0, l_2	λ_4
x_5	l_0, l_1	λ_2

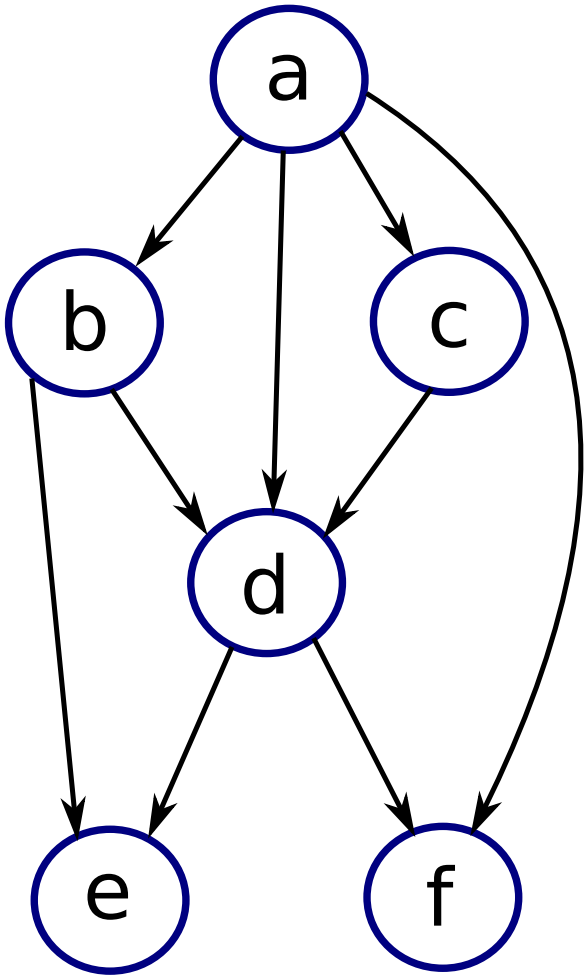
with

$\lambda_0 = l_0, \lambda_1 = l_1, \lambda_2 = \{l_0, l_1\}, \lambda_3 = l_2, \lambda_4 = \{l_0, l_2\}$

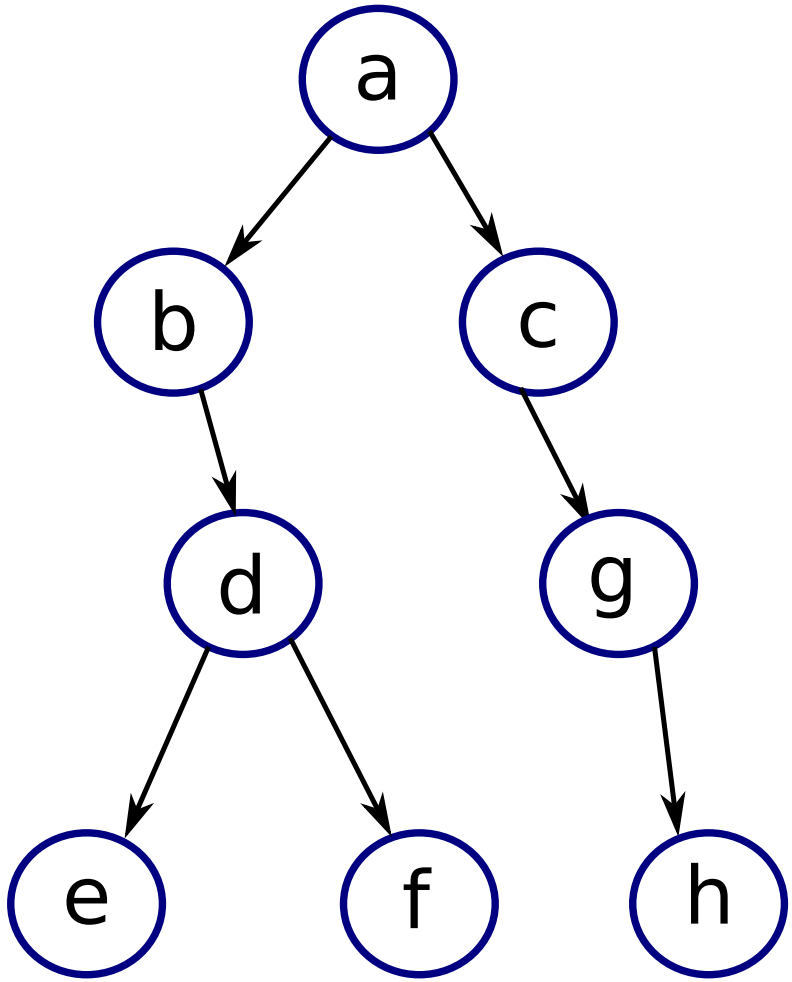
Outline

Hierarchy : tree or DAG

DAG = Directed Acyclic Graph



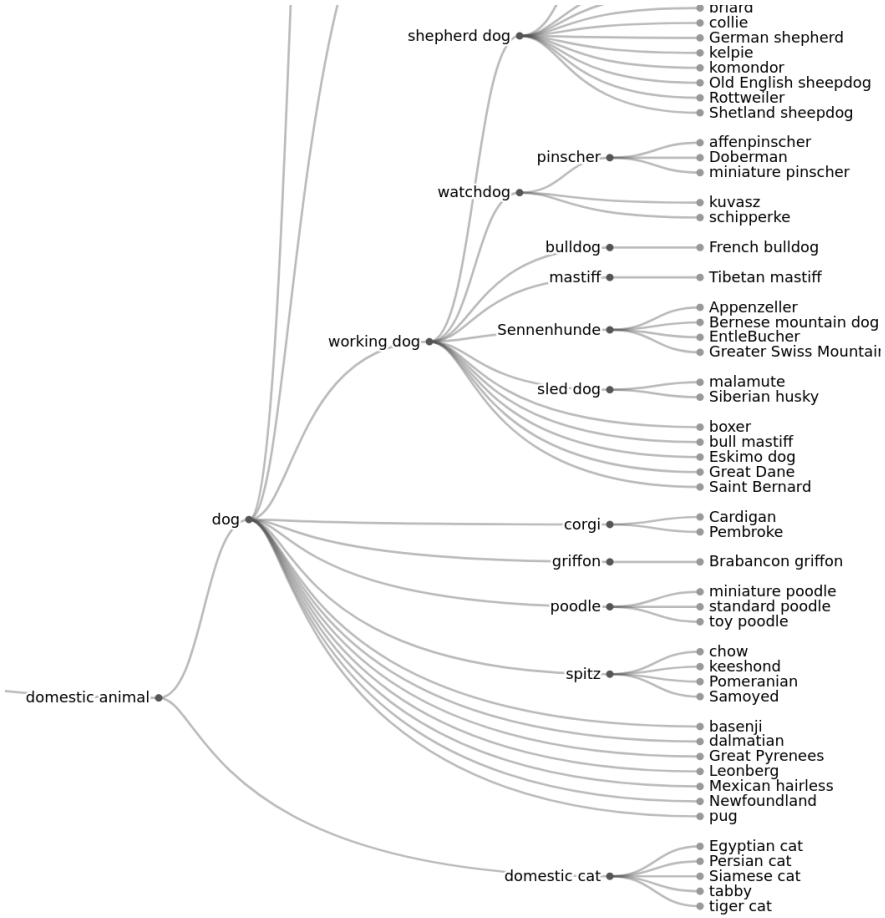
a DAG



a tree

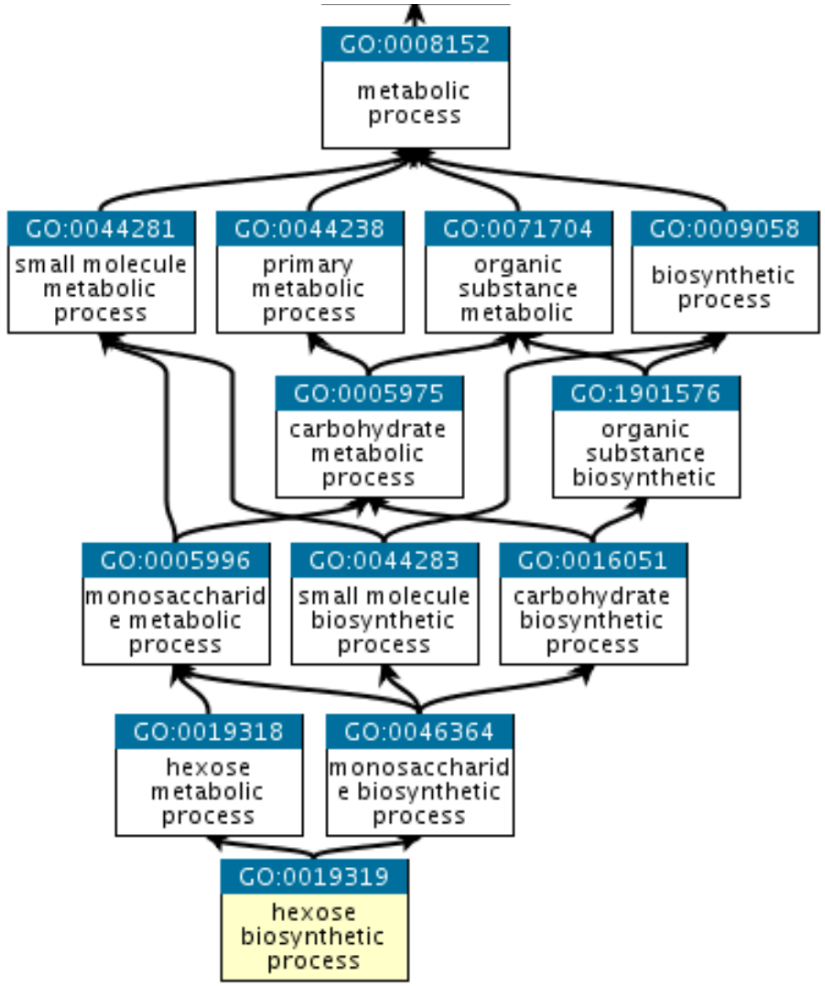
Example of tree

- Imagenet tree



Example of DAG

- from ontologies
  - Example : Gene Ontology
    `https://geneontology.org/docs/ontology-documentation/`



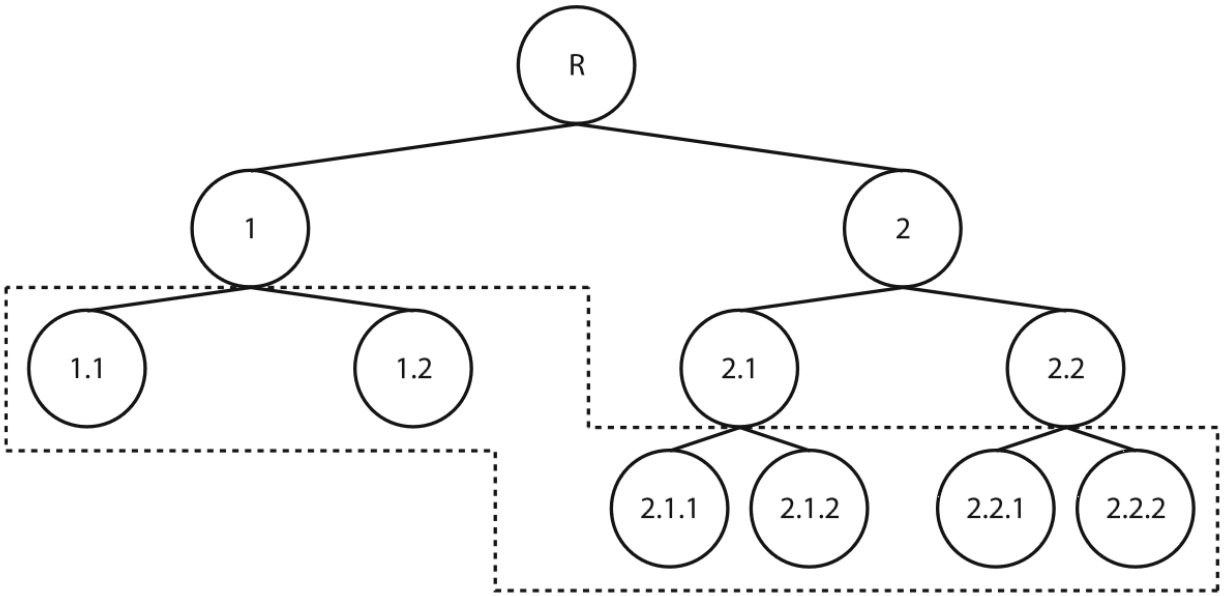
Outline

Hierarchical classification

- Hierarchical classification is a sub-category of structured classification
- outputs defined over a class taxonomy
  - DAG or tree with a 'is a' relationship
- Exploration of the hierarchy [Silla and Freitas, Data Min. Kn. Disc. 2011] :
  - top-down
  - bottom-up
  - local versus global

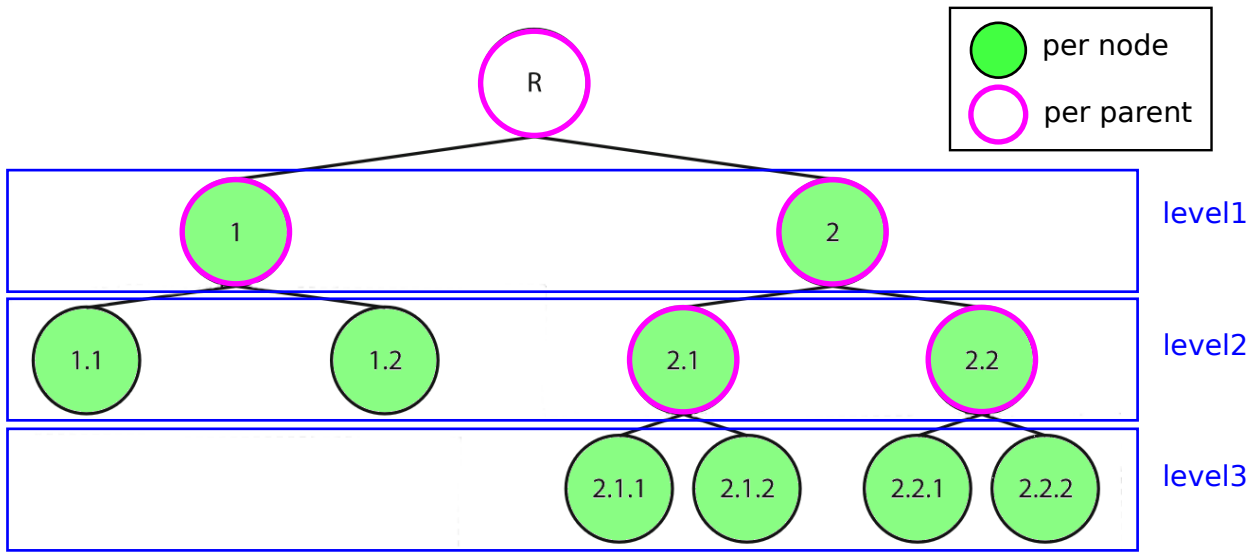
Flat classification (or bottom-up)

- ignore the class hierarchy
- consider only leafs
- at prediction : when a leaf class is predicted, then labels from ancestors are also selected
  - eg : if 2.1.2 is predicted, then also 2.1 and 2



Local classifiers (or top-down)

- per node
  - different methods for positive/negative selections
  - inconsistency
- per parent node : avoid inconsistency predictions
- per level : a multiclass classifier per level
  - need to correct inconsistency



Global classifiers (or big-bang)

- consider the entire class hierarchy at once during the training phase
- could use the top-down approach at testing

Leveraging Class Hierarchies

Making Better Mistakes: Leveraging Class Hierarchies with Deep Networks

Luca Bertinetto* Romain Mueller* Konstantinos Tertikas Sina Samangooei Nicholas A. Lord*

{luca.bertinetto, romain.mueller, konstantinos.tertikas, sina, nick.lord}@five.ai

$$\min \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\phi(x_i, \theta), y(C_i)) + \mathcal{R}(\theta)$$

Leveraging Class Hierarchies

Making Better Mistakes: Leveraging Class Hierarchies with Deep Networks

Luca Bertinetto* Romain Mueller* Konstantinos Tertikas Sina Samangooei Nicholas A. Lord*
{luca.bertinetto, romain.mueller, konstantinos.tertikas, sina, nick.lord}@five.ai

$$\min \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\phi(x_i, \theta), y(C_i)) + \mathcal{R}(\theta)$$

$\mathcal{L}^H$ hierarchical losses

ϕ^H hierarchically-informed architecture changed

$y^H(C_i)$ label embedding

$\mathcal{R}^H$ regularisation

H : class relationship

Outline

Regularisation : $\mathcal{R}^H$

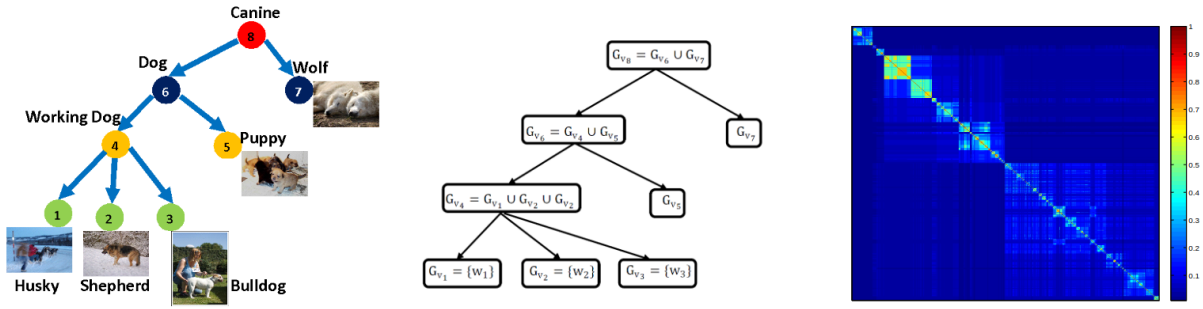


Figure 1: (a) Image category hierarchy in ImageNet; (b) Overlapping group structure; (3) Semantic relatedness measure between image categories.

- overlapping-group-lasso penalty in [Zhao etal Neurips 2011] :

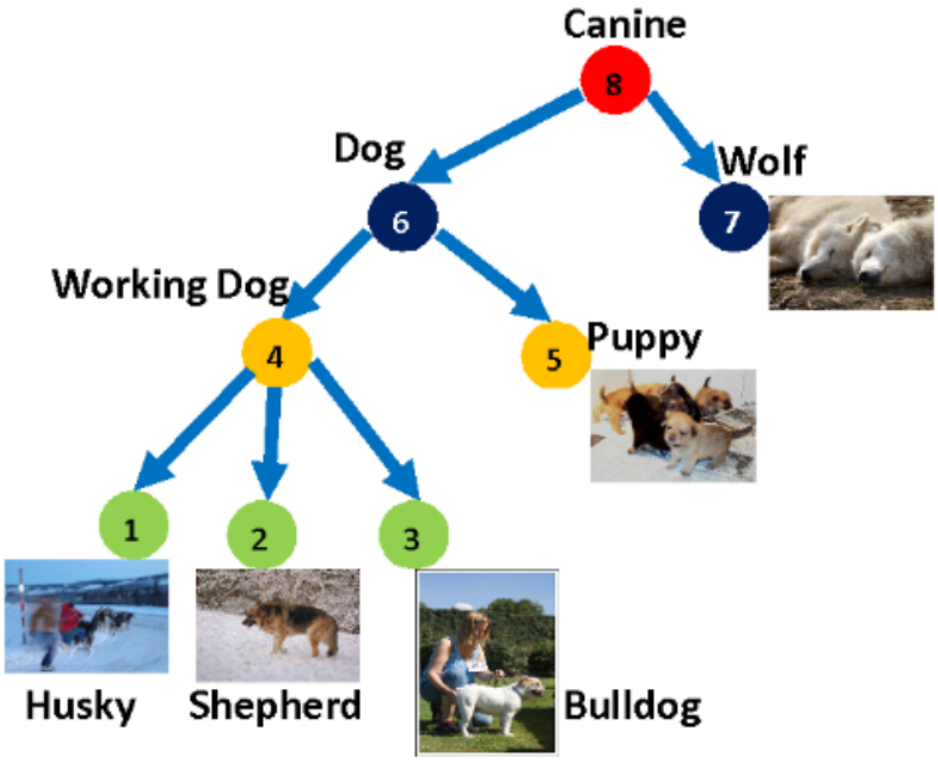
$$\Omega(\mathbf{W}) = \sum_j \sum_{v \in \mathcal{V}} \gamma_v \| \|\mathbf{w}_{jG_v}\| \|_2$$

  - where group G_v composed of all leaf nodes in the subtree rooted at v
  - where $\mathbf{w}_{jG_v}$ is the weight coefficients w_{jk}, $k \in G_v$ for input j, associated with categories in G_v. γ_v is associated to group G_v and reflects the strength of correlation within the group.

Outline

LCA : Lowest Common Ancestor

also called LCS : Lowest Common Subsumer



- height of a node : longest path from that node to a leaf
- Husky and Bulldog :
  - lowest common ancestor :
    - Working Dog
  - LCA = 1
- Husky and Puppy :
  - lowest common ancestor :
    - Dog
  - LCA = 2
- for a metric : divide by the height of the tree
  - $d(u,v) = \frac{height(lcs(u,v)}{\max_{w \in V} height(w)}$
  - not applicable to DAG : violation of triangle inequality

Label embeddings : y^H

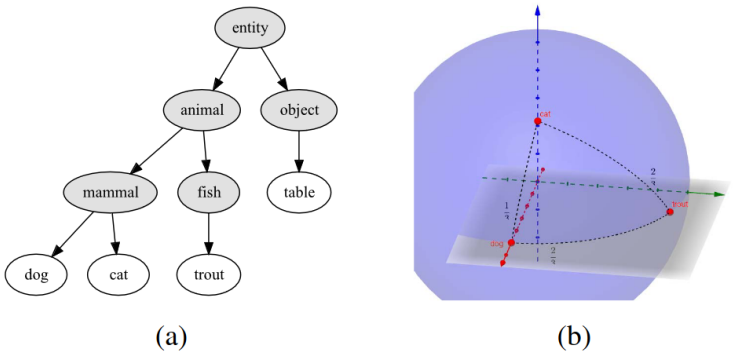


Figure 2: (a) A toy hierarchy and (b) an embedding of 3 classes from this hierarchy with their pair-wise $d_{\mathcal{G}}$.

- Label embeddings on an unit hypersphere [Barz etal WACV 2019] :
  - with the constraint that scalar product between two class embedding is related to the similarity measure. $s(u, v) = 1 - d(u, v)$

$$\forall 1 \leq i, j \leq n \quad : \quad \varphi(c_i)^T \varphi(c_j) = s(c_i, c_j) \tag{1}$$

$$\forall 1 \leq i \leq n \quad : \quad \|\varphi(c_i)\| = 1 \tag{2}$$

  - iterative algorithm
    - starts with $\varphi(c_1) = [1, 0, 0, \ldots, 0]^T$
    - for class i, (i-1) constraints from (2) where $j < i$ solved for (i-1) first components.
    - ith component added for normalisation (2) : $\varphi(c_i)_i = \sqrt{1 - \|\varphi(c_i\|^2}$

Label embeddings : y^H

- Soft labels ([Making Better Mistakes, CVPR 2020]) :

$$y_A^{soft}(C) = \frac{\exp(-\beta d(A, C))}{\sum_{B \in C} exp(-\beta d(B, C))}$$

  - where d is a distance eg LCA divided by the height of the tree
    - LCA(M,N) : height of the lowest common ancestor between class M and class N
  - β big : one-hot encoding
  - β small : uniform
  - More infos on label smoothing : [When Does Label Smoothing Help, Google, Neurips 2020]

Outline

Hierarchical losses : $\mathcal{L}^H$

- Mapping images onto class centroids [Barz etal WACV 2019] :
  - $\mathcal{L}_{CORR+CLS} = \mathcal{L}_{CORR} + \lambda \mathcal{L}_{CLS}$
    - CORR :

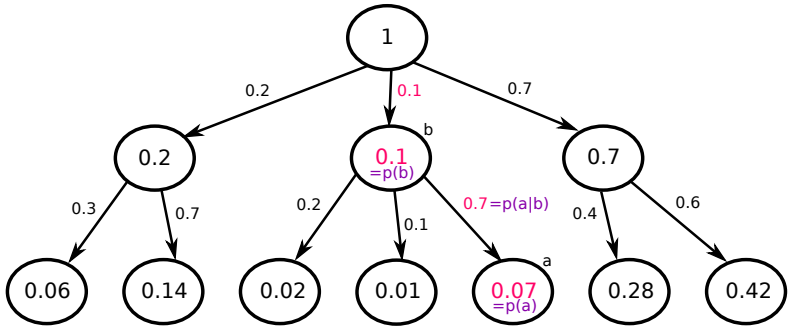
$$\mathcal{L}_{CORR}(B) = \frac{1}{m} \sum_{b=1}^{m} (1 - \psi(I_b))^T \varphi(c_{y_b})$$

    - find ψ for mapping images I_b onto class centroids $\varphi(c_{y_b})$
    - CLS : categorical cross-entropy loss
    - $\lambda = 0.1$ in the paper's experiments

Semantic embedding [Barz etal WACV 2019]

Figure 1: Comparison of image retrieval results on CIFAR-100 [18] for 3 exemplary queries using features extracted from a variant of ResNet-110 [13] trained for classification and semantic embeddings learned by our method. The border colors of the retrieved images correspond to the semantic similarity between their class and the class of the query image (with dark green being most similar and dark red being most dissimilar). It can be seen that hierarchy-based semantic image embeddings lead to much more semantically consistent retrieval results.

Hierarchical losses : $\mathcal{L}^H$



- HXE : Hierarchical cross-entropy
  - suppose that hierarchy $\mathcal{H}$ is a tree
  - $p(C) = \prod_{l=0}^{h-1} p(C^l|C^{l+1})$ where h is the height of node C and we suppose that $p(C^h) = 1$
  - $\mathcal{L}_{HXE}(p, C) = -\sum_{l=0}^{h-1} \lambda(C^l) \log p(C^l|C^{l+1})$ where $\lambda(C^l)$ is the weight associated with the edge node $C^{l+1} \rightarrow C^l$
    - if all λ are equal to 1 : cross entropy
    - could also be used with only class probability using $p(C^l|C^{l+1}) = \frac{\sum_{A \in Leaves(C^l)} p(A)}{\sum_{B \in Leaves(C^{l+1})} p(B)}$
    - in ([Making Better Mistakes, CVPR 2020]) : $\lambda(C) = \exp(-\alpha h(C))$,

C-HMCNN : Coherent Hierarchical Multi-Level

[Giunchiglia, Lukasiewicz, Neurips 2020]

- coding the hierarchy with constraints :
  - at node A, local model m_A. Data x belongs to class A if $m_A(x) \geq \tau$, threshold.
  - if B is a subclass of A, then construct m_A such that $m_A(x) \geq m_B(x), \forall x$
  - a solution : MCM (Max Constraint Module) after h
    - $MCM_B = h_B$ and $MCM_A = \max(h_A, h_B)$
    - thus, if $MCM_B(x) \geq \tau$, then $MCM_A(x) \geq MCM_B(x) \geq \tau$
- General case : D_A is the set of subclasses from A
  - $MCM_A = \max\limits_{B \in D_A} h_B$
  - local loss : $MCLoss_A = -y_A \ln(\max\limits_{B \in D_A}(y_B h_B)) - (1 - y_A)\ln(1 - MCM_A)$
    - almost binary-crossentropy except for data belonging to A : only those belonging to B also are considered ($y \in \{0, 1\}$)
    - avoiding to be stuck in bad local minima when numerous ancestors
- global loss : $MCLoss = \sum\limits_{A \in S} MCLoss_A$ where S is the set of hierarchically structured classes

Outline

Hierarchical architectures : ϕ^H

- [Redmon and Farhadi. Yolo9000: better, faster, stronger. CVPR 2017]
  - tree of classifiers outputting conditional probabilities, with the product of the conditionals along a given leaf's ancestry representing its posterior

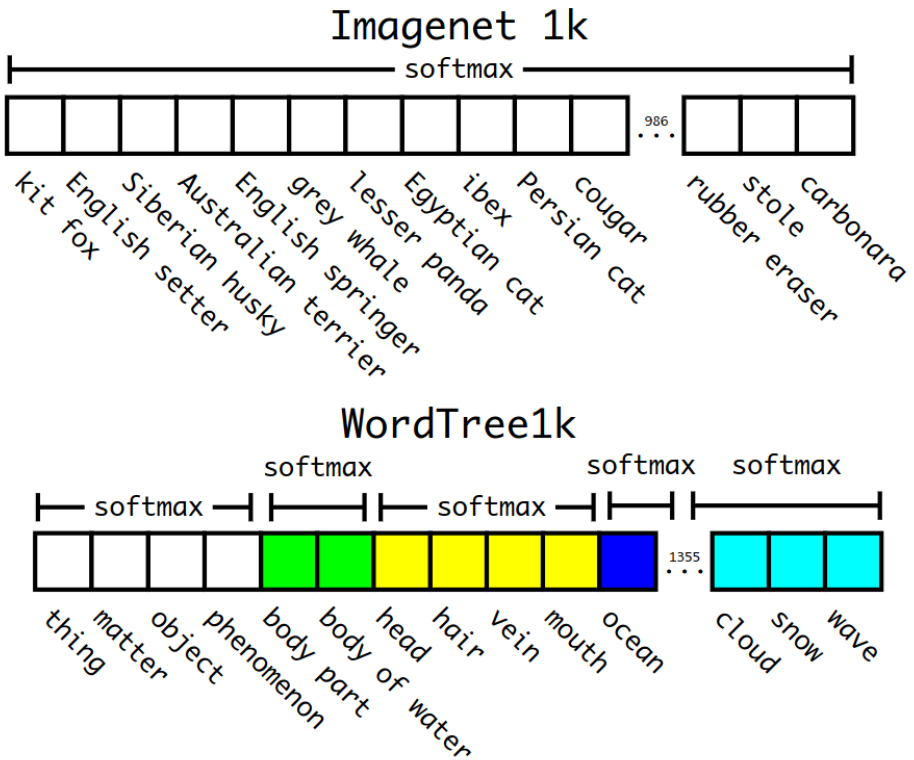
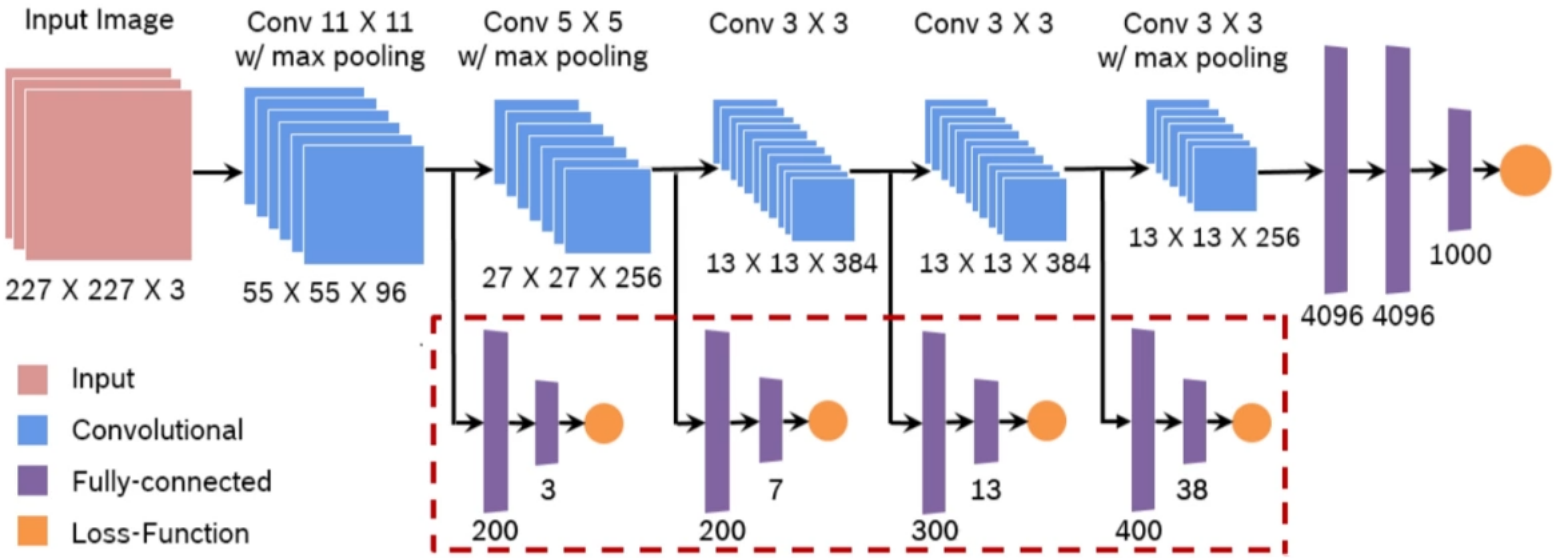


Figure 5: Prediction on ImageNet vs WordTree. Most ImageNet models use one large softmax to predict a probability distribution. Using WordTree we perform multiple softmax operations over co-hyponyms.

- Wordnet pruned into a tree
- classifier at every parent node
  - one softmax per sibling group
  - cross-entropy loss over leaf posteriors

Hierarchical architectures : ϕ^H

- Hierarchy-Aware CNN



  - [Do Convolutional Neural Networks Learn Class Hierarchy? Bilal etal, IEEE Trans. Visu and CG 2018]
  - see video at `https://vimeo.com/228263798`

Hierarchical architectures : ϕ^H : many others

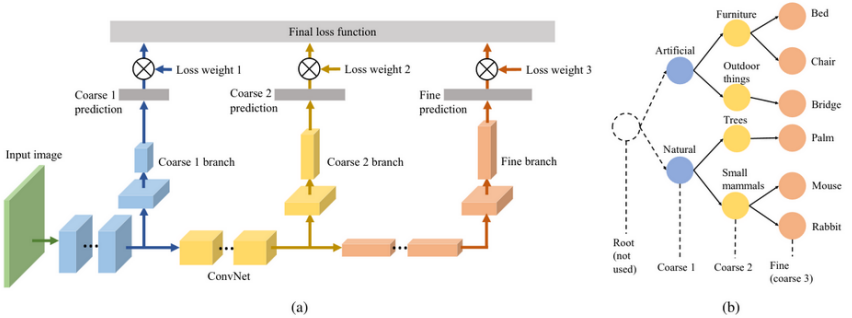


Figure 1: (a) Architecture of Branch Convolutional Neural Network (B-CNN). The network at the bottom can be an arbitrary ConvNet. There can be multiple branch networks and each of them outputs a coarse prediction. The final loss function is a weighted summation of all coarse losses. (b) A sample hierarchical label tree where classes are taken from CIFAR-100 dataset.

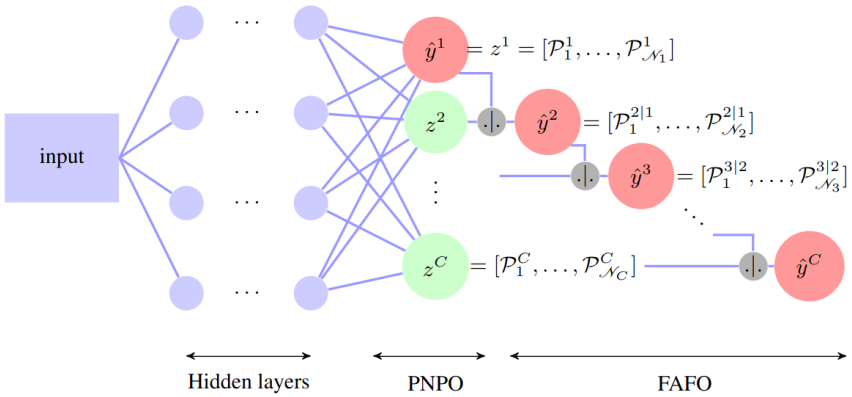
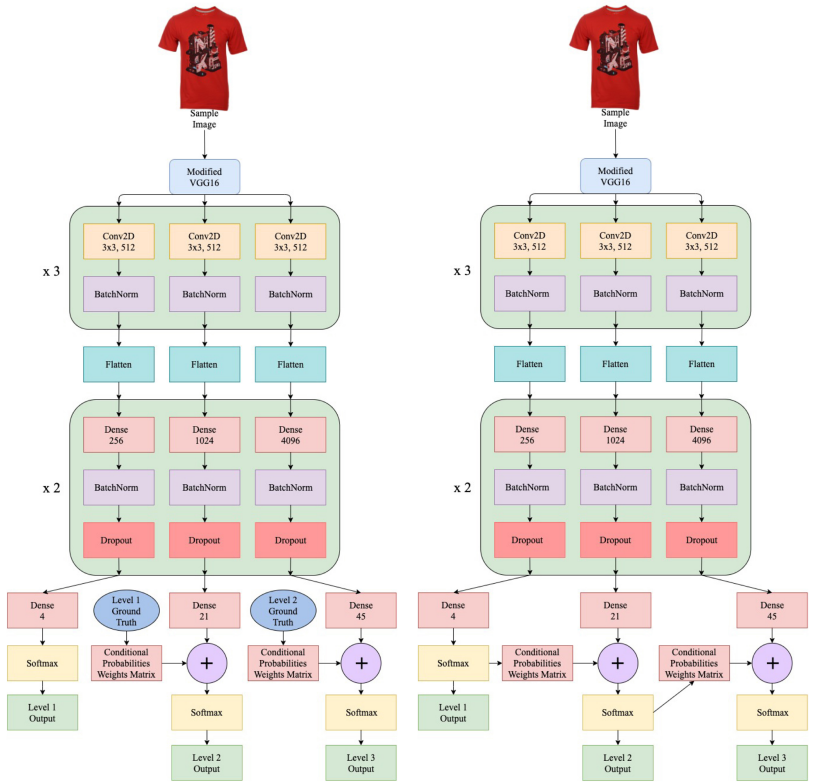


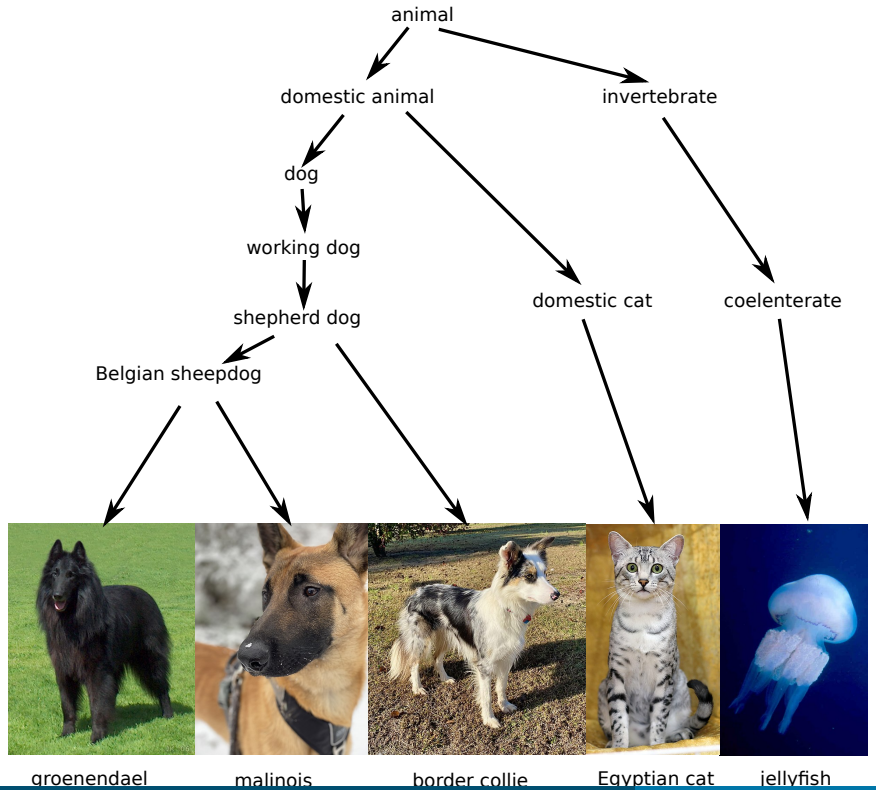
Fig. 1. The GH-CNN architecture. The nodes ⓘ represent the Bayesian adjustement defined by equation (3).



Outline

Metrics : top-k

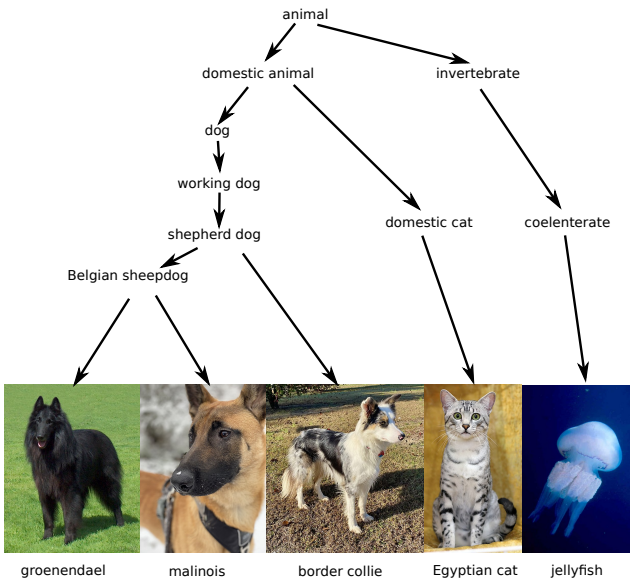
- Top-k error
  - correctly classified if the true label is among the top k predictions
  - all mistakes are considered equally
- Ideas :
  - Is it as bad to missclassify a viola to a violin as to missclassify a viola to a cat ?
  - In top-k accuracy, $k > 1$, are all bad classes equivalent ? Is it all bad if the true class do no belong the the first k classes ?



- top-2 for a groenendael :
  - malinois and border collie
  - groenendael and jellyfish
  - jellyfish and egyptian cat
  - malinois and groenendael
  - egyptian cat and border collie

Metrics : Hierarchical measures

- Hierarchical measures : weighting the severity of mistakes
  - LCA : height of the lowest common ancestor between true class and predicted class
  - hierarchical distance of a mistake : LCA when only one class is predicted
  - average hierarchical distance of top-k : average of LCA between the true class and each of the k best predictions



- top-2 for a groenendael :
  - malinois and border collie : (1+2)/6/2=0.25
  - groenendael and jellyfish : (0+6)/6/2=0.5
  - jellyfish and egyptian cat : (6+5)/6/2=0.92
  - malinois and groenendael : (1+0)/6/2=0.08
  - egyptian cat and border collie : (5+2)/6/2=0.58

Metrics : P_{LCA}, R_{LCA} and F_{LCA}

- Precision, Recall and F-measure :

$$P_{LCA} = \frac{pred_{aug} \cap true_{aug}}{pred_{aug}}; R_{LCA} = \frac{pred_{aug} \cap true_{aug}}{true_{aug}}; F_{LCA} = 2\frac{P_{LCA}.R_{LCA}}{P_{LCA} + R_{LCA}}$$

from `https://arxiv.org/pdf/1306.6802.pdf`

  - 'malinois' :
    - LCA = 'belgian sheepdog'
    - $pred_{aug}$ = {'malinois','belgian sheepdog'}
    - $true_{aug}$ = {'groenendael','belgian sheepdog'}
    - $P_{LCA} = \frac{1}{2} = R_{LCA} = F_{LCA}$
  - 'border collie' or 'egyptian cat'
    - LCA = 'domestic animal'
    - $pred_{aug}$ = {'border collie', 'shepherd dog', 'working dog','dog','domestic animal', 'egyptian cat', 'domestic cat' }
    - $true_{aug}$ = {'groenendael','belgian sheepdog','shepherd dog','working dog','dog','domestic animal'}
    - $P_{LCA} = \frac{4}{7} = 0.57$, $R_{LCA} = \frac{4}{6} = 0.67$, $F_{LCA} = 0.62$

Datasets

- Animals with attributes 2 (AwA no more available). Originally build as successor of AwA but for Zero-Shot Learning.
  - notion of concepts if AwA similar and described in Learning Systems of Concepts with an Infinite Relational Model, Kemp etal AAAI 2006
- tieredImageNet-H (subset of ImageNet + WordNet hierarchy), 608 classes, tree of height 13
  - informations are given in the sup. mat. but not the dataset itself
- iNaturalist-H : images of organisms, biological taxonomy, 1010 classes, tree of height 8
- Cifar100 + WordNet ontology
- BreakHis dataset "Breast Cancer Histopathological Images"
  - 3 levels : Benign/Malign + subtypes (4 for each), 7909 images
- fashion MNIST
- [Serrano-Pérez and Enrique Sucar, Artificial datasets for hierarchical classification, Journal of Expert Systems with Applications, Vol. 182, 2021]
- [ImClefA and ImClefD] : ImageCLEF2007 and ImageCLEF2008 ($\simeq$ 2000 images) with IRMA codes ($\simeq$ 200 codes), 4 lev. of hierarch.
  - available there with other HMC datasets : `https://github.com/EGiunchiglia/C-HMCNN/tree/master/HMC_data`

Questions

- Leaf classification : what is the added value of a hierarchy for the classification of leafs ?
  - classical metric
  - better mistakes
  - does it help to find the dog specie if you know that it is a dog ?
- Intermediate nodes : does the leaf and children nodes help compare to classical multi-class classification ?
  - classification of dogs and cats : it is worth to consider dog's and cat's species ?
- Hierarchy :
  - WordNet or field knowledge (biological taxonomy ...)
  - Other hierarchy