

**HiClass: a Python library for
local hierarchical classification
compatible with scikit-learn**

<https://arxiv.org/pdf/2112.06560.pdf>



Classificação Hierárquica



Introdução

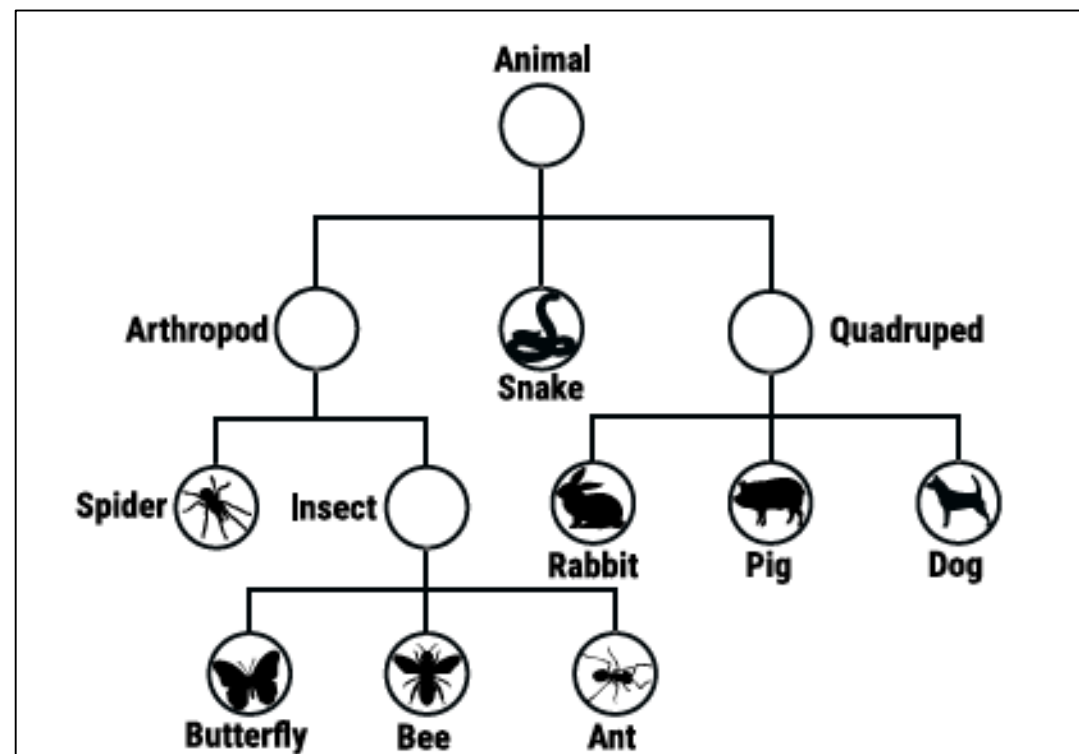
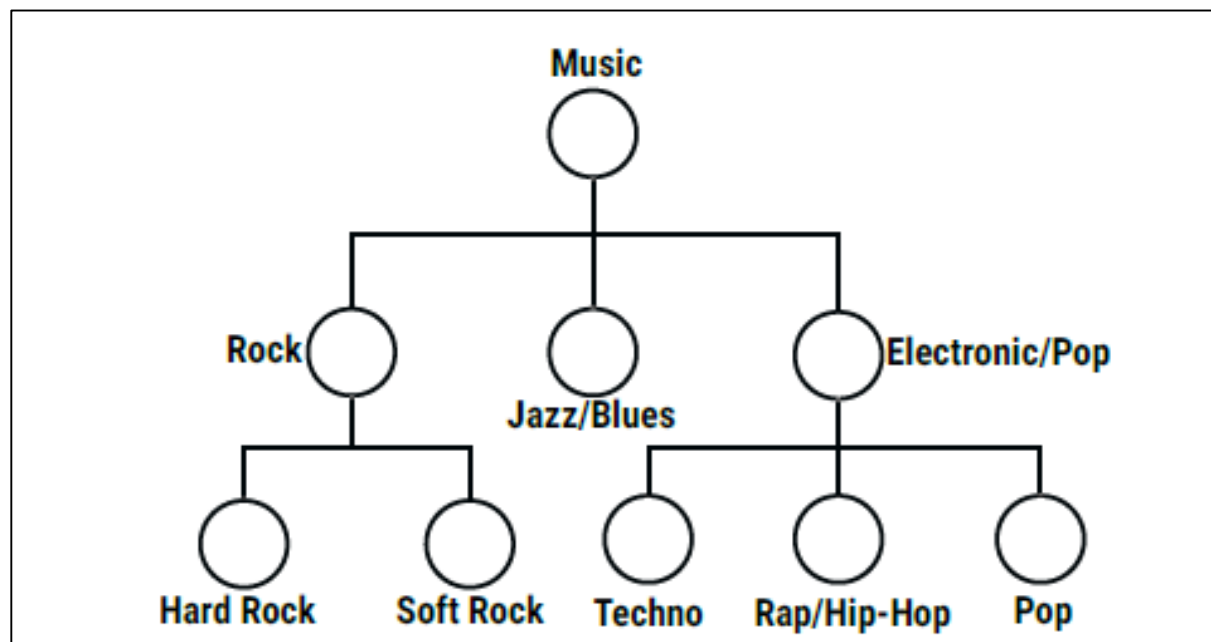
- Na classificação hierárquica, as classes podem apresentar uma relação de taxonomia ou dependência, formando subclasses e superclasses
- Se a hierarquia das classes é conhecida ou pode ser construída, sua consideração pode levar à indução de classificadores com maior acurácia preditiva



Introdução

- Exemplos:
 - Categorização de textos
 - Um texto que trata de uma partida de futebol pode ser classificado na seção de esportes, que, por sua vez, pode também incluir tópicos como basquete, futebol, tênis, entre outros ligados ao tema
 - Classificação de gêneros musicais, de animais, de imagens, etc.

Introdução





Introdução

- *É **importante destacar** que, embora uma hierarquia de classes possa ser utilizada como uma possibilidade de solução de problemas multiclasse, a mesma não se configura como um problema de classificação hierárquica*
- *A classificação hierárquica se configura apenas quando as classes possuem uma relação taxonômica inerente*
- *Ao se induzir modelos de classificação para problemas hierárquicos, a relação hierárquica entre as classes deve ser levada em consideração*



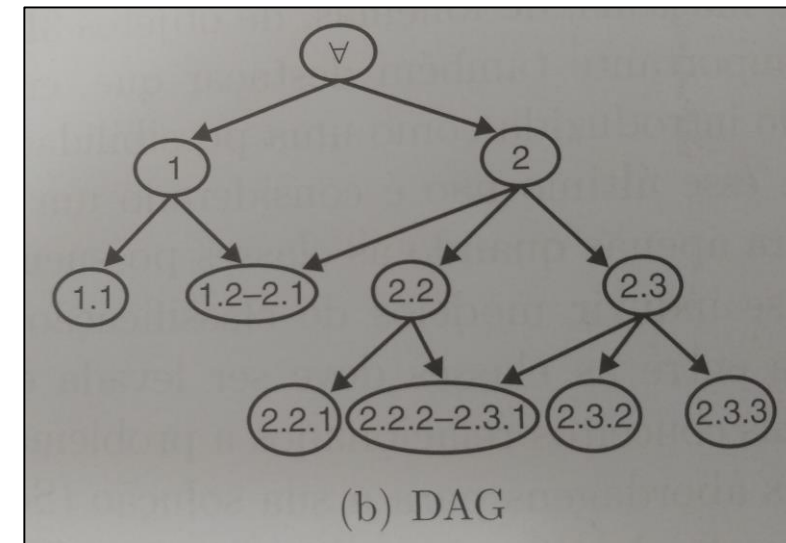
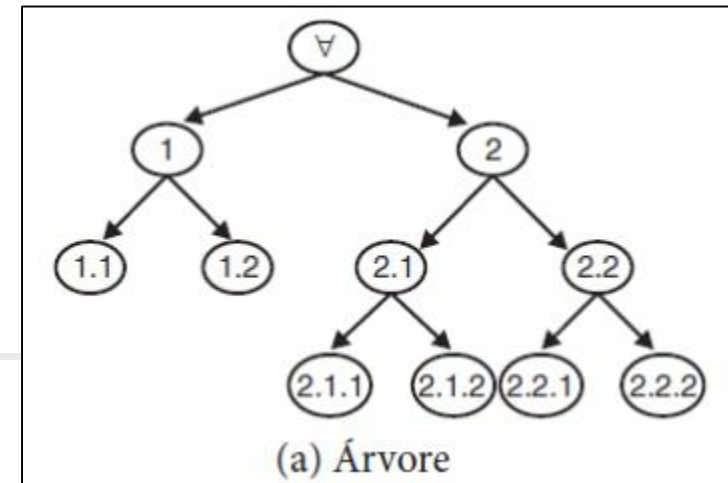
Tipos de Problemas

- Inicialmente é necessário identificar as características do problema para que seja possível escolher uma estratégia algorítmica apropriada à solução do problema
 - Tipo de hierarquia:
 - Árvore
 - DAG
 - Caminho das rotulações:
 - Múltiplo (multirrótulo)
 - Único
 - Profundidade das rotulações:
 - Completa (nós folha)
 - Parcial

Tipos de Problemas

Tipo de Hierarquia

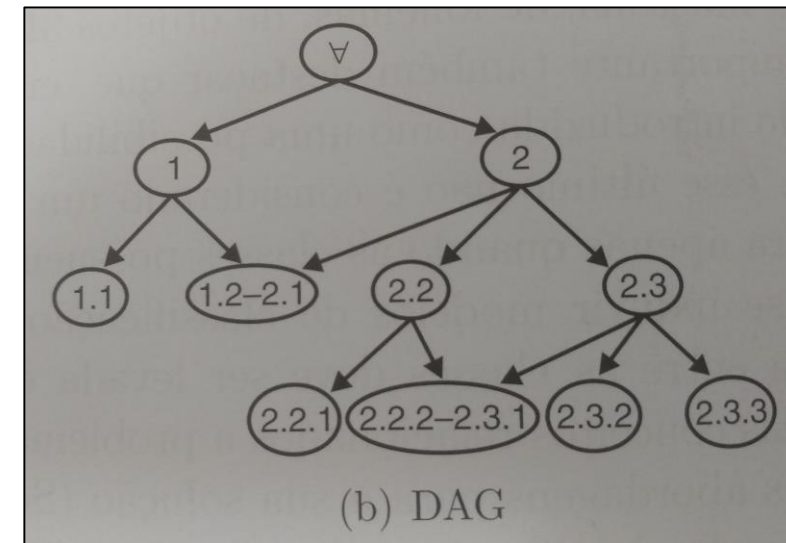
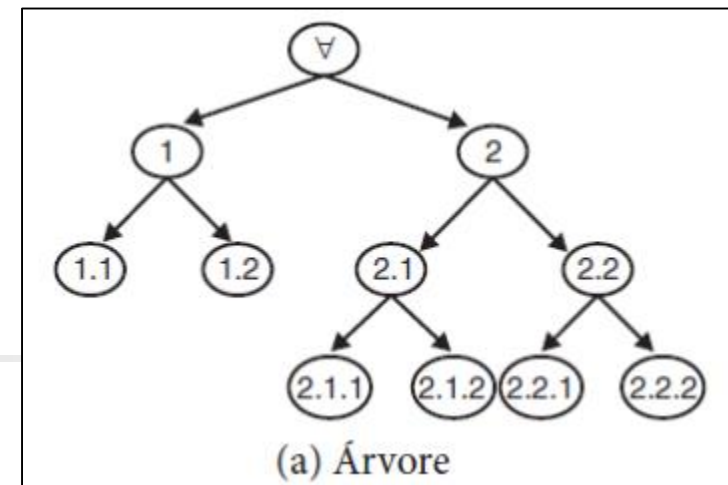
- Distinção entre **DAG** e **árvore**:
 - No DAG um nó pode possuir mais de um pai
 - As árvores são um caso particular de DAG em que todos os nós têm um único pai



Tipos de Problemas

Tipo de Hierarquia

- Em ambas as estruturas, cada nó corresponde a uma classe
- Iniciando pelo nó raiz, as classes são particionadas recursivamente em subclasses
- O conteúdo dos nós indica a classe associada a ele





Tipos de Problemas

Caminho das Rotulações

- Se os dados podem ou não seguir mais de um caminho na hierarquia
- Se os dados puderem seguir mais de um caminho na hierarquia tem-se um problema de classificação hierárquica **multirrótulo**; caso contrário, **monorrótulo**

Tipos de Problemas

Profundidade das Rotulações

- Na rotulação **completa** todo objeto possui rótulos de classes em todos os níveis em um caminho da hierarquia, da raiz a uma folha (predição mandatória em nós folha)
- Na rotulação **parcial** existem exemplos para os quais a classificação para em um nó interno da hierarquia, tipicamente porque a classificação em níveis mais profundos ainda é desconhecida para eles (predição não mandatória em nós folha)

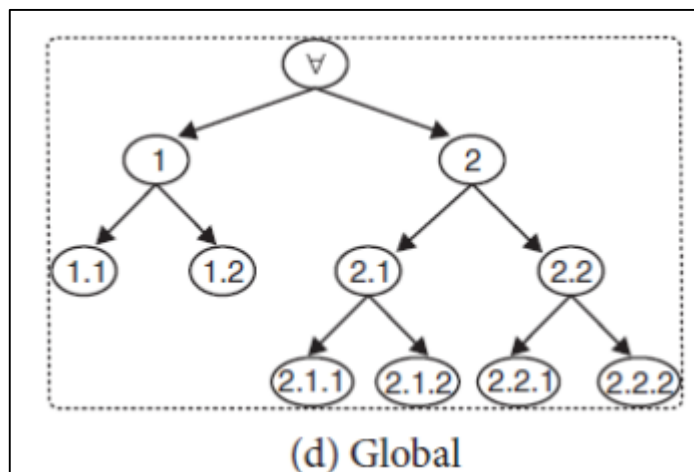
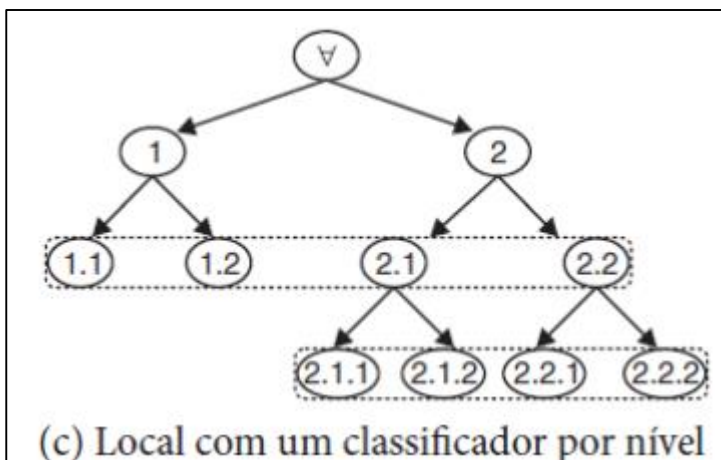
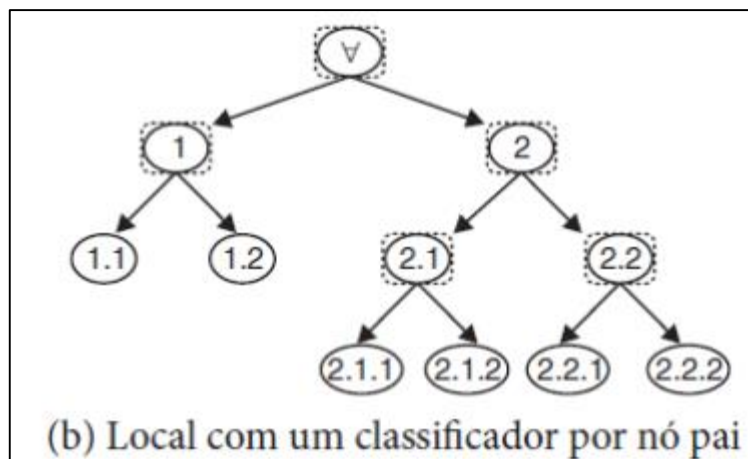
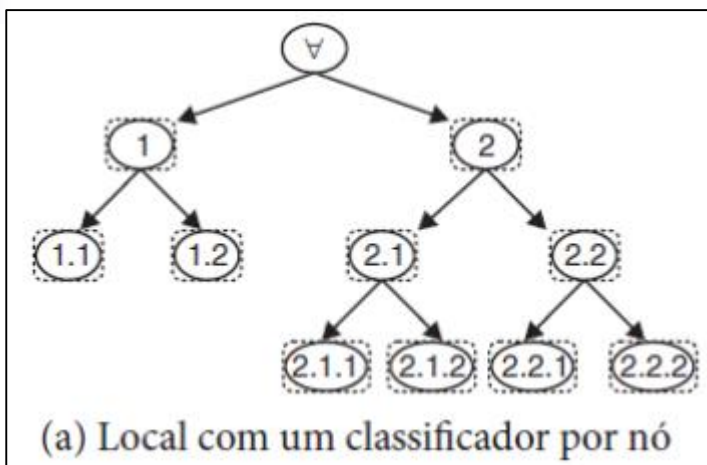


Algoritmos para Classificação Hierárquica

- É possível fazer uma distinção dos algoritmos existentes em relação a quatro características:
 - O tipo de hierarquia com que o algoritmo pode lidar, árvore ou DAG
 - Se ele pode rotular os dados de acordo com múltiplos caminhos na hierarquia (multirrótulo) ou não (monorrótulo)
 - A profundidade das predições do algoritmo, se são sempre em nós folha ou não
 - Os **tipos** de classificadores que são usados na resolução do problema hierárquico: **locais** ou **globais**

Algoritmos para Classificação Hierárquica

Tipos de Classificadores Usados





Algoritmos para Classificação Hierárquica

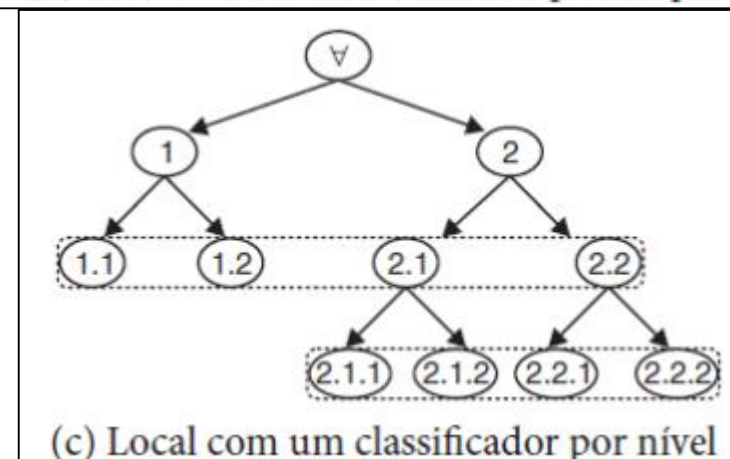
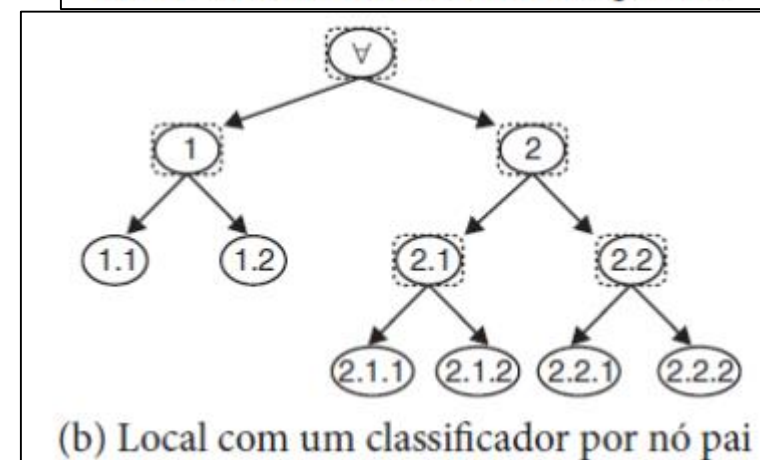
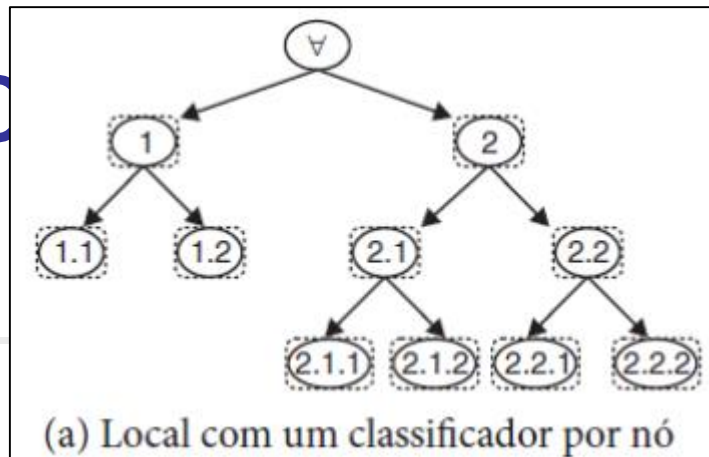
Tipos de Classificadores Usados

- Existe ainda uma quinta abordagem que consiste em usar um único classificador plano, geralmente distinguindo somente as classes em nós folha
 - Corresponde a transformar o problema hierárquico em um único problema de classificação plana
- A predição em uma classe 1.2, por exemplo, também indica que esse exemplo em particular pertence à classe 1
- Essa abordagem possui a desvantagem de poder ter que distinguir um grande número de classes (todas as folhas) sem explorar as informações de relacionamentos hierárquicos presentes (predição mandatória em nós folha)

Algoritmos para Classificação

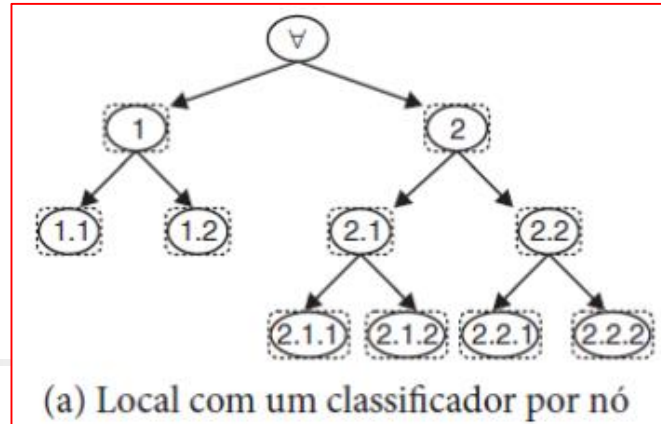
Classificadores Locais

- Abordagens que usam informações locais da hierarquia
- O uso de informação local permite o emprego direto de classificadores planos na solução do problema hierárquico



Algoritmos para Classificação

Classificadores Locais

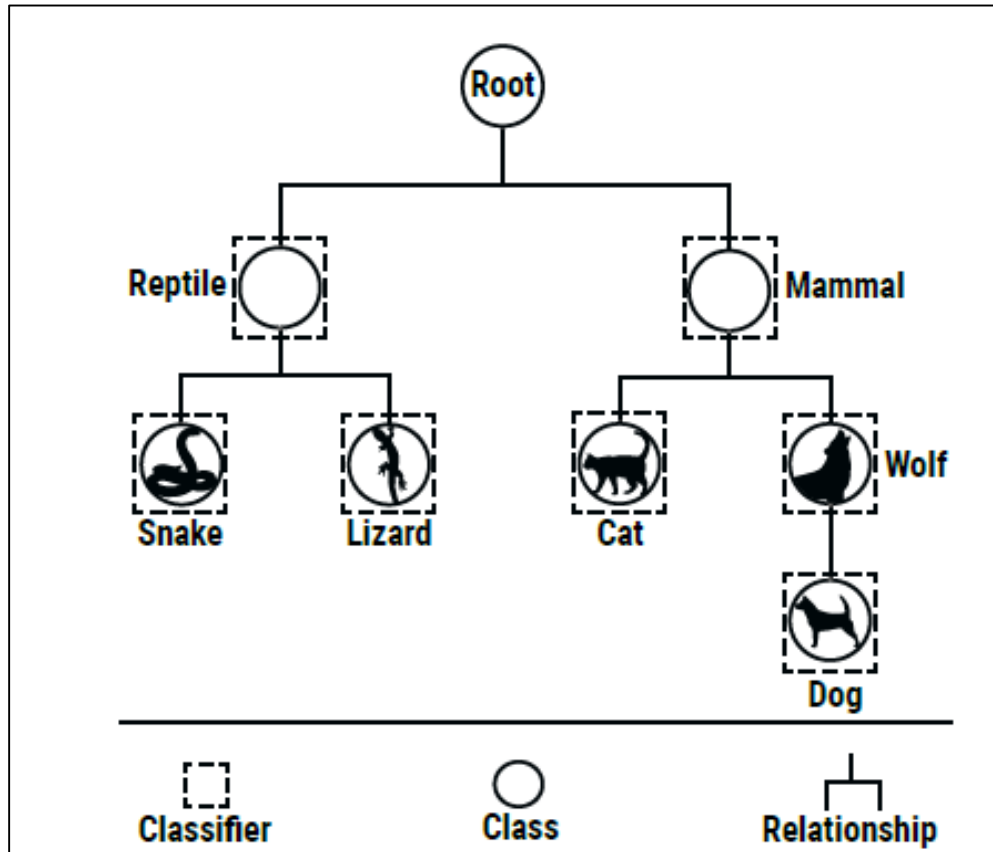


- O uso de um classificador plano por nó é a estratégia mais usada e implica na combinação de classificadores binários, um por nó da hierarquia, a menos da raiz
- Diferentes estratégias podem ser usadas para determinar que exemplos serão considerados positivos e negativos em cada nó
 - Uma das políticas mais usadas é a de irmãos, em que, para um dado nó representando a classe c_j , os exemplos positivos são aqueles com rótulo c_j e de suas subclasses, enquanto os negativos são os de nós irmãos de c_j e de suas subclasses
 - Considerando o exemplo, o classificador para a classe 2.1 tem como objetos positivos os de rótulo 2.1, 2.1.1 e 2.1.2, enquanto como negativos os de rótulo 2.2, 2.2.1 e 2.2.2

HiClass: a Python library for local hierarchical classification compatible with scikit-learn

<https://arxiv.org/pdf/2112.06560.pdf>

Local Classifier per node



- Given a hypothetical test example, the local classifier per node firstly queries the binary classifiers at nodes “Reptile” and “Mammal”
- Supposing that in this hypothetical situation the probability of the test example belonging to class “Reptile” is 0.8, and the probability of belonging to class “Mammal” is 0.5, then class “Reptile” is selected for the first level
- At the next level, only the classifiers at nodes “Snake” and “Lizard” are queried, and again the one with the highest probability is chosen

HiClass: a Python library for local hierarchical classification compatible with scikit-learn

<https://arxiv.org/pdf/2112.06560.pdf>

Local Classifier per node

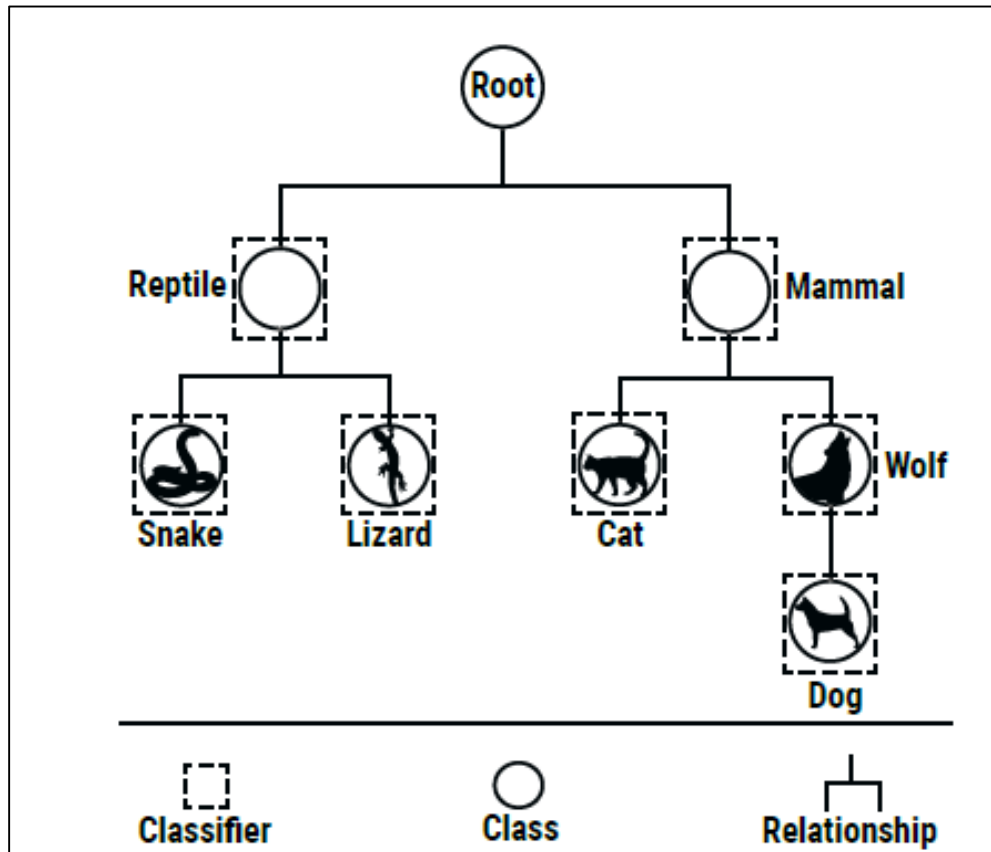


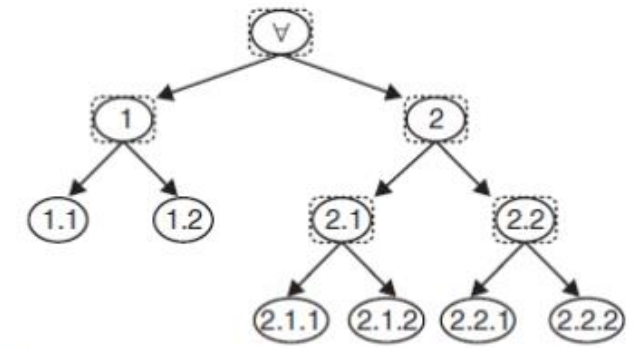
Table S4. Sets of positive and negative examples for each policy, given the class "Wolf".

Policy	$Tr^+(c_{Wolf})$	$Tr^-(c_{Wolf})$
Exclusive	Wolf	Reptile, Snake, Lizard, Mammal, Cat, Dog
Less exclusive	Wolf	Reptile, Snake, Lizard, Mammal, Cat
Less inclusive	Wolf, Dog	Reptile, Snake, Lizard, Mammal, Cat
Inclusive	Wolf, Dog	Reptile, Snake, Lizard, Cat
Siblings	Wolf, Dog	Cat
Exclusive siblings	Wolf	Cat

There are multiple ways to define the set of positive and negative examples for training the binary classifiers. In HiClass it was implemented 6 policies

Algoritmos para Classificação

Classificadores Locais



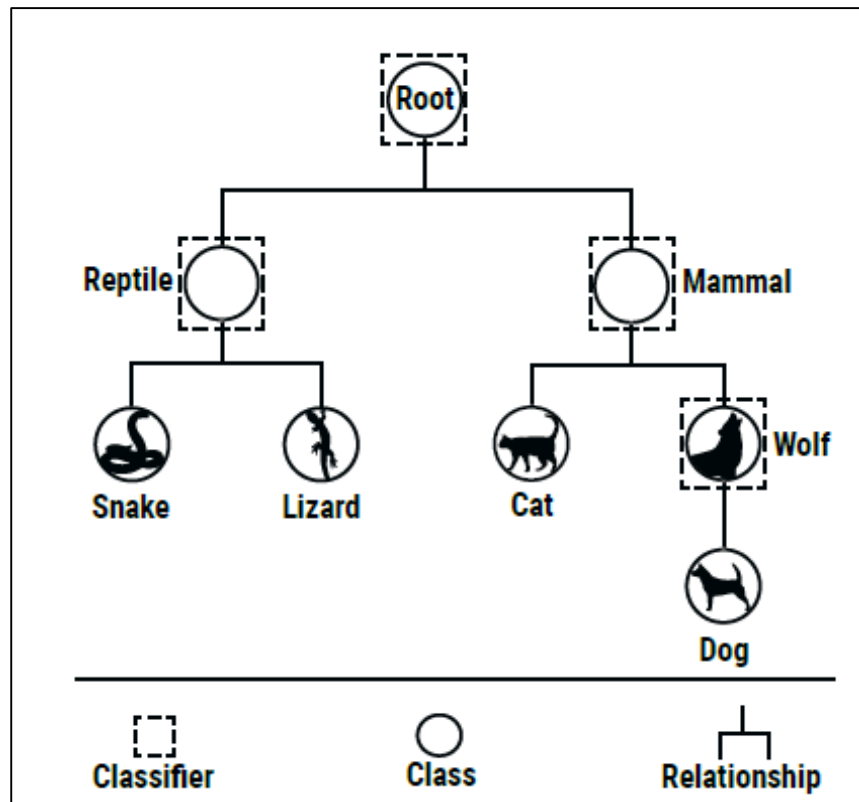
(b) Local com um classificador por nó pai

- Em relação ao uso de um classificador plano por nó pai, cada um desses nós deve distinguir as subclasses contidas em seus nós filhos, usando classificadores multiclasse ou combinações de classificadores binários (decomposição multi-classe)

HiClass: a Python library for local hierarchical classification compatible with scikit-learn

<https://arxiv.org/pdf/2112.06560.pdf>

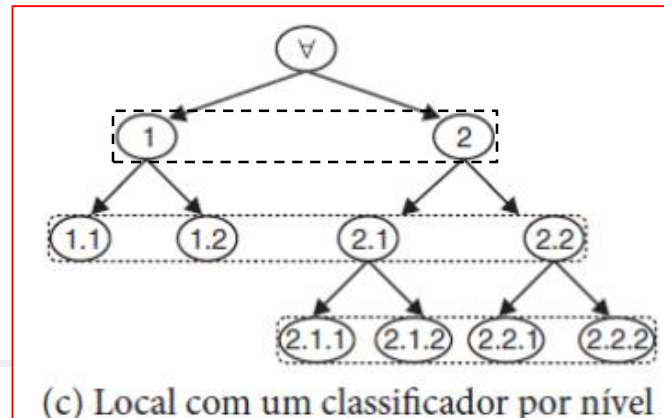
Local Classifier Per Parent Node



Assuming that the classifier located at the root node decides that a test example belongs to class "Mammal", then the next level can only be predicted by the classifier located at node "Mammal", which in turn will ultimately decide if the test example belongs either to the class "Cat" or "Wolf/Dog"

Algoritmos para Classificação

Classificadores Locais

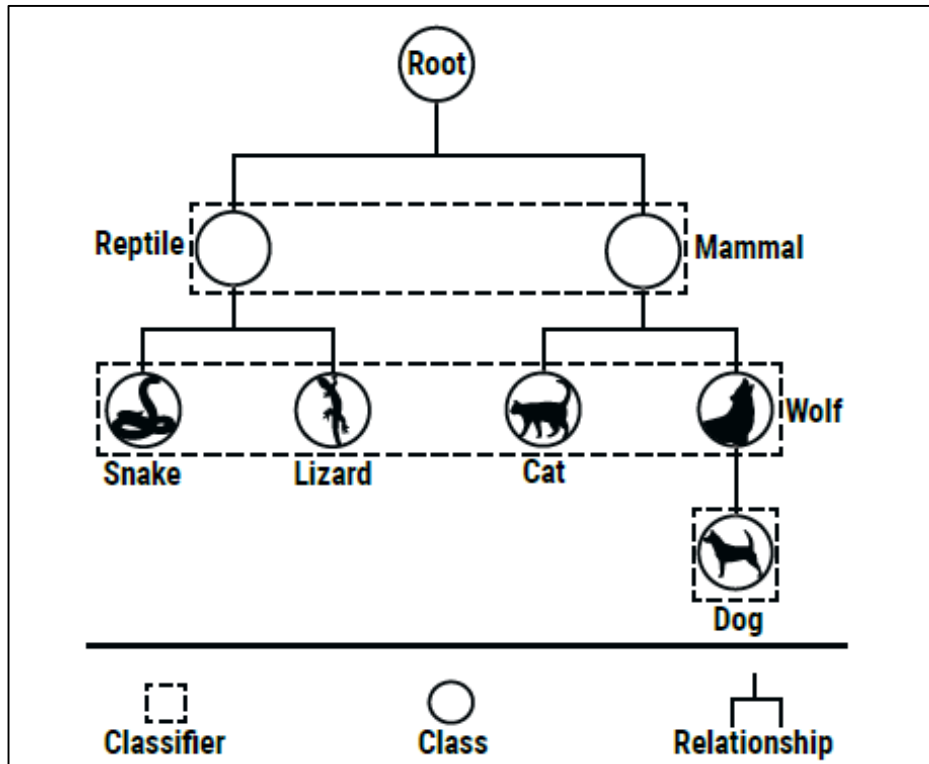


- Neste caso tem-se o uso de um classificador plano multi-classe por nível da hierarquia (três no exemplo considerado)
- Essa abordagem é a menos utilizada, pois podem haver conflitos nas predições realizadas para os diferentes níveis

HiClass: a Python library for local hierarchical classification compatible with scikit-learn

<https://arxiv.org/pdf/2112.06560.pdf>

Local Classifier Per Level



- Supposing that for a given test example the classifier at the first level returns the probabilities 0.91 and 0.7 for classes "Reptile" and "Mammal", respectively, then the one with the highest probability is considered as the correct prediction, which in this case is class "Reptile"
- For the second level, only the probabilities for classes "Snake" and "Lizard" are considered and the one with the highest probability is the final prediction



Algoritmos para Classificação Hierárquica

Classificadores Locais

- Na etapa de teste, partindo-se dos classificadores em níveis iniciais da hierarquia, as **predições** obtidas são usadas para escolher um novo classificador até atingir um nó folha ou por fim decidir parar a predição em um nível intermediário da hierarquia (mais usada, conhecida como top-down)
- Uma desvantagem dessa estratégia é que, quando um exemplo é erroneamente classificado em um determinado nível, esse erro será provavelmente propagado para os níveis seguintes



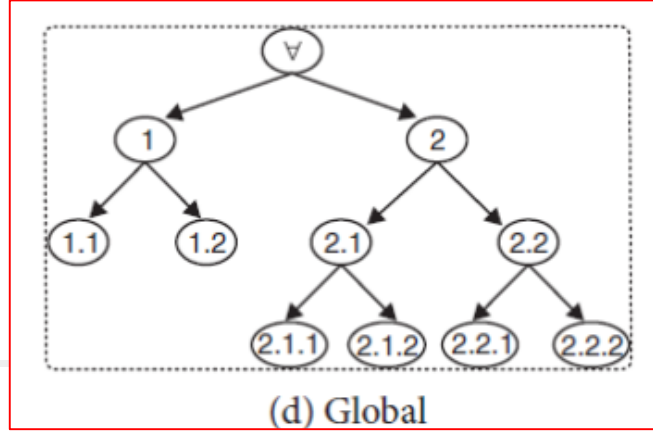
Algoritmos para Classificação Hierárquica

Classificadores Locais

- Se o problema não exige predições mandatórias em nós folha, algum mecanismo pode ser empregado para parar a classificação em algum nível
 - Por exemplo, o classificador no próximo nível é consultado somente se a confiança na predição atual é maior que um determinado limiar

Algoritmos para Classificação

Classificadores Globais



- Os classificadores globais consideram a hierarquia como um todo na etapa de treinamento e não possuem a modularidade característica das abordagens locais
- O modelo final obtido pode ser menor do que aquele formado por múltiplos classificadores locais. Assim, geralmente um único modelo é induzido
- Uma solução trivial é transformar o problema em um problema plano multirrótulo. Para tal, são adicionados a cada exemplo de treinamento os rótulos dos nós antecessores



Avaliação de Classificadores Hierárquicos

- Dada a particularidade das predições terem que seguir uma hierarquia predefinida, as medidas de avaliação de classificadores hierárquicos devem ser adaptadas
 - Por exemplo, é possível considerar predições em níveis mais profundos mais úteis, dando pesos diferentes a classificações obtidas em níveis distintos da hierarquia
 - Outra possibilidade é permitir classificações parcialmente corretas, em que as predições são corretas para apenas alguns níveis da hierarquia



Avaliação de Classificadores Hierárquicos

- As medidas de avaliação hierárquicas podem ser genericamente divididas como baseadas:
 - Em distância
 - Em profundidade
 - Em similaridade (semântica) entre as classes e a hierarquia
 - **Nas relações de descendência e/ou ancestralidade das classes na hierarquia** (mais usadas)



Avaliação de Classificadores Hierárquicos

- **Nas relações de descendência e/ou ancestralidade das classes na hierarquia**

O conjunto
Ancestral(Z)
inclui a classe Z

$$P_H = \frac{|Ancestral(c_p) \cap Ancestral(c_v)|}{|Ancestral(c_p)|}$$

A raiz da
hierarquia não
é considerada
um ancestral

$$R_H = \frac{|Ancestral(c_p) \cap Ancestral(c_v)|}{|Ancestral(c_v)|}$$

Em ambas, considera-se o número de ancestrais comuns existentes entre a(s) classe(s) verdadeira(s) do exemplo e aquela(s) predita(s) pelo classificador

Para obtenção da precisão e revocação em um conjunto contendo n dados, soma-se as medidas calculadas para cada objeto individual



HiClass: a Python library for local hierarchical classification compatible with scikit-learn

- HiClass is an open-source Python library for **local hierarchical classification**
 - <https://arxiv.org/pdf/2112.06560.pdf>
 - <https://github.com/mirand863/hiclass>