

Decomposição de Problemas Multi-classe



Introdução

- Diversas técnicas de AM foram originalmente formuladas para problemas binários
 - SVMs, RNAs Perceptron
- A generalização para problemas multiclasse pode ser realizada, basicamente, por meio de duas estratégias:
 - **Combinando preditores gerados em subproblemas binários (estratégia decomposicional) (em geral, a utilizada)**
 - Realizando-se adaptações nos algoritmos originais das técnicas consideradas



Introdução

- Demais motivações:
 - Alguns algoritmos não são adequados a problemas com um número elevado de classes
 - Existem algoritmos que operam com múltiplas classes mas que contêm procedimentos internos restritos a problemas de duas classes
 - Divisão do subconjunto de atributos nominais no CART
 - Mesmo que o algoritmo opere com múltiplas classes, o uso de um procedimento decomposicional pode reduzir a complexidade computacional envolvida na solução do problema total, uma vez que dividi-se o problema em subtarefas mais simples



Técnicas Decomposicionais

- Empregam dois passos
 - Fase de Decomposição
 - Ocorre antes do aprendizado
 - Consiste em obter múltiplos subproblemas binários, os quais são considerados na obtenção de um conjunto de modelos
 - Fase de Reconstrução (comentada em conjunto com a anterior)
 - Ocorre depois da predição
 - Refere-se à forma como as saídas dos classificadores binários são combinadas na determinação da classe de um exemplo



Fase de Decomposição

- Um-contra-todos (OAA – one-against-all)
- Todos-contra-todos (um-contra-um) (OAO – one-against-one)
- Códigos de Correção de Erros de Saída (ECOC – error-correcting output codes)
- Decomposições Hierárquicas

<https://scikit-learn.org/stable/modules/svm.html#svm>

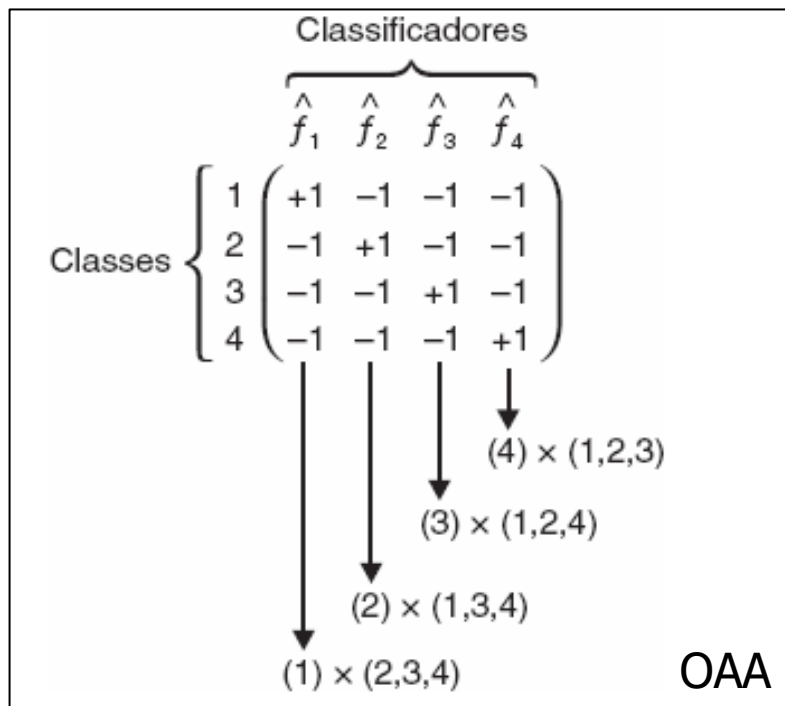
<https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC>

<https://scikit-learn.org/stable/api/sklearn.multiclass.html>

<https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/>

Fase de Decomposição

Matriz de códigos



Matrizes de diferentes decomposições para um problema com quatro classes

$$\begin{pmatrix} +1 & +1 & +1 & +1 & +1 & +1 \\ -1 & -1 & -1 & -1 & +1 & +1 \\ -1 & -1 & +1 & +1 & -1 & -1 \\ -1 & +1 & -1 & +1 & -1 & +1 \end{pmatrix}$$

(b) ECOC

$$\begin{pmatrix} +1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 \\ -1 & -1 & +1 & -1 \\ -1 & -1 & -1 & +1 \end{pmatrix}$$

(c) OAA

$$\begin{pmatrix} +1 & +1 & +1 & 0 & 0 & 0 \\ -1 & 0 & 0 & +1 & +1 & 0 \\ 0 & -1 & 0 & -1 & 0 & +1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{pmatrix}$$

(d) OAO



Um-contratodos

$$\begin{pmatrix} +1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 \\ -1 & -1 & +1 & -1 \\ -1 & -1 & -1 & +1 \end{pmatrix}$$

(c) OAA

- Dado um problema com k classes, k classificadores binários são gerados
- Cada um desses classificadores é treinado de maneira a distinguir uma classe c das demais
- Apresenta desvantagens quando a proporção de exemplos de uma classe é muito pequena em relação à do conjunto formado pelos dados das outras classes
 - Esse tipo de desbalanceamento pode dificultar a indução de um classificador que apresente bom desempenho no reconhecimento da classe considerada



Um-contratodos

- A test instance is classified by combining the predictions made by the binary classifiers
- A voting scheme is typically employed to combine the predictions, where the class that receives the highest number of votes is assigned to the test instance
 - If an instance is classified as negative, then all classes except for the positive class receive a vote
 - This may lead to ties among the different classes
 - Another possibility is to transform the outputs of the binary classifiers into probability estimates and then assign the test instance to the class that has the highest probability



Todos-contra-todos

$$\begin{pmatrix} +1 & +1 & +1 & 0 & 0 & 0 \\ -1 & 0 & 0 & +1 & +1 & 0 \\ 0 & -1 & 0 & -1 & 0 & +1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{pmatrix}$$

(d) OAO

- Dadas k classes, $\frac{k(k-1)}{2}$ classificadores binários são gerados
 - Realiza-se uma combinação par-a-par
 - Cada classificador é responsável por diferenciar um par de classes (i, j) , em que $i \neq j$
 - Embora o número de classificadores gerados seja da ordem de k^2 , o treinamento de cada um deles envolve dados de apenas duas classes
 - Mesmo com um número elevado de classes, o tempo total despendido na geração dos classificadores geralmente não é grande



Todos-contra-todos

$$\begin{pmatrix} +1 & +1 & +1 & 0 & 0 & 0 \\ -1 & 0 & 0 & +1 & +1 & 0 \\ 0 & -1 & 0 & -1 & 0 & +1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{pmatrix}$$

(d) OAO

- Dadas k classes, $\frac{k(k-1)}{2}$ classificadores binários são gerados
 - Um problema é que a resposta de um classificador, para um par de classes (i, j) , não fornece informação quando o exemplo não pertence às classes i ou j
 - Suponha um problema com 10 classes
 - Dos 45 problemas de decisão binária, somente 9 podem classificar corretamente o exemplo teste
 - Todos os outros 36 irão classificar o exemplo de maneira errada
 - Portanto, neste caso, $(k-1)(k-2)/2$ irão classificar incorretamente qualquer exemplo, e somente $k-1$ podem prover a classificação correta



Todos-contra-todos

- A test instance is classified by combining the predictions made by the binary classifiers
- A voting scheme is typically employed to combine the predictions, where the class that receives the highest number of votes is assigned to the test instance



Códigos de Correção de Erros de Saída

- A potential problem with the previous two approaches is that they may be sensitive to binary classification errors
 - If at least of one of the binary classifiers makes a mistake in its prediction, then the classifier may end up declaring a tie between classes or making a wrong prediction



Códigos de Correção de Erros de Saída

- The error-correcting output coding (ECOC) method provides a more robust way for handling multiclass problems
- The method is inspired by an information-theoretic approach for sending messages across noisy channels
 - The idea is to add redundancy into the transmitted message by means of a codeword, so that the receiver may detect errors in the received message and perhaps recover the original message if the number of errors is small



Códigos de Correção de Erros de Saída

Class	Codeword						
y_1	1	1	1	1	1	1	1
y_2	0	0	0	0	1	1	1
y_3	0	0	1	1	0	0	1
y_4	0	1	0	1	0	1	0

- Each class is represented by a unique bit string of length n known as its codeword
- We then train n binary classifiers to predict each bit of the codeword string
- The predicted class of a test instance is given by the codeword whose Hamming distance is closest to the codeword produced by the binary classifiers
 - The Hamming distance between a pair of bit strings is given by the number of bits that differ



Códigos de Correção de Erros de Saída

Class	Codeword						
y_1	1	1	1	1	1	1	1
y_2	0	0	0	0	1	1	1
y_3	0	0	1	1	0	0	1
y_4	0	1	0	1	0	1	0

- Consider a multiclass problem where the 4 classes were encoded using seven bit codewords (see table)
- Each bit of the codeword is used to train a binary classifier
- If a test instance is classified as (0,1,1,1,1,1,1) by the binary classifiers, then the Hamming distance between the codeword and y_1 is 1, while the Hamming distance to the remaining classes is 3
- The test instance is therefore classified as y_1

<https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OutputCodeClassifier.html>

<https://scikit-learn.org/stable/api/sklearn.multiclass.html>

$$\begin{pmatrix} +1 & +1 & +1 & +1 & +1 & +1 \\ -1 & -1 & -1 & -1 & +1 & +1 \\ -1 & -1 & +1 & +1 & -1 & -1 \\ -1 & +1 & -1 & +1 & -1 & +1 \end{pmatrix}$$

(b) ECOC



Códigos de Correção de Erros de Saída

- An interesting property of ECOC is that if the minimum Hamming distance between any pair of codewords is d , then any $\lfloor (d-1)/2 \rfloor$ errors in the output code can be corrected using its nearest codeword
 - In the example, the classifier may tolerate errors made by one of the seven binary classifiers
 - If there is more than one classifier that makes a mistake, then the classifier may not be able to compensate for the error
- An important issue is how to design the appropriate set of codewords for different classes



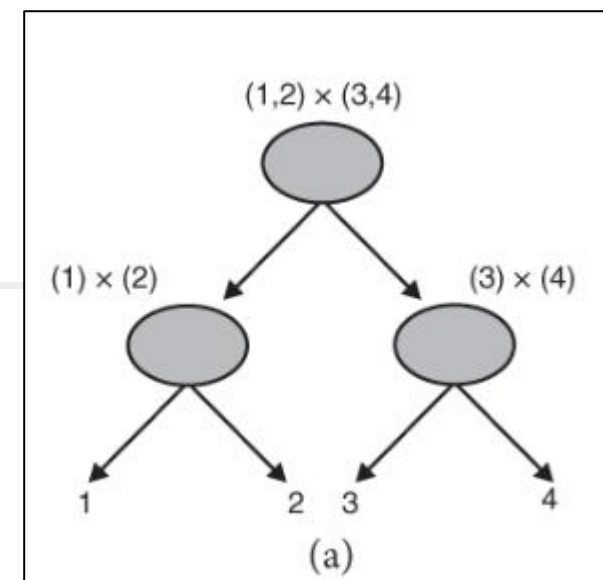
Decomposições Hierárquicas

- A introdução de uma hierarquia em uma aplicação multiclasse pode reduzir a complexidade de sua solução
- A ideia é realizar, inicialmente, discriminações mais gerais, as quais são refinadas sucessivamente até a obtenção da classificação final

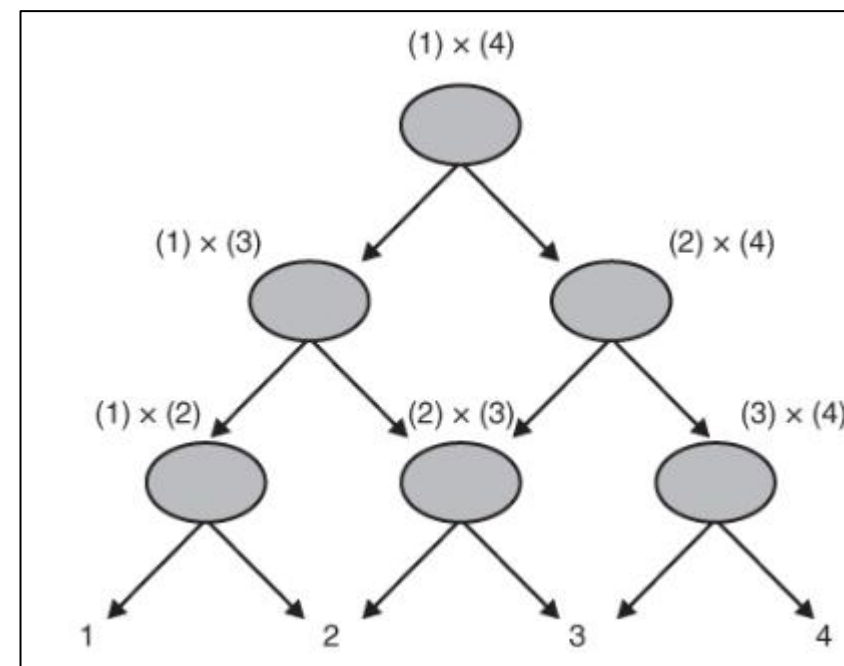
Decomposições Hierárquicas

- Veja os exemplos
- Em ambas as estruturas, cada nó interno corresponde a um classificador binário que distingue dois subconjuntos de classes, enquanto os nós terminais, denominados folhas, representam as classes individuais

Árvore direcionada binária



Grafo direcionado acíclico



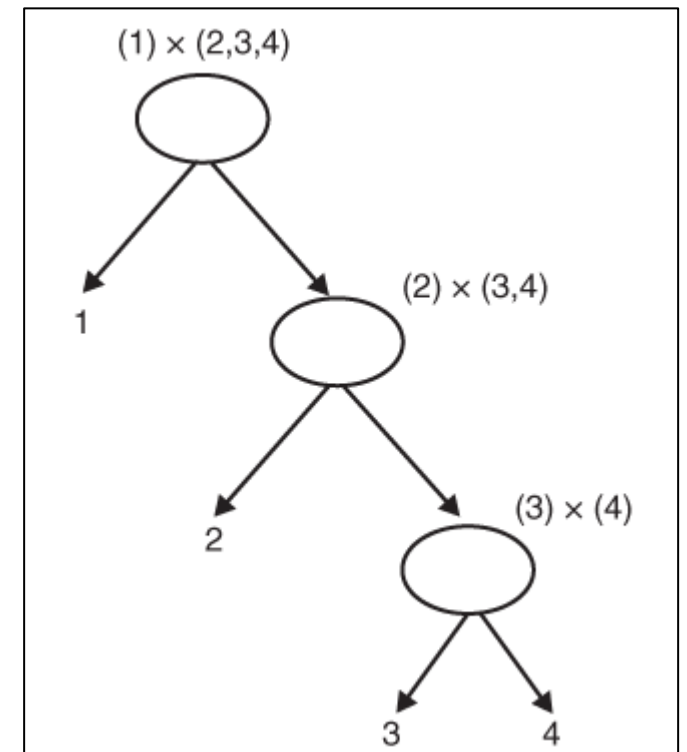
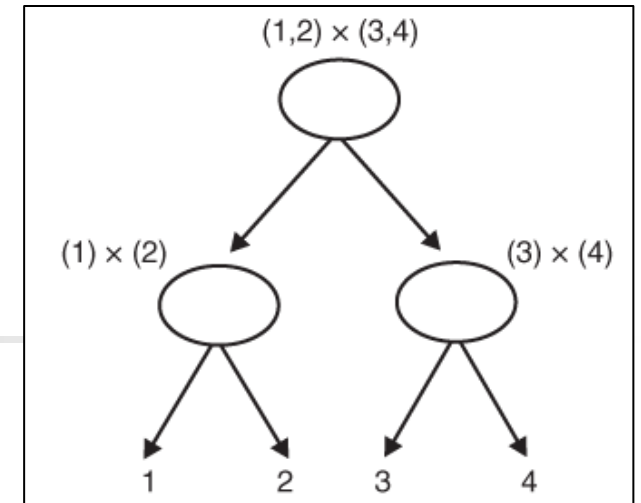


Árvores Direcionadas Binárias

- As árvores possuem $k-1$ classificadores binários e, portanto, envolvem o treinamento de $k-1$ preditores
- Os nós de níveis inferiores envolvem menos classes e, portanto, menos dados de treinamento para os classificadores binários correspondentes

Árvores Direcionadas Binárias

- Para um problema com $k \geq 3$ classes, existem $\prod_{i=3}^k 2i - 3$ estruturas de árvores distintas
- Assim, a estrutura da árvore, ou seja, a maneira que os classificadores são dispostos na árvore e onde eles se encontram, influencia o seu resultado



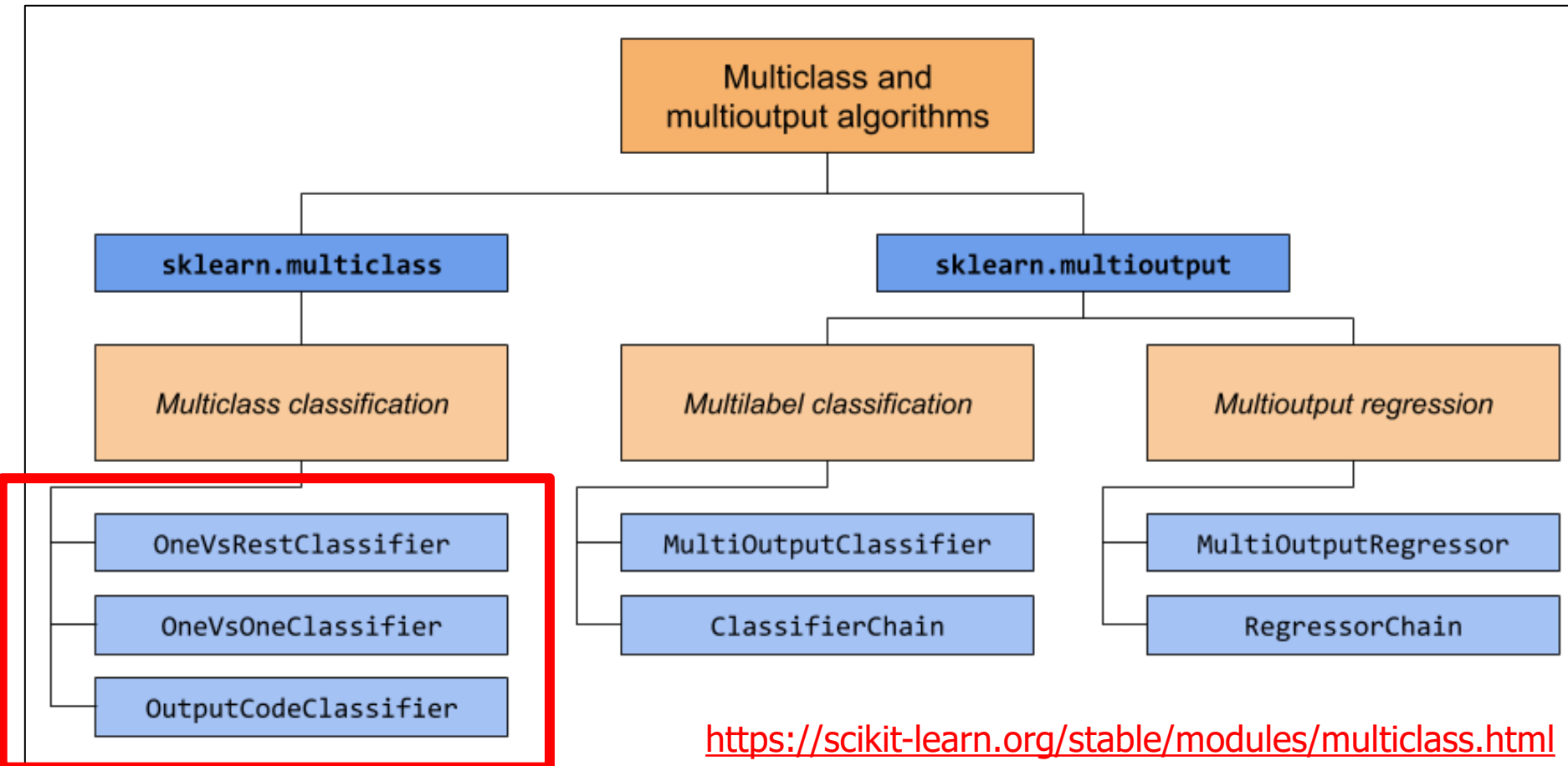


Árvores Direcionadas Binárias

- Os trabalhos se diferenciam então no processo de obtenção das partições binárias das classes em cada nó da árvore e, conseqüentemente, na determinação de sua estrutura
- Algumas soluções:
 - Agrupar as classes com centros mais similares em cada nível da hierarquia com o algoritmo k-médias, com $k = 2$
 - Realizar um agrupamento hierárquico das classes de acordo com sua similaridade
 - Diferentes critérios podem ser utilizados para medir a similaridade das classes, tal como a distância entre seus centroides, a sua separabilidade, entre outros

SKLearn

	Number of targets	Target cardinality	Valid type of target
Multiclass classification	1	>2	'multiclass'
Multilabel classification	>1	2 (0 or 1)	'multilabel-indicator'
Multiclass-multioutput classification	>1	>2	'multiclass-multioutput'
Multioutput regression	>1	Continuous	'continuous-multioutput'



<https://scikit-learn.org/stable/modules/multiclass.html>