Clasificación Multi-Etiqueta

Eduardo Morales

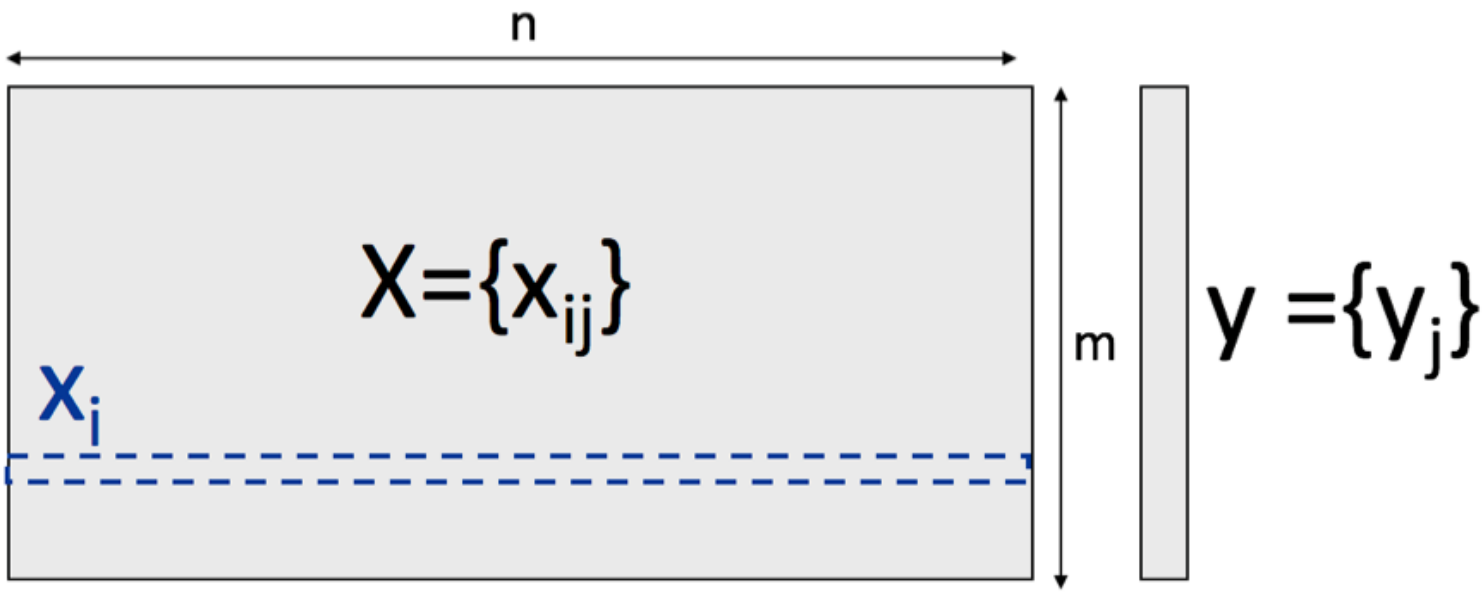
INAOE

Contenido

Clasificación Multi-Clase

- Los algoritmos de aprendizaje que hemos visto hasta ahora, inducen un modelo, usando ejemplos de entrenamiento, para predecir el valor de una clase.
  Dados:
  $$D = (\vec{x}_i, y_i)_{1...N}, \vec{x}_i \in \mathcal{R}^d; y_i \in C$$
  Encontrar:
  $$f : \mathcal{R}^d \rightarrow C$$
- Clasificación binaria:
  $$f : \mathcal{R}^d \rightarrow \{-1, 1\}$$
- Clasificación multiclase:
  $$f : \mathcal{R}^d \rightarrow \{C_1, \ldots, C_k\}$$

Clasificación Multi-Clase



Clasificación Multi-Etiqueta

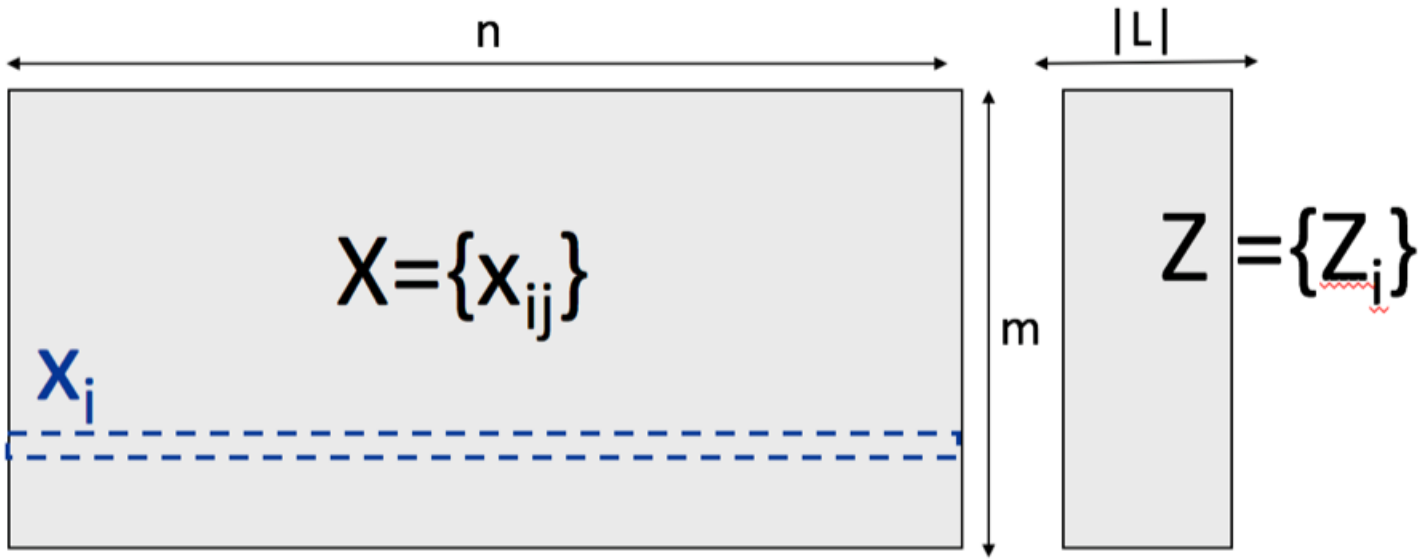
- En clasificación multi-etiqueta lo que queremos es predecir un conjunto de valores
- Dado:

$$D = (\vec{x}_i, Z_i)_{1...N}, \vec{x}_i \in \mathcal{R}^d; Z_i \subseteq L$$

- Encontrar:

$$f : \mathcal{R}^d \rightarrow Z, Z \subseteq L = \{1, \ldots, K\}$$

Clasificación Multi-Etiqueta



Algunos Ejemplos

Data type	Application	Resource	Labels Description (Examples)	References
text	categorization	news article	Reuters topics (agriculture, fishing)	[16]
		web page	Yahoo! directory (health, science)	[17]
		patent	WIPO (paper-making, fibreboard)	[18, 19]
		email	R&D activities (delegation)	[20]
		legal document	Eurovoc (software, copyright)	[21]
		medical report	MeSH (disorders, therapies)	[22]
		radiology report	ICD-9-CM (diseases, injuries)	[23]
		research article	Heart conditions (myocarditis)	[24]
		research article	ACM classification (algorithms)	[25]
		bookmark	Bibsonomy tags (sports, science)	[26]
		reference	Bibsonomy tags (ai, kdd)	[26]
		adjectives	semantics (object-related)	[27]
image	semantic annotation	pictures	concepts (trees, sunset)	[1, 2, 3]
video	semantic annotation	news clip	concepts (crowd, desert)	[4]
audio	noise detection	sound clip	type (speech, noise)	[28]
	emotion detection	music clip	emotions (relaxing-calm)	[11, 14]
structured	functional genomics	gene	functions (energy, metabolism)	[7, 6, 8]
	proteomics	protein	enzyme classes (ligases)	[19]
	directed marketing	person	product categories	[15]

Clasificación Multi-Etiqueta

Existen dos enfoques generales para clasificación multi-etiqueta:

1. Transformación: Transforman el problema en varios problemas de clasificación multiclase
2. Adaptación: Adaptan algoritmos para lidear con conjuntos de clases

Métodos de Transformación

- Copia: Reemplaza cada ejemplo multi-etiqueta ($\vec{x}_i$, Y_i) en $|Y_i|$ ejemplos de una sola etiqueta
- Directamente o de forma pesada ($\frac{1}{|Y_i|}$)

Example	Attributes	Label set
1	x_1	$\{\lambda_1, \lambda_4\}$
2	x_2	$\{\lambda_3, \lambda_4\}$
3	x_3	$\{\lambda_1\}$
4	x_4	$\{\lambda_2, \lambda_3, \lambda_4\}$

Original ML problem

Ex.	Label
1a	λ_1
1b	λ_4
2a	λ_3
2b	λ_4
3	λ_1
4a	λ_2
4b	λ_3
4c	λ_4

Transformed ML problem (unweighted)

Ex.	Label	Weight
1a	λ_1	0.50
1b	λ_4	0.50
2a	λ_3	0.50
2b	λ_4	0.50
3	λ_1	1.00
4a	λ_2	0.33
4b	λ_3	0.33
4c	λ_4	0.33

Transformed ML problem (weighted)

Métodos de Transformación

- Copia seleccionada: Copia y selecciona una de las clases
- La más frecuente (*max*), menos frecuente (*min*), en forma aleatoria (*random*), ignorando los ejemplos multi-etiqueta (*ignore*)

Example	Attributes	Label set
1	$\mathbf{x_1}$	$\{\lambda_1, \lambda_4\}$
2	$\mathbf{x_2}$	$\{\lambda_3, \lambda_4\}$
3	$\mathbf{x_3}$	$\{\lambda_1\}$
4	$\mathbf{x_4}$	$\{\lambda_2, \lambda_3, \lambda_4\}$

Original ML problem

Max

Ex.	Label
1	λ_4
2	λ_4
3	λ_1
4	λ_4

Min

Ex.	Label
1	λ_1
2	λ_3
3	λ_1
4	λ_2

Rand

Ex.	Label
1	λ_1
2	λ_4
3	λ_1
4	λ_3

Transformed ML problem

Ex.	Label
3	λ_1

Ignore approach

Métodos de Transformación

- Conjunto potencia (*powerset*): Simple y muy usado, en donde considera cada subconjunto diferente de clases como una nueva clase de un nuevo problema de clasificación multi-clase

Example	Attributes	Label set
1	$\mathbf{x}_1$	$\{\lambda_1, \lambda_4\}$
2	$\mathbf{x}_2$	$\{\lambda_3, \lambda_4\}$
3	$\mathbf{x}_3$	$\{\lambda_1\}$
4	$\mathbf{x}_4$	$\{\lambda_2, \lambda_3, \lambda_4\}$

Original ML problem

Ex.	Label
1	$\lambda_{1,4}$
2	$\lambda_{3,4}$
3	λ_1
4	$\lambda_{2,3,4}$

Transformed ML problem

Label Powerset

- ¿Cómo clasificamos? Si el clasificador nos da una probabilidad de salida, las podemos repartir en las clases y sumarlas

c	$p(c\|\mathbf{x})$	λ_1	λ_2	λ_3	λ_4
$\lambda_{1,4}$	0.7	1	0	0	1
$\lambda_{3,4}$	0.2	0	0	1	1
λ_1	0.1	1	0	0	0
$\lambda_{2,3,4}$	0.0	0	1	1	1
	$\sum_c p(c\|\mathbf{x})\lambda_j$	0.8	0.0	0.2	0.9

RA*k*EL

- Random k-label sets construye un ensemble de “Label Powersets”, cada clasificador construido con un subconjunto pequeño de clases
- Ventajas: Mantiene las correlaciones entre las clases y mantiene el número de clases reducido
- De nuevo ordena las salidas de los clasificadores

Binary Relevance

- Es un método popular que genera n clasificadores binarios, uno por cada valor (i) de las clases
- Cada clasificador se entrena con todos los datos originales, considerando ejemplos positivos a los que tienen la clase i, y negativos el resto ($j \neq i$), y lo hace para todas las clases

Example	Attributes	Label set
1	$\mathbf{x_1}$	$\{\lambda_1, \lambda_4\}$
2	$\mathbf{x_2}$	$\{\lambda_3, \lambda_4\}$
3	$\mathbf{x_3}$	$\{\lambda_1\}$
4	$\mathbf{x_4}$	$\{\lambda_2, \lambda_3, \lambda_4\}$

Original ML problem

Ex.	Label
1	λ_1
2	$\neg\lambda_1$
3	λ_1
4	$\neg\lambda_1$

Ex.	Label
1	$\neg\lambda_2$
2	$\neg\lambda_2$
3	$\neg\lambda_2$
4	λ_2

Ex.	Label
1	$\neg\lambda_3$
2	λ_3
3	$\neg\lambda_3$
4	λ_3

Ex.	Label
1	λ_4
2	λ_4
3	$\neg\lambda_4$
4	λ_4

Data sets generated by BR

Ranking by Pairwise Comparison

- Transforma el problema multiclase en $\frac{q(q-1)}{2}$ conjunto de clases binarias (uno para cada par de clases)
- Cada conjunto de datos contiene ejemplos de alguna de las clases, pero no de las dos
- Dada una nueva instancia se corre en todos los clasificadores y se cuentan los votos recibidos para cada clase

Original ML problem

Example	Attributes	Label set
1	x_1	$\{\lambda_1, \lambda_4\}$
2	x_2	$\{\lambda_3, \lambda_4\}$
3	x_3	$\{\lambda_1\}$
4	x_4	$\{\lambda_2, \lambda_3, \lambda_4\}$

λ_1 vs. λ_2

Ex.	Label
1	$\lambda_{1,\neg 2}$
3	$\lambda_{1,\neg 2}$
4	$\lambda_{\neg 1,2}$

(a)

λ_1 vs. λ_3

Ex.	Label
1	$\lambda_{1,\neg 3}$
2	$\lambda_{\neg 1,3}$
3	$\lambda_{1,\neg 3}$
4	$\lambda_{\neg 1,3}$

(b)

λ_1 vs. λ_4

Ex.	Label
2	$\lambda_{\neg 1,4}$
3	$\lambda_{1,\neg 4}$
4	$\lambda_{\neg 1,4}$

(c)

λ_2 vs. λ_3

Ex.	Label
2	$\lambda_{\neg 2,3}$

(d)

λ_2 vs. λ_4

Ex.	Label
1	$\lambda_{\neg 2,4}$
2	$\lambda_{\neg 2,4}$

(e)

λ_3 vs. λ_4

Ex.	Label
1	$\lambda_{\neg 3,4}$

(f)

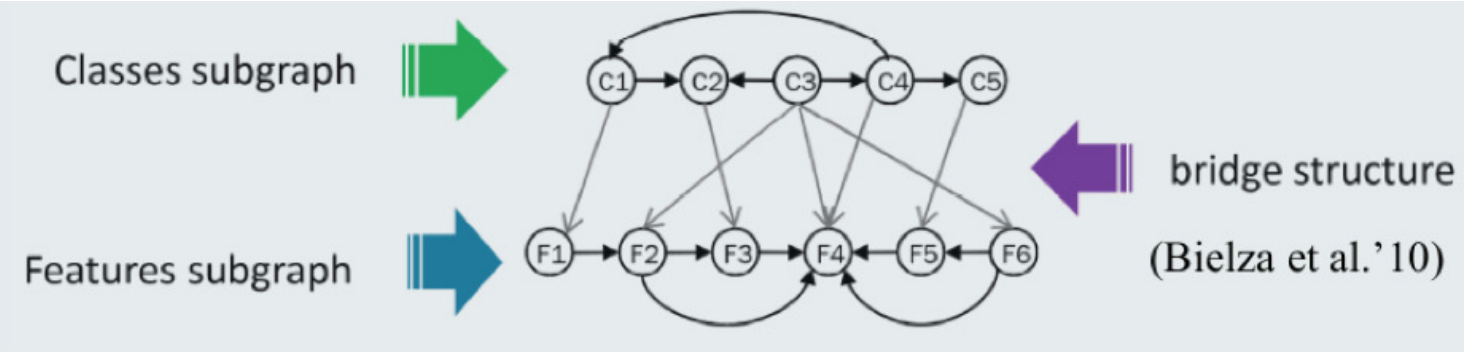
Adaptación de Algoritmos

Se han realizado adaptaciones a varios algoritmos para poder lidear con ejemplos multi-etiquetas:

- Árboles de decisión (permite a las hojas tener más de una clase y modifica la medida de entropía)
- Boosting (Adaboost): Evalua considerando múltiples clases
- Campos aleatorios de Markov: Lo modifican para considerar co-ocurrencia de etiquetas
- Redes neuronales: Adaptan *back-propagation* para considerar multi-etiquetas
- SVM: Generan *n* clasificadores binarios, sus predicciones se usan como atributos para nuevos clasificadores binarios
- kNN: Encuentra vecinos más cercanos tomando en cuenta la frecuencia de las clases

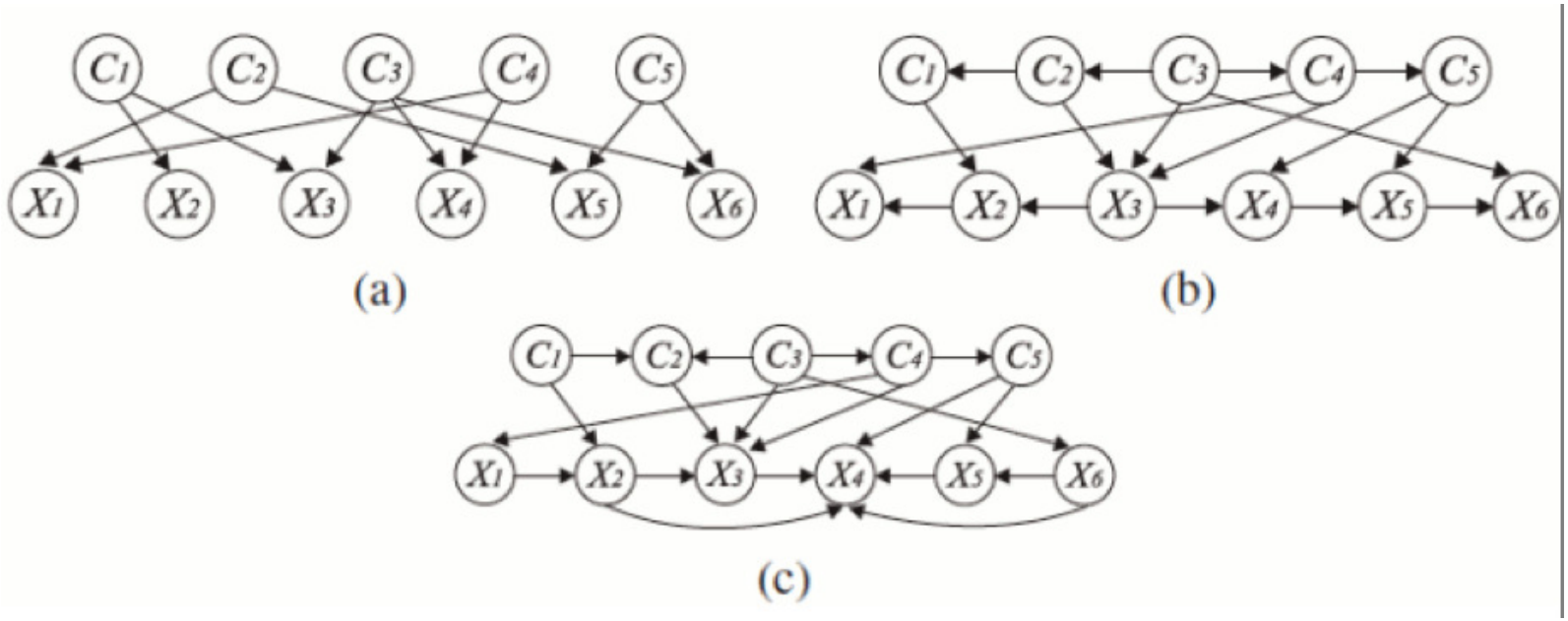
Multi-Dimensional Bayesian Classifiers

- Una red de clasificación bayesiana multi-dimensional es una red bayesiana con una topología restringida



- Se pueden crear diferentes estructuras y estrategias de aprendizaje para cada sub-grafo.

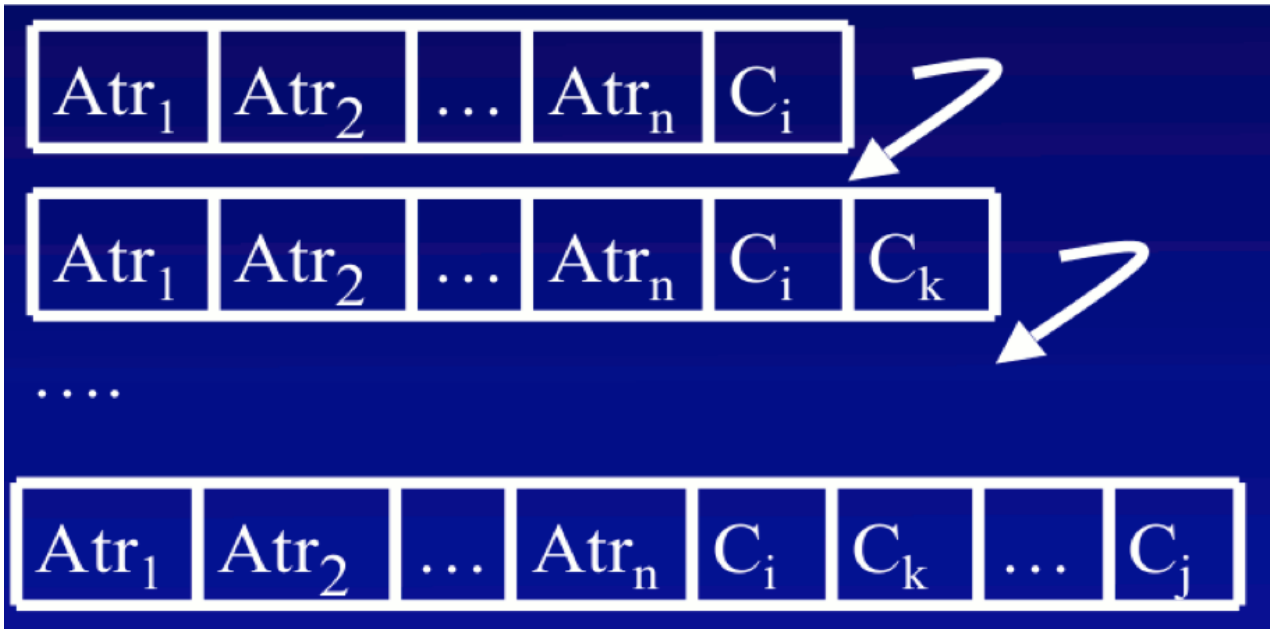
Multi-Dimensional Bayesian Classifiers



- Tree-augmented MBCs (van der Gaag, 2006)
- Poly-tree structures (de Waal, 2007; Zaragoza, 2011)
- Greedy approaches for filter, wrapper and hybrid (Bielza, 2010)
- Based on Markov blanquets (Borchani, 2011)

Chain Classifiers

- La idea de los clasificadores en cadena es por un lado tener clasificadores simples (binarios) y considerar las dependencias entre las clases



- Se crea una “cadena” de clasificadores, en donde los atributos de clasificadores binarios se aumentan con las predicciones de los clasificadores anteriores en la cadena

Chain Classifiers

(a) BR's transformation

h :	x →	y
h_1:	[0,1,0,1,0,0,1,1,0]	1
h_2:	[0,1,0,1,0,0,1,1,0]	0
h_3:	[0,1,0,1,0,0,1,1,0]	0
h_4:	[0,1,0,1,0,0,1,1,0]	1
h_5:	[0,1,0,1,0,0,1,1,0]	0

(b) CC's transformation

h :	x′ →	y
h_1:	[0,1,0,1,0,0,1,1,0]	1
h_2:	[0,1,0,1,0,0,1,1,0,1]	0
h_3:	[0,1,0,1,0,0,1,1,0,1,0]	0
h_4:	[0,1,0,1,0,0,1,1,0,1,0,0]	1
h_5:	[0,1,0,1,0,0,1,1,0,1,0,0,1]	0

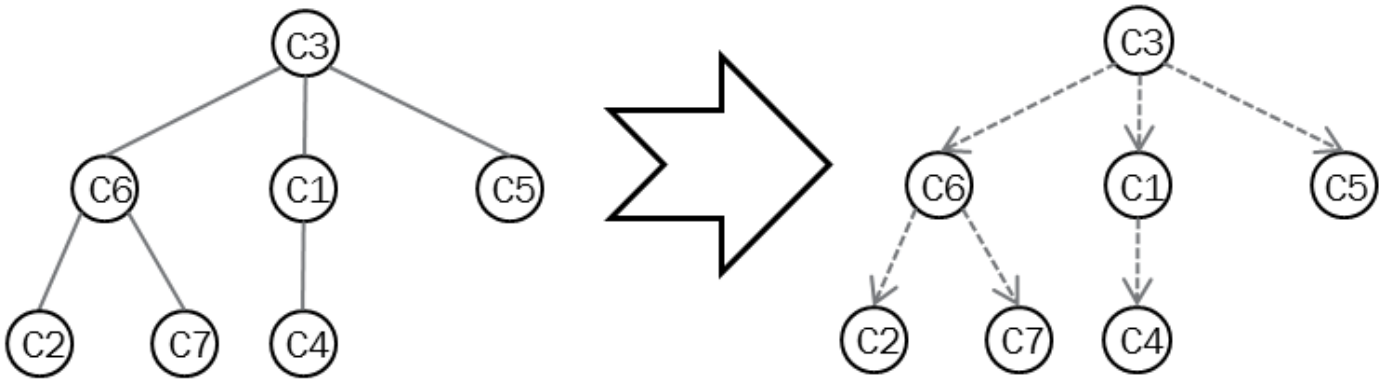
- El orden de la cadena es relevante si existen dependencias entre las clases
- Como no se sabe cuál debe de ser el orden se crea un ensamble con muchos ordenes de clases generados aleatoriamente
- Se usa un voto simple de las clases predichas por todos los ensambles usando un umbral

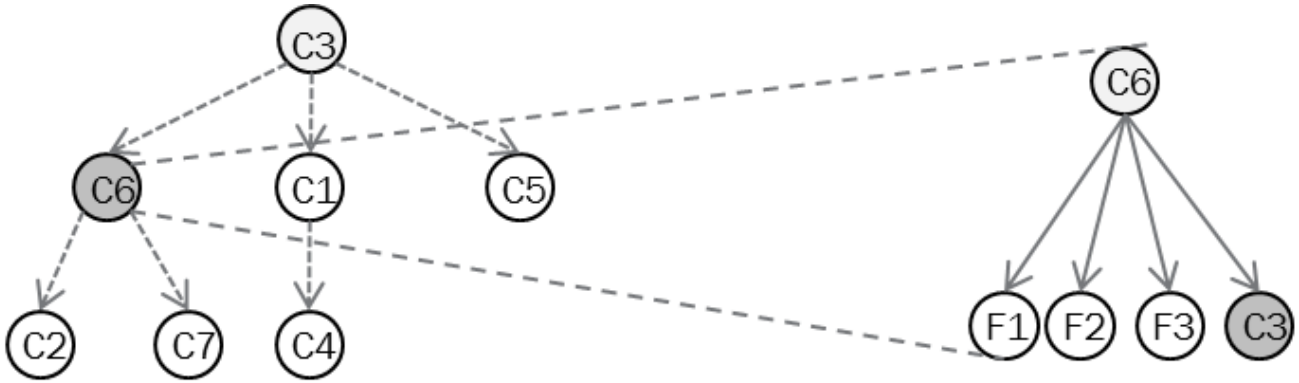
Bayesian Chain Classifier (BCC)[1]

- La idea es determinar un orden con base en dependecias y limitar el número de atributos usados para los clasificadores en la cadena
- Pasos:
  1. Obtener una estructura de dependencias (red bayesiana) para las clases
  2. Crear una clasificador en cadena tomando en cuenta ésta estructura (sólo incorpora los padres de cada clase como atributos adicionales)

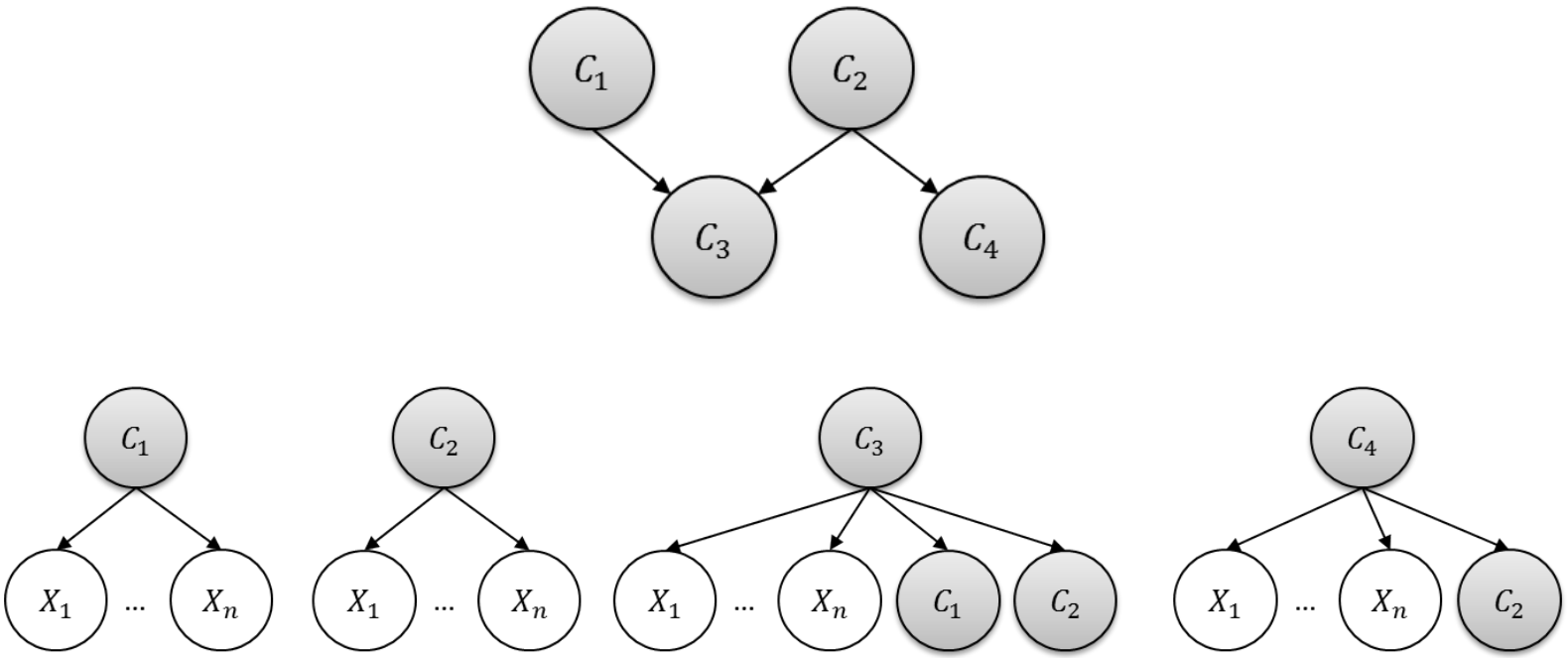
[1] J.H. Zaragoza, L.E. Sucar, E.F. Morales, C. Bielza, P. Larrañaga (2011). Bayesian Chain Classifiers for Multidimensional Classification. *Proc. of the International Joint Conference on Artificial Intelligence* (IJCAI-2011), pp. 2192-2197.

Bayesian Chain Classifier





Bayesian Chain Classifier



Jerárquicos

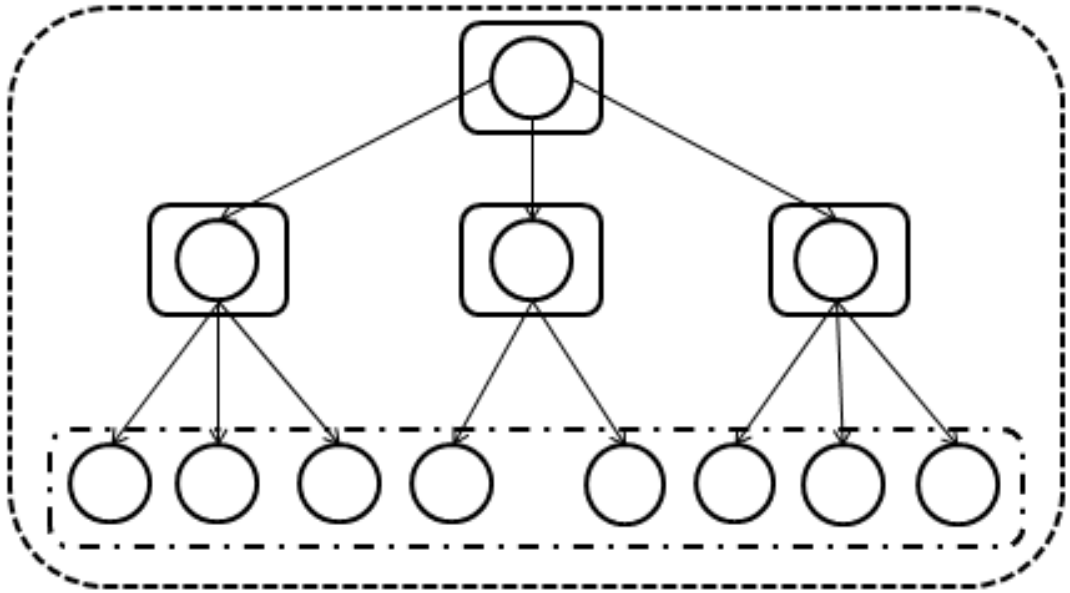
- A veces las clases están organizadas en una jerarquía
- Algunos algoritmos aprovechan esa información adicional (dependencias jerárquicas conocidas)
- Clasificación por:
  1. Tipo de jerarquía: (i) Árbol o (ii) DAG
  2. Profundidad de clasificación: (i) *mandatory leaf-node prediction* o (ii) *non mandatory leaf-node prediction*
  3. Esquema de exploración: (i) Local o (ii) Global

Local o *Top-Down*

- El entrenamiento se puede hacer de difrentes formas:
  1. Clasificación binaria en cada nodo (excepto el nodo raíz)
  2. Usar un clasificador multi-clase en cada nodo padre
  3. Usar un clasificador multi-clase por nivel
  4. Usar un clasificador multi-clase sólo para las hojas
- Normalmente se usa el mismo clasificador en toda la jerarquía
- *Inconsistency problem*: Un error en algún nivel de la jerarquía se propaga a todos sus descendientes
- El problema es porque los clasificadores se consideran indpendientes entre sí

Tipos de Clasificadores

Tipos: Flat, Global, Local



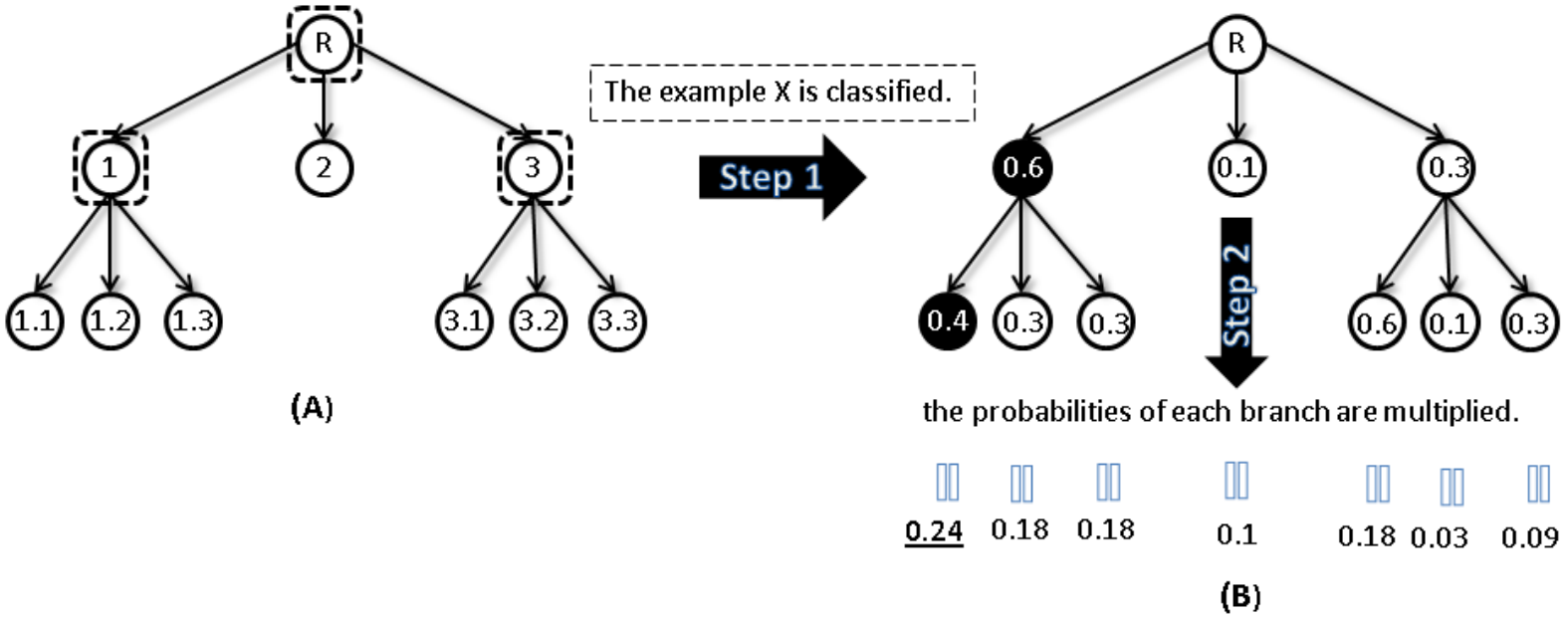
Jerárquico (MHC)[2]

- Aprende un clasificador multiclase para cada nodo padre
- Con un nueva instancia usa todos los clasificadores para predecir las clases en todos los nodos y combina los resultados de todas los caminos
- Regresa el camino con probabilidad más alta
- Se puede decidir parar la clasificación hasta cierto nivel (*non mandatory leaf-node prediction*)

[2] J. Hernández, L.E. Sucar, E.F. Morales (2014). Multidimensional hierarchical classification. *Expert Systems with Applications* 41 (17): 7671-7677.

Jerárquico (MHC)

La combinación aquí es multiplicando, pero se pueden pensar en otras formas



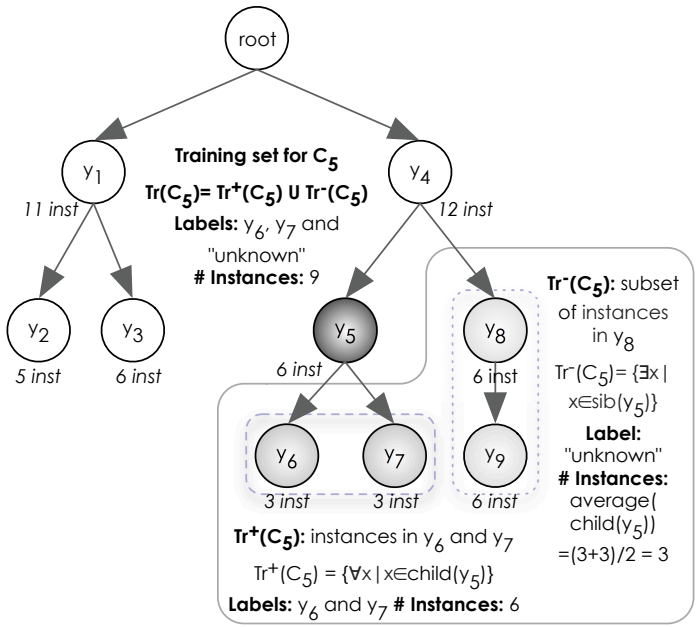
Jerárquico (HMC)[3]

- Usar ideas de clasificadores multi-etiqueta
- Aprovechar propiedades: Un ejemplo que pertenece a una clase, también pertenence a todas sus super-clases (y un negativo se propaga a todas sus sub-clases)
- Incluir las predicciones de las clases de los padres en los atributos de los hijos (*chain classifier*)

[3] M. Ramírez-Corona, L.E. Sucar, E.F. Morales (2016). Hierarchical multilabel classification based on path evaluation, *International Journal of Approximate Reasoning* 68: 179-193.

Jerárquico (HMC)

- Usar ejemplos negativos de nodos cercanos para balancear las clases



Jerárquico (HMC)

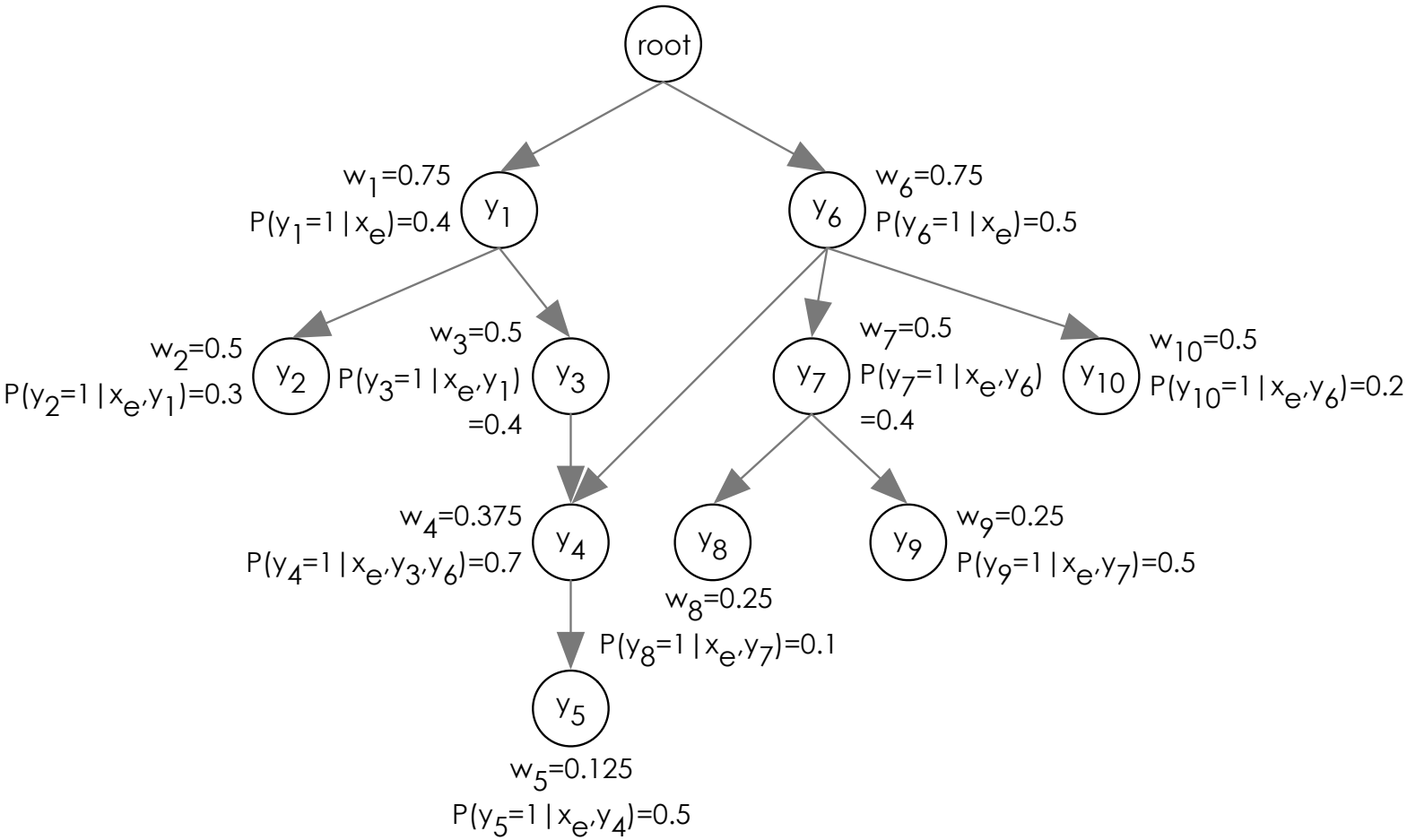
- Merging rule: considera el nivel en el árbol y predicción de cada nodo:

$$level(y_i) = 1 + \frac{\sum_{j=1}^{m} level(pa(y_i)_j)}{|pa(y_i)|}$$

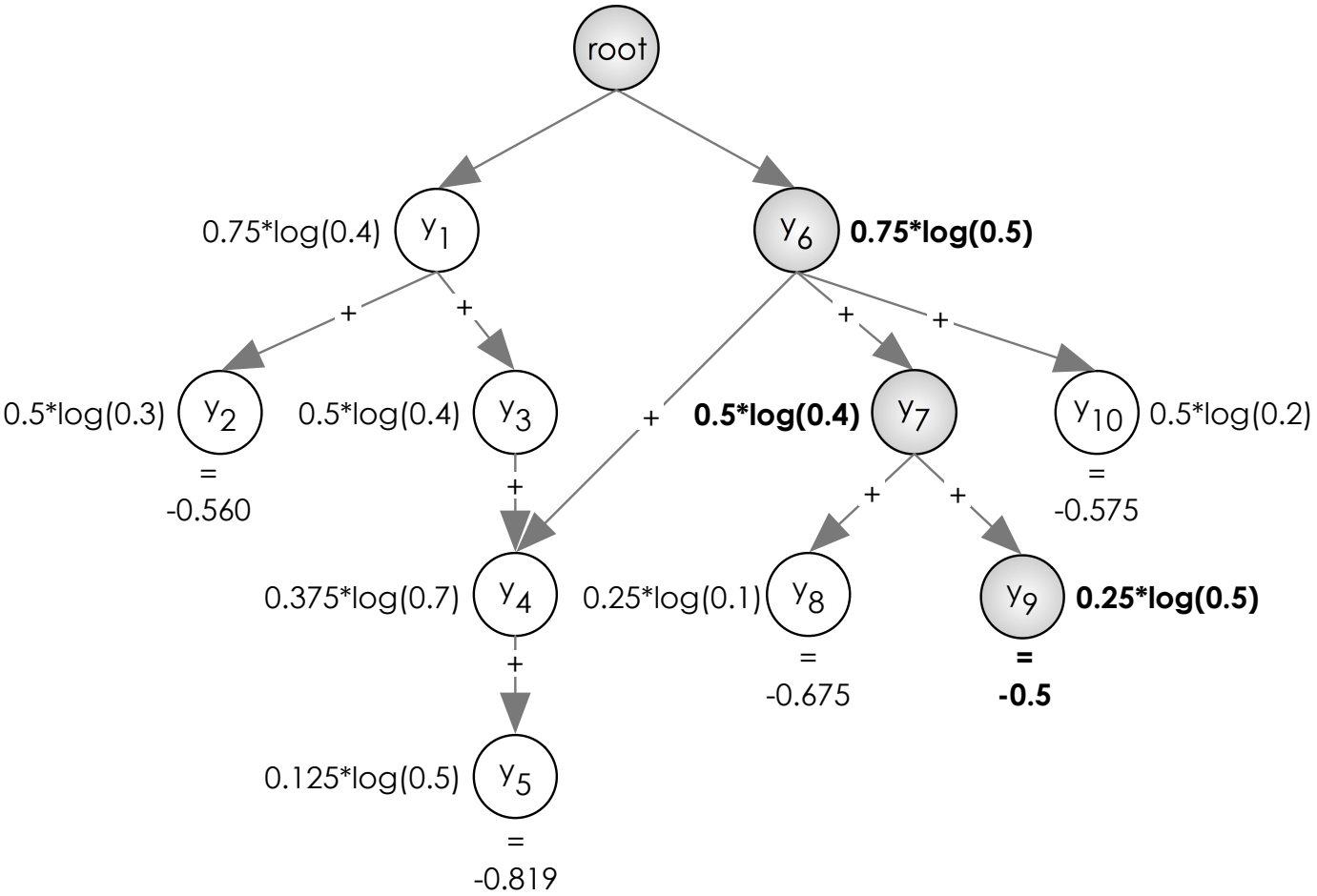
$$w(y_i) = 1 - \frac{level(y_i)}{maxLevel + 1}$$

$$score = \sum_{i=1}^{p} w(y_i) * log(P(y_i|x_e, pa(y_i)))$$

Jerárquico (HMC)



Jerárquico (HMC)



Medidas de Evaluación

Para los clasificadores multi-etiqueta se han propuesto diferentes medidas de evaluación:

- *Mean accuracy* (por clase para d clases):

$$overline{Acc}_d = \frac{1}{d}\sum_{j=1}^{d} Acc_j = \frac{1}{d}\sum_{j=1}^{d}\frac{1}{N}\sum_{i=1}^{N}\delta(c'_{ij}, c_{ij})$$

  donde $\delta(c'_{ij}, c_{ij}) = 1$ si $c'_{ij} = c_{ij}$ and 0 en otro caso

- *Global accuracy* (por ejemplo):

$$Acc = \frac{1}{N}\sum_{i=1}^{N}\delta(\mathbf{c'_i}, \mathbf{c_i})$$

  donde $\mathbf{c_i}$ es el vector d-dimensional de las clases y $\delta(\mathbf{c'_i}, \mathbf{c_i}) = 1$ si $\mathbf{c'_i} = \mathbf{c_i}$ y 0 en otro caso

Medidas de Evaluación

- *Multilabel accuracy* (también llamado de Jaccard):

$$\text{ML-Acc} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\mathbf{c}_i \wedge \mathbf{c}'_i|}{|\mathbf{c}_i \vee \mathbf{c}'_i|}$$

- *F-measure*:

$$\text{F-measure} = \frac{1}{d}\sum_{j=1}^{d}\frac{2p_j r_j}{(p_j + r_j)}$$

Medidas de Evaluación Jerárquicas

- Exact-Match:

$$ExactMatch = \frac{1}{N}\sum_{i=1}^{N} 1_{Y_i=\hat{Y}_i}$$

- Accuracy:

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N} \frac{\left|Y_i \cap \hat{Y}_i\right|}{\left|Y_i \cup \hat{Y}_i\right|}$$

- Hamming-Loss and Hamming-Accuracy:

$$HammingLoss = \frac{1}{N|L|}\sum_{i=1}^{N} \left|Y_i \oplus \hat{Y}_i\right|$$

donde $\oplus$ es *or exclusivo*
Hamming accuracy (H-Accuracy) se define como:
$H - Accuracy = 1 - HammingLoss$.

Medidas de Evaluación Jerárquicas

- F1-measure: Para multi-etiqueta, refiniendo precisión y recuerdo

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

Donde: *Precision*: $\frac{|z_i \wedge \hat{z}_i|}{|\hat{z}_i|}$ y *Recall*: $\frac{|z_i \wedge \hat{z}_i|}{|z_i|}$

- F1-macro D: mide el desempeño promedio por instancia

$$F1 macro\, D = \frac{1}{N} \sum_{i=0}^{N} F1(z_i, \hat{z}_i)$$

- F1-macro L: mide el desempeño promedio por clase

$$F1 macro\, L = \frac{1}{|L|} \sum_{i=0}^{|L|} F1(z_i, \hat{z}_i)$$

Medidas de Evaluación Jerárquicas

- Gain-Loose Balance: premia nodes bien clasificados y castiga los mal, considerando el número de hermanos y la profundidad en la jerarquía

$$GLB = \frac{\sum_{i=0}^{n_p}(1-\frac{1}{N_i})(1-w_i)}{\sum_{i=0}^{n_t}(1-\frac{1}{N_i})(1-w_i)} - \left(\sum_{i=0}^{n_{fp}}\frac{1}{N_i}w_i + \sum_{i=0}^{n_{fn}}\frac{1}{N_i}w_i\right)$$

Conocimiento el posible valor máximo y mínimo se puede normalizar:

$$NGLB = \frac{(GLB - min)}{max - min}$$

Selección de Atributos

- A partir de los atributos originales selecciona un subconjunto de estos
- La meta es seleccionar el subconjunto S más pequeño de todos los atributos F, tal que $P(C|S) \approx P(C|F)$
- Ventajas esperadas:
  1. Mejorar el desempeño predictivo
  2. Construir modelos más eficientemente
  3. Mejorar entendimiento sobre los modelos generados

Selección de Atributos

En general, los algoritmos de selección de atributos se distinguen por su forma de evaluar atributos y los podemos clasificar en tres:

1. Filtros (*filters*): seleccionan/evalúan los atributos en forma independiente del algoritmo de aprendizaje
2. Wrappers: usan el desempeño de algún clasificador para determinar lo deseable de un subconjunto
3. Híbridos: usan una combinación de los dos criterios de evaluación en diferentes etapas del proceso de búsqueda.

Selección de Atributos en Problemas Multi-Etiqueta

- *Filter*: Transforman el problema en uno o más de una sóla clase y se usa algún algoritmo de selección de atributos tipo filtro. Después se sigue algún esquema de “agregación”
- *Wrapper*: se pueden aplicar directamente con algún algoritmo de clasificación multi-etiqueta
- También se han propuesto variantes de algoritmos de extracción de atributos como LDA

Meka

- MEKA: A Multi-Label Extension to WEKA
- Algunos de los algoritmos que tiene son:
  1. Binary Relevance
  2. Chain classifier
  3. metaBagging
  4. Bayesian chain classifier (BCC)
  5. RAkEL
  6. . . .
- http://meka.sourceforge.net