

Clasificación Multi-Etiqueta

Eduardo Morales, Hugo Jair Escalante

INAOE

Contenido

Introducción

Transformación

Adaptación

Evaluación

Selección

- 1 Introducción
- 2 Transformación
- 3 Adaptación
- 4 Evaluación
- 5 Selección

Clasificación Multi-Etiqueta

Introducción

Transformación

Adaptación

Evaluación

Selección

- Los algoritmos de aprendizaje que hemos visto hasta ahora, inducen un modelo, usando ejemplos de entrenamiento, para predecir el valor de una clase.

Dados:

$$D = (\vec{x}_i, y_i)_{1 \dots N}, \vec{x}_i \in \mathcal{R}^d; y_i \in \mathcal{C}$$

Encontrar:

$$f : \mathcal{R}^d \rightarrow \mathcal{C}$$

- Clasificación binaria:

$$f : \mathcal{R}^d \rightarrow \{-1, 1\}$$

- Clasificación multiclase:

$$f : \mathcal{R}^d \rightarrow \{C_1, \dots, C_k\}$$

Clasificación Multi-Etiqueta

Introducción

Transformación

Adaptación

Evaluación

Selección

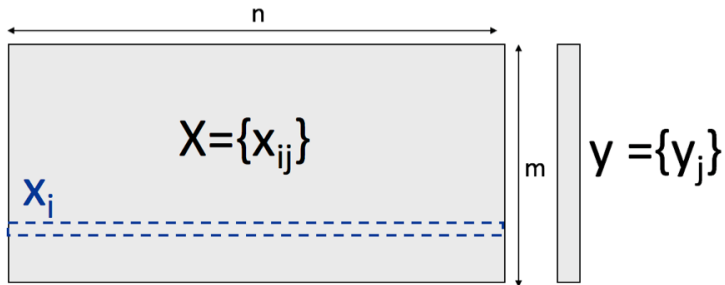
- Dado:

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

- Encontrar:

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

Clasificación Multi-Clase



Introducción

Transformación

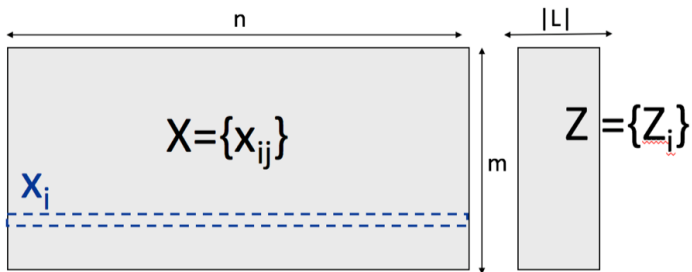
Adaptación

Evaluación

Selección

Clasificación Multi-Etiqueta

Introducción
Transformación
Adaptación
Evaluación
Selección



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: Características

Introducción

Transformación

Adaptación

Evaluación

Selección

- 1 Se tiene un conjunto finito de etiquetas menor que el número de atributos
- 2 Las etiquetas pueden estar correlacionadas
- 3 Los datos pueden estar desbalanceados

La base de datos de películas Imdb

Introducción

Transformación

Adaptación

Evaluación

Selección

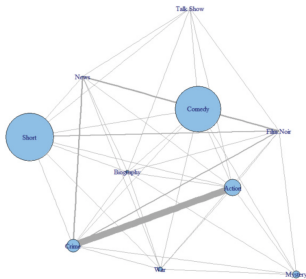


Fig. 1. Co-occurrence graph of labels in the imdb dataset.

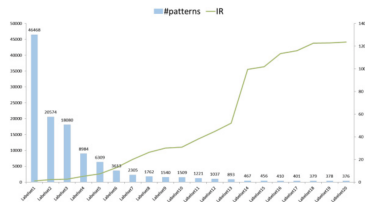


Fig. 2. Examples and Imbalance Ratio (IR) per labelset in the imdb dataset.

Ejemplo

Introducción

Transformación

Adaptación

Evaluación

Selección

EXAMPLE	FEATURES	SINGLE-LABEL BINARY	SINGLE-LABEL MULTICLASS	MULTILABEL OUTPUT				
		$Y \in \mathcal{L} = \{0, 1\}$	$Y \in \mathcal{L} = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$	y_1	y_2	y_3	y_4	$Y \subseteq \mathcal{L} = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$
1	$\bar{\mathbf{x}}_1$	1	λ_2	1	1	0	1	$\{\lambda_1, \lambda_2, \lambda_4\}$
2	$\bar{\mathbf{x}}_2$	0	λ_4	0	0	0	1	$\{\lambda_4\}$
3	$\bar{\mathbf{x}}_3$	0	λ_3	0	1	1	1	$\{\lambda_2, \lambda_3, \lambda_4\}$
4	$\bar{\mathbf{x}}_4$	1	λ_1	1	0	1	0	$\{\lambda_1, \lambda_3\}$
5	$\bar{\mathbf{x}}_5$	0	λ_3	0	1	1	0	$\{\lambda_2, \lambda_3\}$
				2	3	3	3	<COUNT

Ejemplo 2



- Clasificación binaria: ¿Es una foto del mar? $\in \{\mathbf{si}, \mathbf{no}\}$
- Clasificación multiclase: ¿Qué hay en la foto? $\in \{\mathbf{mar}, \mathbf{gente}, \mathbf{árboles}, \mathbf{montañas}\}$
- Clasificación multi-etiqueta: ¿Qué clases son relevantes en la foto? $\in \{\mathbf{mar}, \mathbf{gente}, \mathbf{árboles}, \mathbf{montañas}\}$

Clasificación Multi-Etiqueta

Introducción

Transformación

Adaptación

Evaluación

Selección

El aprendizaje multi-etiqueta pueda involucrar dos tareas:

- 1 Clasificación multi-etiqueta (MLL): Dado un dato, regresar un subconjunto de etiquetas asociadas a los datos (y su complemento)
- 2 Ordenamiento de etiquetas (LR): Dado un dato regresa todas las etiquetas ordenadas por relevancia

Clasificación Multi-Etiqueta

Introducción

Transformación

Adaptación

Evaluación

Selección

Existen dos enfoques generales para clasificación multi-etiqueta:

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

Algoritmos

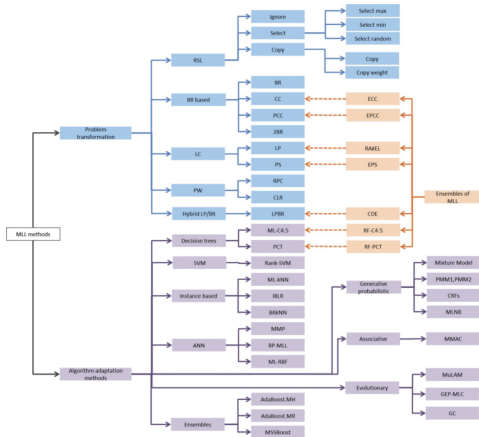
Introducción

Transformación

Adaptación

Evaluación

Selección



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)

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	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

Max		Min		Rand	
Ex.	Label	Ex.	Label	Ex.	Label
1	λ_4	1	λ_1	1	λ_1
2	λ_4	2	λ_3	2	λ_4
3	λ_1	3	λ_1	3	λ_1
4	λ_4	4	λ_2	4	λ_3

Transformed ML problem

Ex.	Label
3	λ_1

Ignore approach

Métodos de Transformación

- Conjunto potencia (*power set*): 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	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
1	$\lambda_{1,4}$
2	$\lambda_{3,4}$
3	λ_1
4	$\lambda_{2,3,4}$

Transformed ML problem

- Problemas: Puede crear desbalances, no clasifica subconjuntos no vistos, y se pueden crear muchas clases

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

RAkEL

Introducción

Transformación

Adaptación

Evaluación

Selección

- 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	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	Ex.	Label	Ex.	Label	Ex.	Label
1	λ_1	1	$-\lambda_2$	1	$-\lambda_3$	1	λ_4
2	$-\lambda_1$	2	$-\lambda_2$	2	λ_3	2	λ_4
3	λ_1	3	$-\lambda_2$	3	$-\lambda_3$	3	$-\lambda_4$
4	$-\lambda_1$	4	λ_2	4	λ_3	4	λ_4

Data sets generated by BR

Binary Relevance

Introducción

Transformación

Adaptación

Evaluación

Selección

- **Ventajas:** Tiene baja complejidad computacional, al suponer independencia entre etiquetas estas se pueden añadir o quitar, se puede paralelizar
- **Desventajas:** Supone independencia de las etiquetas, después de la transformación se pueden crear desbalances que se incrementa con alta dimensionalidad en el número de etiquetas

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	$\lambda_{1,-2}$
3	$\lambda_{1,-2}$
4	$\lambda_{-1,2}$

(a)

 λ_1 vs. λ_3

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

(b)

 λ_1 vs. λ_4

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

(c)

 λ_2 vs. λ_3

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

(d)

 λ_2 vs. λ_4

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

(e)

 λ_3 vs. λ_4

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

(f)

Adaptación de Algoritmos

Introducción

Transformación

Adaptación

Evaluación

Selección

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

- Árboles de decisión (permite a las hojas tener más de una clase y usa una medida de entropía multi-etiqueta)
- Boosting (Adaboost): Evalua considerando múltiples clases
- Campos aleatorios de Markov: Lo modifican para considerar co-ocurrencia de etiquetas

Adaptación de Algoritmos (cont.)

Introducción

Transformación

Adaptación

Evaluación

Selección

- 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 (las clases más comunes)
- Ensamblados de clasificadores multi-etiqueta, usando variantes de Random Forest

Multi-Dimensional Bayesian Classifiers

Introducción

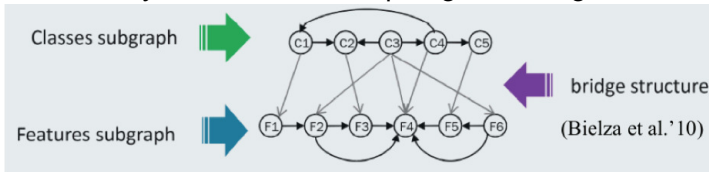
Transformación

Adaptación

Evaluación

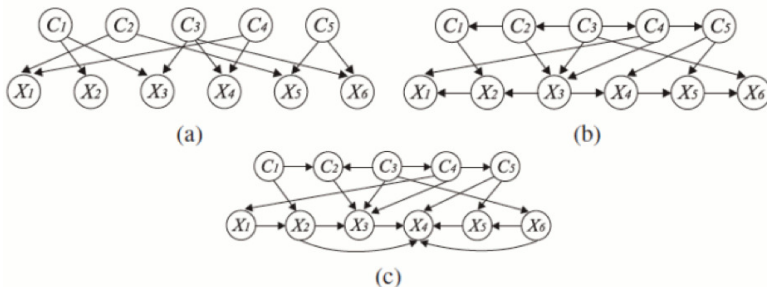
Selección

- 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.

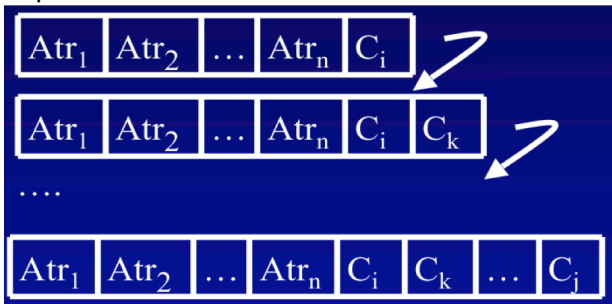
Multi-Dimensional Bayesian Classifiers



- 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 blankets (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	$x \rightarrow$	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	$x' \rightarrow$	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

Chain Classifiers: Otras opciones

Introducción

Transformación

Adaptación

Evaluación

Selección

- Monte Carlo chain
- Usar una búsqueda con un parámetro de temperatura
- Usar una población
- Algún algoritmo de búsqueda (e.g., beam search)

Bayesian Chain Classifiers

Introducción

Transformación

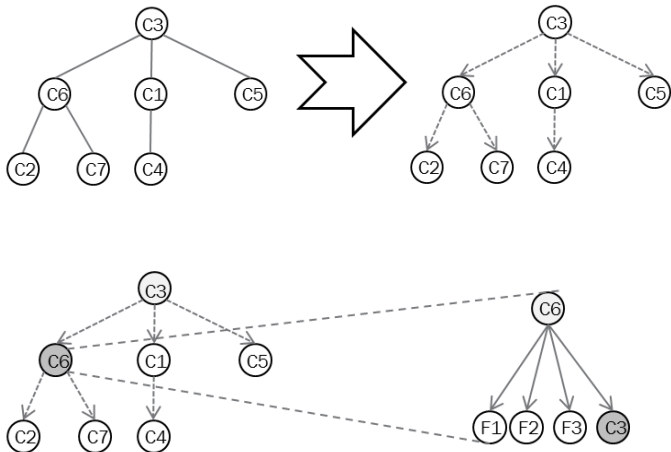
Adaptación

Evaluación

Selección

- La idea es determinar un orden con base en dependencias 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 un clasificador en cadena tomando en cuenta esta estructura (sólo incorpora los padres de cada clase como atributos adicionales)

Bayesian Chain Classifiers



Introducción

Transformación

Adaptación

Evaluación

Selección

Bayesian Chain Classifiers

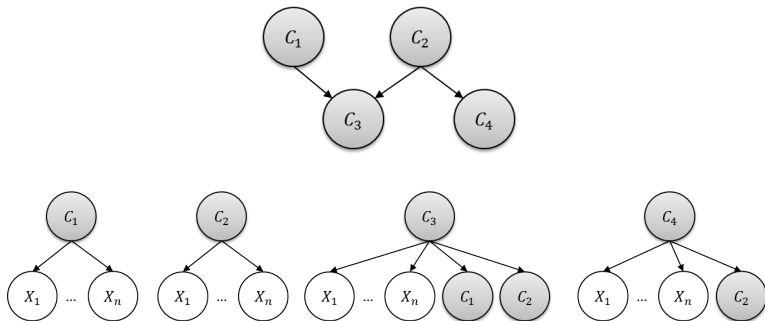
Introducción

Transformación

Adaptación

Evaluación

Selección



Jerárquicos

Introducción

Transformación

Adaptación

Evaluación

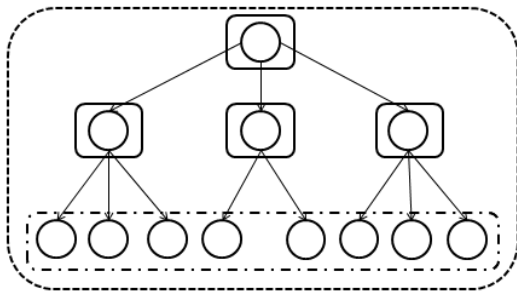
Selección

- A veces las clases están organizadas en una jerarquía
- Algunos algoritmos aprovechan esa información adicional (dependencias jerárquicas conocidas)
- Clasificación:
 - 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 diferentes 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 surge porque los clasificadores se consideran independientes entre sí

Tipos de Clasificadores



Introducción

Transformación

Adaptación

Evaluación

Selección

Jerárquicos

Introducción

Transformación

Adaptación

Evaluación

Selección

- 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*)

Jerárquicos

Introducción

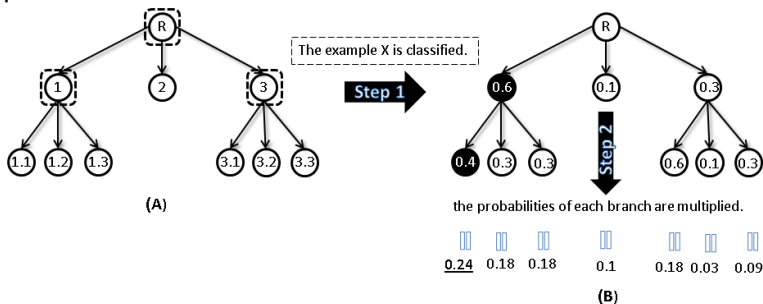
Transformación

Adaptación

Evaluación

Selección

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



Jerárquicos

Introducción

Transformación

Adaptación

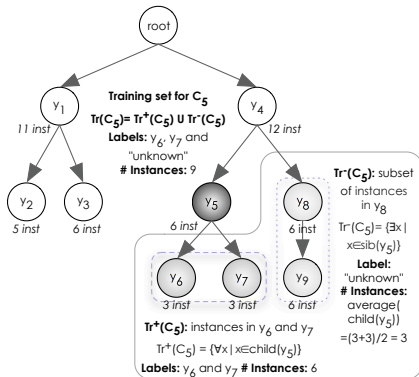
Evaluación

Selección

- Usar ideas de clasificadores multi-etiqueta
- Aprovechar propiedades: Un ejemplo que pertenece a una clase, también pertenece 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*)

Jerárquicos

- Usar ejemplos negativos de nodos cercanos para balancear las clases



Jerárquicos

Introducción

Transformación

Adaptación

Evaluación

Selección

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

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

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

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

Jerárquicos

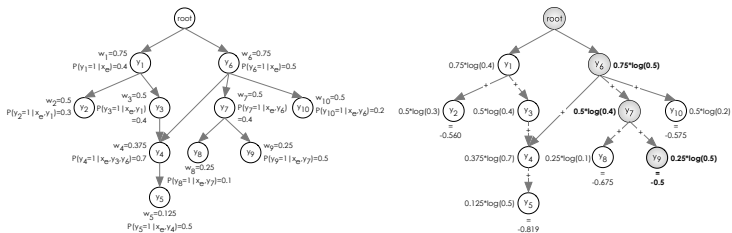
Introducción

Transformación

Adaptación

Evaluación

Selección



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

Introducción

Transformación

Adaptación

Evaluación

Selección

- *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

- Exact-Match:

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

- Accuracy:

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

- Hamming-Loss and Hamming-Accuracy:

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

donde \oplus es *exclusive or*

Hamming accuracy (H-Accuracy) se define como:

H-Accuracy = $1 - HammingLoss$.

Medidas de Evaluación

- F1-measure: Para multi-etiqueta, redefiniendo 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

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

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

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

Medidas de Evaluación Jerárquicas

Introducción

Transformación

Adaptación

Evaluación

Selección

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

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

Conociendo el posible valor máximo y mínimo se puede normalizar:

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

Medidas de Evaluación para ordenamiento de etiquetas

Introducción

Transformación

Adaptación

Evaluación

Selección

- One-error: Cuántas veces el mejor evaluado no estaba en el conjunto de etiquetas
- Coverage: La profundidad para cubrir todas las etiquetas asociadas a un ejemplo
- Ranking-loss: El promedio en que pares de etiquetas están desordenadas
- ...

Selección de Atributos

Introducción

Transformación

Adaptación

Evaluación

Selección

- 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

Introducción

Transformación

Adaptación

Evaluación

Selección

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

- Los algoritmos *wrapper* se pueden aplicar directamente (con algún algoritmo de clasificación multi-etiqueta)
- Transforman el problema en uno o más de una sola clase y se usa algún algoritmo de selección de atributos clásico, después se sigue algún esquema de “agregación”
- También se han propuesto variantes de algoritmos de extracción de atributos como LDA

Meka

Introducción

Transformación

Adaptación

Evaluación

Selección

- 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>