

Notas para reu: 30-03-22

Temas:

- Lectura de "Towards Class-Imbalance Aware Multi-Label Learning"
- Lectura de "Integration of deep learning model and feature selection for multi-label classification"
- Data augmentation
- Errores de clasificación: ¿son los mismos al variar el threshold?

Towards Class-Imbalance Aware Multi-Label Learning

- El paper incorpora conceptos como Label Density y Label Cardinality.
- El enfoque propuesto, COCOA, se compara con los enfoques de transformación del problema que ya probamos (algoritmos lineales, no lineales y de ensemble).
- Se usa como principal métrica el F1-Score. Esto es razonable en general, pues es una media armónica entre precision y recall. Sin embargo, a nosotros podría interesarnos más el recall, caso en el que podría convenir usar F0.5-Score.
- Se realizan experimentos sobre 18 datasets distintos, con distinto tipo de entrada. De ellos, el con la menor label density alcanza un valor de 0.036, poco más del doble de nuestro caso (0.015). Se trata de Bibtex, un dataset de texto. El mejor F-Score es de 0.326 por RML y el segundo 0.318 por COCOA.
- También se incluyen dos datasets de imágenes, Corel5k y Scene. COCOA resulta ser el mejor y segundo mejor algoritmo para cada caso, con F-score de 0.728 y 0.196, respectivamente.
- Me llama la atención que se mencionan los **árboles de decisión** como un buen clasificador base para problemas de class imbalance. **Debería realizar experimentos con ellos para ver si mejora nuestro estado actual.**
- No creo que COCOA sirva para nuestro caso. El algoritmo trata de abarcar el desbalance de clases pero no en una misma clase si no en pares de clases, para así contribuir a la forma en que se descubren las relaciones entre labels.
 - En este escenario, frecuentemente hay un alto desbalance pues es difícil tener muchos patrones con ambas etiquetas (tenemos algunos con una, otros con la otra y solo algunos con ambas).
 - Lo que propone es, para una etiqueta A, generar K predictores multiclase que predigan a su vez A y alguna otra etiqueta B_i, transformando el problema de multilabel en multiclase: 0 si no aparece ni A ni B, 1 si solo aparece A, 2 si solo aparece B, 3 si aparecen ambas. Y la mejora que propone es reducir las cuatro clases a solo 3, combinando el caso 1 y 3 (es decir: cada vez que aparezca el label que me interesa predecir, asumo que viene también con su label correlacionado).
 - Qué tendría que pasar para que este enfoque nos sirva: **que un label facil de predecir (probablemente aquellos mas frecuentes) arrastre siempre labels especificos (mas dificiles)**
- No encontré implementaciones de COCOA, pero podría codearlo si lo consideramos necesario.

Integration of deep learning model and feature selection for multi-label classification

- Enfoque basado en deep learning
- Graph-based feature selection para reducir dimensiones
- Reducir las dimensiones de los datos como una forma de reducir también la cantidad de datos necesarios para entrenar (razonable)

Data Augmentation

- Pasamos de 775 patrones a 3285
- La frecuencia de cada etiqueta aumentó muchísimo. Ahora con t=15 tenemos 144 labels, mientras que con t=60 tenemos 57.
- Para el caso desbalanceado, los resultados no mejoran, pero cambia levemente el ranking de algoritmos
 - Precision alrededor de 0.7, recall alrededor de 0.4
- Para el caso balanceado (solo BR) los resultados empeoran levemente.
-

Errores de clasificación: ¿son los mismos al variar el threshold?

- Se prueba con Binary Relevance sobre Logistic Regression, en t=40 (HL: 0.82) y t=60 (HL: 0.944).
- Caso t=40:

label	precision	recall	f1-score	support
panel	0.75	0.72	0.73	46
horizontal	0.76	0.76	0.76	55
ornament	0.65	0.55	0.59	20
hatched	0.43	0.43	0.43	7
vertical	0.39	0.78	0.52	9
circle	0.22	0.29	0.25	7
metopal	0.71	0.50	0.59	20
filling	0.78	0.68	0.72	31
lozenge	0.20	0.11	0.14	9
enclosing	0.42	0.61	0.50	18
cross-hatched	0.37	0.35	0.36	20
double	0.47	0.80	0.59	10
triangle	0.27	0.60	0.37	5
concentric	0.37	0.44	0.40	16
chain	0.50	0.33	0.40	9
meander	0.38	0.60	0.46	5
bar	0.60	0.23	0.33	13
solid	0.32	0.47	0.38	15
dotted	0.00	0.00	0.00	5
outline	0.72	0.93	0.81	14
line	0.07	0.11	0.08	9
cross	0.78	0.93	0.85	15
dot	0.60	0.75	0.67	4
hook	0.50	0.50	0.50	8
micro avg	0.55	0.59	0.57	370
macro avg	0.47	0.52	0.48	370
weighted avg	0.57	0.59	0.57	370
samples avg	0.55	0.56	0.53	370

- Caso t=60:

precision	recall	f1-score	support	support
panel	0.75	0.72	0.73	46
horizontal	0.76	0.76	0.76	55
ornament	0.65	0.55	0.59	20
hatched	0.39	0.78	0.52	9
vertical	0.71	0.50	0.59	20
circle	0.78	0.68	0.72	31
metopal	0.42	0.61	0.50	18
filling	0.37	0.35	0.36	20
lozenge	0.47	0.80	0.59	10
enclosing	0.37	0.44	0.40	16
cross-hatched	0.38	0.60	0.46	5
double	0.32	0.47	0.38	15
triangle	0.72	0.93	0.81	14
concentric	0.78	0.93	0.85	15
micro avg	0.60	0.66	0.63	294
macro avg	0.56	0.65	0.59	294
weighted avg	0.63	0.66	0.64	294
samples avg	0.57	0.61	0.57	294

En general, los resultados no cambian de un threshold a otro. Algunas excepciones - tales como el caso de la etiqueta hatched - se asume pueden deberse a la reconstrucción de train y test size de forma aleatoria.