Beijing Forest Studio
北京理工大学信息系统及安全对抗实验中心

多标签学习

硕士研究生　张睿智

2021年08月22日

内容提要

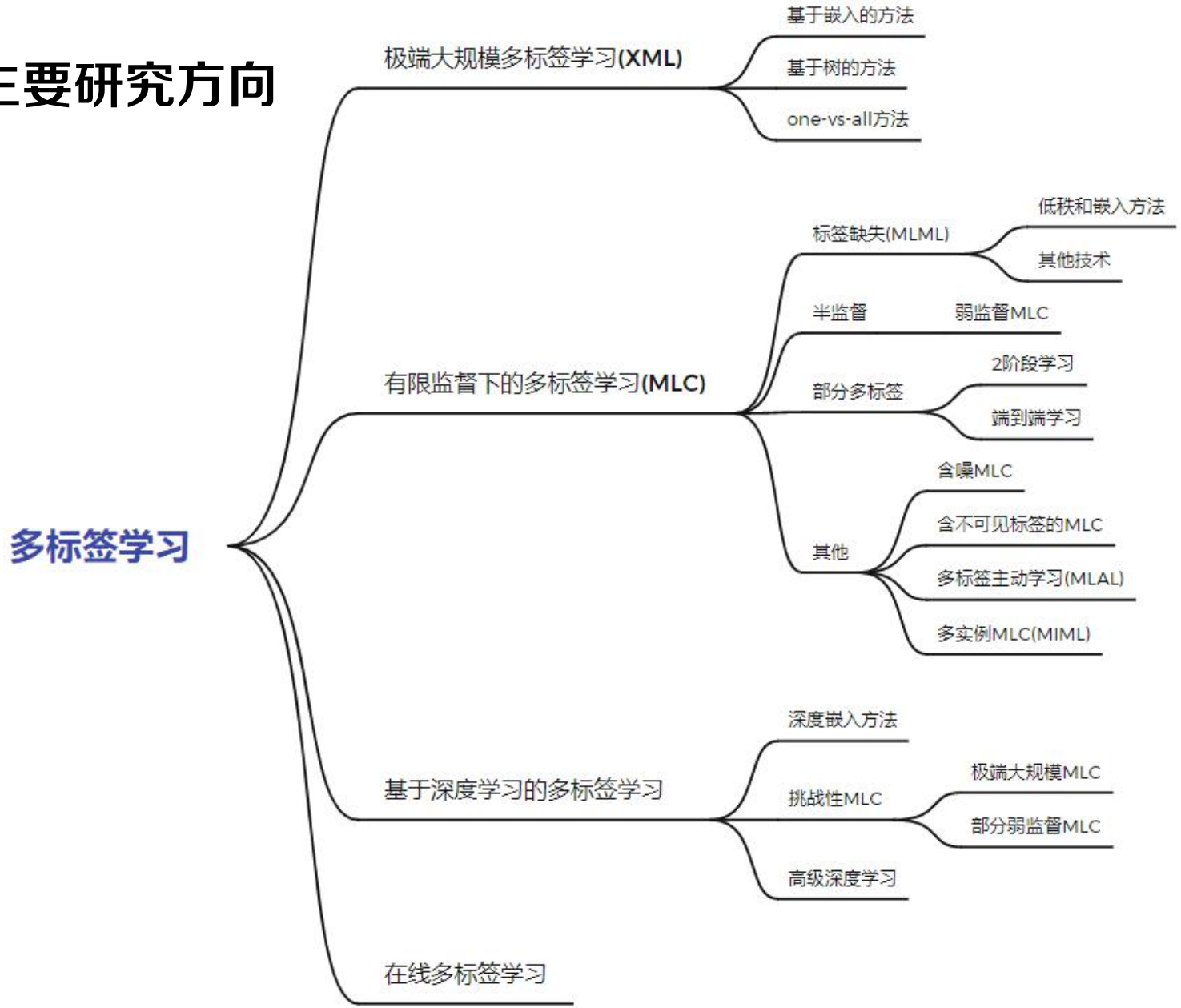
- 背景简介
- 基本概念
  - 主动学习
  - AR和AE语言模型
- 算法原理
- 应用总结
- 参考文献

预期收获

- 预期收获
  - 1. 了解多标签学习的意义和概念
  - 2. 了解基于深度学习的多标签学习原理
  - 3. 了解多标签学习常用方法
  - 4. 了解多标签学习的主要应用

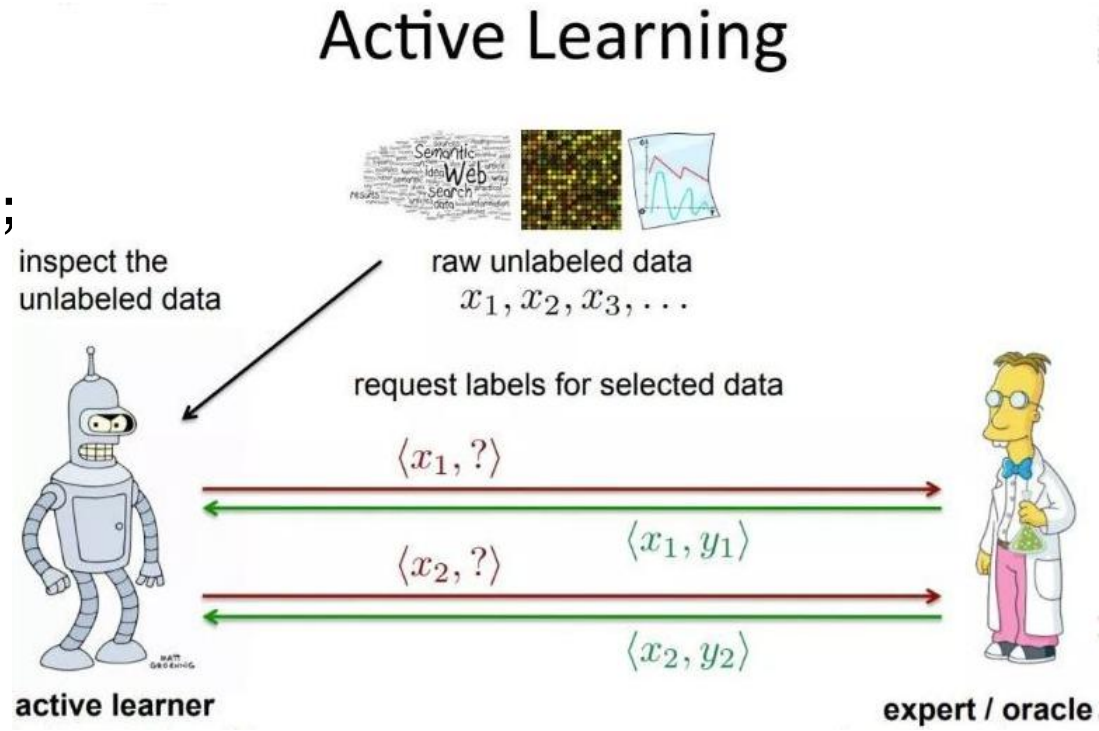
背景简介

- 多标签学习主要研究方向



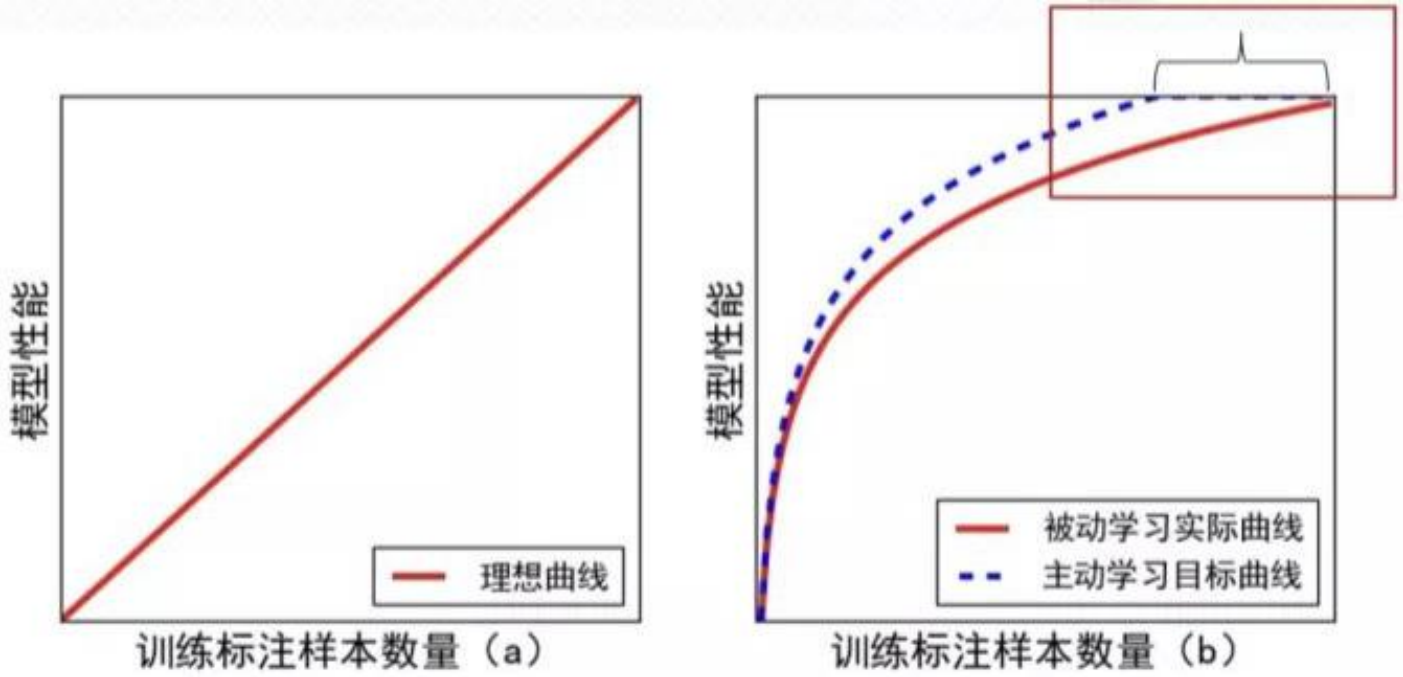
基本概念

- 主动学习（Active Learning）
  - 目标：以尽可能少的标注样本达到目标性能
  - 实现流程：
    - 1. 模型训练：机器学习模型训练及预测；
    - 2. 数据候选集提取：获取较难分类的样本数据；
    - 3. 人工标注：依据专家经验人工确认和审核；
    - 4. 获得候选集的标注数据：获得更有价值的样本数据；
    - 5. 模型更新：通过增量学习或者重新学习的方式更新模型，将人工经验融入机器学习模型中，逐步提升模型的效果。



基本概念

- 主动学习和被动学习的关系
  - 被动学习：模型只学习给定的样本
  - 主动学习：模型参与样本的选择，并优先学习最能够提升当前模型性能的样本
  - 主动学习核心解决的问题是使用尽可能少的标注数据达到模型的瓶颈性能，从而减少不必要的标注成本

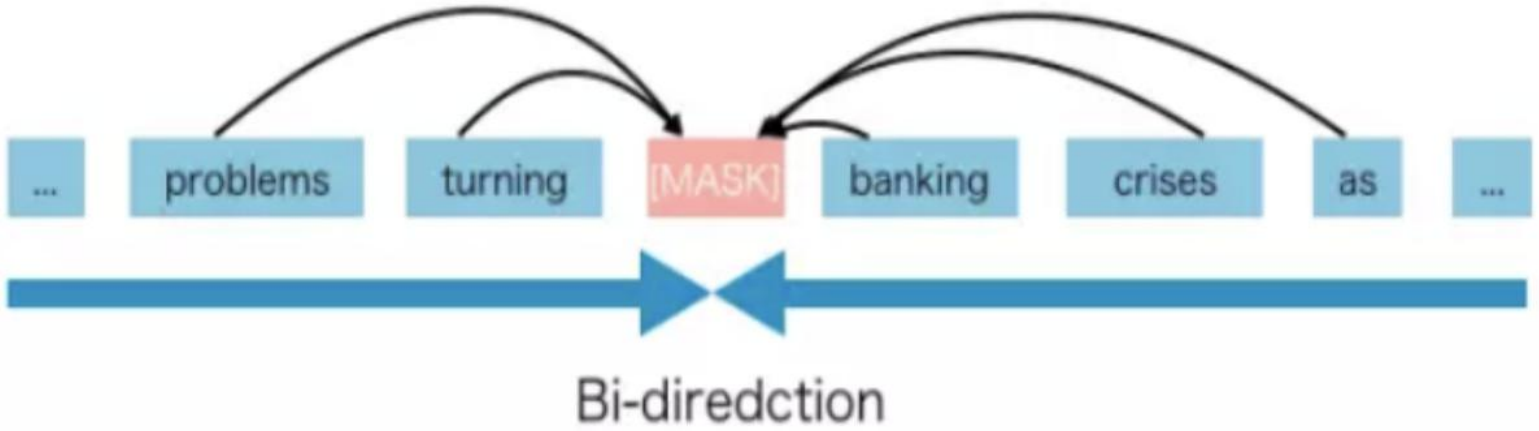


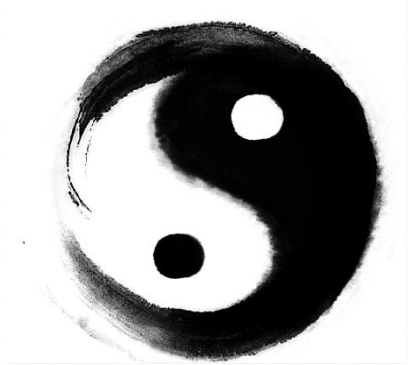
基本概念

- 主动学习和半监督学习的关系
  - 相同点：部分标注
    - 都是从未标记样例中挑选部分价值量高的样例，标注后补充到已标记样例集中来提高分类器精度。
  - 不同点：学习方式
    - 半监督学习一般不需要人工参与，通过具有一定分类精度的基准分类器实现对未标注样例的自动标注，标注代价相对较小。
    - 主动学习需要人工准确标注，减少错误引入，但标注代价相对较大。

基本概念

- AR和AE语言模型
  - 自回归语言模型（Autoregressive LM）
    - 只能获取单向信息，即只能根据上文预测下一个单词，或根据下文预测前一个单词
    - GPT，GPT2，XLNet，ELMO
  - 自编码语言模型（Autoencoder LM）
    - 可以获取双向信息，能同时看到预测位置的上文和下文
    - Bert





算法原理

算法原理 X-Transformer

T	在有限资源条件以及极大规模数据集下进行多标签学习
I	原始文本
P	1. 标签聚类 2.使用Transformer框架进行微调，将实例映射到标签簇 3. 对预测的多个标签簇内进行排序得到最终标签
O	样本的多个标签

P	将Transformer框架应用到多标签学习
C	有限资源条件下的大规模多标签学习
D	如何在保证模型性能的前提下减小模型规模
L	KDD2020

算法原理

- 提出问题
  - 标签空间过大以及标签稀疏导致XLNet等大型模型无法在有限资源下应用到该领域
- 解决思路
  - 将难以处理的XMC问题分解成一组输出空间小得多的可行子问题，缓解了标签稀疏性问题

	XLNet-large model (# params)			(batch size, sequence length)=(1,128)			
problem	encoder	classifier	total	load model	+forward	+backward	+optimizer step
GLUE (MNLI)	361 M	2 K	361 M	2169 MB	2609 MB	3809 MB	6571 MB
XMC (1M)	361 M	1,025 M	1,386 M	6077 MB	6537 MB	OOM	OOM

Table 1: On the left of are the model sizes (numbers of parameters) when applying the XLNet-large model to the MNLI problem vs. the XMC (1M) problem; on the right is the GPU memory usage (in megabytes) in solving the two problems, respectively. The results were obtained on a recent Nvidia 2080Ti GPU with 12GB memory. OOM stands for out-of-memory.

算法原理 原理框图

- 实现过程：
  - 1.使用语义标签索引将标签进行聚类，减小标签输出空间；
  - 2.使用Transformer在XMC子问题上进行微调，将实例映射到标签簇；
  - 3.在聚类和Transformer的输出上训练线性排序器，在预测的标签簇内重新排序标签。

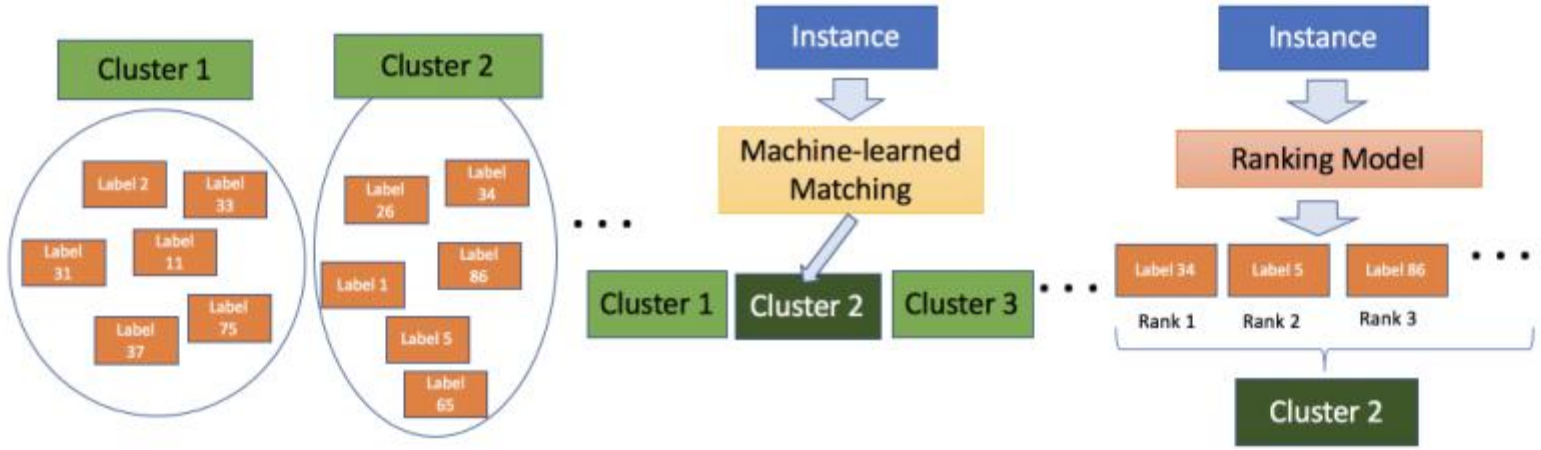


Figure 3: The proposed X-Transformer framework. First, Semantic Label Indexing reduces the large output space. Transformers are then fine-tuned on the XMC sub-problem that maps instances to label clusters. Finally, linear rankers are trained conditionally on the clusters and Transformer's output in order to re-rank the labels within the predicted clusters.

- 语义标签索引
  - 1.标签特征表示
    - 使用标签描述信息表示标签（例如维基百科数据集中类别的简短描述；亚马逊购物网站上的搜索查询）。

$$\psi_{\text{text-emb}}(l) = \frac{1}{|\text{text}(l)|} \sum_{w \in text(l)} \phi_{\text{xlnet}}(w)$$

    - 正实例特征聚合（PIFA），即提取被该标签标记的所有样本的特征和作为该标签的特征信息。

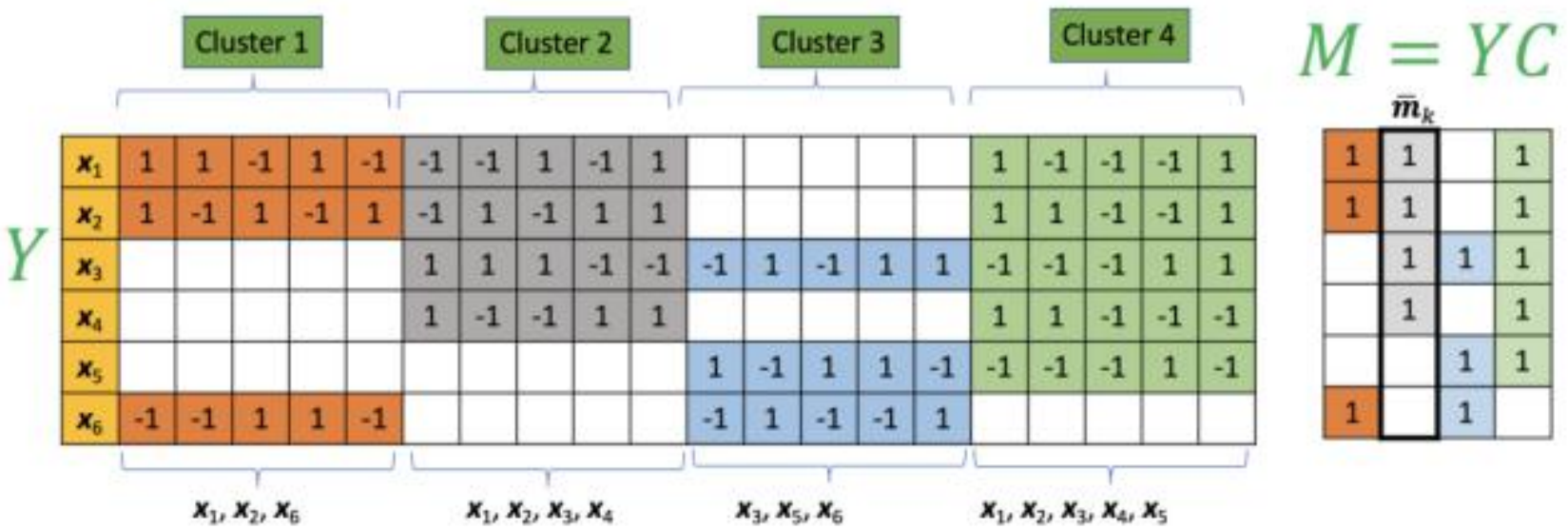
$$\psi_{\text{pifa-tfidf}}(l) = \mathbf{v}_l / \|\mathbf{v}_l\|, \ \mathbf{v}_l = \sum_{i:y_{il}=1} \phi_{\text{tf-idf}}(\mathbf{x}_i), \quad l = 1, \ldots, L,$$

$$\psi_{\text{pifa-neural}}(l) = \mathbf{v}_l / \|\mathbf{v}_l\|, \ \mathbf{v}_l = \sum_{i:y_{il}=1} \phi_{\text{xlnet}}(\mathbf{x}_i), \quad l = 1, \ldots, L.$$

  - 2.使用二叉平衡树对标签进行分层聚类，形成一个有K（标签簇数）个叶节点的标签树

算法原理 标签排序

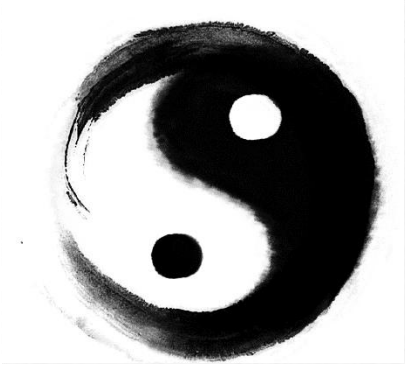
- 标签排序
  - 目标：尽可能使用最低的计算复杂度对实例和匹配到的标签簇中的标签之间进行相关性建模
  - 方法：
    - TFN
      - 只使用含该标签的样例子集训练分类器，减小计算量



算法原理 实验结果

- 对比实验结果
  - 在除Wiki-500K外的3个测试数据集上均取得最佳性能

Methods	Prec@1	Prec@3	Prec@5	Methods	Prec@1	Prec@3	Prec@5
Eurlex-4K				Wiki10-31K			
AnnexML [24]	79.66	64.94	53.52	AnnexML [24]	86.46	74.28	64.20
DiSMEC [1]	83.21	70.39	58.73	DiSMEC [1]	84.13	74.72	65.94
PfastreXML [8]	73.14	60.16	50.54	PfastreXML [8]	83.57	68.61	59.10
Parabel [20]	82.12	68.91	57.89	Parabel [20]	84.19	72.46	63.37
eXtremeText [29]	79.17	66.80	56.09	eXtremeText [29]	83.66	73.28	64.51
Bonsai [9]	82.30	69.55	58.35	Bonsai [9]	84.52	73.76	64.69
MLC2seq [16]	62.77	59.06	51.32	MLC2seq [16]	80.79	58.59	54.66
XML-CNN [12]	75.32	60.14	49.21	XML-CNN [12]	81.41	66.23	56.11
AttentionXML [32]	87.12	73.99	61.92	AttentionXML [32]	87.47	78.48	69.37
$\phi_{\text{pre-xlnet}}$ + Parabel	33.53	26.71	22.15	$\phi_{\text{pre-xlnet}}$ + Parabel	81.77	64.86	54.49
ϕ_{tfidf} + Parabel	81.71	69.15	58.11	ϕ_{tfidf} + Parabel	84.27	73.20	63.66
$\phi_{\text{fnt-xlnet}} \oplus \phi_{\text{tfidf}}$ + Parabel	84.09	71.50	60.12	$\phi_{\text{fnt-xlnet}} \oplus \phi_{\text{tfidf}}$ + Parabel	87.35	78.24	68.62
X-Transformer	**87.22**	**75.12**	**62.90**	X-Transformer	**88.51**	**78.71**	**69.62**
AmazonCat-13K				Wiki-500K			
AnnexML [24]	93.54	78.36	63.30	AnnexML [24]	64.22	43.15	32.79
DiSMEC [1]	93.81	79.08	64.06	DiSMEC [1]	70.21	50.57	39.68
PfastreXML [8]	91.75	77.97	63.68	PfastreXML [8]	56.25	37.32	28.16
Parabel [20]	93.02	79.14	64.51	Parabel [20]	68.70	49.57	38.64
eXtremeText [29]	92.50	78.12	63.51	eXtremeText [29]	65.17	46.32	36.15
Bonsai [9]	92.98	79.13	64.46	Bonsai [9]	69.26	49.80	38.83
MLC2seq [16]	94.26	69.45	57.55	MLC2seq [16]	-	-	-
XML-CNN [12]	93.26	77.06	61.40	XML-CNN [12]	-	-	-
AttentionXML [32]	95.92	82.41	67.31	AttentionXML [32]	76.95	**58.42**	**46.14**
$\phi_{\text{pre-xlnet}}$ + Parabel	80.96	63.92	50.72	$\phi_{\text{pre-xlnet}}$ + Parabel	31.83	20.24	15.76
ϕ_{tfidf} + Parabel	92.81	78.99	64.31	ϕ_{tfidf} + Parabel	68.75	49.54	38.92
$\phi_{\text{fnt-xlnet}} \oplus \phi_{\text{tfidf}}$ + Parabel	95.33	82.77	67.66	$\phi_{\text{fnt-xlnet}} \oplus \phi_{\text{tfidf}}$ + Parabel	75.57	55.12	43.31
X-Transformer	**96.70**	**83.85**	**68.58**	X-Transformer	**77.28**	57.47	45.31



算法原理

算法原理 APLC-XLNet

T	在标签分布不平衡的情况下进行多标签学习
I	原始文本
P	1.使用预训练的XLNet微调生成文本表示 2.将标签按照出现频率排序并分为头簇和几个尾簇 3.使用全连接层将样本映射到标签空间
O	样本的多个标签

P	尾部标签正样本数量少；模型规模大
C	样本稀疏及类别不平衡下的多标签学习
D	如何在保证模型性能的前提下减小模型规模
L	ICML2020

算法原理

- 提出问题
  - 标签分布遵循齐普夫定律
  - 在数据集Wiki-500K中，只有2%的标签有超过100条训练样本
  - 频繁标签只占标签集的20%，但出现概率大约75%
- 解决思路
  - 将标签集划分为一个头簇和几个尾簇
  - 头簇由出现最频繁的标签组成
  - 出现不频繁的标签被分到尾簇中

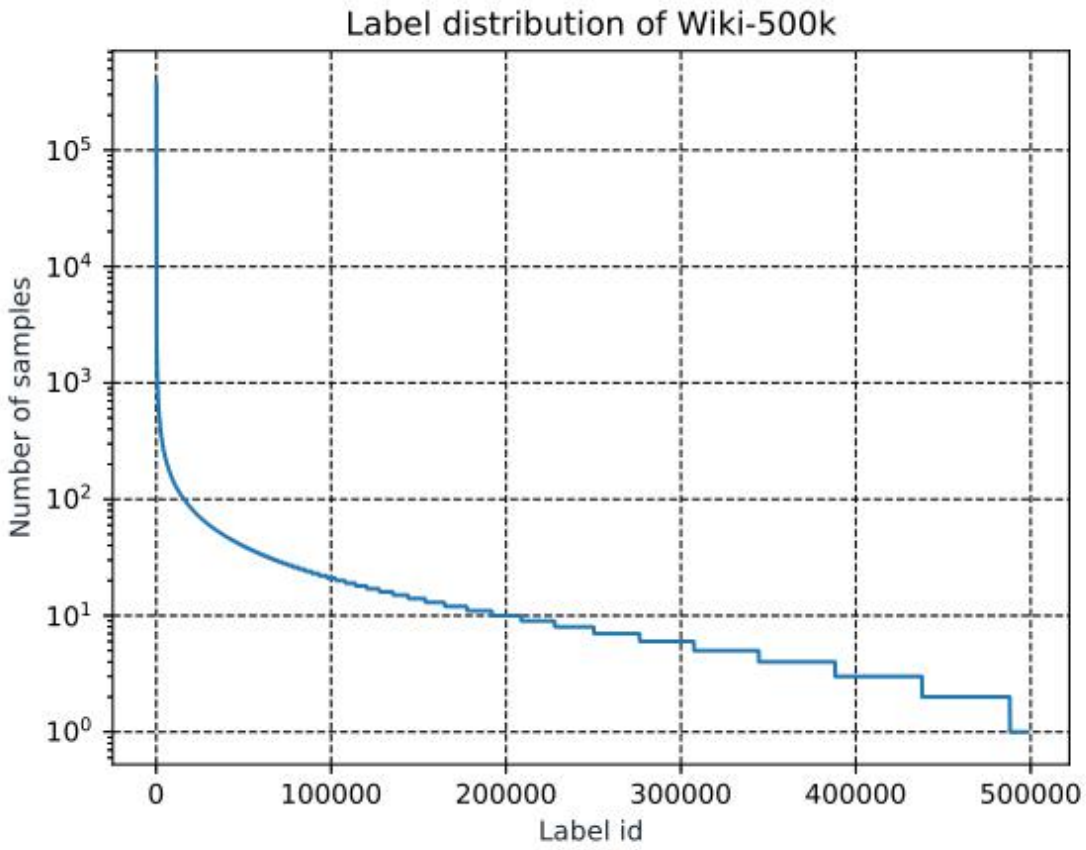
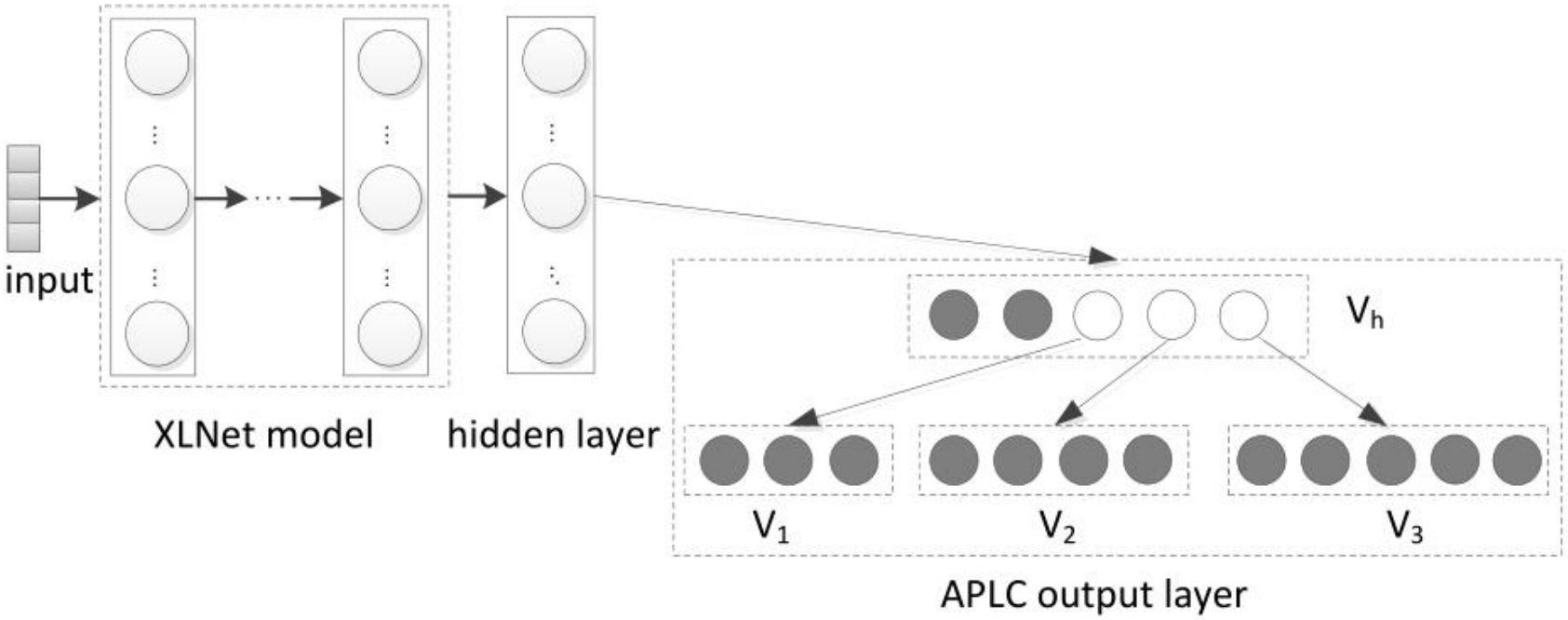


Figure 2. Label distribution of dataset Wiki-500k follows Zipf's Law. Label ids are sorted in descending order by the number of occurrence in the dataset.

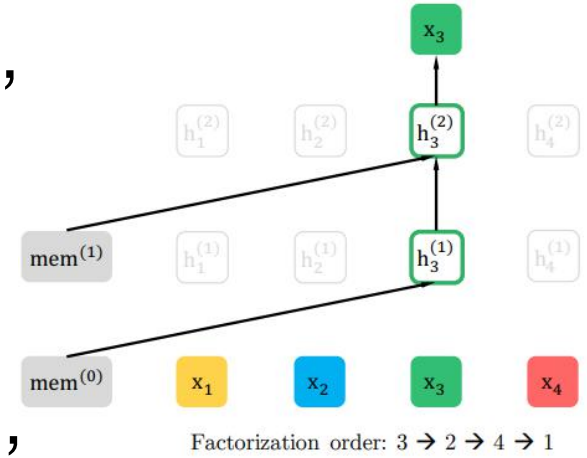
算法原理 原理框图

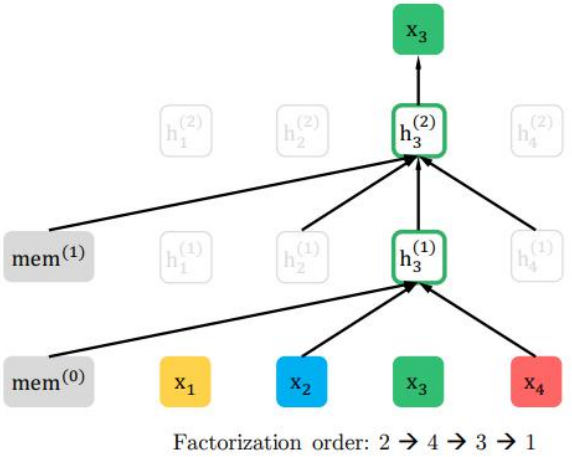
- 实现过程：
  - 1.使用预训练的XLNet模型并微调来学习文本表示；
  - 2.使用自适应概率标签簇将标签按照出现频率划分到头簇和尾簇中；
  - 3.使用一个全连接层连接XLNet的池化层和APLC模型的输出层，将样本的特征空间映射到标签空间。

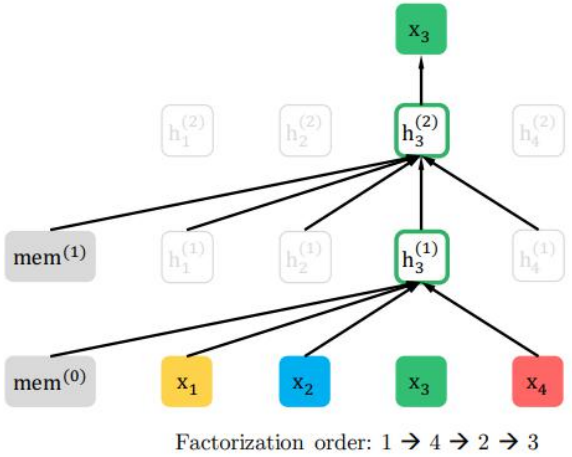


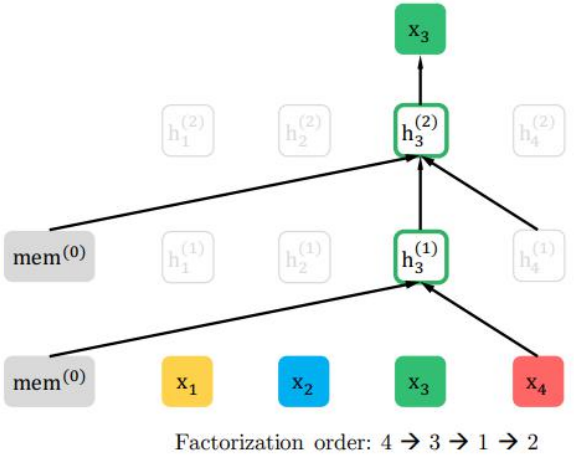
算法原理 XLNet

- XLNet:
  - 广义自回归预训练语言模型（AR），通过在因式分解顺序的所有排列上最大化期望对数似然来捕获双向上下文
  - 有序列[1,2,3,4]，如果预测目标是3，先对该序列进行因式分解，有24种排列方式
  - 第一种情况3的左边没有其他值，所以无需做对应计算
  - 第二种情况3的左边有2与4，因此结果是$p(3) = p(3|2) * p(3|2,4)$



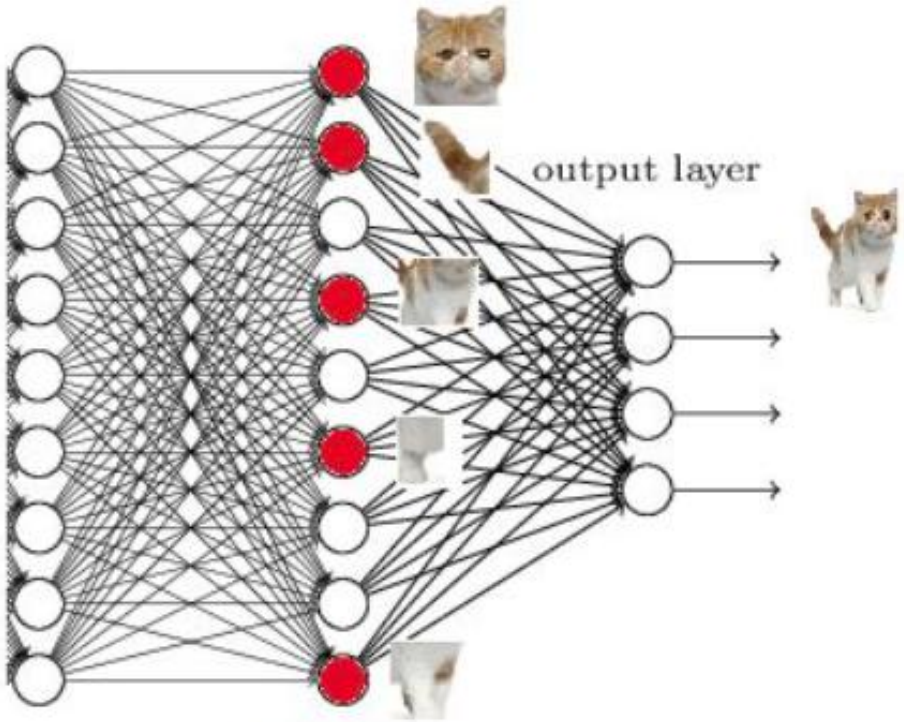






算法原理 全连接层

- 在XLNet的池化层和APLC输出层之间设置全连接层
  - 当输出标签的数量不多时，该层中的神经元数量设置为与池化层相同，以获得最佳的文本表示。
  - 当处理极端标签时（标签空间过大），则该层中的神经元数量比池化层少，以减小模型规模，并使计算更有效。



- 自适应概率标签簇
  - 生成二级树，将头簇作为根节点
  - 一个标签只可能出现在某一簇中
  - 通过概率的链式法则，一个标签的概率可以表示为：

$$p(y_j|x) = \begin{cases} p(y_j|x), & \text{if } y_j \in V_h \\ p(V_t|x)p(y_j|V_t, x), & \text{if } y_j \in V_t \end{cases}$$

  - 训练时会计算每条样本与头簇中每个标签的匹配程度（概率）。
  - 当训练样本的标签位于尾簇中时才访问尾簇，计算其中每个标签的概率。

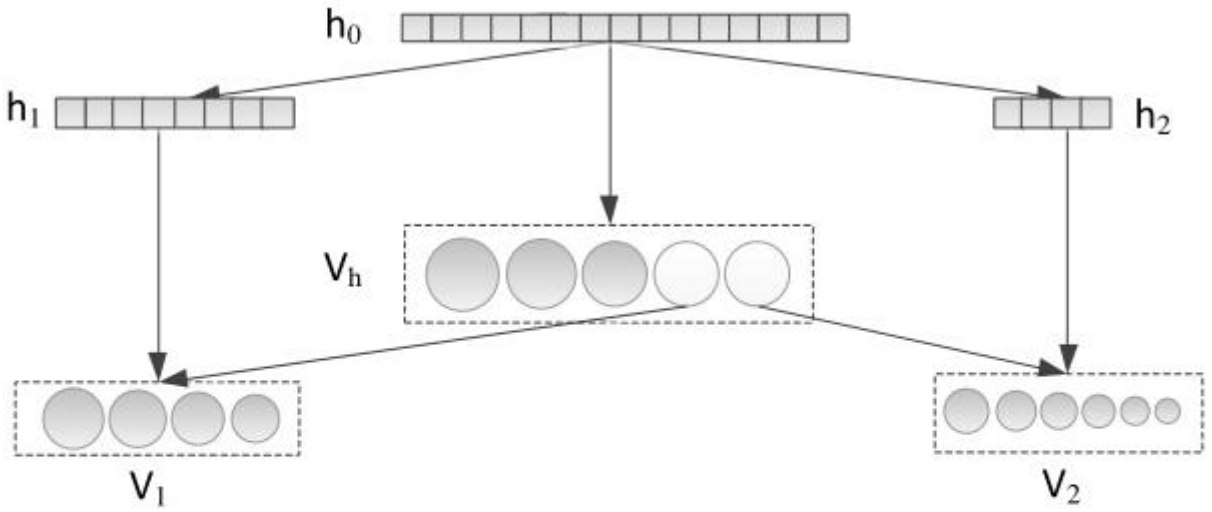


Figure 3. Architecture of the 3-cluster APLC. h denotes the hidden state. V_h denotes the head cluster. V_1 and V_2 denote the two tail clusters. The size of the leaf node indicates the frequency of the label.

算法原理 标签簇

- 自适应概率标签簇
  - 对于分类器频繁访问的头部聚类，高维的隐藏状态可以保持较高的预测精度
  - 对于尾部聚类，由于分类器很少访问它们，因此通过除以因子q (q ⩾ 1)来降低维数
  - 通过这种方式，可以显著减小模型规模，并保持良好的性能。

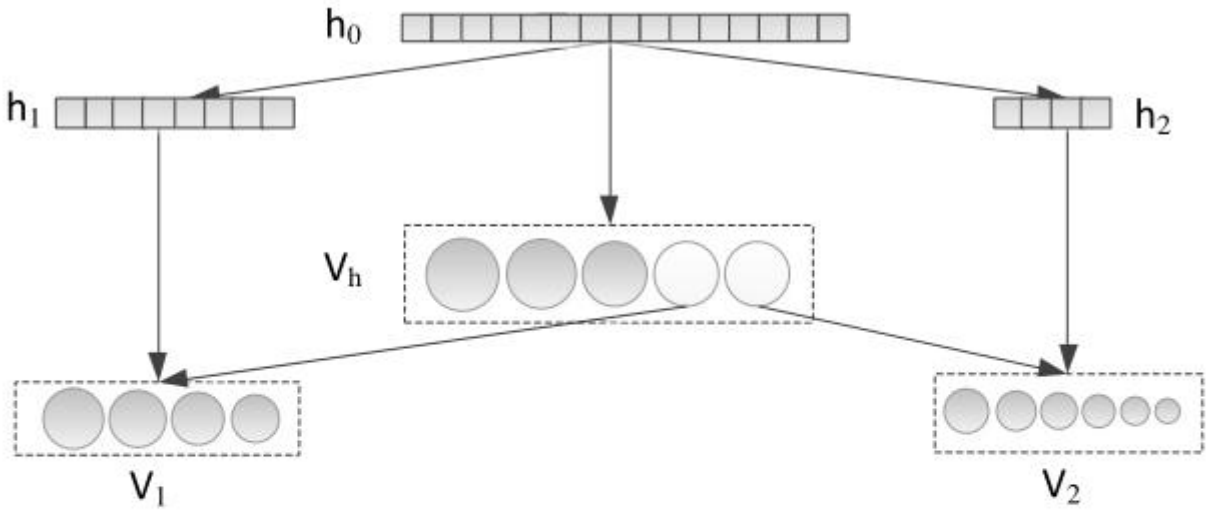


Figure 3. Architecture of the 3-cluster APLC. h denotes the hidden state. V_h denotes the head cluster. V_1 and V_2 denote the two tail clusters. The size of the leaf node indicates the frequency of the label.

算法原理 实验结果

- 对比实验结果
  - 所有数据集上的结果都优于嵌入方法（SLEEC、AnnexML）
  - 在三个数据集上(EURLex-4k、AmazonCat-13k和Wiki10-31k)，该方法比三种基于树的方法表现得更好
  - 深度学习方法AttentionXML在数据集Amazon-670k上的性能最好

Dataset		SLEEC	AnnexML	DisMEC	PfastreXML	Parabel	Bonsai	XML-CNN	AttentionXML	APLC-XLNet
EURLex-4k	P@1	79.26	79.66	82.40	75.45	81.73	83.00	76.38	87.14	**87.72**
	P@3	64.30	64.94	68.50	62.70	68.78	69.70	62.81	**75.18**	74.56
	P@5	52.33	53.52	57.70	52.51	57.44	58.40	51.41	**62.58**	62.28
AmazonCat-13k	P@1	90.53	93.55	93.40	91.75	93.03	92.98	93.26	92.62	**94.56**
	P@3	76.33	78.38	79.10	77.97	79.16	79.13	77.06	77.56	**79.82**
	P@5	61.52	63.32	64.10	63.68	64.52	64.46	61.40	62.74	**64.60**
Wiki10-31k	P@1	85.88	86.50	85.20	83.57	84.31	84.70	84.06	86.04	**89.44**
	P@3	72.98	74.28	74.60	68.61	72.57	73.60	73.96	77.54	**78.93**
	P@5	62.70	64.19	65.90	59.10	63.39	64.70	64.11	68.48	**69.73**
Wiki-500k	P@1	53.60	63.86	70.20	56.25	68.52	69.20	59.85	72.62	**72.83**
	P@3	34.51	40.66	50.60	37.32	49.42	49.80	39.28	**51.02**	50.50
	P@5	25.85	29.79	**39.70**	28.16	38.55	38.80	29.81	39.41	38.55
Amazon-670k	P@1	35.05	42.08	44.70	39.46	44.89	**45.50**	35.39	45.45	43.46
	P@3	31.25	36.65	39.70	35.81	39.80	40.30	31.93	**40.63**	38.82
	P@5	28.56	32.76	36.10	33.05	36.00	36.50	29.32	**36.92**	35.32

算法原理 实验结果

- 其他实验结果
  - 不同标签聚类数量对结果的影响
  - 不同标签集划分方法对结果的影响

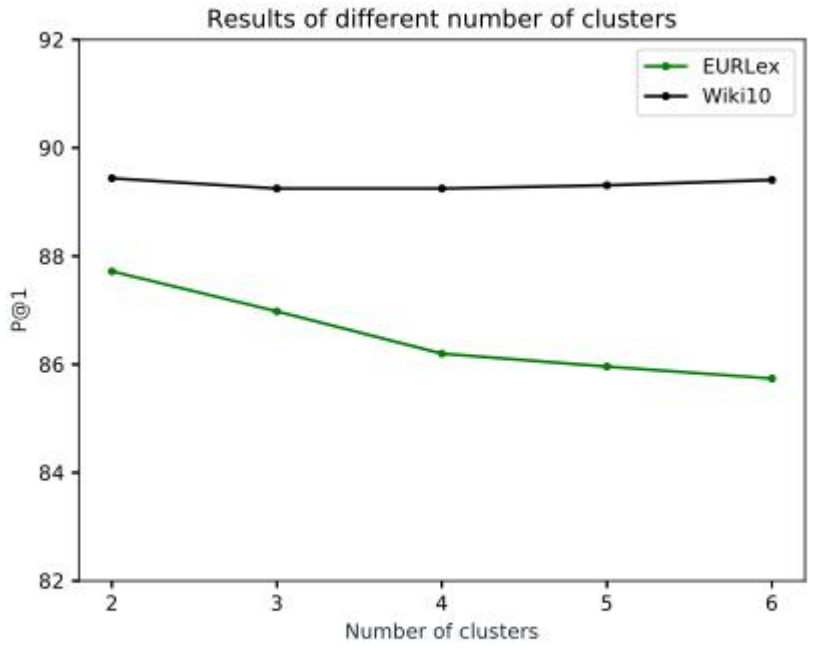


Figure 4. Precision $P@1$ on dataset EURLex and Wiki10, as a function of the number of clusters.

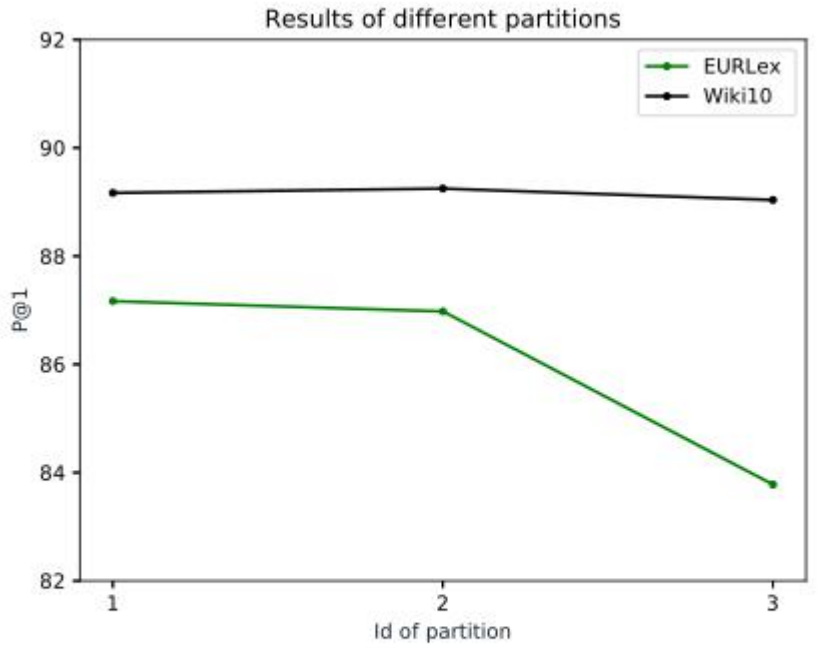
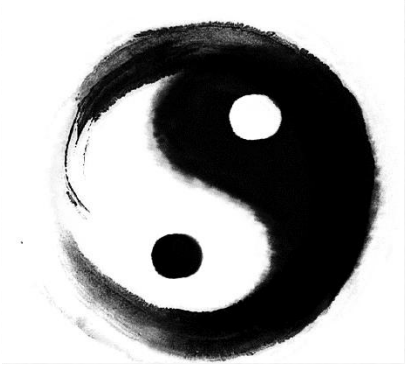


Figure 5. Precision $P@1$ of different partitions on dataset EURLex and Wiki10. Id 1, 2 and 3 denote the partition (0.7, 0.2, 0.1), (0.33, 0.33, 0.34) and (0.1, 0.2, 0.7) respectively.



应用总结

应用总结

- 多标签学习常见应用
  - 图像和视频识别注释
  - 面部动作单元(AU)识别
  - 用户画像
  - 城市应急管理
  - …

参考文献

[1] Liu W , X Shen, Wang H , et al. The Emerging Trends of Multi-Label Learning[J]. 2020.
[2] H Ye, Chen Z, Wang D H, et al. Pretrained Generalized Autoregressive Model with Adaptive Probabilistic Label Clusters for Extreme Multi-label Text Classification[J]. 2020.
[3] Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S Dhillon. Taming pretrained transformers for extreme multi-label text classification. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 3163–3171, 2020.

大成若缺，其用不弊。
大盈若冲，其用不穷。
大直若屈。大巧若拙。
大辩若讷。静胜躁，寒
胜热。清静为天下正。

谢谢！