

机器学习讨论班

2018年暑期

多标签学习

潘路明

2018/08/01

内容概要

- 基本概念
- 问题转化（problem transformation）法
 - 转化为二类分类问题
 - 转化为多类分类问题
 - 转化为标签排序（label ranking）问题
- 算法适应（algorithm adaptation）法
 - 惰性学习
 - 决策树
 - 神经网络
- 试验

基本概念： 图像中的目标探测



自然风景图

湖

树

山

多标签学
习

基本概念：文章标签

公司将继续密切关注该事项的后续发展，并严格按照相关法律法规及时履行信息披露义务。有关公司信息以公司指定披露媒体《中国证券报》、《上海证券报》、《证券时报》、《证券日报》及上海证券交易所网站刊登的公告为准。

敬请广大投资者注意投资风险。

特此公告。

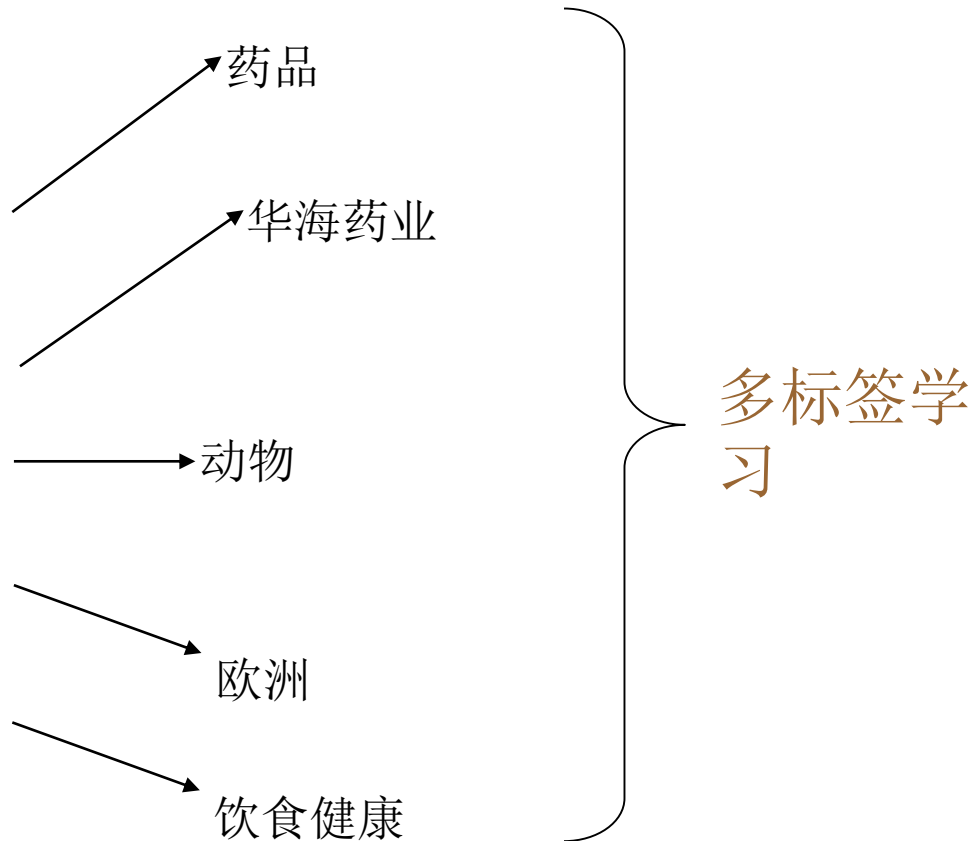
浙江华海药业股份有限公司

董事会

2018年7月30日

🔍 药品 / 华海药业 / 动物 / 欧洲 / 饮食健康

☆ 收藏 □ 举报



今日头条的新闻，来自：

https://www.toutiao.com/a6583838706946277896/?iid=39342979272&app=news_article&article_category=stock×tamp=1532931422

单标签（二分类） V.S. 多标签

X_1	X_2	X_3	X_4	X_5	Y
1	0.1	3	1	0	0
0	0.9	1	0	1	1
0	0.0	1	1	0	0
1	0.8	2	0	1	1
1	0.0	2	0	1	0
0	0.0	3	1	1	?

单标签 $Y \in \{0, 1\}$

X_1	X_2	X_3	X_4	X_5	Y
1	0.1	3	1	0	$\{\lambda_2, \lambda_3\}$
0	0.9	1	0	1	$\{\lambda_1\}$
0	0.0	1	1	0	$\{\lambda_2\}$
1	0.8	2	0	1	$\{\lambda_1, \lambda_4\}$
1	0.0	2	0	1	$\{\lambda_4\}$
0	0.0	3	1	1	?

多标签 $Y \subseteq \{\lambda_1, \dots, \lambda_L\}$

单标签（二分类）和多标签的向量化

X_1	X_2	X_3	X_4	X_5	Y
1	0.1	3	1	0	0
0	0.9	1	0	1	1
0	0.0	1	1	0	0
1	0.8	2	0	1	1
1	0.0	2	0	1	0
0	0.0	3	1	1	?

单标签 $Y \in \{0, 1\}$

X_1	X_2	X_3	X_4	X_5	Y_1	Y_2	Y_3	Y_4
1	0.1	3	1	0	0	1	1	0
0	0.9	1	0	1	1	0	0	0
0	0.0	1	1	0	0	1	0	0
1	0.8	2	0	1	1	0	0	1
1	0.0	2	0	1	0	0	0	1
0	0.0	3	1	1	?	?	?	?

多标签 $[Y_1, \dots, Y_L] \in 2^L$

形式化定义

- $\mathbf{X} \in \mathbf{R}^m$: 特征向量（输入）
- $\mathbf{Y} \in \mathbf{R}^d$: 类别向量（输出）
 - $\mathbf{x} = \{x_1, \dots, x_m\}$: 特征向量实例
 - $\mathbf{y} = \{y_1, \dots, y_d\}$: 类别向量实例
- 多标记分类器 $h(\cdot): \mathbf{X} \rightarrow 2^d$ 或者 $P(\mathbf{Y}=\mathbf{y} \mid \mathbf{X}=\mathbf{x}) = P(\mathbf{y} \mid \mathbf{x})$

多标签学习的评价指标

■ 基于样本的评价指标 (example-based metrics)

- Subset accuracy
- Hamming loss
- One-error
- Coverage
- Ranking loss
- Average precision

■ 基于类别的多标签评价指标

- 宏平均
- 微平均

难点

- 类标数量不确定，有些样本可能只有一个类标，有些样本的类标可能高达几十甚至上百个
- 类标之间相互依赖，例如包含蓝天类标的样本很大概率上包含白云，如何解决类标之间的依赖性问题也是一大难点
- 效率：标签空间

考察标签相关性的不同方式

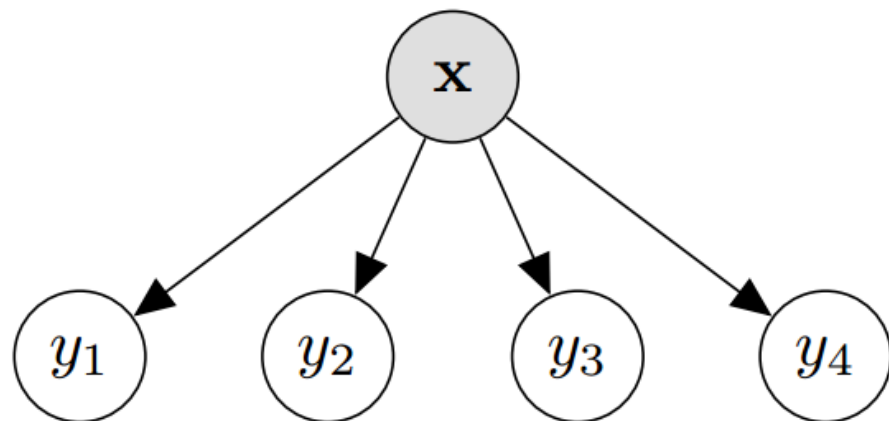
- 一阶（First-order）策略：将多标签学习转化为多个单标签学习问题，每个标签单独处理，完全忽略标签之间的相关信息
- 二阶（Second-order）策略：考虑两两标签之间的相关性
- 高阶（High-order）策略：考虑到更高阶的标签相关性，如：处理任一标签对其他所有标签的影响；处理一组随机标签组合的相关性

解决问题的角度

- 问题转换（Problem Transformation）法：改造数据适应算法
 - 转化为二类分类问题：Binary Relevance（一阶）及其改进
 - 转化为多类分类问题：标签幂集（Label Powerset，高阶）及其改进
 - 转化为标签排序问题：Calibrated Label Ranking（二阶）

- 算法适应（Algorithm Adaptation）法：改造算法适应数据
 - 惰性学习：ML-KNN（一阶）
 - 决策树：ML-DT
 - 神经网络：BP-MLL（二阶）

Binary Relevance



$$\hat{y}_j = h_j(\mathbf{x}) = \operatorname{argmax}_{y_j \in \{0,1\}} p(y_j | \mathbf{x})$$

(其中, $j = 1, \dots, d$)

$$\begin{aligned}\hat{\mathbf{y}} &= \mathbf{h}(\mathbf{x}) = [\hat{y}_1, \dots, \hat{y}_4] \\ &= \left[\operatorname{argmax}_{y_1 \in \{0,1\}} p(y_1 | \mathbf{x}), \dots, \operatorname{argmax}_{y_4 \in \{0,1\}} p(y_4 | \mathbf{x}) \right] \\ &= [f_1(\mathbf{x}), \dots, f_4(\mathbf{x})] = f(\mathbf{W}^\top \mathbf{x})\end{aligned}$$

Binary Relevance: 数据集转换

X	Y_1	Y_2	Y_3	Y_4
$\mathbf{x}^{(1)}$	0	1	1	0
$\mathbf{x}^{(2)}$	1	0	0	0
$\mathbf{x}^{(3)}$	0	1	0	0
$\mathbf{x}^{(4)}$	1	0	0	1
$\mathbf{x}^{(5)}$	0	0	0	1

1、将数据集转换为d个二类分类问题（每个标签各一个）



X	Y_1	X	Y_2	X	Y_3	X	Y_4
$\mathbf{x}^{(1)}$	0	$\mathbf{x}^{(1)}$	1	$\mathbf{x}^{(1)}$	1	$\mathbf{x}^{(1)}$	0
$\mathbf{x}^{(2)}$	1	$\mathbf{x}^{(2)}$	0	$\mathbf{x}^{(2)}$	0	$\mathbf{x}^{(2)}$	0
$\mathbf{x}^{(3)}$	0	$\mathbf{x}^{(3)}$	1	$\mathbf{x}^{(3)}$	0	$\mathbf{x}^{(3)}$	0
$\mathbf{x}^{(4)}$	1	$\mathbf{x}^{(4)}$	0	$\mathbf{x}^{(4)}$	0	$\mathbf{x}^{(4)}$	1
$\mathbf{x}^{(5)}$	0	$\mathbf{x}^{(5)}$	0	$\mathbf{x}^{(5)}$	0	$\mathbf{x}^{(5)}$	1

2、为每个问题训练一个二类分类器

转换方式：拷贝和选择

样本	属性	标记
1	x_1	$\{l_1, l_4\}$
2	x_2	$\{l_3, l_4\}$
3	x_3	$\{l_1\}$
4	x_4	$\{l_2, l_3, l_4\}$

样本	标签
1a	l_1
1b	l_4
2a	l_3
2b	l_4
3	l_1
4a	l_2
4b	l_3
4c	l_4

(a)copy

样本	标签	权重
1a	l_1	0.5
1b	l_4	0.5
2a	l_3	0.5
2b	l_4	0.5
3	l_1	1
4a	l_2	0.33
4b	l_3	0.33
4c	l_4	0.33

(b)copy-weight

样本	标签
1	l_4
2	l_4
3	l_1
4	l_4

(c)select-max

样本	标签
1	l_1
2	l_3
3	l_1
4	l_2

(d)select-min

样本	标签
1	l_1
2	l_4
3	l_1
4	l_3

(e)select-random

Binary Relevance 的局限

- Binary Relevance 忽略了标签之间的依赖，比如：

$$p(\mathbf{y}|\mathbf{x}) \propto p(\mathbf{x}) \prod_{j=1}^L p(y_j|\mathbf{x})$$

并不总是成立。

比如在对电影类型分类时，有：

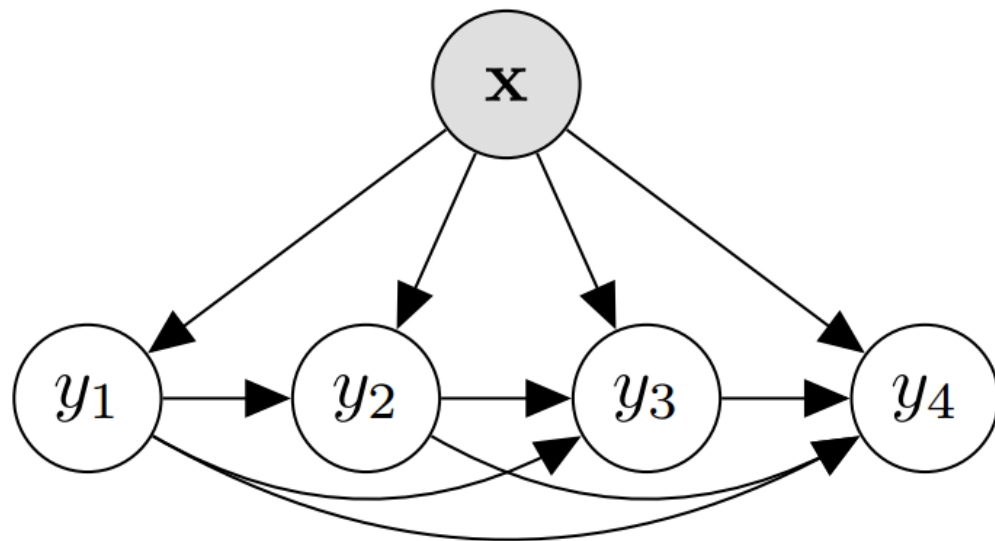
$$p(Y_{romance} | \mathbf{X}) \neq p(Y_{romance} | \mathbf{X}, Y_{horror})$$

- 数据不平衡：

$$p(Y_{romance}) \gg p(\bar{Y}_{romance})$$

克服标签依赖: Classifier Chains (CC)

对标签之间的依赖建模:



$$p(\mathbf{y}|\mathbf{x}) \propto p(\mathbf{x}) \prod_{j=1}^L p(y_j|\mathbf{x}, y_1, \dots, y_{j-1})$$

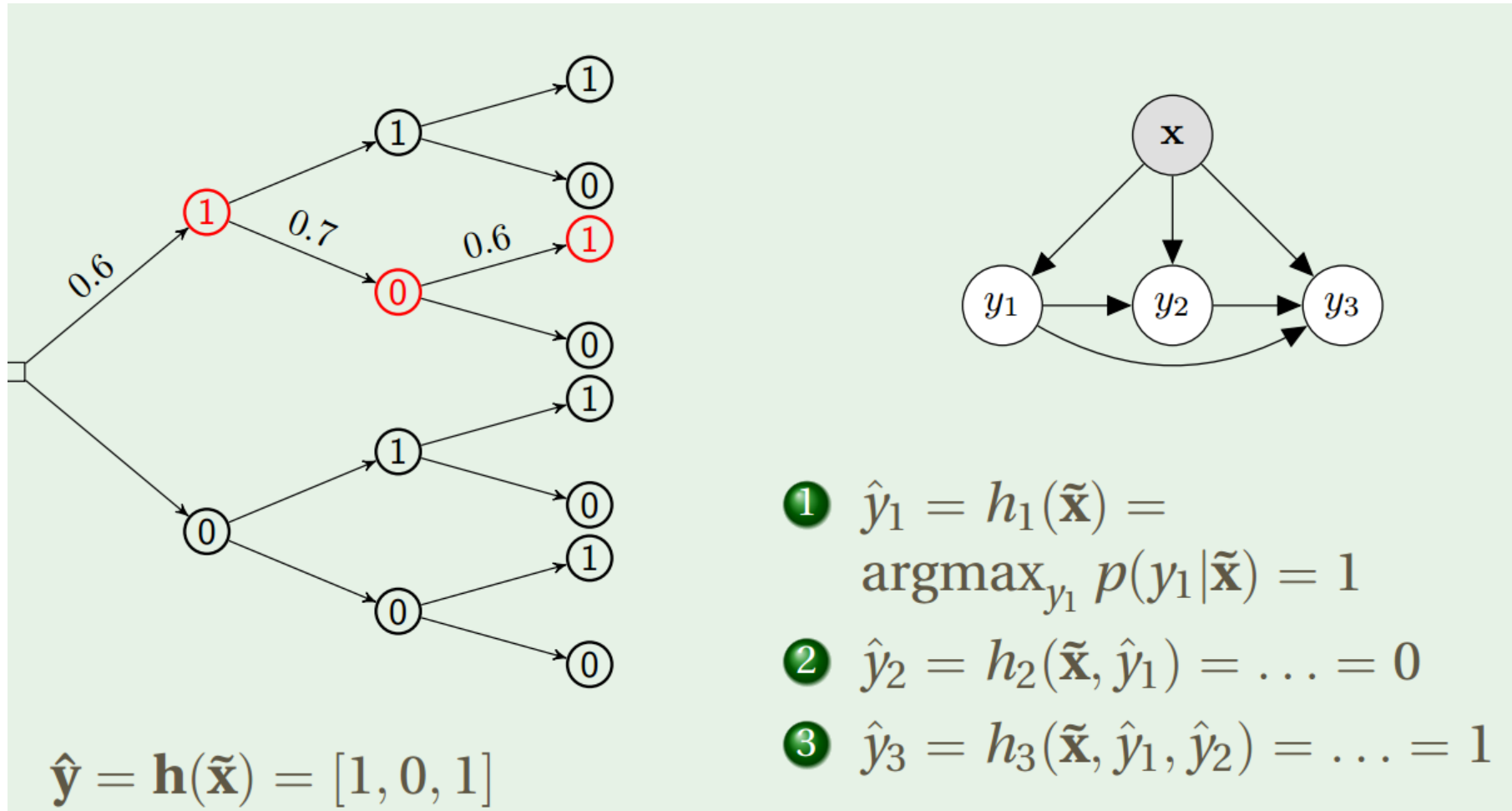
$$\hat{\mathbf{y}} = \operatorname{argmax}_{\mathbf{y} \in \{0,1\}^L} p(\mathbf{y}|\mathbf{x})$$

Classifier Chains: Stack

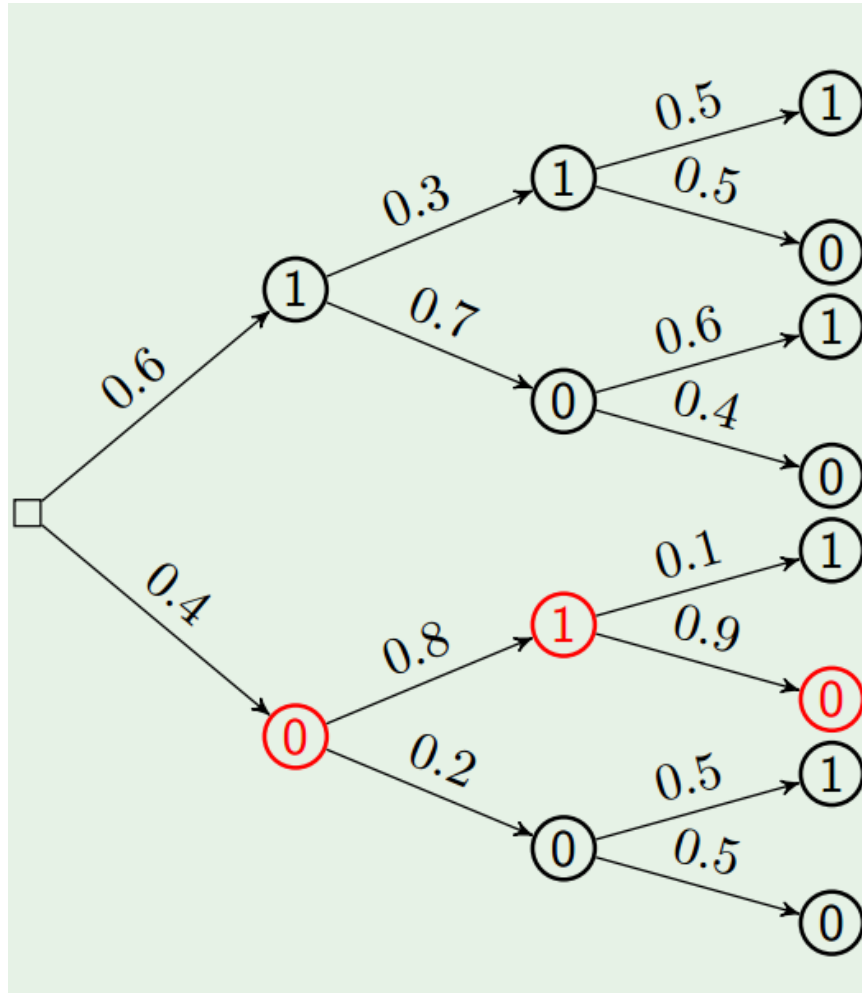
X	Y_1	X	Y_1	Y_2	X	Y_1	Y_2	Y_3	X	Y_1	Y_3	Y_3	Y_4
$\mathbf{x}^{(1)}$	0	$\mathbf{x}^{(1)}$	0	1	$\mathbf{x}^{(1)}$	0	1	1	$\mathbf{x}^{(1)}$	0	1	1	0
$\mathbf{x}^{(2)}$	1	$\mathbf{x}^{(2)}$	1	0	$\mathbf{x}^{(2)}$	1	0	0	$\mathbf{x}^{(2)}$	1	0	0	0
$\mathbf{x}^{(3)}$	0	$\mathbf{x}^{(3)}$	0	1	$\mathbf{x}^{(3)}$	0	1	0	$\mathbf{x}^{(3)}$	0	1	0	0
$\mathbf{x}^{(4)}$	1	$\mathbf{x}^{(4)}$	1	0	$\mathbf{x}^{(4)}$	1	0	0	$\mathbf{x}^{(4)}$	1	0	0	1
$\mathbf{x}^{(5)}$	0	$\mathbf{x}^{(5)}$	0	0	$\mathbf{x}^{(5)}$	0	0	0	$\mathbf{x}^{(5)}$	0	0	0	1

- 和Binary Relevance 类似，创建d个子问题，但是每训练一个分类器之前都需要把上一次的分类结果作为特征加入训练

Greedy Classifier Chain



Bayes Optimal Classifier Chain



① $p(\mathbf{y} = [0, 0, 0]) = 0.040$

② $p(\mathbf{y} = [0, 0, 1]) = 0.040$

③ $p(\mathbf{y} = [0, 1, 0]) = 0.288$

④ ...

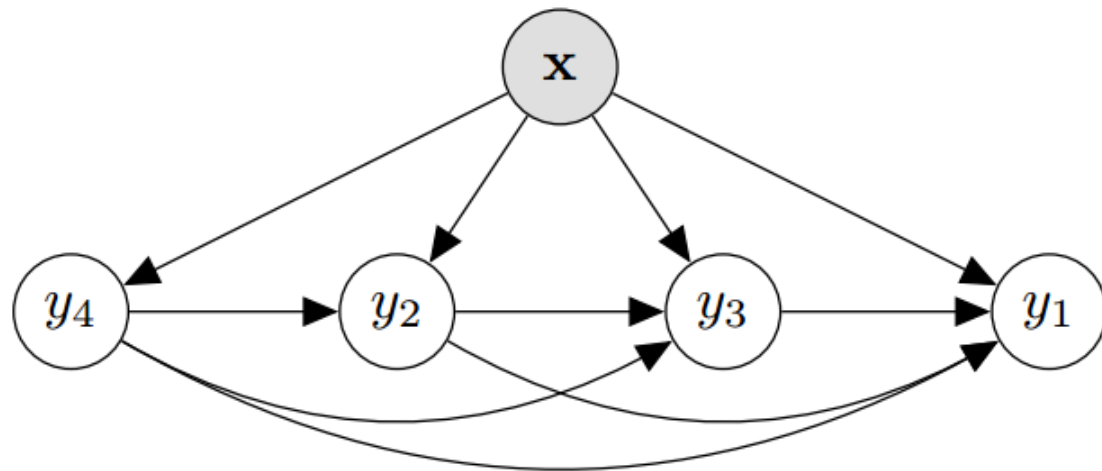
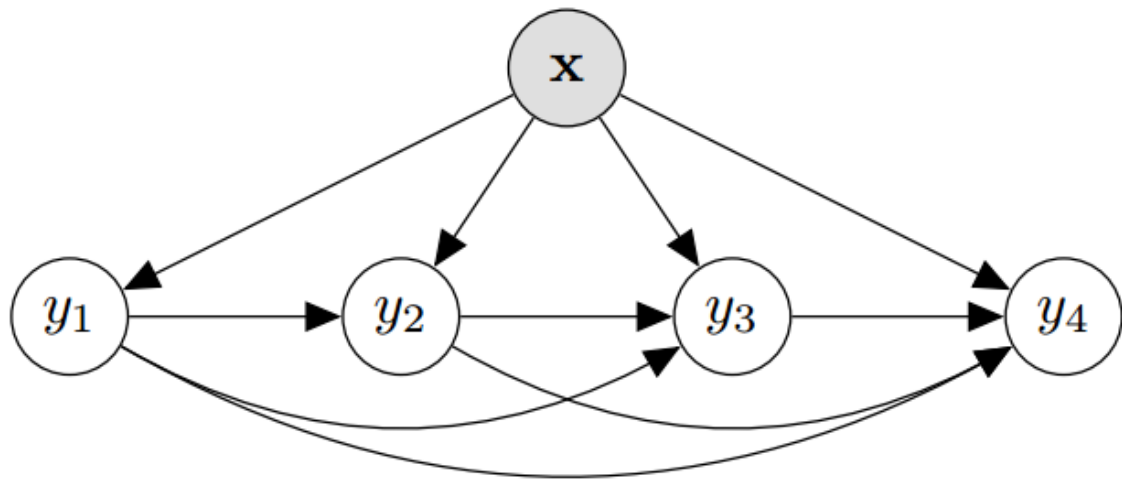
⑥ $p(\mathbf{y} = [1, 0, 1]) = 0.252$

⑦ ...

⑧ $p(\mathbf{y} = [1, 1, 1]) = 0.090$

return $\operatorname{argmax}_{\mathbf{y}} p(\mathbf{y}|\tilde{\mathbf{x}})$

Classifier Chains: 标签前后顺序



理想状态: $p(\mathbf{y}|\mathbf{x}) = p(y_1|\mathbf{x})p(y_2|y_1, \mathbf{x}) = p(y_2|\mathbf{x})p(y_1|y_2, \mathbf{x})$

样本有限且有噪声: $\hat{p}(y_1|\mathbf{x})\hat{p}(y_2|y_1, \mathbf{x}) \neq \hat{p}(y_2|\mathbf{x})\hat{p}(y_1|y_2, \mathbf{x})$

贪心CC: $\hat{p}(y_2|y_1, \mathbf{x}) \approx \hat{p}(y_2|\hat{y}_1, \mathbf{x}) = \hat{p}(y_2|y_1 = \underset{y_1}{\operatorname{argmax}} \hat{p}(y_1|\mathbf{x})|\mathbf{x})$

Label Powerset: 数据集转换

X	Y_1	Y_2	Y_3	Y_4
$\mathbf{x}^{(1)}$	0	1	1	0
$\mathbf{x}^{(2)}$	1	0	0	0
$\mathbf{x}^{(3)}$	0	1	1	0
$\mathbf{x}^{(4)}$	1	0	0	1
$\mathbf{x}^{(5)}$	0	0	0	1

1、将数据集转换1个多分类问题，可能有 2^d



X	$Y \in 2^L$
$\mathbf{x}^{(1)}$	0110
$\mathbf{x}^{(2)}$	1000
$\mathbf{x}^{(3)}$	0110
$\mathbf{x}^{(4)}$	1001
$\mathbf{x}^{(5)}$	0001

2、为每个问题训练一个二类分类器

Label Powerset 的局限

- 复杂度： 2^d 种组合
- 数据集平衡：每个类别的样本很少
- 过拟合：无法预测训练集中没出现过的类别

Label Powerset的改进: Pruned Problem Transformation

Doc.	Labels ($S \subseteq L$)
d_1	{Sports, Science}
d_2	{Environment, Science, Politics}
d_3	{Sports}
d_4	{Environment, Science}
d_5	{Science}
d_6	{Sports}
d_7	{Environment, Science}
d_8	{Politics}
d_9	{Politics}
d_{10}	{Science}

Doc.	Labels ($S \subseteq L$)
d_1	{Sports, Science}
d_2	{Environment, Science, Politics}

Doc.	Labels ($S \subseteq L$)
d_1	{Sports}
d_1	{Science}
d_2	{Environment, Science}
d_2	{Politics}

Pruned Problem Transformation : 数据集转换

Doc.	Labels ($S \subseteq L$)
d_1	{Sports, Science}
d_2	{Environment, Science, Politics}
d_3	{Sports}
d_4	{Environment, Science}
d_5	{Science}
d_6	{Sports}
d_7	{Environment, Science}
d_8	{Politics}
d_9	{Politics}
d_{10}	{Science}



Doc.	Labels ($S \subseteq L$)
d_1	{Sports}
d_1	{Science}
d_2	{Environment, Science}
d_2	{Politics}
d_3	{Sports}
d_4	{Environment, Science}
d_5	{Science}
d_6	{Sports}
d_7	{Environment, Science}
d_8	{Politics}
d_9	{Politics}
d_{10}	{Science}

Random k-Labelsets: 数据集转换

\mathbf{X}	$Y \in 2^L$
$\mathbf{x}^{(1)}$	0110
$\mathbf{x}^{(2)}$	1000
$\mathbf{x}^{(3)}$	0110
$\mathbf{x}^{(4)}$	1001
$\mathbf{x}^{(5)}$	0001



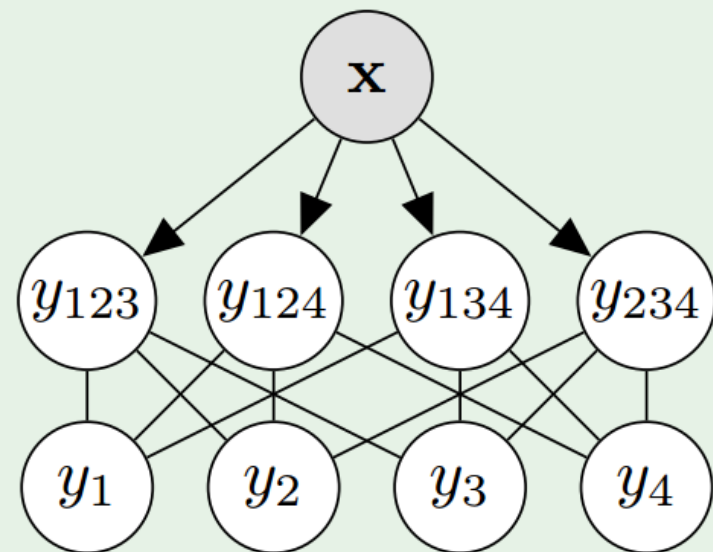
\mathbf{X}	$Y_{123} \in 2^k$
$\mathbf{x}^{(1)}$	011
$\mathbf{x}^{(2)}$	100
$\mathbf{x}^{(3)}$	011
$\mathbf{x}^{(4)}$	100
$\mathbf{x}^{(5)}$	000

\mathbf{X}	$Y_{124} \in 2^k$
$\mathbf{x}^{(1)}$	010
$\mathbf{x}^{(2)}$	100
$\mathbf{x}^{(3)}$	010
$\mathbf{x}^{(4)}$	101
$\mathbf{x}^{(5)}$	001

\mathbf{X}	$Y_{234} \in 2^k$
$\mathbf{x}^{(1)}$	110
$\mathbf{x}^{(2)}$	000
$\mathbf{x}^{(3)}$	110
$\mathbf{x}^{(4)}$	001
$\mathbf{x}^{(5)}$	001

Random k-Labelsets的集成：投票

	\hat{y}_1	\hat{y}_2	\hat{y}_3	\hat{y}_4
$\mathbf{h}^1(\tilde{\mathbf{x}})$	1	1	1	
$\mathbf{h}^2(\tilde{\mathbf{x}})$		0	1	0
$\mathbf{h}^3(\tilde{\mathbf{x}})$	1		0	0
$\mathbf{h}^4(\tilde{\mathbf{x}})$	1	0		0
score	0.75	0.25	0.75	0
$\hat{\mathbf{y}}$	1	0	1	0



标签排序：数据集转换

X	Y_1	Y_2	Y_3	Y_4
$\mathbf{x}^{(1)}$	0	1	1	0
$\mathbf{x}^{(2)}$	1	0	0	0
$\mathbf{x}^{(3)}$	0	1	0	0
$\mathbf{x}^{(4)}$	1	0	0	1
$\mathbf{x}^{(5)}$	0	0	0	1



X	$Y_{1 \vee 2}$
$\mathbf{x}^{(1)}$	0
$\mathbf{x}^{(2)}$	1
$\mathbf{x}^{(3)}$	0
$\mathbf{x}^{(4)}$	1

X	$Y_{1 \vee 4}$
$\mathbf{x}^{(2)}$	1
$\mathbf{x}^{(5)}$	0

X	$Y_{2 \vee 3}$
$\mathbf{x}^{(3)}$	1

X	$Y_{2 \vee 4}$
$\mathbf{x}^{(1)}$	1
$\mathbf{x}^{(3)}$	1
$\mathbf{x}^{(4)}$	0
$\mathbf{x}^{(5)}$	0

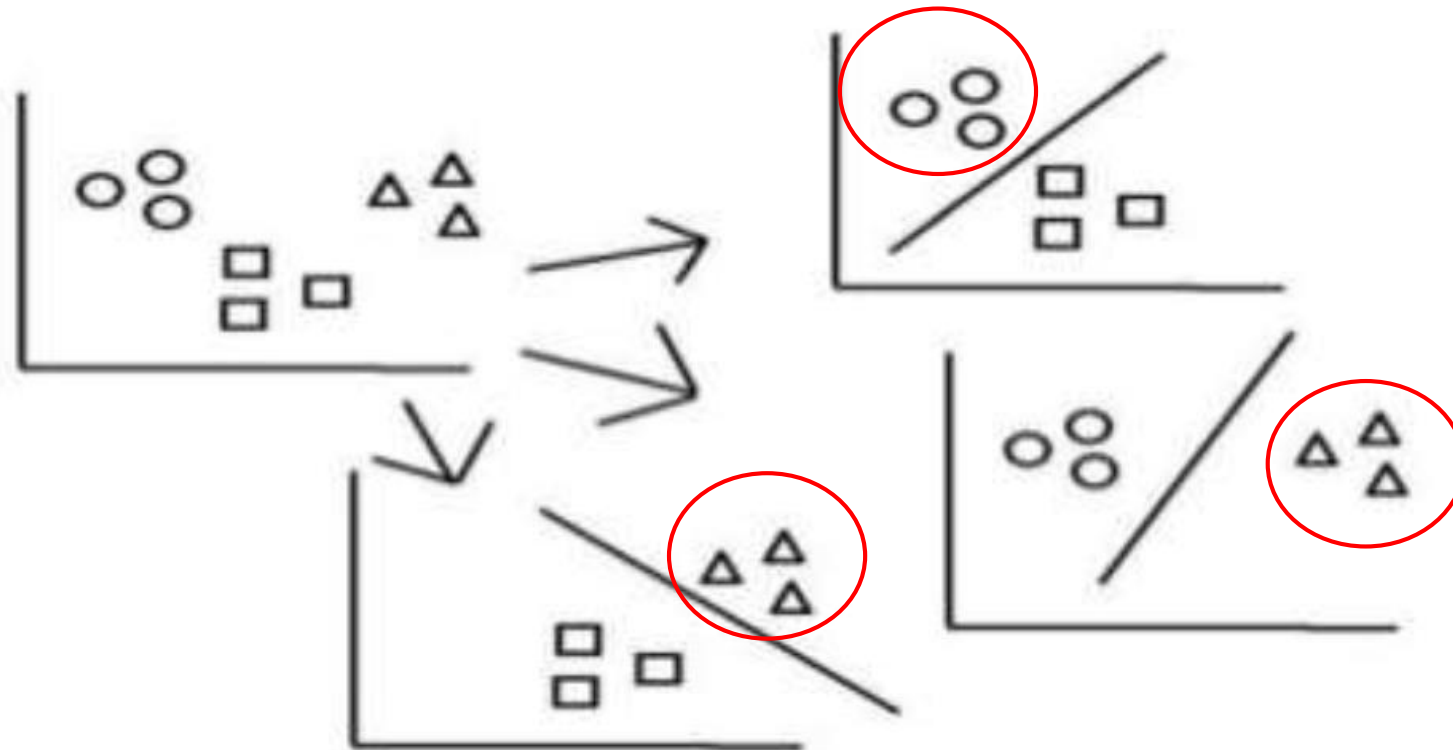
X	$Y_{1 \vee 3}$
$\mathbf{x}^{(1)}$	0
$\mathbf{x}^{(2)}$	1
$\mathbf{x}^{(4)}$	1

X	$Y_{3 \vee 4}$
$\mathbf{x}^{(1)}$	1
$\mathbf{x}^{(4)}$	0
$\mathbf{x}^{(5)}$	0

构造 $\frac{d \cdot (d-1)}{2}$ 个二类分类子问题（all
V.S. all），并训练 $\frac{d \cdot (d-1)}{2}$ 个分类器

如何通过若干个分类器做预测？

标签排序：One V.S. One



标签排序：Calibrated Label Ranking (CLR)

- 用一个“虚拟标签” λ_0 作为“人工分割点”来分割标签排序：

$$\lambda_1 \succ \lambda_3 \succ \lambda_0 \succ \lambda_4 \succ \lambda_2 \dots$$

虚拟标签位于所有相关标签之后，位于所有无关标签之前

Calibrated Label Ranking: λ_j 的计算

训练集构造: $\mathcal{D}_{jk} = \{(\mathbf{x}_i, \psi(Y_i, y_j, y_k)) \mid \phi(Y_i, y_j) \neq \phi(Y_i, y_k), 1 \leq i \leq d\}$

(其中, $1 \leq j \leq k \leq d$)

$$\text{where } \psi(Y_i, y_j, y_k) = \begin{cases} +1, & \text{if } \phi(Y_i, y_j) = +1 \text{ and } \phi(Y_i, y_k) = -1 \\ -1, & \text{if } \phi(Y_i, y_j) = -1 \text{ and } \phi(Y_i, y_k) = +1 \end{cases}$$

$$\lambda_j \text{ 的计算: } \zeta(\mathbf{x}, y_j) = \sum_{k=1}^{j-1} \mathbb{I}[g_{kj}(\mathbf{x}) \leq 0] + \sum_{k=j+1}^q \mathbb{I}[g_{jk}(\mathbf{x}) > 0] \quad (1 \leq j \leq q)$$

$g_{kj}(\mathbf{x})$ 为二类分类问题的分类器。当前类别 i 时, 对于序号比 i 小的类 $g_{kj}(\mathbf{x}) \leq 0$ 表示它被分到类别 i 的概率, 对于序号比 i 大的类, $g_{kj}(\mathbf{x}) > 0$ 表示它被分到类别 i 的概率

问题转换法总结

- 对于d个标签的多标签问题，可以从以下3个角度看：
 - d个二类分类问题： Binary Relevance
 - 2^d 个类的多分类问题： Label Powerset
 - 或者上面2种方法的结合

- 通常的流程：
 - 将原问题转换为子问题：二分类或多分类问题
 - 应用一些现有的基分类器
 - 正则化：可选
 - 集成：可选

算法适应法

■ 如何做？

- 选择1个现有的算法
- 修改使之适应多标签分类

■ 通常的流程：

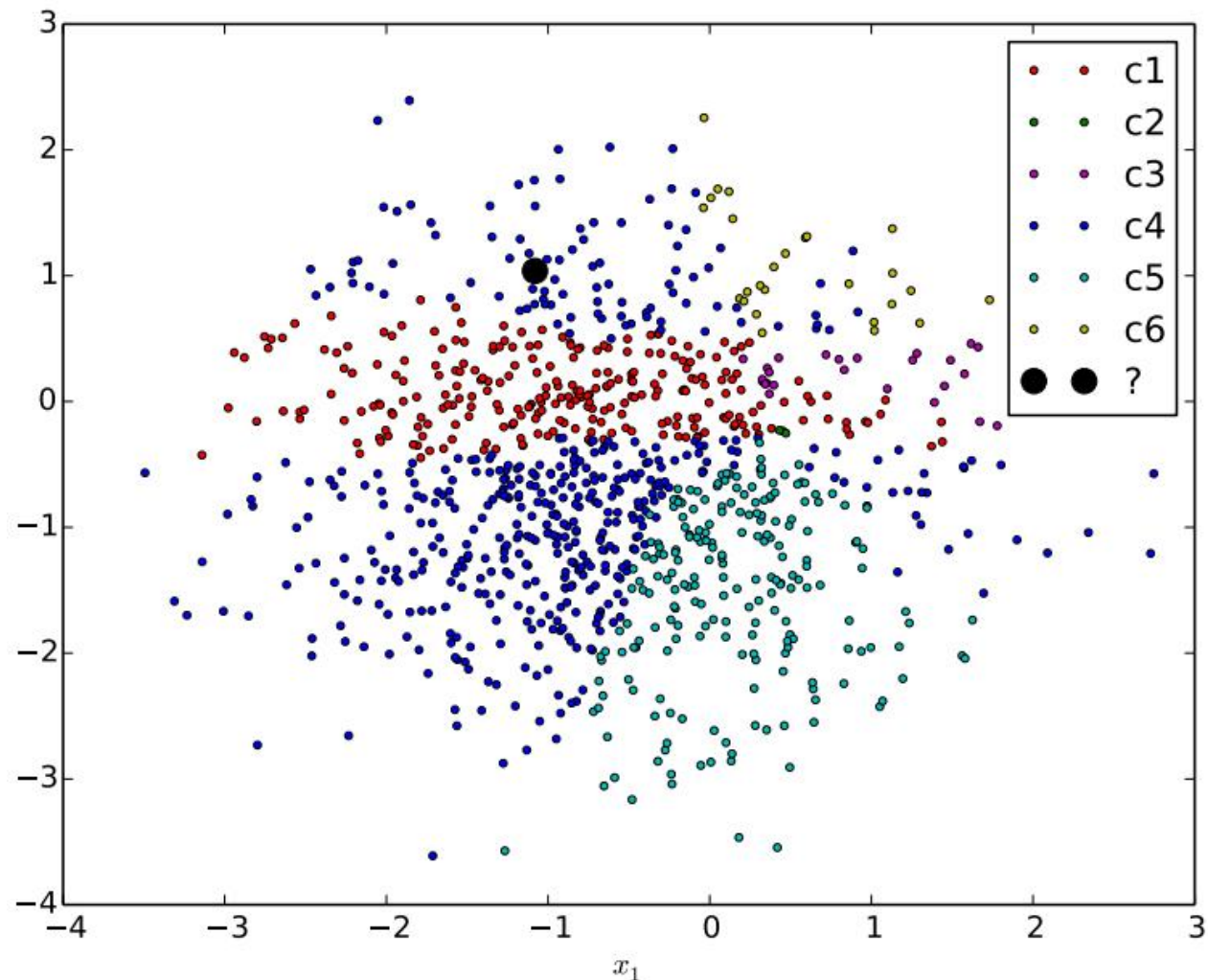
- 优点：单一模型，扩展性强
- 缺点：模型性能依赖于特定领域

ML-kNN: 回顾 kNN

样本 \hat{x} 的类型为它 k 个最近邻居中的大多数类:

$$\hat{y} = \operatorname{argmax}_y \sum_{i \in N_k} y^{(i)}$$

其中, N_k 包括了与 \hat{x} 最近的样本, 记为 $y^{(i)}$



ML-kNN: 最大化后验概率

$$p(y_j = 1 | \mathbf{x}) = \frac{1}{k} \sum_{i \in N_k} y_j^{(i)}$$

$$\hat{y}_j = \operatorname{argmax}_{y_j \in \{0,1\}} [p(y_j | \mathbf{x}) > 0.5]$$

引入事件 C_j : N_k 个最近邻居中, y_j 作为其相关标记的样本个数, 则有 $0 \leq C_j \leq k$:

$$C_j = \sum_{i \in N_k} y_j^{(i)}$$

计算: $P(y_j = 1 | C_j)$

ML-kNN: 贝叶斯公式

$$P(y_j = 1 | C_j) = \frac{P(y_j=1) \cdot P(C_j | y_j=1)}{P(C_j)}$$

$$P(y_j = 0 | C_j) = \frac{P(y_j=0) \cdot P(C_j | y_j=0)}{P(C_j)}$$

$$h(\cdot) = \frac{P(y_j=1 | C_j)}{P(y_j=0 | C_j)} = \frac{P(y_j=1) \cdot P(C_j | y_j=1)}{P(y_j=0) \cdot P(C_j | y_j=0)} > 1 ?$$

ML-kNN: 计算先验概率

$$P(y_j = 1) = \frac{s + \sum_{i=1}^m [y_i \in Y_j]}{s \times 2 + m};$$

$$P(y_j = 0) = 1 - P(y_j = 1);$$

其中， m 为训练集样本个数， s 为平滑参数

ML-kNN: 计算条件概率

因为 $0 \leq C_j \leq k$, 确定2个长度为 $k+1$ 的数组: $k_j[r]$ 和 $\bar{k}_j[r]$

$$k_j[r] = \sum_{i=1}^m [y_j \in Y_j] \cdot [\delta_j(x_j) = r], 0 \leq r \leq k$$

$$\bar{k}_j[r] = \sum_{i=1}^m [y_j \notin Y_j] \cdot [\delta_j(x_j) = r], 0 \leq r \leq k$$

其中, $\delta_j(x_i) = \sum_{i \in N_k} y_j^{(i)}$, 统计了第 i 个训练样本的 k 近邻中, 将 y_j 作为其相关标签的近邻个数

ML-kNN是几阶？

$$h(\cdot) = \frac{P(y_j=1 | C_j)}{P(y_j=0 | C_j)}$$

$$= \frac{P(y_j=1) \cdot P(C_j | y_j=1)}{P(y_j=0) \cdot P(C_j | y_j=0)} > 1 ?$$

$[\vec{y}_t, \vec{r}_t] = \text{ML-kNN}(S, k, t, s)$

%Computing the prior probabilities $P(H_i^l)$

- (1) **for** $l \in \mathcal{Y}$ **do**
- (2) $P(H_1^l) = (s + \sum_{i=1}^m \vec{y}_{x_i}(l)) / (s \times 2 + m)$;
- (3) $P(H_0^l) = 1 - P(H_1^l)$;

%Computing the posterior probabilities $P(E_j^l | H_i^l)$

- (4) **Identify** $N(x_i)$, $i \in \{1, \dots, m\}$;
- (5) **for** $l \in \mathcal{Y}$ **do**
- (6) **for** $j \in \{0, \dots, k\}$ **do**
- (7) $c[j] = 0$; $c'[j] = 0$;
- (8) **for** $i \in \{1, \dots, m\}$ **do**
- (9) $\delta = \vec{C}_{x_i}(l) = \sum_{a \in N(x_i)} \vec{y}_{x_a}(l)$;
- (10) **if** $(\vec{y}_{x_i}(l) == 1)$ **then** $c[\delta] = c[\delta] + 1$;
- (11) **else** $c'[\delta] = c'[\delta] + 1$;
- (12) **for** $j \in \{0, \dots, k\}$ **do**
- (13) $P(E_j^l | H_1^l) = \frac{s + c[j]}{s \times (k+1) + \sum_{p=0}^k c[p]}$;
- (14) $P(E_j^l | H_0^l) = \frac{s + c'[j]}{s \times (k+1) + \sum_{p=0}^k c'[p]}$;

%Computing \vec{y}_t and \vec{r}_t

- (15) **Identify** $N(t)$;
- (16) **for** $l \in \mathcal{Y}$ **do**
- (17) $\vec{C}_t(l) = \sum_{a \in N(t)} \vec{y}_{x_a}(l)$;
- (18) $\vec{y}_t(l) = \arg \max_{b \in \{0,1\}} P(H_b^l) P(E_{\vec{C}_t(l)}^l | H_b^l)$;
- (19) $\vec{r}_t(l) = P(H_1^l | E_{\vec{C}_t(l)}^l)$
 $= P(H_1^l) P(E_{\vec{C}_t(l)}^l | H_1^l) / P(E_{\vec{C}_t(l)}^l)$
 $= \frac{P(H_1^l) P(E_{\vec{C}_t(l)}^l | H_1^l)}{\sum_{b \in \{0,1\}} P(H_b^l) P(E_{\vec{C}_t(l)}^l | H_b^l)}$;

回顾C4.5决策树学习算法

用增益率来进行决策树的划分属性选择

启发式规则：先从候选划分属性中找出信息增益高于平均水平的属性，再从其中选择增益率最高的

$$\text{Gain_ratio}(D, a) = \frac{\text{Gain}(D, a)}{\text{IV}(a)}$$

$$\text{Gain}(D, a) = \text{Ent}(D) - \sum_{v=1}^V \frac{|D^v|}{|D|} \text{Ent}(D^v) .$$

$$\text{IV}(a) = - \sum_{v=1}^V \frac{|D^v|}{|D|} \log_2 \frac{|D^v|}{|D|}$$

属性 a 的固有值，即为数据集 D 关于特征 a 的熵。属性 a 取值越多，熵值就会越大

ML-DT

- 将决策树C4.5的熵计算扩展到多个标签:

$$entropy(S) = - \sum_{i=1}^N p(c_i) \log p(c_i) \quad (\text{单个类标签, 多选一})$$



$$- \sum_{i=1}^N ((p(c_i) \log p(c_i)) + (q(c_i) \log q(c_i))) \quad \text{其中, } q(c_i) = 1 - p(c_i)$$

(每个标签都二选一)

- 构造过程和C4.5决策树一样
- 允许叶节点有多个标签

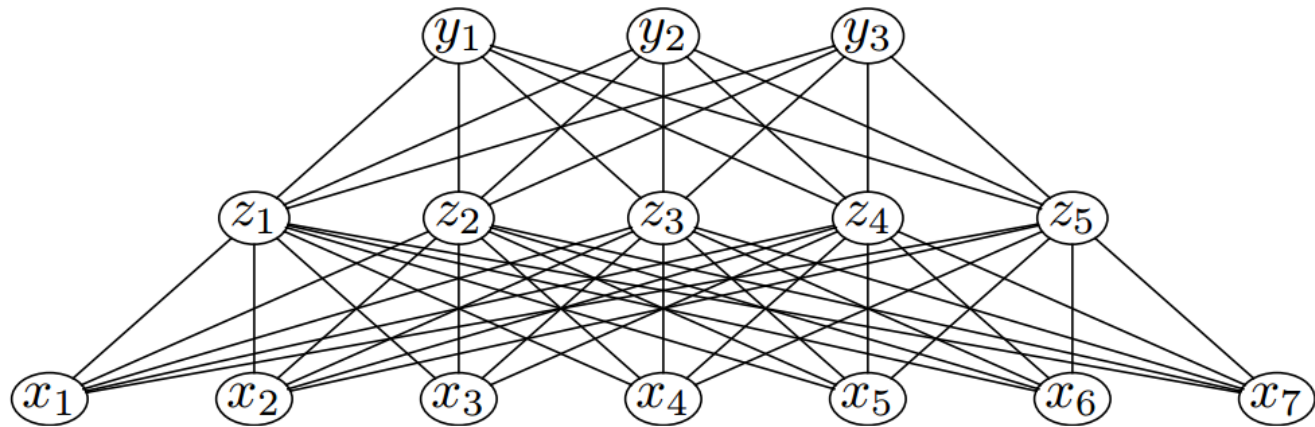
神经网络：BP-MLL

■ BP神经网络

- 反向传播，输出层有 **多个输出**
- 训练：梯度下降 + 错误反向传播
- 一个基于排序的损失函数：

$$E = \sum_{i=1}^m E_i = \sum_{i=1}^m \frac{1}{|Y_i| |\bar{Y}_i|} \sum_{(k,l) \in Y_i \times \bar{Y}_i} \exp(-(c_k^i - c_l^i))$$

- 1个隐藏层
- 每个标签对应一个输出层神经元




BP-MLL的改进：交叉熵

$$J_{PWE}(\Theta; \mathbf{x}, \mathbf{y}) = \frac{1}{|\mathbf{y}||\bar{\mathbf{y}}|} \sum_{(p,n) \in \mathbf{y} \times \bar{\mathbf{y}}} \exp(-(o_p - o_n)) \quad (\text{非凸损失函数})$$

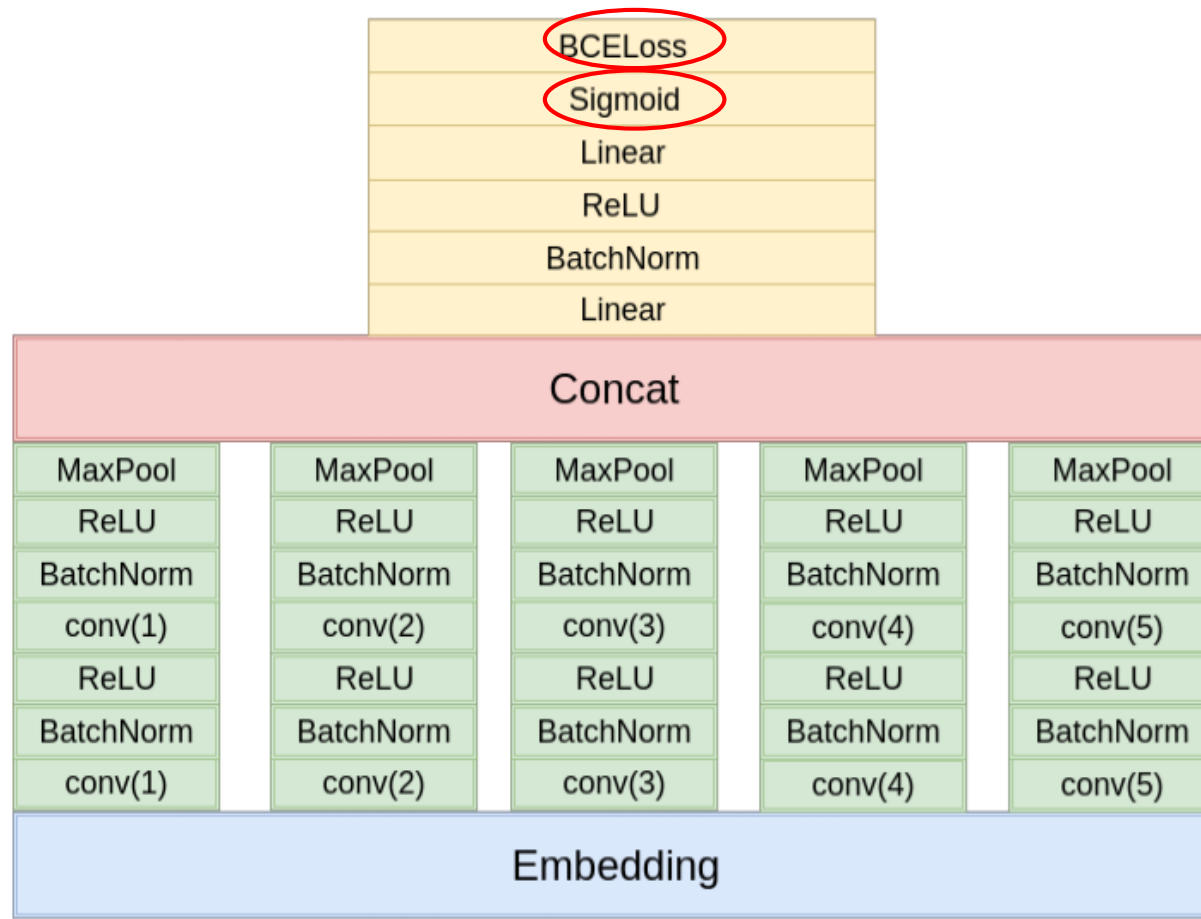
$$J_{log}(\Theta; \mathbf{x}, \mathbf{y}) = w(\mathbf{y}) \sum_l \log(1 + e^{-\dot{y}_l z_l}) \quad (\text{More Consistent, 仍是非凸})$$

$$\begin{aligned} \text{令 } w(\mathbf{y}) = \mathbf{1}: J_{log}(\Theta; \mathbf{x}, y_l) &= \log(1 + e^{-\dot{y}_l z_l}) = -\log\left(\frac{1}{1 + e^{-\dot{y}_l z_l}}\right) \\ &= -\log\left(\frac{1}{1 + e^{-\dot{y}_l z_l}}\right) = \begin{cases} -\log(\sigma(z_l)) & \text{if } \dot{y} = 1 \\ -\log(\sigma(-z_l)) & \text{if } \dot{y} = -1 \end{cases} \quad (\text{log 损失中: } \dot{y} \text{ 取 } -1 \text{ 或 } 1) \end{aligned}$$

如果输出层使用Sigmoid函数激活: $o_l = \sigma(z_l)$ 

$$\begin{aligned} &= J_{CE}(\Theta; \mathbf{x}, y_l) = -(y_l \log o_l + (1 - y_l) \log(1 - o_l)) \quad (y_l \text{ 取 } 0 \text{ 或 } 1) \\ &\quad (\text{二分类交叉熵损失函数}) \end{aligned}$$

二分类交叉熵的应用



试验： MEKA

■ 工具： MEKA

- 主页： <https://waikato.github.io/meka/>
- Jesse Read: <http://jmread.github.io/>

■ 数据集： Language Log

- 来源： <https://waikato.github.io/meka/datasets/>

Dataset	L	N	LC	PU	Description and Original Source(s)
Language Log	75	1460	1.18	0.208	Articles posted on the Language Log

- **N** = The number of examples (training+testing) in the datasets
- **L** = The number of predefined labels relevant to this dataset
- **LC** = Label Cardinality. Average number of labels assigned per document
- **PU** = Percentage of documents with Unique label combinations

参考文献

Zhang, M. L., & Zhou, Z. H. (2006). Multilabel neural networks with applications to functional genomics and text categorization. *IEEE transactions on Knowledge and Data Engineering*, 18(10), 1338-1351.

Zhang, M. L., & Zhou, Z. H. (2007). ML-KNN: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7), 2038-2048.

Clare, A., & King, R. D. (2001, September). Knowledge discovery in multi-label phenotype data. *In European Conference on Principles of Data Mining and Knowledge Discovery (pp. 42-53)*. Springer, Berlin, Heidelberg.

Nam, J., Kim, J., Mencía, E. L., Gurevych, I., & Fürnkranz, J. (2014, September). Large-scale multi-label text classification—revisiting neural networks. *In Joint european conference on machine learning and knowledge discovery in databases (pp. 437-452)*. Springer, Berlin, Heidelberg.

李娜, 潘志松, & 周星宇. (2016). 基于多标记重要性排序的分类器链算法. *模式识别与人工智能*, 29(6), 567-575.

周志华, & 张敏灵. (2009). *MIML 多示例多标记学习* [J] (Doctoral dissertation).