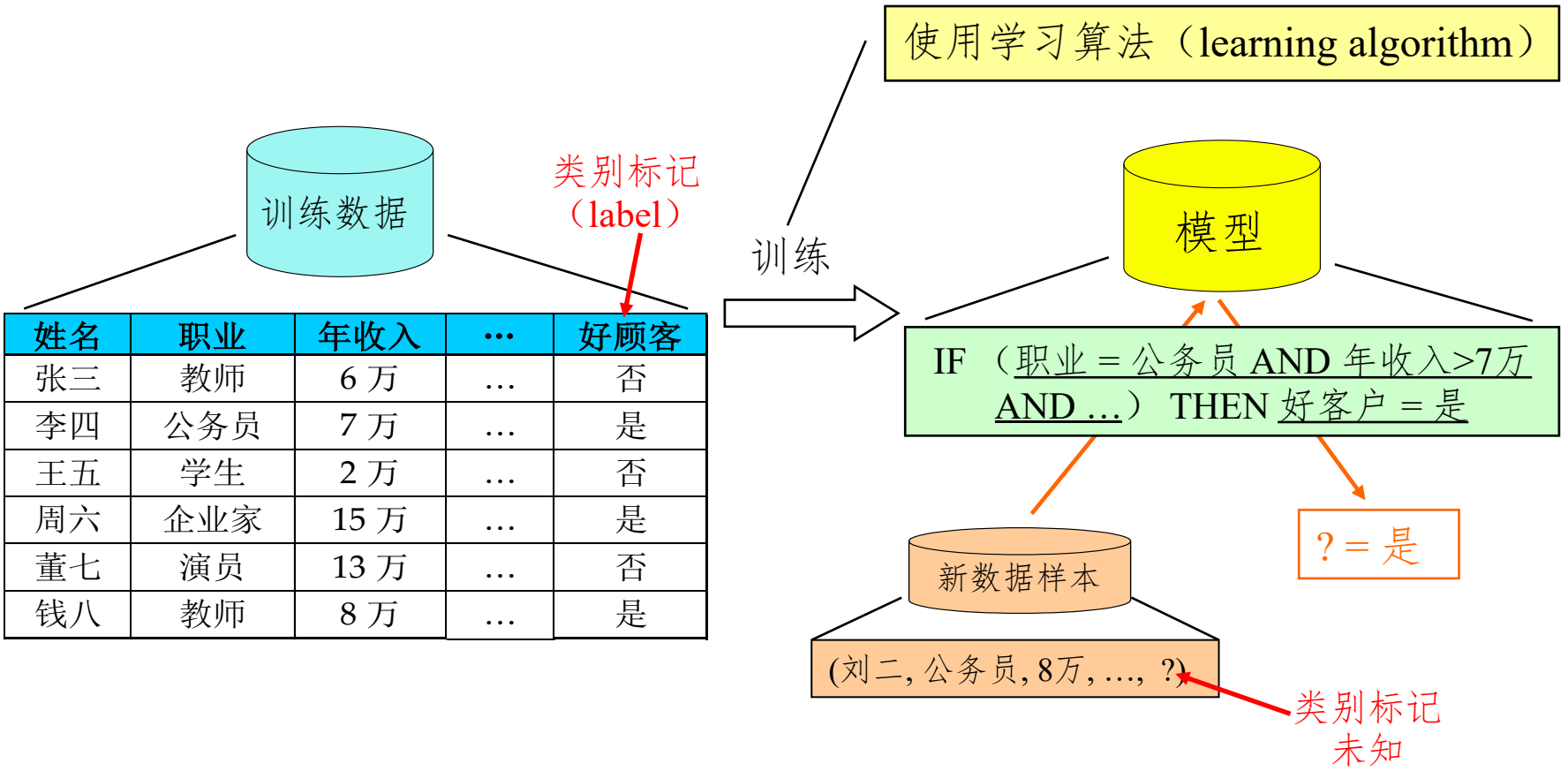
高级机器学习

多标记学习

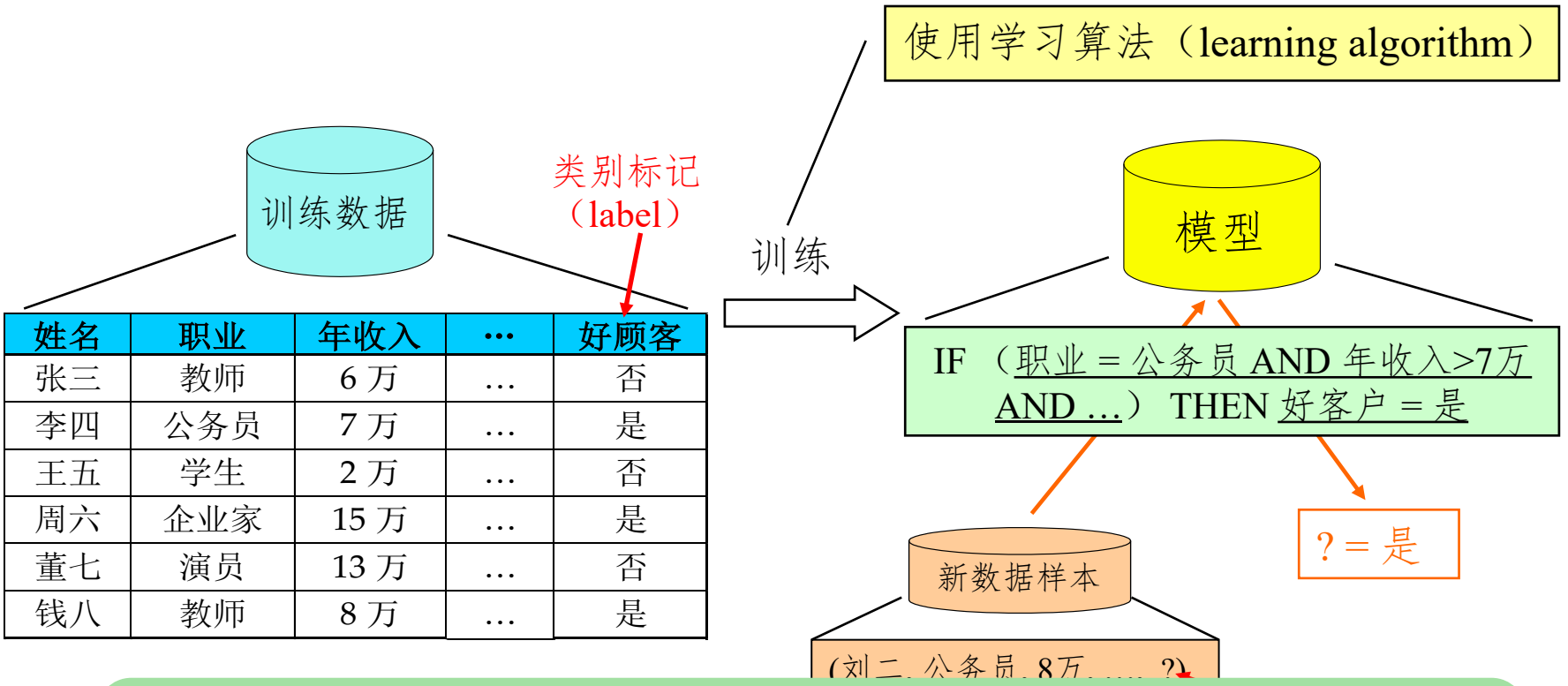
提纲

- 多标记学习
  - 经典算法
- 大规模多标记学习
  - 主流算法
- 总结

典型的机器学习过程


使用学习算法（learning algorithm）
训练数据
类别标记
（label）
训练
模型
姓名 职业 年收入 … 好顾客
张三 教师 6 万 … 否
李四 公务员 7 万 … 是
王五 学生 2 万 … 否
周六 企业家 15 万 … 是
董七 演员 13 万 … 否
钱八 教师 8 万 … 是
IF （职业 = 公务员 AND 年收入>7万 AND …） THEN 好客户 = 是
? = 是
新数据样本
(刘二, 公务员, 8万, …, ?)
类别标记
未知

典型的机器学习过程



姓名	职业	年收入	…	好顾客
张三	教师	6 万	…	否
李四	公务员	7 万	…	是
王五	学生	2 万	…	否
周六	企业家	15 万	…	是
董七	演员	13 万	…	否
钱八	教师	8 万	…	是

基本假设：每条数据仅与单个标记相关联

多标记数据示例

图片

相关标记：建筑、草、人

文本

2022年北京冬季奥运会 编辑 99+ 讨论 上传视频

同义词 2022年冬奥会一般指2022年北京冬季奥运会

第24届冬季奥林匹克运动会（英语：XXIV Winter Olympic Games；法语：XXIVes Jeux Olympiques d'hiver），又称2022年北京冬季奥运会。

2015年7月31日，托马斯·巴赫宣布2022年冬季奥运会的主办城市是北京。北京申办成功后，北京成为奥运史上第一个举办过夏季奥林匹克运动会和冬季奥林匹克运动会的城市，也是继1952年挪威的奥斯陆举办之后时隔整整70年后第二个举办冬奥会的首都城市。

北京与张家口成为中国第一个获得过冬季奥林匹克运动会举办权的城市 [1] 。

北京冬季奥运会设7个大项，15个分项 [2] ，109个小项。北京将主办冰上项目，张家口将主办雪上项目，延庆协办张家口举办雪上项目。

2017年12月15日20时22分，北京2022年冬奥会会徽“冬梦”和冬残奥会会徽“飞跃”，正式亮相。

第23届冬季奥林匹克运动会于2018年2月25日晚在平昌奥林匹克体育场闭幕。北京市市长陈吉宁接过奥运会会旗，标志着冬奥会进入“北京周期”。2018年8月8日，2022年北京冬奥会和冬残奥会吉祥物全球征集启动仪式在北京隆重举行。 [3-6] 2018年11月16日，北京冬奥会可持续性咨询和建议委员会正式成立并举办首次全体会议。 [7]

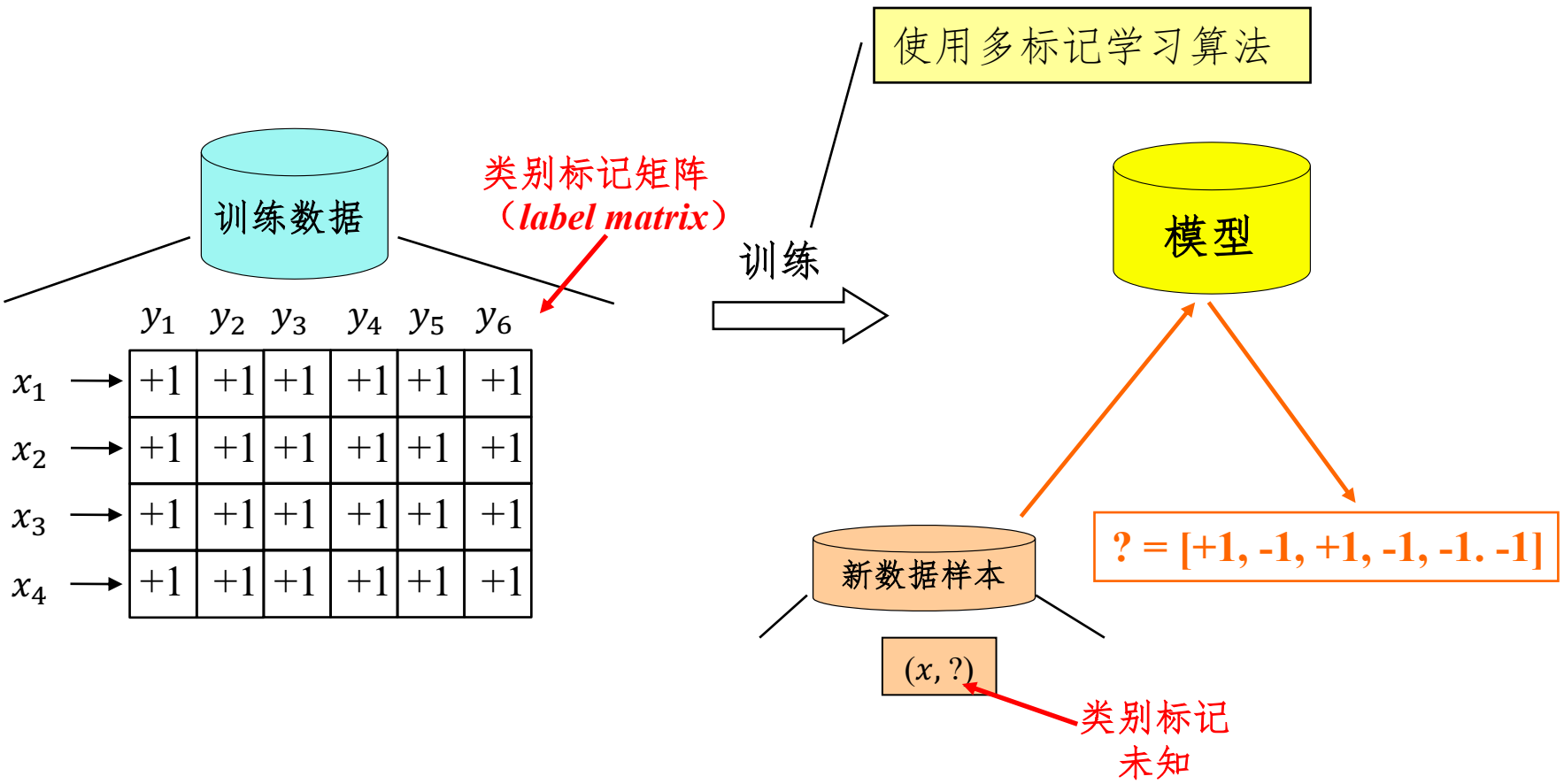
2019年11月6日上午，北京冬奥组委可持续性管理体系认证证书颁发仪式在北京冬奥组委首钢办公区举行。

2020年5月8日上午9点，北京冬奥会色彩系统和核心图形正式发布 [8] ；7月30日，“冬奥之声”全球传播活动正式启动 [9] 。

相关标记：奥运会、北京、体育

真实场景：每条数据可与多个标记相关联

多标记学习过程


使用多标记学习算法
训练数据
类别标记矩阵
（label matrix）
训练
模型
y_1 y_2 y_3 y_4 y_5 y_6
x_1 +1 +1 +1 +1 +1 +1
x_2 +1 +1 +1 +1 +1 +1
x_3 +1 +1 +1 +1 +1 +1
x_4 +1 +1 +1 +1 +1 +1
新数据样本
? = [+1, -1, +1, -1, -1. -1]
(x, ?)
类别标记
未知

经典多标记学习算法

- **将多标记转化为单标记问题**
  - 一阶方法: Binary Relevance (BR)
  - 二阶方法: Calibrated Label Ranking (CLR)
  - 高阶方法: Random k-labelsets (RAKEL)

- **多标记学习专有算法**
  - 二阶方法: Rank-SVM
  - 高阶方法: LEAD, ECC, CCE

经典多标记学习算法

- 将多标记转化为单标记问题
  - 一阶方法: Binary Relevance (BR)
  - 二阶方法: Calibrated Label Ranking (CLR)
  - 高阶方法: Random k-labelsets (RAKEL)

- 多标记学习专有算法
  - 二阶方法: Rank-SVM
  - 高阶方法: LEAD, ECC, CCE

Binary Relevance (BR) [Boutell et al., PRJ04]

基本思路：将多标记学习转化为多个独立的二分类问题

BR 算法

- 训练 q 个二分类模型，每个二分类器用来判别样本是否与标记 y_j 相关，$1 \le j \le q$
- 标记 y_j 分类器的训练集构造
  - 对于样本x_i，如果 $Y_{ij} = +1$ 且 $Y_{ik} = -1$，则 x_i 为正样本
  - 对于样本x_i，如果 $Y_{ij} = -1$ 且 $Y_{ik} = +1$，则 x_i 为负样本

Binary Relevance (BR) [Boutell et al., PRJ04]

基本思路：将多标记学习转化为多个独立的二分类问题

BR 算法

- 训练 q 个二分类模型，每个二分类器用来判别样本是否与标记 y_j 相关，$1 \le j \le q$
- 标记 y_j 分类器的训练集构造
  - 对于样本x_i，如果 $Y_{ij} = +1$ 且 $Y_{ik} = -1$，则 x_i 为正样本
  - 对于样本x_i，如果 $Y_{ij} = -1$ 且 $Y_{ik} = +1$，则 x_i 为负样本

测试时，综合所有二分类器的预测结果得到最终输出

Binary Relevance (BR) [Boutell et al., PRJ04]

基本思路：将多标记学习转化为多个独立的二分类问题

BR

- 训
- 标

优点: 方法简单、容易实现

缺点: 忽略标记关联, 可能导致性能损失

测试时，综合所有二分类器的预测结果得到最终输出

Calibrated Label Ranking (CLR) [Fürnkranz et al. MLJ08]

基本假设：将多标记学习转化为每对标签的排序问题

标签对排序算法

- 训练 $q(q-1)/2$ 个二分类模型，每个二分类器用来判别标签对 $(y_j, y_k), 1 \le j \le k \le q.$
- 标签对(y_j, y_k) 训练集构造
  - 对于样本x_i, 如果 $Y_{ij} = +1$ 且 $Y_{ik} = -1$, 则 x_i 为正样本
  - 对于样本x_i, 如果 $Y_{ij} = -1$ 且 $Y_{ik} = +1$, 则 x_i 为负样本
  - 忽略不满足任一上述条件的样本

Calibrated Label Ranking (CLR) [Fürnkranz et al. MLJ08]

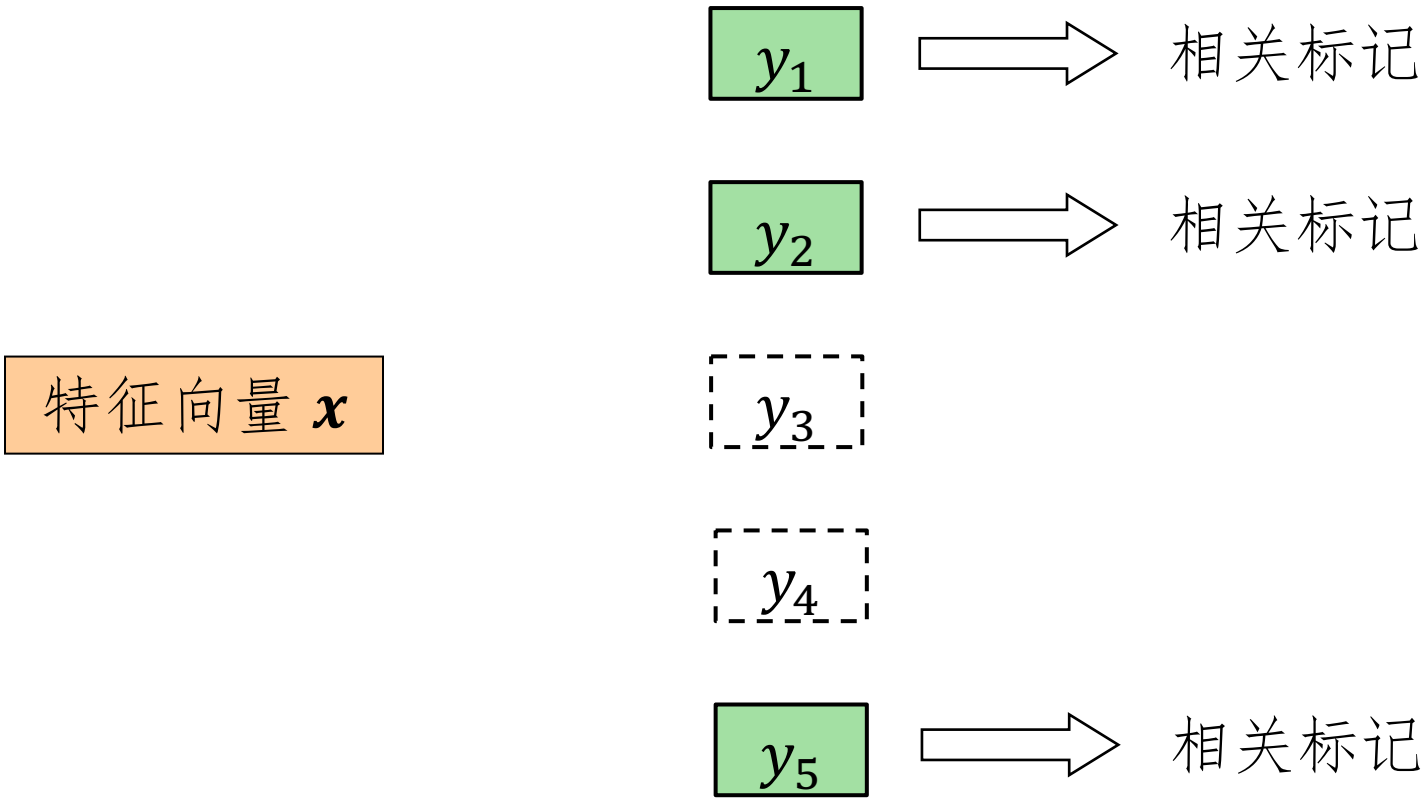
基本假设：将多标记学习转化为每对标签的排序问题

标签对排序算法

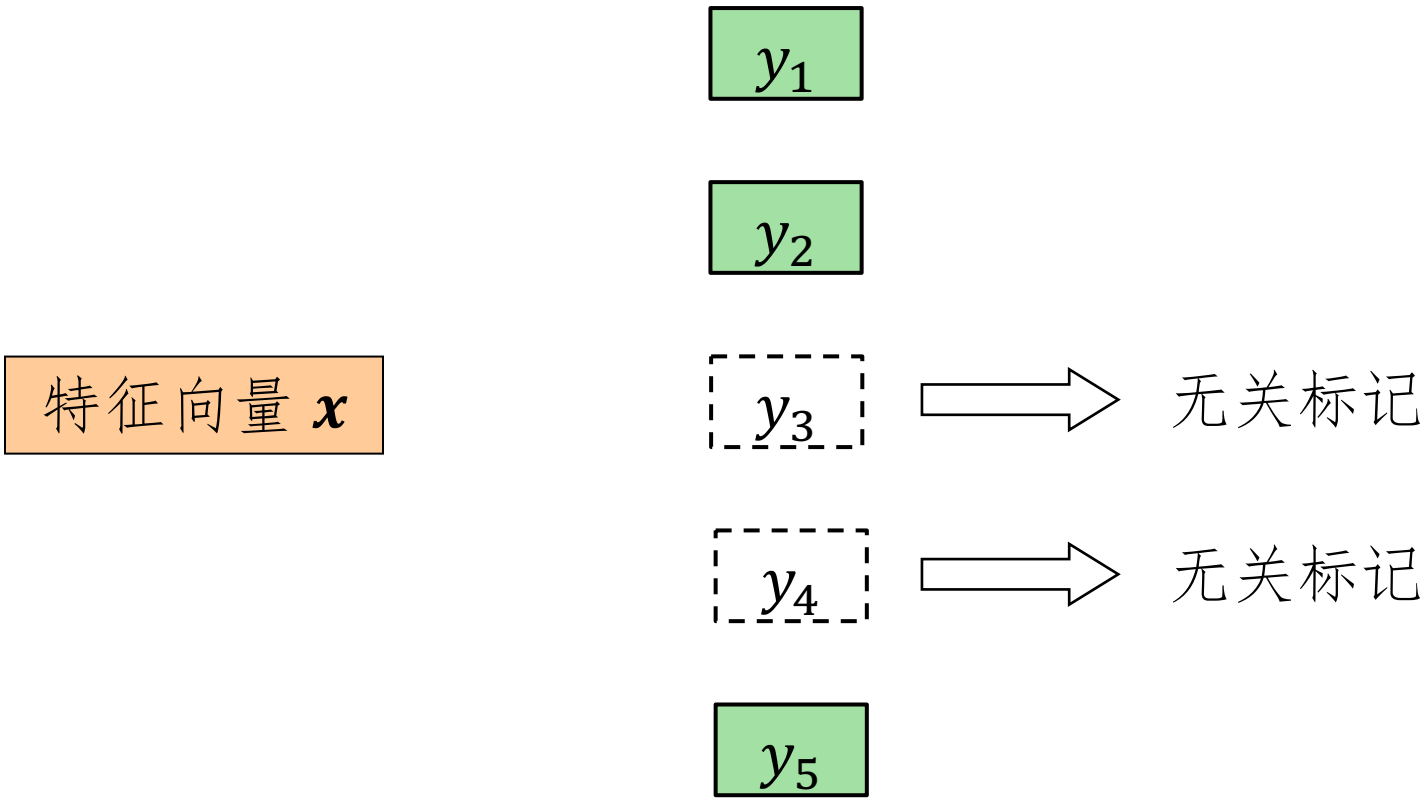
- 训练 $q(q-1)/2$ 个二分类模型，每个二分类器用来判别标签对 $(y_j, y_k), 1 \le j \le k \le q.$
- 标签对(y_j, y_k) 训练集构造
  - 对于样本x_i, 如果 $Y_{ij} = +1$ 且 $Y_{ik} = -1$, 则 x_i 为正样本
  - 对于样本x_i, 如果 $Y_{ij} = -1$ 且 $Y_{ik} = +1$, 则 x_i 为负样本
  - 忽略不满足任一上述条件的样本

测试时，所有二分类模型进行投票，根据得票数对标记排序

Calibrated Label Ranking (CLR) [Fürnkranz et al. MLJ08]

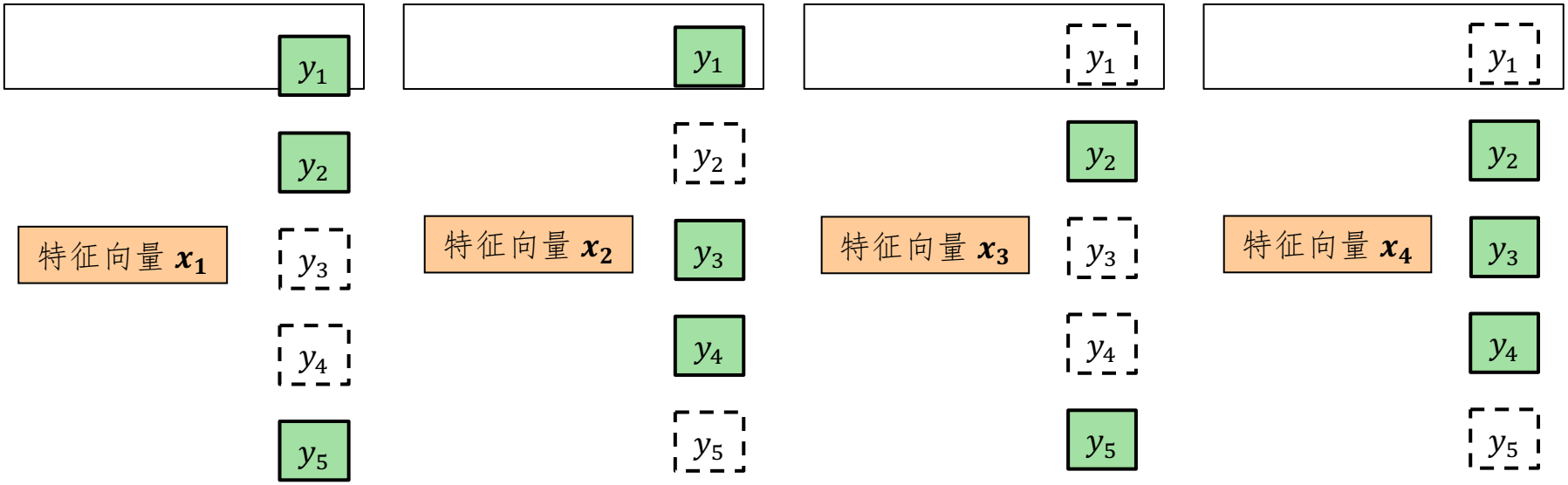


Calibrated Label Ranking (CLR) [Fürnkranz et al. MLJ08]

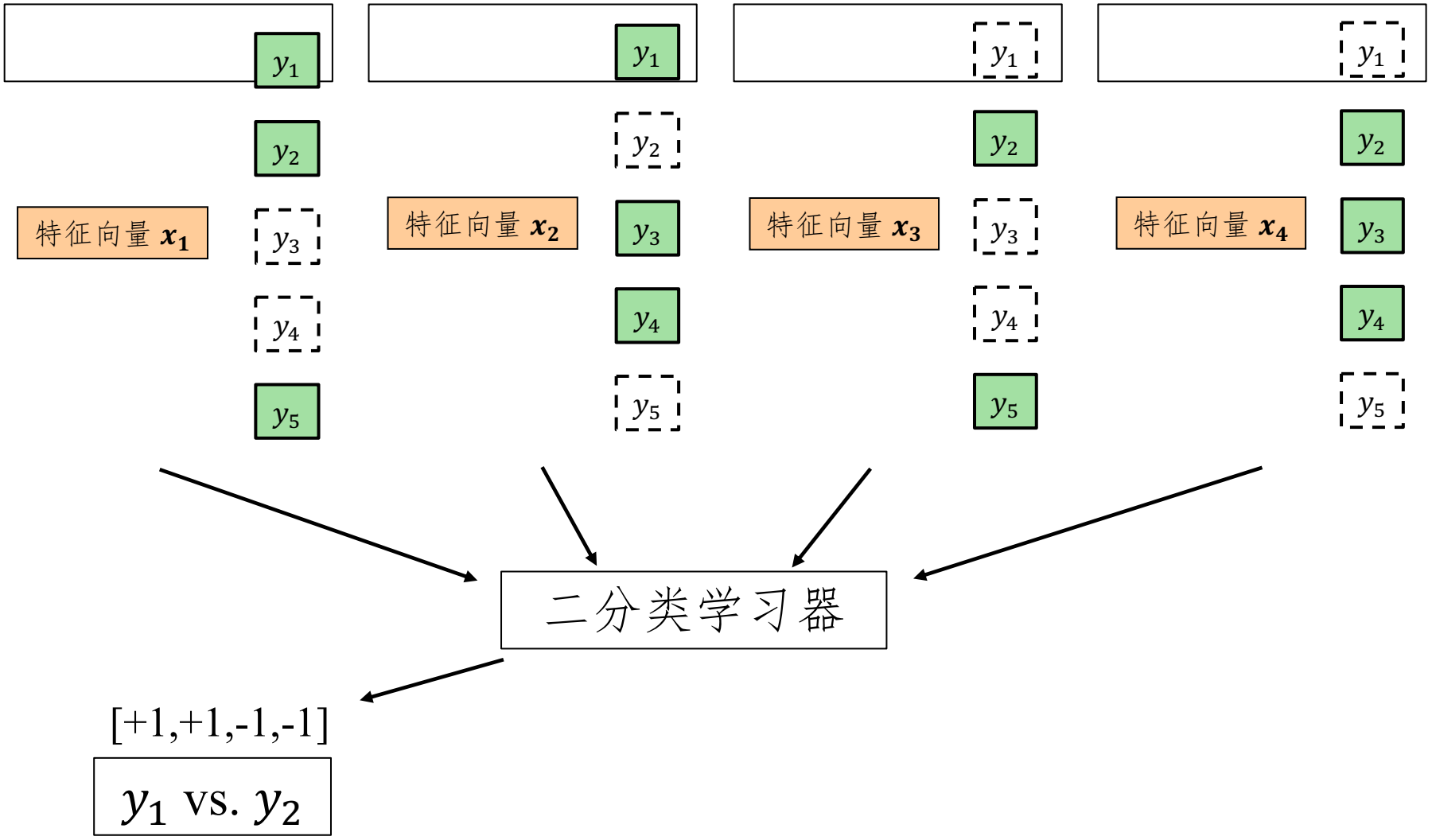


Calibrated Label Ranking (CLR) [Fürnkranz et al. MLJ08]

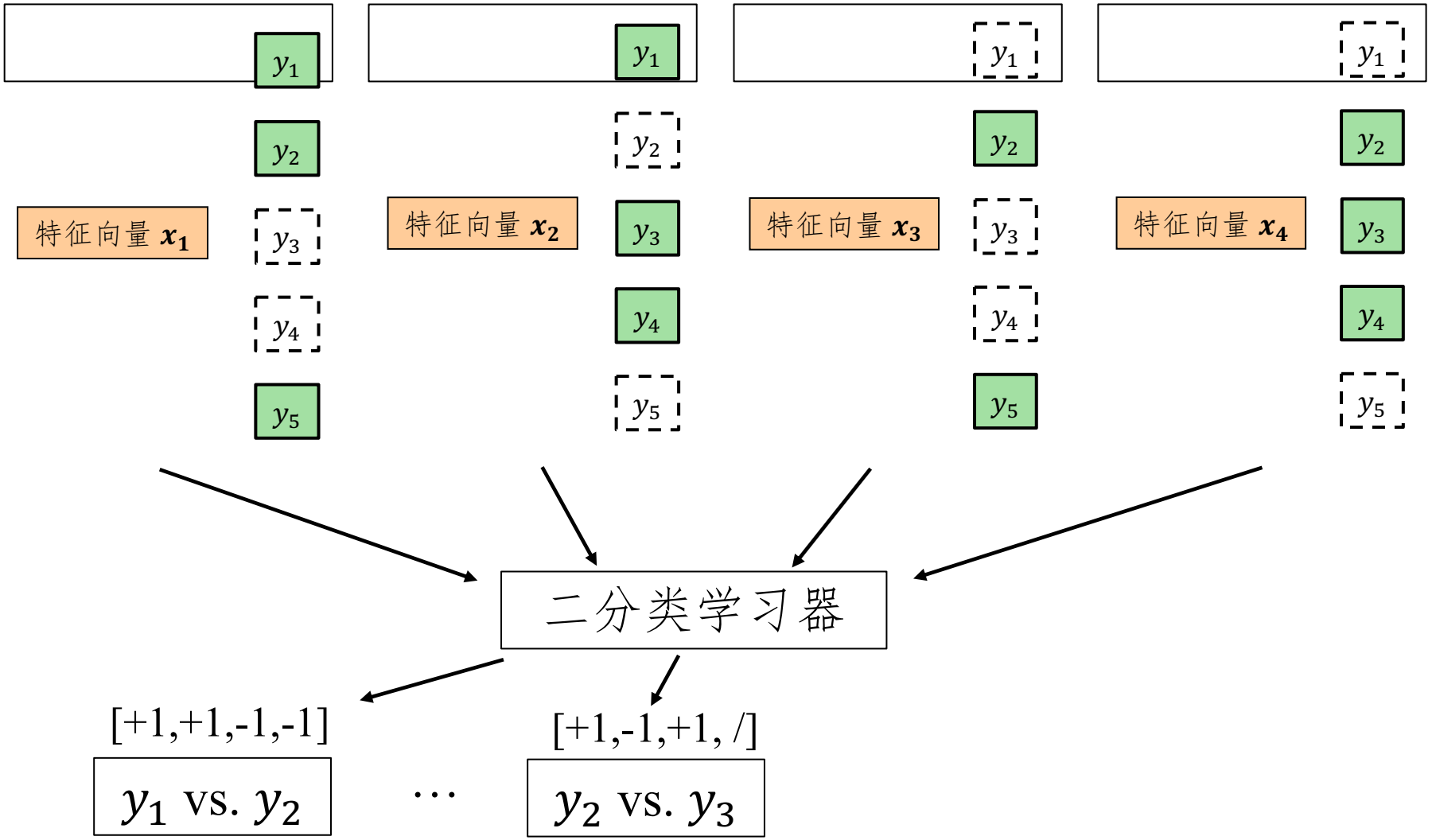
多标记学习训练数据示意图



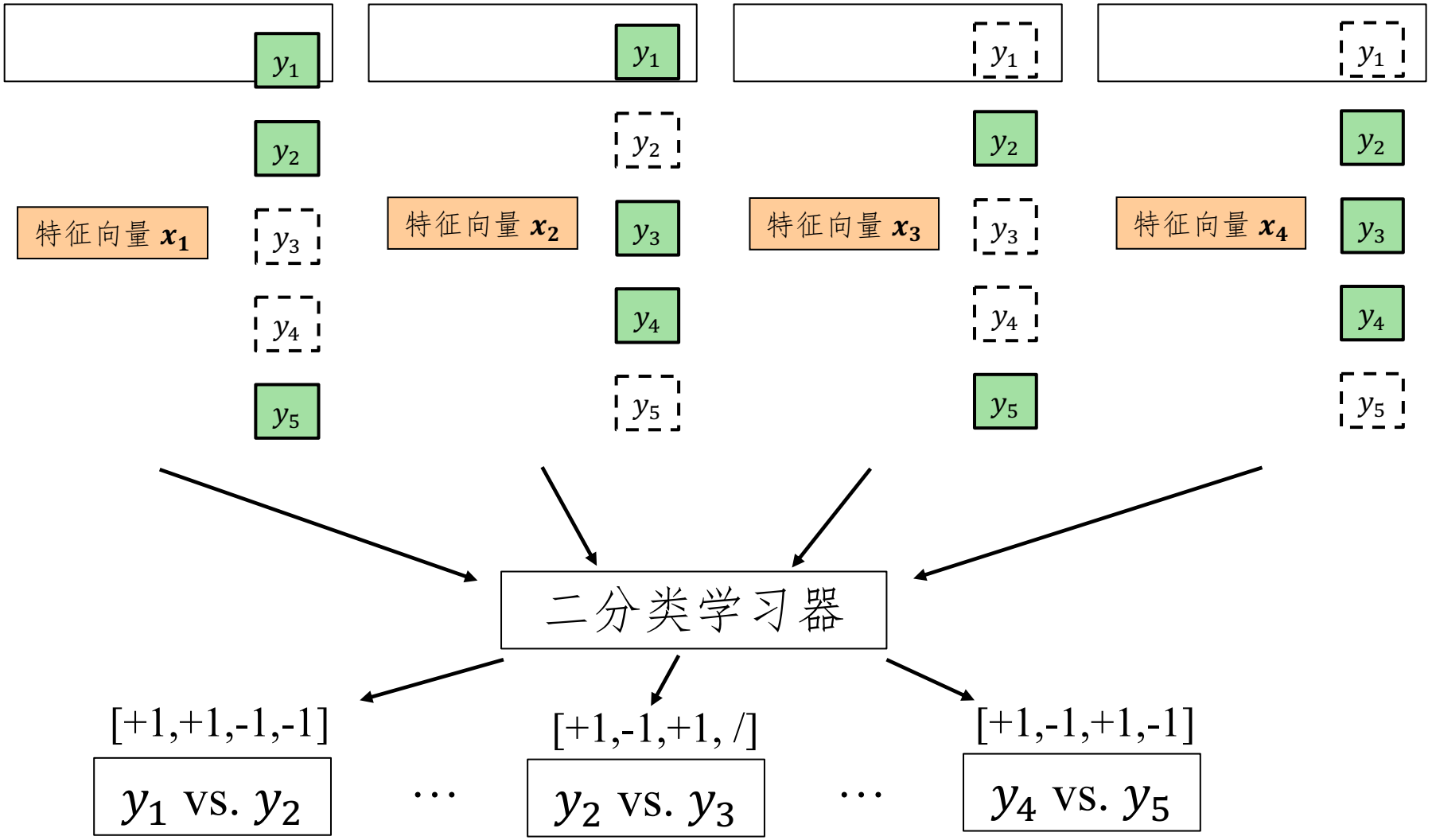
Calibrated Label Ranking (CLR) [Fürnkranz et al. MLJ08]



Calibrated Label Ranking (CLR) [Fürnkranz et al. MLJ08]



Calibrated Label Ranking (CLR) [Fürnkranz et al. MLJ08]



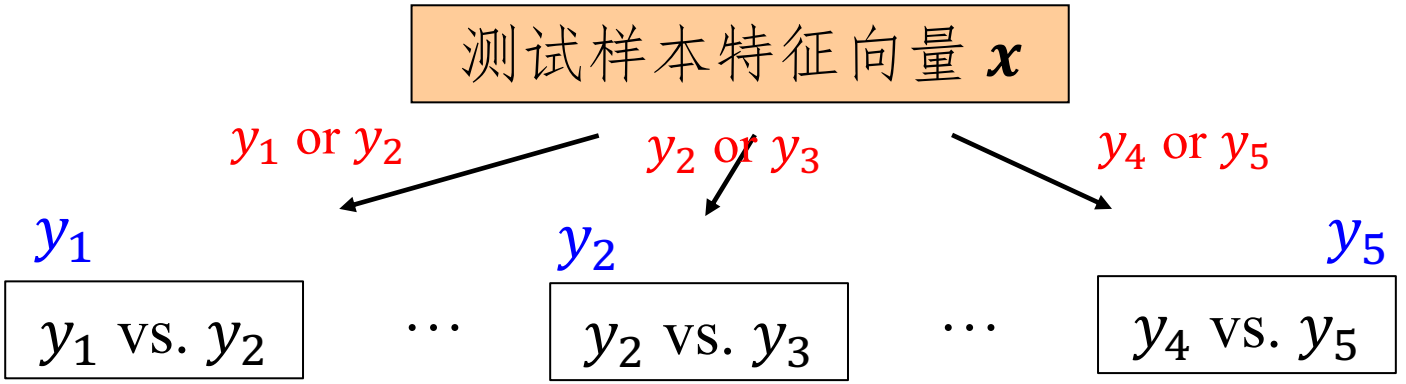
Calibrated Label Ranking (CLR) [Fürnkranz et al. MLJ08]

标记	票数
y_1	2
y_2	4
y_3	1
y_4	0
y_5	3

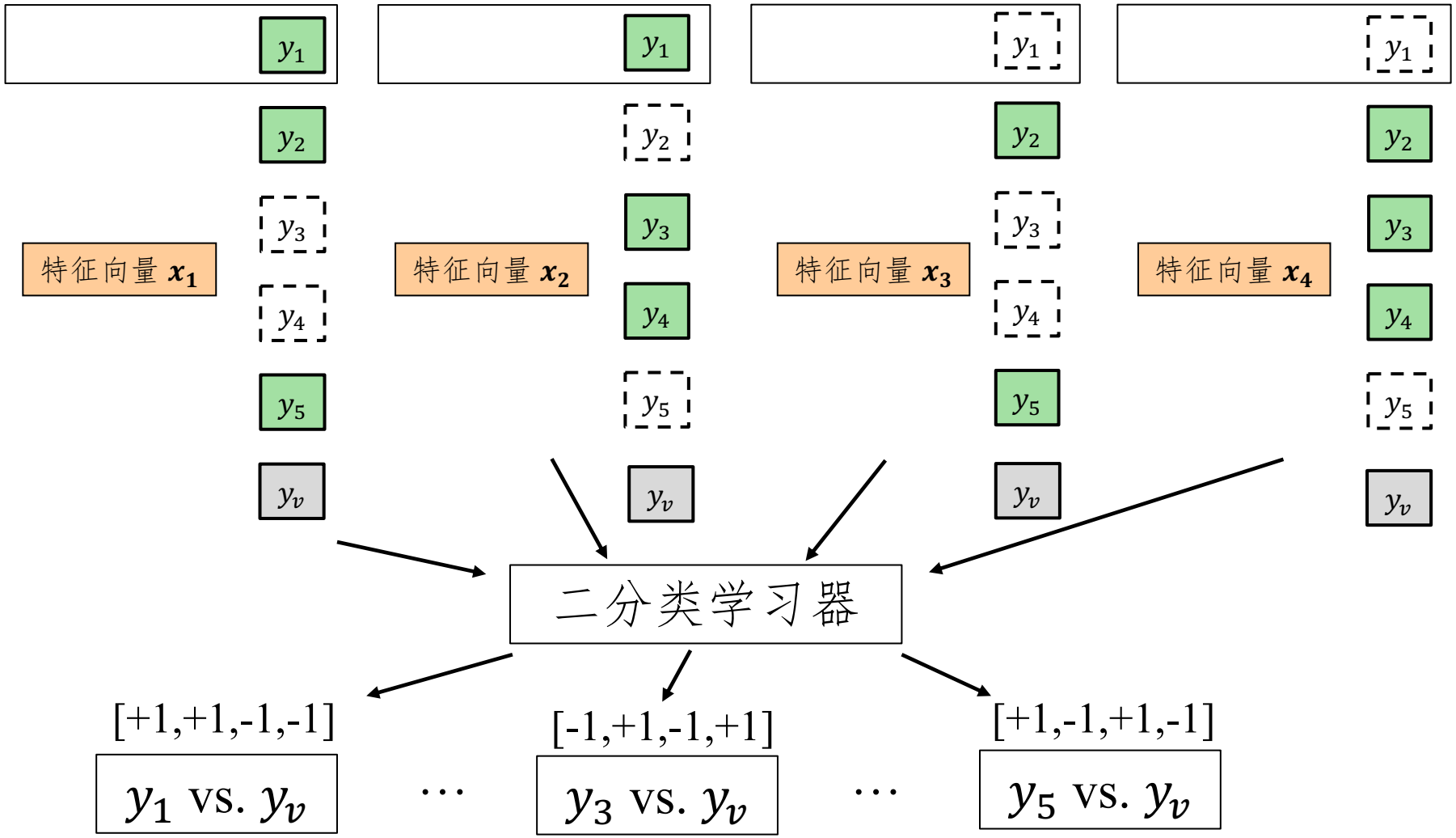
排序

$y_2 \succ y_5 \succ y_1 \succ y_3 \succ y_4$

但应选取多少个标记呢?



Calibrated Label Ranking (CLR) [Fürnkranz et al. MLJ08]



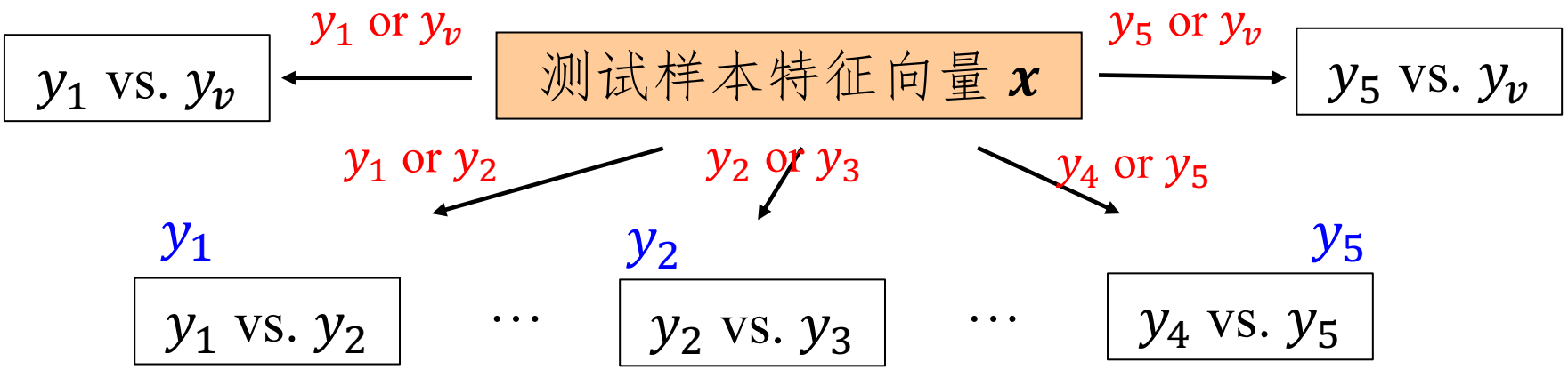
Calibrated Label Ranking (CLR) [Fürnkranz et al. MLJ08]

标记	票数
y_1	2
y_2	4
y_3	1
y_4	0
y_5	3

校准后的排序

$$y_2 \succ y_5 \succ y_v \succ y_1 \succ y_3 \succ y_4$$

y_v 前的标记作为预测结果



Random k-Labelsets (RAKEL) [Tsoumakas, TKDE11]

基本假设：将多标记学习转化为多个单标记多分类问题的集成

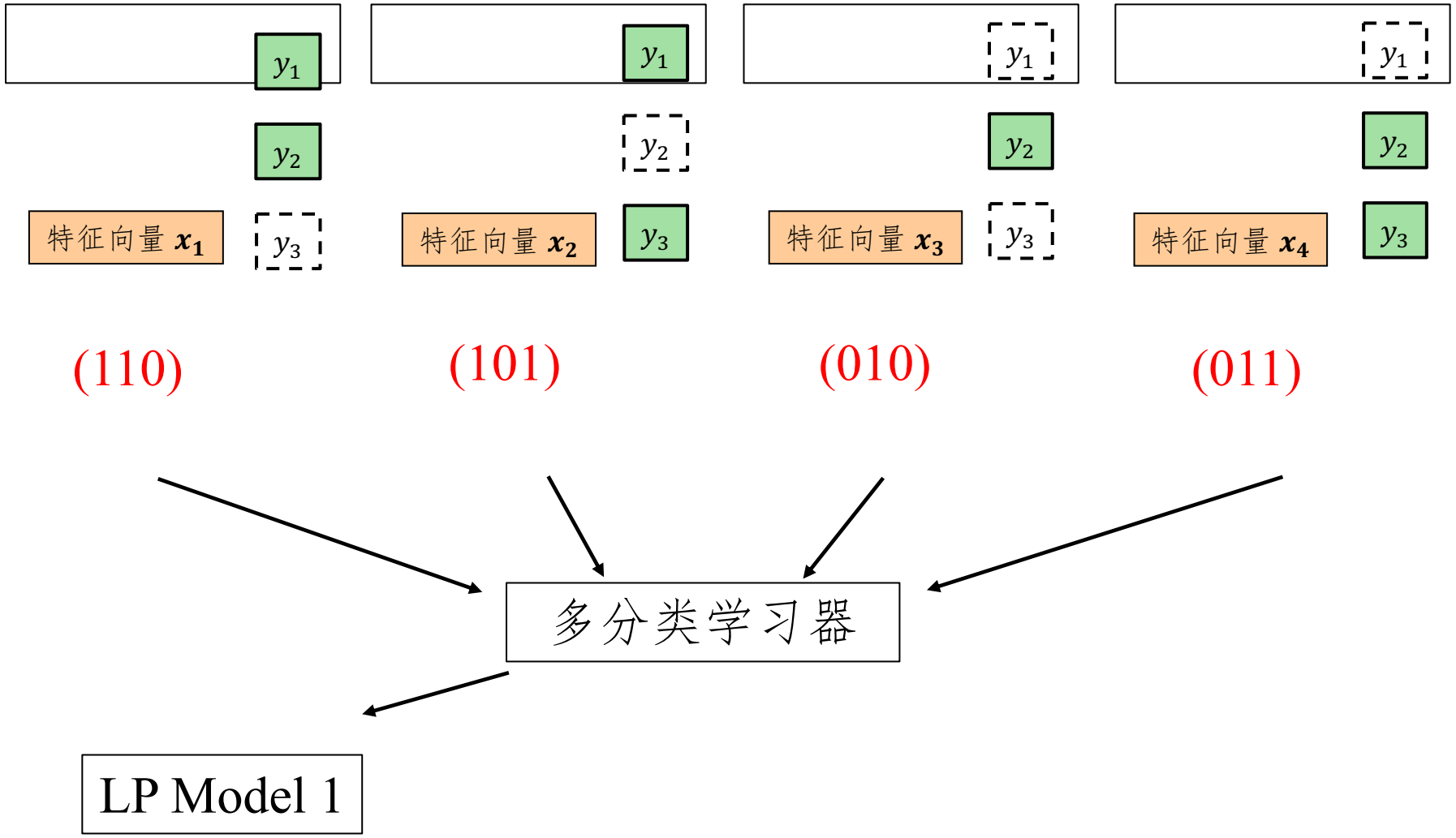
Label Powerset (LP) 的不足

- 无法预测训练集未出现的标记组合
- 当标记数量大时，算法复杂度高

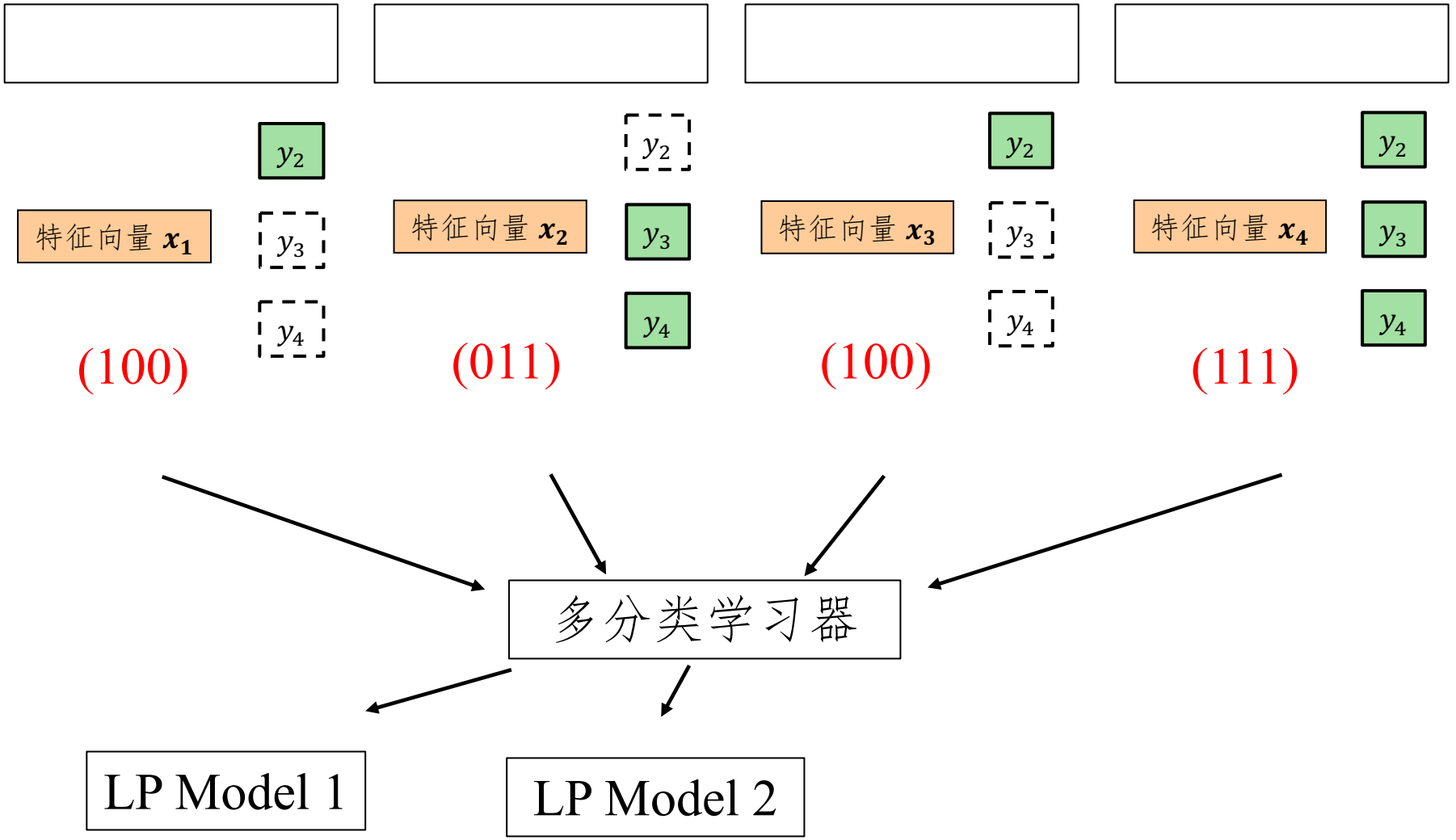
k-Labelsets

- 随机选取 k 的值 (如 k=3), 然后调用 LP 算法
- 通过选取多个 k, 构建多个 LP 模型,并通过投票法或阈值法进行预测

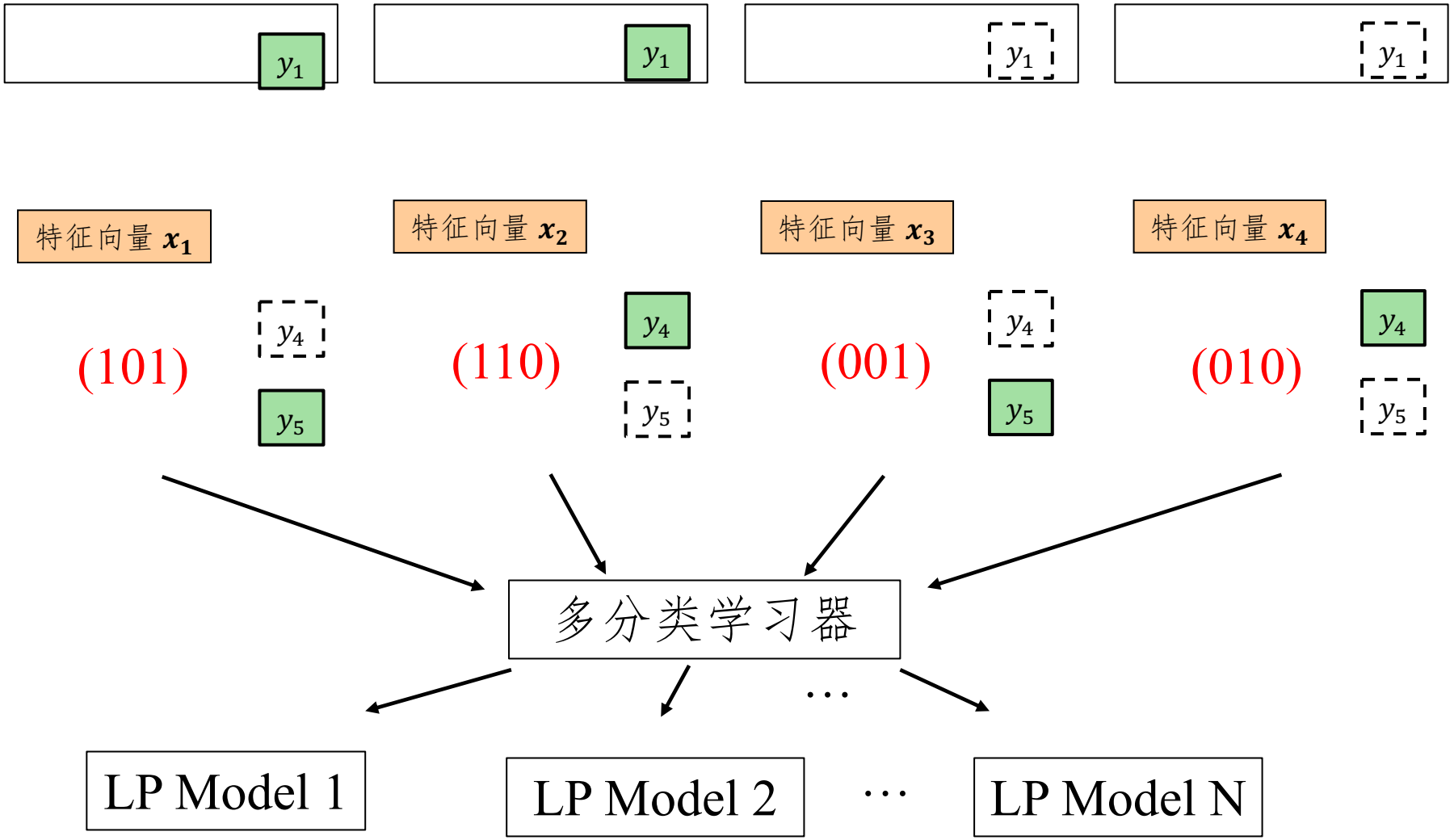
Random k-Labelsets (RAKEL) [Tsoumakas, TKDE11]



Random k-Labelsets (RAKEL) [Tsoumakas, TKDE11]



Random k-Labelsets (RAKEL) [Tsoumakas, TKDE11]

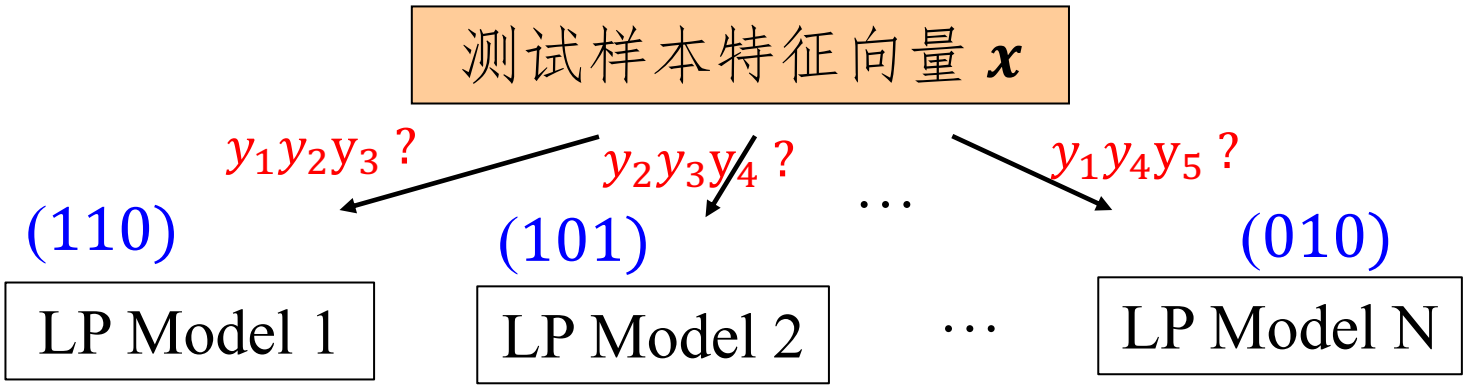


Random k-Labelsets (RAKEL) [Tsoumakas, TKDE11]

LP Model	k-Labelset	预测结果				
		y_1	y_2	y_3	y_4	y_5
h_1	$\{y_1, y_2, y_3\}$	1	1	0	-	-
h_2	$\{y_2, y_3, y_4\}$	-	1	0	1	-
h_3	$\{y_2, y_4, y_5\}$	-	0	-	1	1
h_4	$\{y_1, y_4, y_5\}$	0	-	-	1	0
平均得票数		1/2	2/3	0/2	3/3	1/2

$\{y_2, y_4\}$

阈值 = 0.5



经典多标记学习算法

- **将多标记转化为单标记问题**
  - 一阶方法: Binary Relevance (BR)
  - 二阶方法: Calibrated Label Ranking (CLR)
  - 高阶方法: Random k-labelsets (RAKEL)

- **多标记学习专有算法**
  - 二阶方法: Rank-SVM
  - 高阶方法: LEAD, ECC, CCE

Rank-SVM [Elisseeff, NeurIPS'02]

基本假设：通过优化排序损失为每个标记训练一个分类器

算法流程

- 为第 j 个标记训练二分类器，模型参数为 (w_j, b_j)
- 测试时，当 $< w_j, x > + b_j > 0$ 时，判定为与第 j 个标记相关

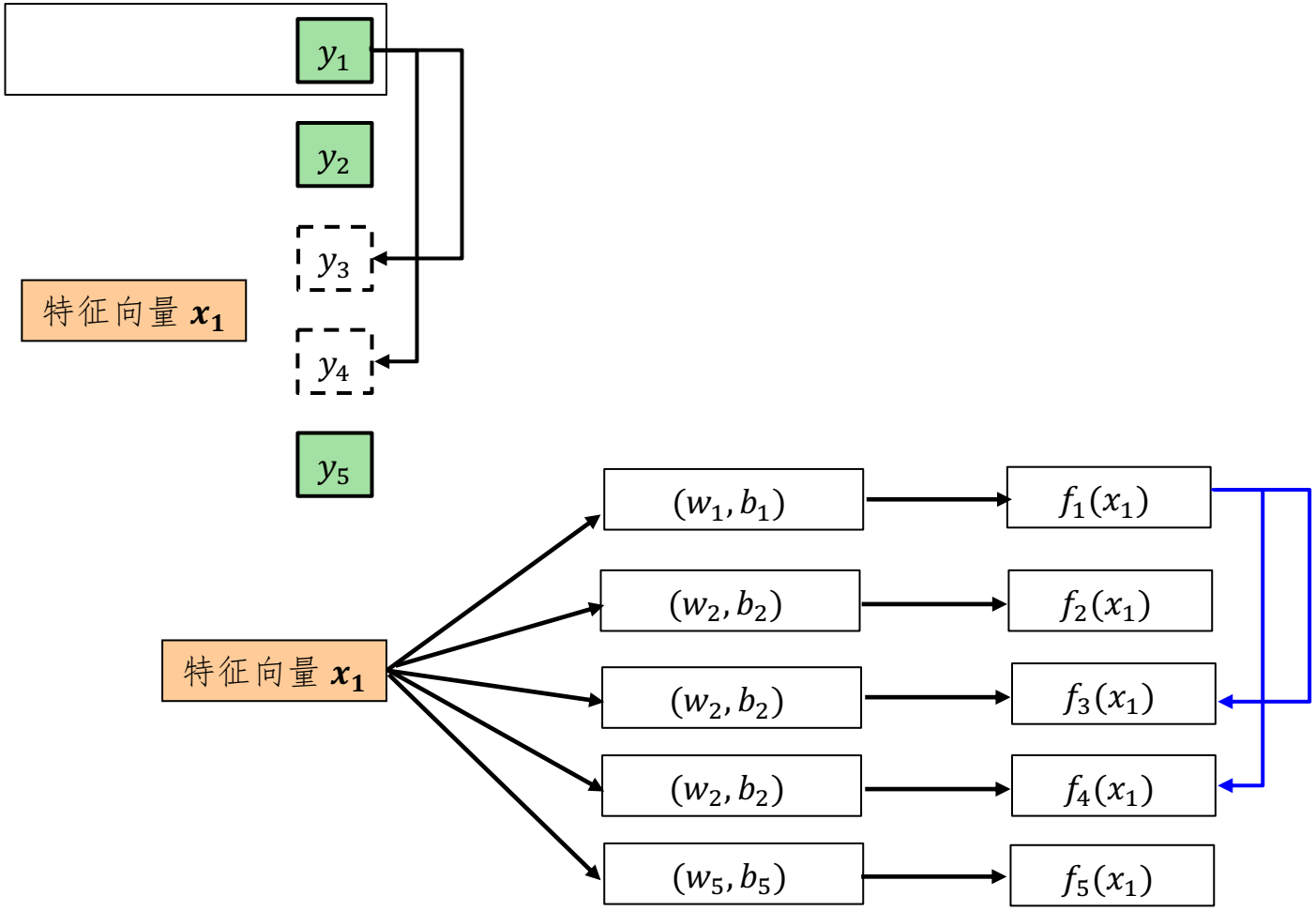
Rank-SVM 优化目标

$$\sum_{j=1}^{q} ||\boldsymbol{w}_j||^2 + C \cdot \sum_{i=1}^{m} \frac{1}{|Y_i||\bar{Y}_i|} \sum_{(j,k)\in Y_i \times \bar{Y}_i} \mathbf{hinge}\left(\langle \boldsymbol{w}_j - \boldsymbol{w}_k, \boldsymbol{x}_i \rangle + b_j - b_k\right)$$

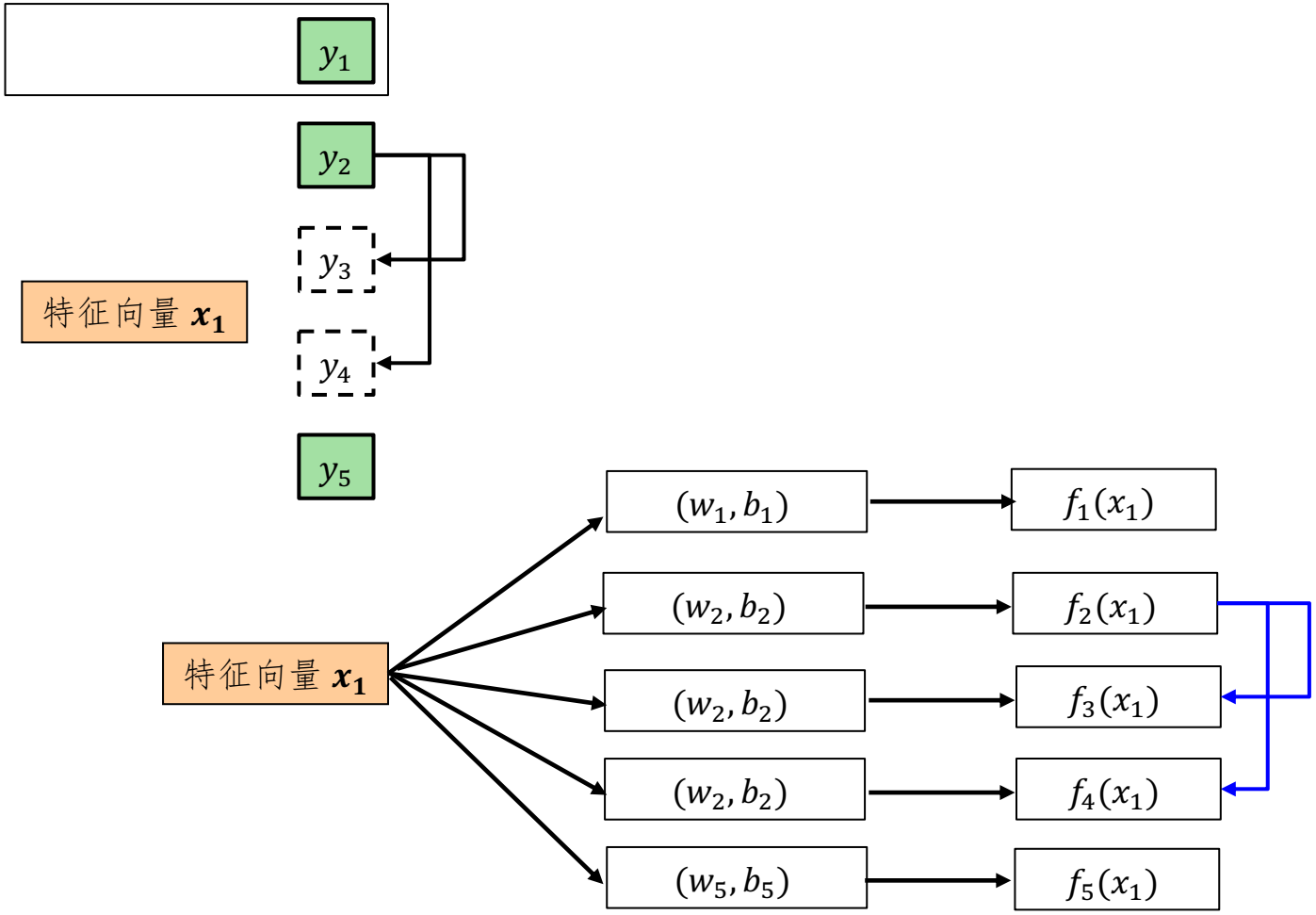
模型复杂度正则

成对排序损失

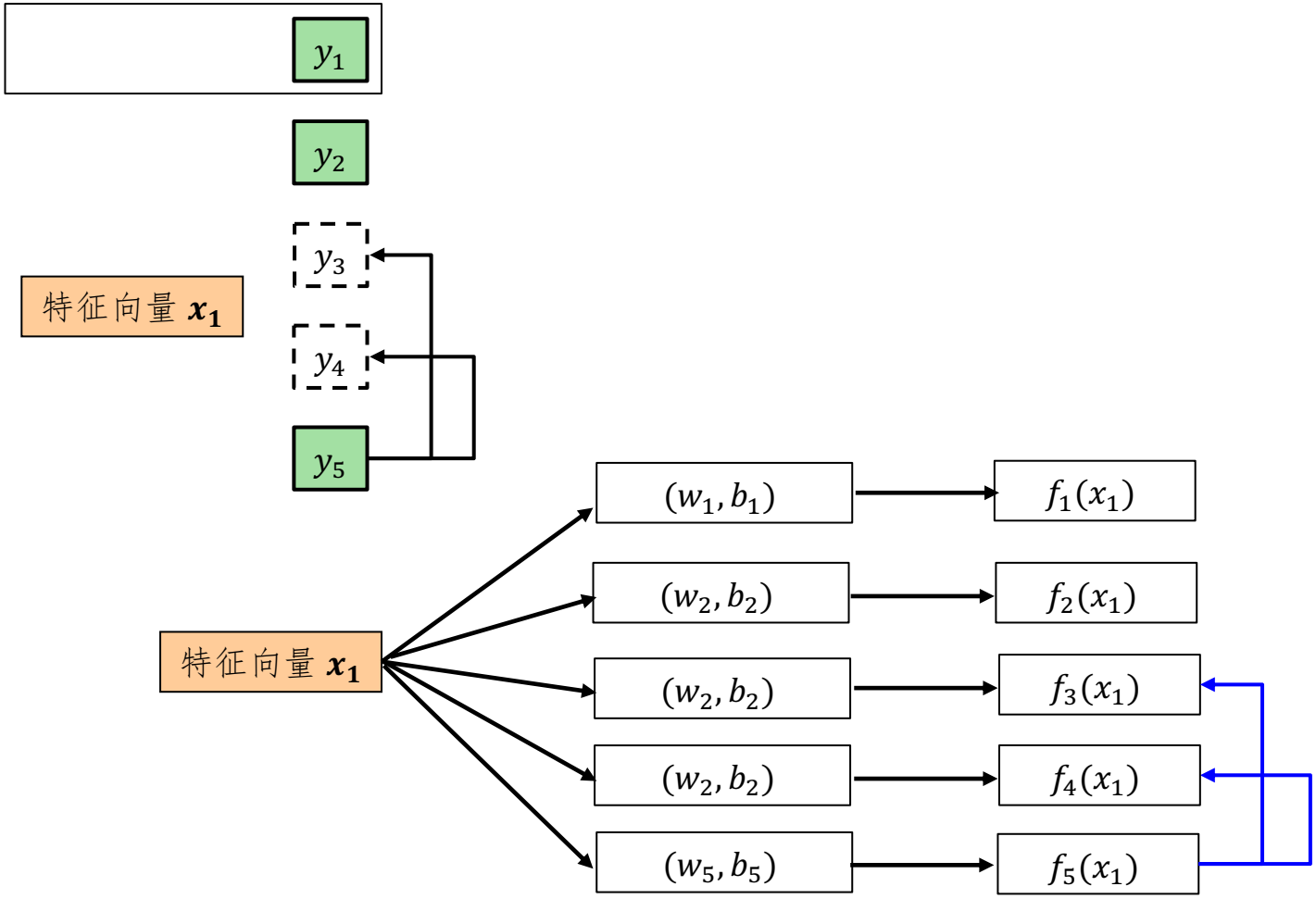
Rank-SVM [Elisseeff, NeurIPS'02]



Rank-SVM [Elisseeff, NeurIPS'02]



Rank-SVM [Elisseeff, NeurIPS'02]



Classifier Chain (CC) [Read, ECML PKDD'09]

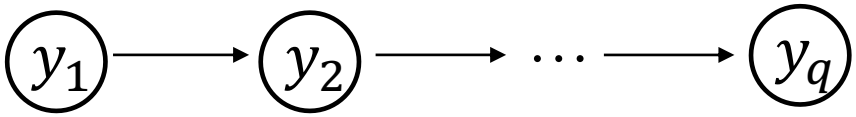
基本思路：通过标记链式排列方式，利用前序标记的预测值增强后序标记的预测性能

算法流程

- 随机生成一个排列，作为标记排列的顺序
- 对于排列中的第 j 个标记 y_j，修改样本特征为 $x_i' = [x_i, \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{j-1}]$，其中 $\hat{y}_k$ 为第 k 个标记的预测结果

Classifier Chain (CC) [Read, ECML PKDD'09]

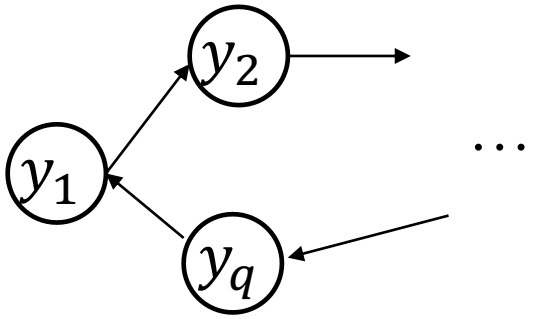
基本思路：通过标记链式排列方式，利用前序标记的预测值增强后序标记的预测性能



ECC: 为了减少生成排列的随机性，
可生成多个标记排列顺序，进行集成

Classifier Circle [Wang, Journal of Software'15]

基本思路：沿着圈结构遍历每个标记若干次充分利用到了它与每个标记或分类器之间的关系



分类器圈相比分类器链的优势：

- 充分利用到了它与每个标记或分类器之间的关系
- 对标记学习次序不敏感

多标记学习常用评价指标

基于样本的评价指标

- 单独为每个测试样本计算指标
- 返回所有测试样本指标的平均值

基于标记的评价指标

- 单独为每个标记计算指标
- 返回所有标记指标的平均值 (包括宏平均、微平均)

多标记学习常用评价指标

Y_i: 样本 x_i 真实的标记向量
P_i: 样本 x_i 预测的标记向量

Subset Accuracy:

$$\frac{1}{m}\sum_{i=1}^{m}[\![P_i = Y_i]\!] \uparrow \qquad ([\![\text{true}]\!] = 1,\ [\![\text{false}]\!] = 0)$$

所有样本中真实标记向量和预测的标记向量完全一致的比例.
当标记数量较多时, 该指标一般较低

多标记学习常用评价指标

Y_i: 样本 x_i 真实的标记向量
P_i: 样本 x_i 预测的标记向量

Hamming Loss:

$$\frac{1}{m}\sum_{i=1}^{m}\frac{|P_i \Delta Y_i|}{q} \downarrow$$

(Δ : 值不相同的位置)

所有样本中每个真实标记和预测标记不同的比例

多标记学习常用评价指标

Y_i: 样本 $\boldsymbol{x}_i$ 真实的标记向量
P_i: 样本 $\boldsymbol{x}_i$ 预测的标记向量

One-error:

$$\frac{1}{m}\sum_{i=1}^{m}[\![[\arg\max_{y\in\mathcal{Y}} f(\boldsymbol{x}_i, y)] \notin Y_i]\!]$$

每个样本中预测置信度最高的标记为不相关标记的比例

基于标记的评价指标

单个标记的混淆矩阵

第 j 个标记		真实值	
		相关	不相关
预测值	相关	TP_j	FP_j
	不相关	FN_j	TN_j

$B(TP_j, FP_j, FN_j, TN_j)$ 表示从混淆矩阵计算得到的二分类指标

例如:

$$\text{Accuracy} = B(TP_j, FP_j, FN_j, TN_j) = \frac{TP_j + TN_j}{TP_j + FP_j + FN_j + TN_j}$$

$$\text{Precision} = B(TP_j, FP_j, FN_j, TN_j) = \frac{TP_j}{TP_j + FP_j}$$

基于标记的评价指标

所有标记的混淆矩阵

第 1 个标记		真实值	
		相关	不相关
预测值	相关	TP_1	FP_1
	不相关	FN_1	TN_1

……

第 q 个标记		真实值	
		相关	不相关
预测值	相关	TP_q	FP_q
	不相关	FN_q	TN_q

Macro-averaging:

$$B_{\text{macro}} = \frac{1}{q}\sum_{j=1}^{q} B(TP_j, FP_j, FN_j, TN_j)$$

Micro-averaging:

$$B_{\text{micro}} = B(\sum_{j=1}^{q} TP_j, \sum_{j=1}^{q} FP_j, \sum_{j=1}^{q} FN_j, \sum_{j=1}^{q} TN_j)$$

多标记学习常用评价指标

如何选择评价指标?

- 没有任务通用的评价指标
- 根据具体任务选择合适的评价指标
  - ✓ 检索任务: 一般选择基于标记的指标, 如 micro avg. precision
  - ✓ 分类任务: 一般选择基于样本的指标, 如 hamming loss

多标记学习常用数据集和算法

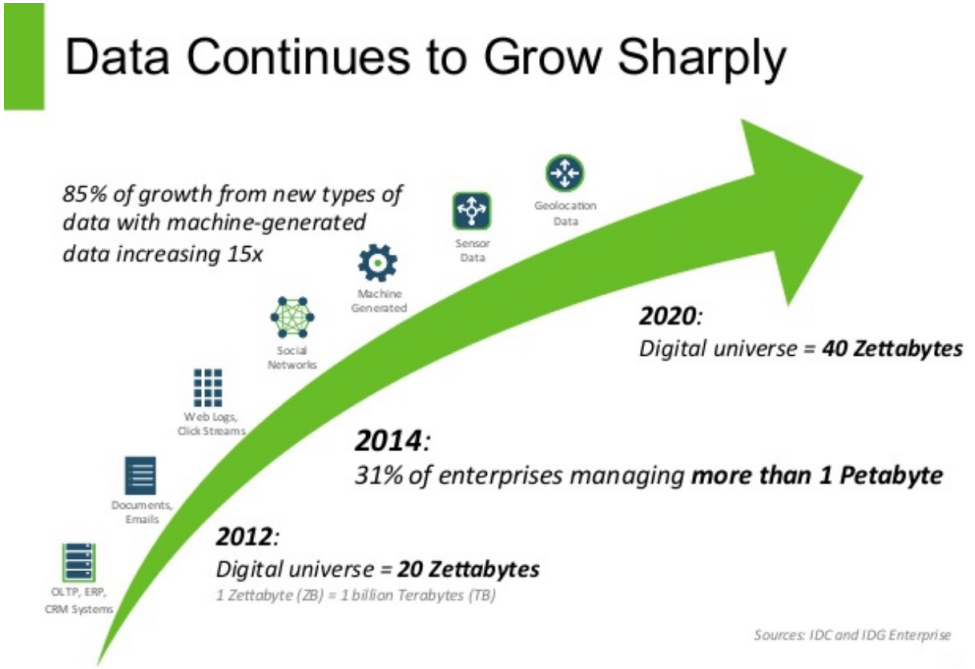
- http://mulan.sourceforge.net/datasets.html
- http://meka.sourceforge.net/#datasets
- http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multilabel.html

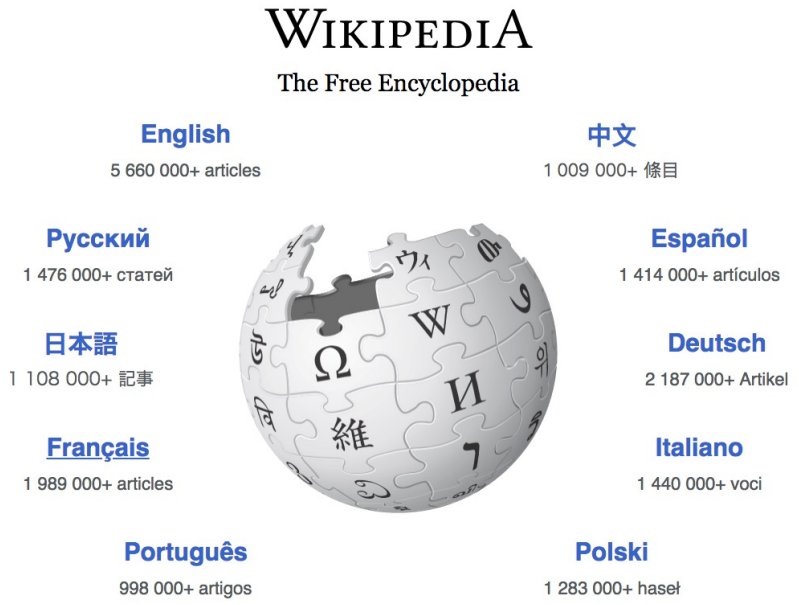
提纲

- 多标记学习
  - 经典算法
- 大规模多标记学习
  - 主流算法
- 总结

大规模多标记学习

随着数据量的快速增涨，多标记学习问题中的标记数量可以达到十万甚至百万级别





例如，维基百科数据集包含 50 万个标记

经典多标记方法时间复杂度

经典算法	Binary Relevance	Classifier Chain	Rank SVM	LEAD	Calibrated Label Ranking
时间复杂度	$O(NDL)$	$O(NDL^2)$	$O(NDL^2)$	$O(NDL^2)$	$O(NDL^2)$

- N: 训练样本数量
- D: 样本特征维度
- L: 标记个数

经典多标记方法时间复杂度

经典算法	Binary Relevance	Classifier Chain	Rank SVM	LEAD	Calibrated Label Ranking
时间复杂度	$O(NDL)$	$O(NDL^2)$	$O(NDL^2)$	$O(NDL^2)$	$O(NDL^2)$

- N: 训练样本数量
- D: 样本特征维度
- L: 标记个数

当 L 较大时 (如 $L = 10^5$), 传统方法无法直接应用!

大规模多标记学习算法分类

- One-vs-All Methods
  - DisMEC
- Embedding-based Methods
  - LEML
  - SLEEC
- Tree-based Methods
  - FastXML
  - Parabel
- Deep Learning-based Methods
  - XML-CNN
  - AttenttionXML

大规模多标记学习算法分类

- One-vs-All Methods
  - DisMEC
- Embedding-based Methods
  - LEML
  - SLEEC
- Tree-based Methods
  - FastXML
  - Parabel
- Deep Learning-based Methods
  - XML-CNN
  - AttenttionXML

One-vs-All Method (DisMEC)

基本思路：忽略标记相关性，每个标记训练一个二分类模型

由于标记数量较多，因此采用线性模型，其优化目标如下：

$$\min_{\mathbf{w}_\ell} \left[||\mathbf{w}_\ell||_2^2 + C \sum_{i=1}^{N} (\max(0, 1 - s_{\ell_i} \mathbf{w}_\ell^T \mathbf{x}_i))^2 \right]$$

其中，$\boldsymbol{w_l}$ 为第 l 个标记的线性模型的参数，s_{l_i} 表示第 i 个样本与第 l 个标记是否相关.

One-vs-All Method (DisMEC)

基本思路：忽略标记相关性，每个标记训练一个二分类模型

由于标记数量较多，因此采用线性模型，其优化目标如下：

$$\min_{\mathbf{w}_\ell} \left[||\mathbf{w}_\ell||_2^2 + C \sum_{i=1}^{N} (\max(0, 1 - s_{\ell_i} \mathbf{w}_\ell^T \mathbf{x}_i))^2 \right]$$

- 加速：为了进一步加速训练，且得益于每个标记相互独立，可以采用并行计算的方式加速训练和测试过程.
- 压缩：为了减少模型参数量，作者发现绝大部分参数分布在零的邻域范围，可以进行裁剪，对模型性能影响很小.

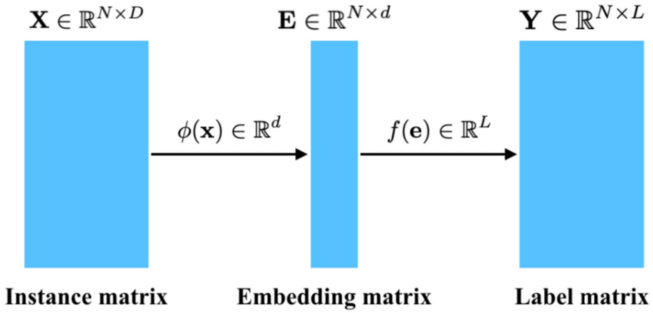
大规模多标记学习算法分类

- One-vs-All Methods
  - DisMEC
- Embedding-based Methods
  - LEML
  - SLEEC
- Tree-based Methods
  - FastXML
  - Parabel
- Deep Learning-based Methods
  - XML-CNN
  - AttenttionXML

Low rank Empirical Risk Minimization (LEML)

基本思路：假设特征到标记的投影矩阵低秩, 利用低秩矩阵分解减少模型参数量

LEML 目标函数如下, 其中 Z 是需求解的投影矩阵 (由于标记间存在关联, 因此假设 Z 低秩):



$$\hat{Z} = \arg\min_{Z} J(Z) = \sum_{i=1}^{n} \sum_{j=1}^{L} \ell(Y_{ij}, f^j(\boldsymbol{x}_i; Z)) + \lambda \cdot r(Z),$$

$$s.t.\ \ \mathrm{rank}(Z) \leq k, \qquad (1)$$

Low rank Empirical Risk Minimization (LEML)

基本思路：假设特征到标记的投影矩阵低秩, 利用低秩矩阵分解减少模型参数量

LEML 目标函数如下, 其中 Z 是需求解的投影矩阵 (由于标记间存在关联, 因此假设 Z 低秩):

$$\hat{Z} = \arg\min_{Z} J(Z) = \sum_{i=1}^{n}\sum_{j=1}^{L} \ell(Y_{ij}, f^j(\boldsymbol{x}_i; Z)) + \lambda \cdot r(Z),$$

$$s.t.\ \ \mathrm{rank}(Z) \leq k, \qquad (1)$$

因 Z 低秩, 因此可以令 $Z = WH^{\mathsf{T}}$, 上述优化目标可以表示为:

$$J_\Omega(W, H) = \sum_{(i,j)\in\Omega} \ell(Y_{ij}, \boldsymbol{x}_i^T W \boldsymbol{h}_j) + \frac{\lambda}{2}\left(\|W\|_F^2 + \|H\|_F^2\right)$$

Low rank Empirical Risk Minimization (LEML)

基本思路：假设特征到标记的投影矩阵低秩，利用低秩矩阵分解减少模型参数量

LEML 目标函数如下，其中 Z 是需求解的投影矩阵 (由于标记间

令特征维度为 D，标记个数为 L，有$Z \in \mathbb{R}^{D\times L}$. 通过低秩假设，即 $Z = WH^{\mathsf{T}}$，其中 $W \in \mathbb{R}^{D\times k}$，$H \in \mathbb{R}^{L\times k}$，且 k≪L，所以需要求解的模型参数量从 $D\times L$ 减少为 $(D+L)k$.

因 Z 低秩，因此可以令 $Z = WH^{\mathsf{T}}$，上述优化目标可以表示为：

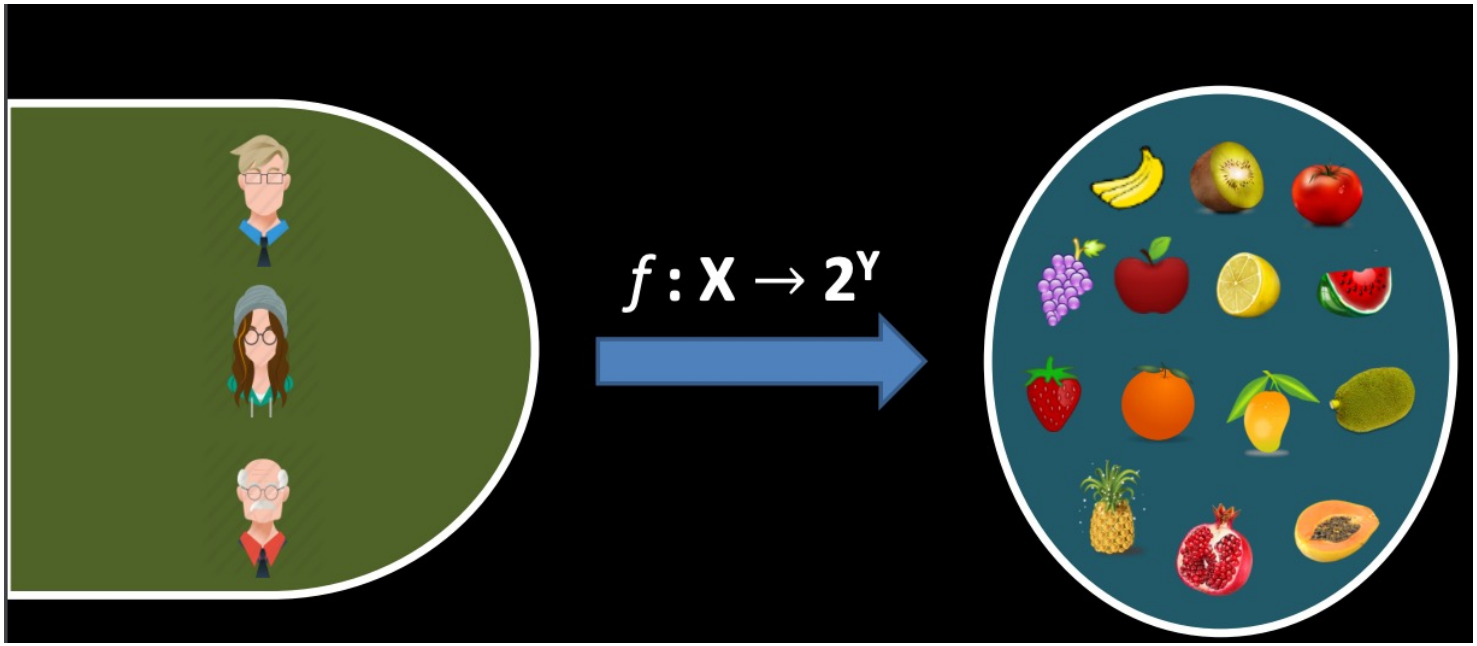
$$J_\Omega(W,H) = \sum_{(i,j)\in\Omega} \ell(Y_{ij}, \boldsymbol{x}_i^T W \boldsymbol{h}_j) + \frac{\lambda}{2}\left(\|W\|_F^2 + \|H\|_F^2\right)$$

大规模多标记学习算法分类

- One-vs-All Methods
  - DisMEC
- Embedding-based Methods
  - LEML
  - SLEEC
- Tree-based Methods
  - FastXML
  - Parabel
- Deep Learning-based Methods
  - XML-CNN
  - AttenttionXML

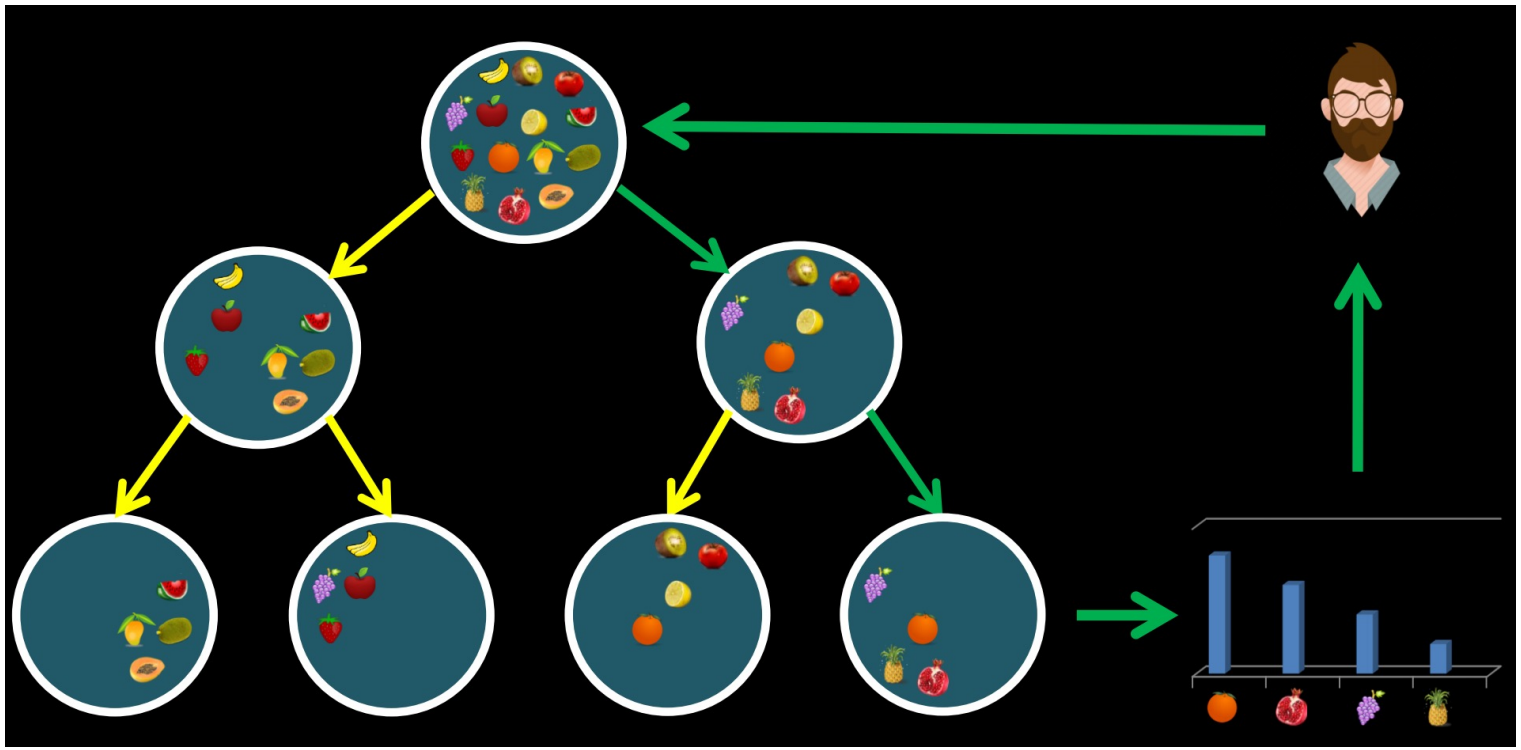
Tree-classifier (FastXML)

问题背景：在推荐领域，需要从大量的商品集中为用户推荐可能偏好的商品子集



Tree-classifier (FastXML)

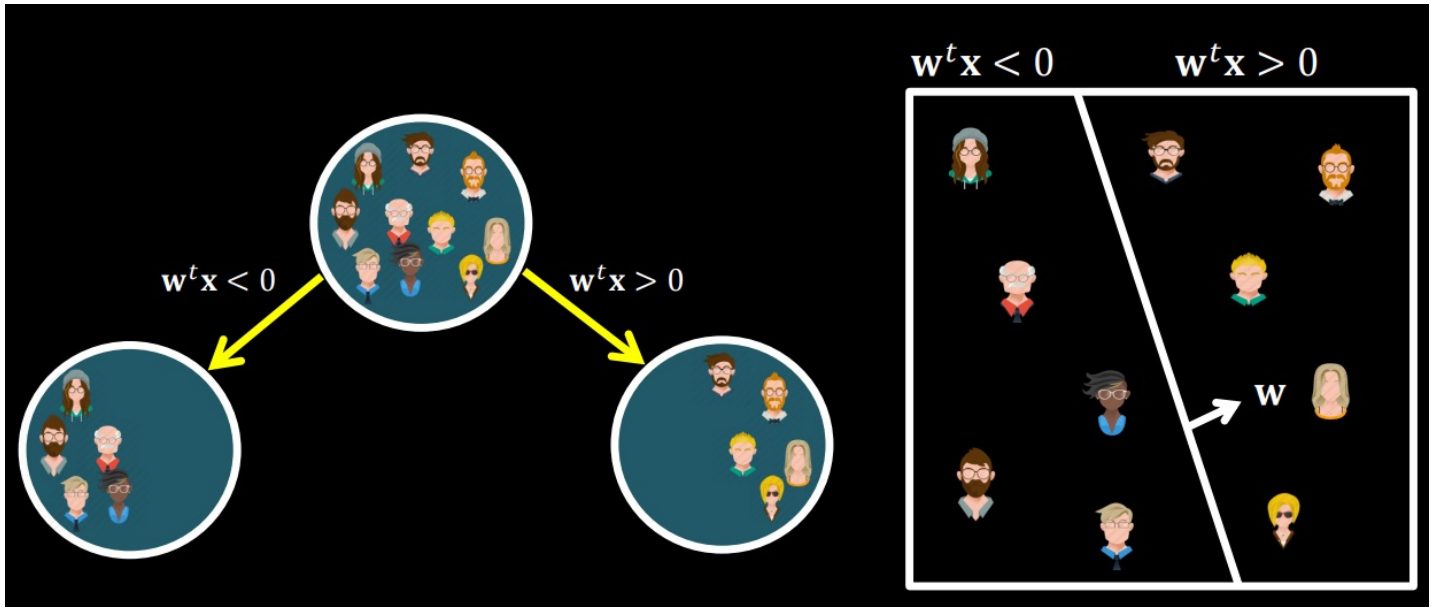
基本思路：标记相似的样本被划分到同一个子节点



Tree-classifier (FastXML)

基本思路：标记相似的样本被划分到同一个子节点

对于树中的每个非叶子结点，FastXML 学习分类器 w 将样本划分到孩子节点



Tree-classifier (FastXML)

基本思路：标记相似的样本被划分到同一个子节点

对于树中的每个非叶子结点，FastXML 目标函数如下，其中 w 将样本划分到孩子节点. 优化 nDCG 指标可以标记类似的样本划分到同一节点.

$$
\begin{aligned}
\min \quad & \|\mathbf{w}\|_1 + \sum_i C_\delta(\delta_i) \log(1 + e^{-\delta_i \mathbf{w}^\top \mathbf{x}_i}) \\
& - C_r \sum_i \tfrac{1}{2}(1 + \delta_i) \mathcal{L}_{\mathrm{nDCG@}L}(\mathbf{r}^+, \mathbf{y}_i) \\
& - C_r \sum_i \tfrac{1}{2}(1 - \delta_i) \mathcal{L}_{\mathrm{nDCG@}L}(\mathbf{r}^-, \mathbf{y}_i) \\
\text{w.r.t.} \quad & \mathbf{w} \in \mathcal{R}^D, \boldsymbol{\delta} \in \{-1, +1\}^L, \mathbf{r}^+, \mathbf{r}^- \in \Pi(1, L)
\end{aligned}
$$

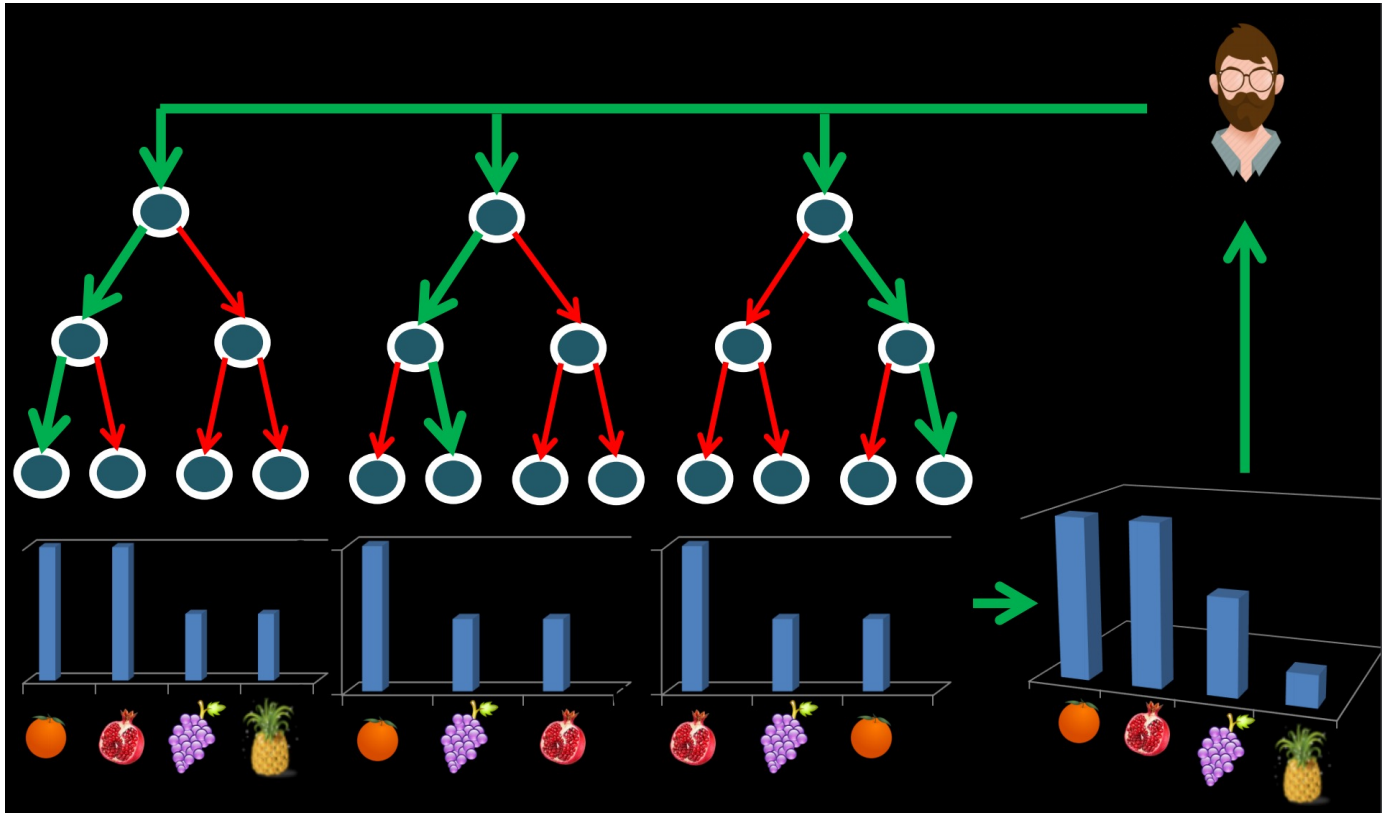
Tree-classifier (FastXML)

对于树中的每个非叶子结点，FastXML 目标函数如下，其中 w 将样本划分到孩子节点. 优化 nDCG 指标可以标记类似的样本划分到同一节点.

$$
\begin{aligned}
\min \quad & \|\mathbf{w}\|_1 + \sum_i C_\delta(\delta_i) \log(1 + e^{-\delta_i \mathbf{w}^\top \mathbf{x}_i}) \\
& - C_r \sum_i \tfrac{1}{2}(1 + \delta_i) \mathcal{L}_{\mathrm{nDCG@}L}(\mathbf{r}^+, \mathbf{y}_i) \\
& - C_r \sum_i \tfrac{1}{2}(1 - \delta_i) \mathcal{L}_{\mathrm{nDCG@}L}(\mathbf{r}^-, \mathbf{y}_i) \\
\text{w.r.t.} \quad & \mathbf{w} \in \mathcal{R}^D, \boldsymbol{\delta} \in \{-1, +1\}^L, \mathbf{r}^+, \mathbf{r}^- \in \Pi(1, L)
\end{aligned}
$$

使用交替优化方法，分别对 $\boldsymbol{w}, \boldsymbol{r}, \boldsymbol{\delta}$ 求解

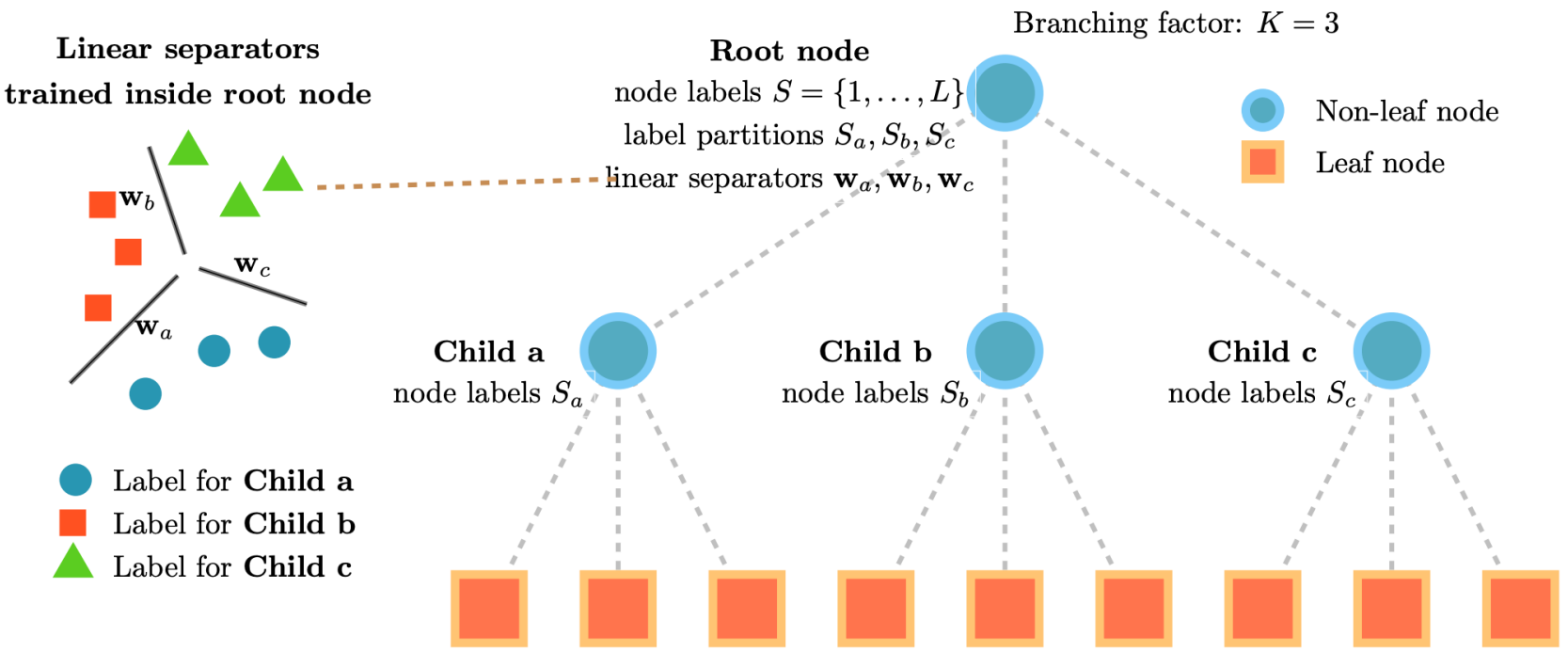
Tree-classifier (FastXML)



可训练多棵树模型，通过集成得到更鲁棒的预测结果

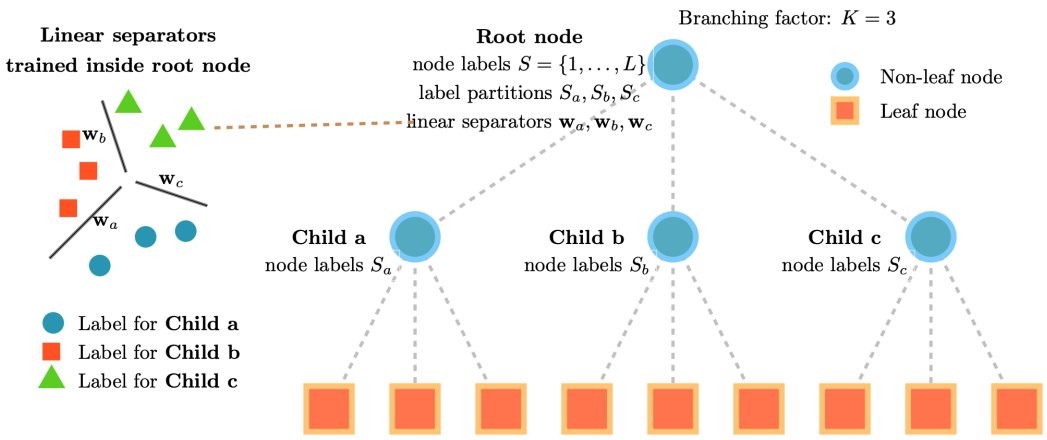
Tree-classifier (Parabel)

基本思路：通过将标记聚类划分成不相交的子集，将问题转化为规模更小的子问题，加速训练过程



Tree-classifier (Parabel)

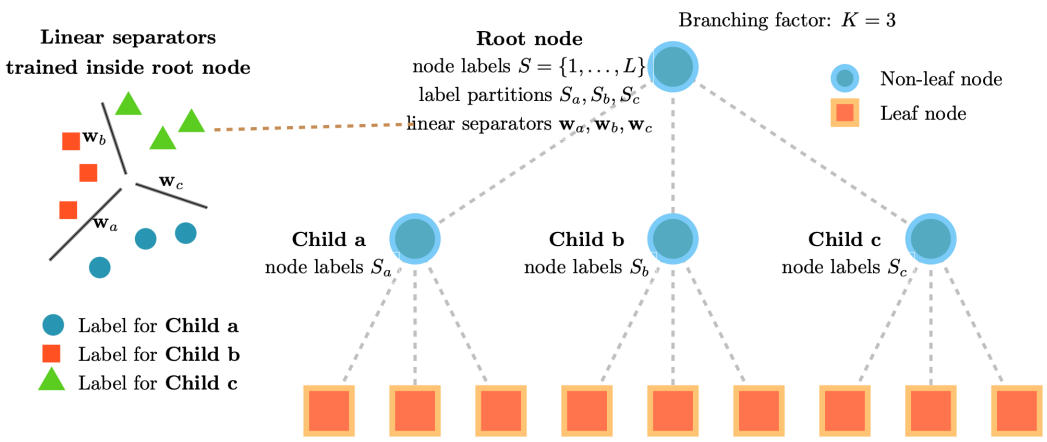
基本思路：通过将标记聚类划分成不相交的子集，将问题转化为规模更小的子问题，加速训练过程



- 标记聚类：通过聚类算法，如 k-means 实现
- 标记特征表示：可通过类别的 word embedding，或该类样本的特征均值得到
- 分类：得到聚类结果后，训练分类器将样本划分到子节点

Tree-classifier (Parabel)

基本思路：通过将标记聚类划分成不相交的子集，将问题转化为规模更小的子问题，加速训练过程



节点划分过程递归进行，知道满足终止条件

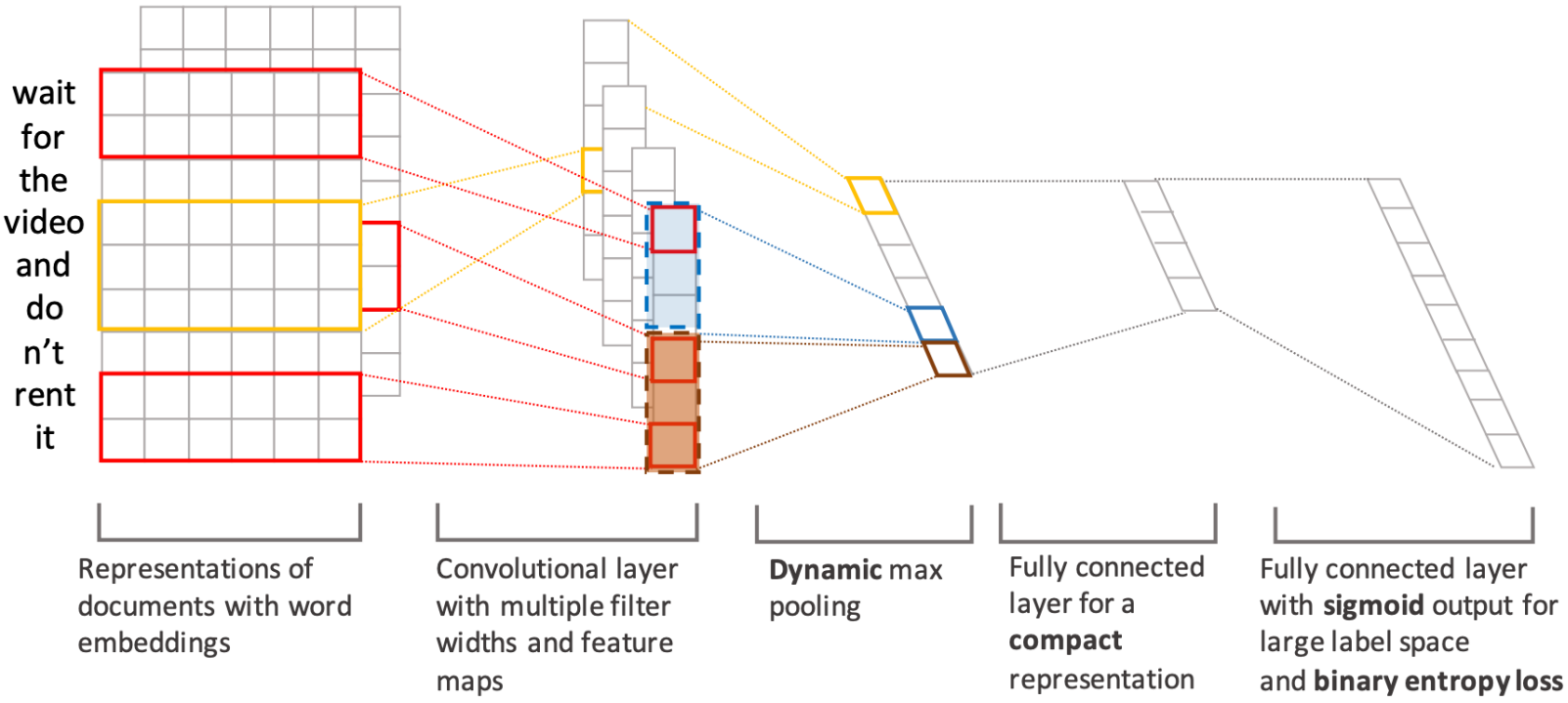
- 标记聚类：通过聚类算法，如 k-means 实现
- 标记特征表示：可通过类别的 word embedding，或该类样本的特征均值得到
- 分类：得到聚类结果后，训练分类器将样本划分到子节点

大规模多标记学习算法分类

- One-vs-All Methods
  - DisMEC
- Embedding-based Methods
  - LEML
  - SLEEC
- Tree-based Methods
  - FastXML
  - Parabel
- Deep Learning-based Methods
  - XML-CNN
  - AttenttionXML

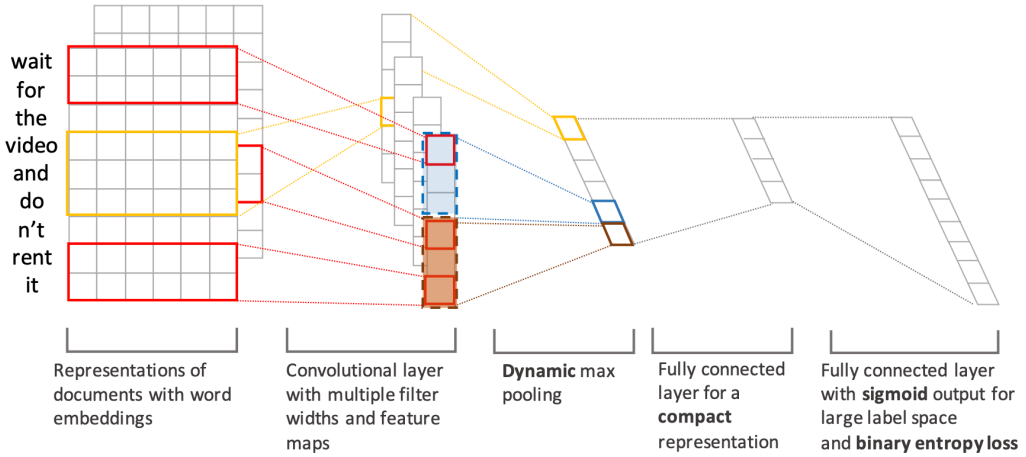
XML-CNN 网络结构 [SIGIR'17]

基本思路：利用卷积操作提取上下文单词的关联



XML-CNN 网络结构 [SIGIR'17]

基本思路：利用卷积操作提取上下文单词的关联



优化 Binary Cross Entropy loss:

$$\min_{\Theta} \quad -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{L}\left[y_{ij}\log(\sigma(f_{ij})) + (1-y_{ij})\log(1-\sigma(f_{ij}))\right]$$

其中, $\sigma(\cdot)$ 为sigmoid函数, Θ 为模型参数.

方法对比

方法类型	训练开销	测试开销	模型大小	精度
One-vs-All	大	大	较小	较高
Embedding-based	大	较小	小	低
Tree-based	小	小	较小	较高
Deep Learning-based	大	较小	大	高

基于深度学习的方法可利用 GPU 资源, 其他方法都基于 CPU 进行运算.

提纲

- 多标记学习
  - 经典算法
- 大规模多标记学习
  - 主流算法
- 总结

总结

- **多标记学习**
  - 一阶方法: 忽略标记之间的相关性
  - 二阶方法: 考虑两两标记之间的相关性
  - 高阶方法: 同时考虑多个标记之间的相关性
- **大规模多标记学习**
  - One-vs-All 方法:忽略标记之间的相关性, 可并行加速
  - Embedding-based 方法: 考虑标记相关性,如低秩假设
  - Tree-based 方法: 训练和推理速度快
  - Deep Learning-based 方法: 精度高