多标签学习算法研究综述

作者：Min-Ling Zhang，Zhi-Hua Zhou

译：Nick Li，Li_zhjun@163.com

摘要

多标签学习研究的问题是，每个样本由一个实例与一组相关联标签来表示。在过去的十年中，这种新兴的机器学习范式已经取得了显著的进展。本文旨在对这一领域进行及时的回顾，重点介绍目前最先进的多标签学习算法。首先，给出了多标签学习的基本概念，包括形式定义和评价指标。第二而且是主要部分，对八种具有代表性的多标签学习算法在常用符号下进行了详细的研究，并进行了相关的分析和讨论。第三，简要总结了几种相关的学习场景。最后，总结了多标签学习的网络资源和开放性研究问题，以供参考。

关键词：多标签学习，标签相关性，问题转换，算法适应

1.引言

传统的监督学习是研究最多的机器学习范式之一，其中每个真实世界的对象（样本）由一个实例（特征向量）表示，并与一个标签关联。形式上，让$\mathcal{X}$表示实例空间，$\mathcal{Y}$表示标签空间，传统监督学习的任务是从训练集$\{(x_i, y_i)|1 \leq i \leq m\}$中学习函数$f : x \to y$。其中，$x_i \in \mathcal{X}$是描述对象属性（特征）的实例，$y_i \in \mathcal{Y}$是描述对象语义的对应标签。因此，传统的监督学习所采用的一个基本假设是，每个样本只属于一个概念，即具有唯一的语义意义。

虽然传统的监督学习是主流和成功的，但由于现实世界的对象可能是复杂的，同时具有多种语义含义，所以在许多学习任务中，上述简化假设并不适用。举几个例子，在文本分类中，新闻文档可以涵盖多个主题，如体育、伦敦奥运会、门票销售和火炬接力；在音乐信息检索中，一首交响乐可以传达钢琴、古典音乐、莫扎特、奥地利等多种信息；在自动视频标注中，一个视频剪辑可能与城市、建筑等场景相关。

为了解释一个真实对象可能具有的多种语义含义，一个直接的解决方案是为该对象分配一组适当的标签，以显式地表达其语义。基于以上考虑，多标签学习的范式自然出现[95]。与传统的监督学习不同，在多标签学习中，每个对象都由一个实例表示，并与一组标签关联，而不是单个标签。任务是学习一个函数，该函数可以预测未知实例的正确标签集。（注释1）

注释1：从广义上讲，多标签学习可以看作是多目标学习的一种可能的实例化[95]，其中每个对象都与多个目标变量（多维输出）相关联[3]。不同类型的目标变量会产生不同的多目标学习实例化，如多标签学习（二元目标）、多维分类（分类/多类目标）、多输出/多元回归（数值目标），甚至结合目标变量类型进行学习。

早期对多标签学习的研究主要集中在多标签文本分类问题上[63，75，97]。在过去的十年中，多标签学习逐渐吸引了机器学习和相关团体的重要注意，而已被广泛应用于不同的问题：自动注释多媒体内容包括图像[5，67，74，85，102]、生物信息学[16，27，107]、web挖掘[51，82]、规则挖掘[84，99]、信息检索[35，114]、标签推荐[50，77]等等。具体来说，近6年（2007-2012），在主要的机器学习相关会议（包括ICML/ECML PKDD/IJCAI/AAAI/KDD/ICDM/NIPS）上发表了标题中含有关键词multi-label（或multilabel）的论文60余篇。

本文对多标签学习这一新兴领域进行了及时的回顾。（注释2）第一部分（第二节）给出了多标签学习的基础知识，包括形式化定义（学习框架、关键挑战、阈值标定）和评价指标（基于实例、基于标签、理论结果）。在第二部分（第三节）中，在常用符号下详细分析了多达八种代表性的多标签算法的技术细节，并进行了必要的分析和讨论。第三部分（第四节）简要总结了几种相关的学习场景。为了总结这篇综述（第五节），我们讨论了在线资源和未来关于

多标签学习的可能研究方向。

注释2：值得注意的是，关于多标签学习技术已有一些很好的综述，如[17，89，91]。与之前在这方面的尝试相比，我们努力提供一个增强了以下方面的丰富版本：a）对更多算法的深入描述；b）最新进展的综合介绍；c）对相关学习场景的简要总结。

2.范式

A.形式定义

1）学习框架

假设$\mathcal{X}=\mathbb{R}^d$（或$\mathbb{Z}^d$）表示d维实例空间，$\mathcal{Y}=\{y_1,y_2,\ldots,y_q\}$表示包含$q$个可能的类标签的标签空间。多标签学习的任务是从多标签训练集$\mathcal{D}=\{(x_i,Y_i)|1\leq i\leq m\}$中学习一个函数$h:X\rightarrow 2^{\mathcal{Y}}$。对于每个多标签样本$(x_i,Y_i)$，$x_i\in\mathcal{X}$是一个$d$维特征向量$(x_{i1},x_{i2},\ldots,x_{id})^{\top}$，$Y_i\subseteq\mathcal{Y}$是与$x_i$关联的一组标签。（注释3）对于任何未知的实例$x\in\mathcal{X}$，多标签分类器$h(\cdot)$预测$h(x)\subseteq\mathcal{Y}$为$x$的一组合适的标签。

注释3：本文将“多标签学习”等价于“多标签分类”，因为分配给每个实例的标签都是二元的。此外，还存在替代的多标签场景，其中除了单个实例外，每个样本都由一组实例[113]或图[54]表示，或者标签空间上可能存在额外的本体知识，如层次结构[2，100]。为了使综述全面而重点突出，假设样本采用单实例表示，并具有平直的类标签。

为了描述任何多标签数据集的属性，可以使用几个有用的多标签指标[72，95]。衡量多标签最自然的方法是标签基数：$LCard(\mathcal{D})=\frac{1}{m}\sum_{i=1}^{m}|Y_i|$，即每个样本标签的平均数量；相应地，标签密度通过标签空间中可能的标签数量来规范化标签基数：$LDen(\mathcal{D})=\frac{1}{|\mathcal{Y}|}\cdot LCard(\mathcal{D})$。另一个流行的多标签测度是标签多样性：$LDiv(\mathcal{D})=|\{Y|\exists x:(x,Y)\in\mathcal{D}\}|$，即出现在数据集的不同的标签集的数量；同样，标签多样性也可以通过样本的数量来标准化，来表示不同标签集的比例：$PLDiv(\mathcal{D})=\frac{1}{|\mathcal{D}|}\cdot LDiv(\mathcal{D})$。

在大多数情况下，多标签学习系统返回的模型对应于一个实值函数$f:\mathcal{X}\times\mathcal{Y}\rightarrow\mathbb{R}$，其中$f(x,y)$可以被视为$y\in\mathcal{Y}$作为$x$的适当标签的置信度。具体地说，给定一个多标签的样本(x,Y)，$f(\cdot,\cdot)$应该对相关标签$y'\in Y$生成较大的输出，而在不相关的标签$y''\notin Y$生成较小的输出，即$f(x,y')>f(x,y'')$。注意，多标签分类器$h(\cdot)$可以由实值函数$f(\cdot,\cdot)$通过$h(x)=\{y|f(x,y)>t(x),\ y\in\mathcal{Y}\}$得到，其中$t:\mathcal{X}\rightarrow\mathbb{R}$作为阈值函数，将标签空间分为相关和不相关的标签集。

为了便于参考，表1列出了本文中使用的主要符号及其数学含义。

表1 主要数学符号概述。

记号	数学含义
$\mathcal{X}$	d维实例空间$\mathbb{R}^d$（或$\mathbb{Z}^d$）
$\mathcal{Y}$	由q个可能的类别标签$\{y_1, y_2, \dots, y_q\}$组成的标签空间
x	d维特征向量$(x_1, x_2, \dots, x_d)^\top (x \in \mathcal{X})$
Y	与x关联的标签集合（$Y \subseteq \mathcal{Y}$）
$\overline{Y}$	Y在$\mathcal{Y}$下的补集
$\mathcal{D}$	多标签训练集$\{(x_i, Y_i) \mid 1 \le i \le m\}$
$\mathcal{S}$	多标签测试集$\{(x_i, Y_i) \mid 1 \le i \le p\}$
$h(\cdot)$	多标签分类器$h: \mathcal{X} \to 2^{\mathcal{Y}}$，其中$h(x)$返回$x$的一组合适的标签
$f(\cdot, \cdot)$	实值函数$f: \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$，其中$f(x, y)$返回$x$为合适的标签的置信度
$rank_f(\cdot, \cdot)$	$rank_f(x, y)$根据$f(x, \cdot)$的降序返回y在$\mathcal{Y}$中的排序
$t(\cdot)$	阈值函数$t: \mathcal{X} \to \mathbb{R}$，其中$h(x) = \{y \mid f(x, y) > t(x), y \in \mathcal{Y}\}$
$\lvert \cdot \rvert$	$\lvert \mathcal{A} \rvert$返回集合$\mathcal{A}$的基数
$[\cdot]$	$[\pi]$返回1当谓词π成立，否则为0
$\phi(\cdot, \cdot)$	$\phi(Y, y)$返回$+1$当$y \in Y$，否则为-1
$\mathcal{D}_j$	从$\mathcal{D}$的第j个类别标签y_i衍生的二分类训练集$\{(x_i, \phi(Y_i, y_j)) \mid 1 \le i \le m\}$
$\psi(\cdot, \cdot, \cdot)$	$\psi(Y, y_j, y_k)$返回$+1$当$y_j \in Y$且$y_k \notin Y$，返回-1当$y_j \notin Y$且$y_k \in Y$
$\mathcal{D}_{jk}$	从$\mathcal{D}$中的标签对(y_j, y_k)衍生的二分类训练集 $\{(x_i, \psi(Y_i, y_j, y_k)) \mid \phi(Y_i, y_j) \ne \phi(Y_i, y_k), 1 \le i \le m\}$
$\sigma_{\mathcal{Y}}(\cdot)$	$\sigma_{\mathcal{Y}}: 2^{\mathcal{Y}} \to \mathbb{N}$从$\mathcal{Y}$的幂集映射到自然数的单射函数（$\sigma_{\mathcal{Y}}^{-1}$为对应的反函数）
$\mathcal{D}_{\mathcal{Y}}^{\uparrow}$	从$\mathcal{D}$衍生的多类别（单标签）训练集$\{(x_i, \sigma_{\mathcal{Y}}(Y_i)) \mid 1 \le i \le m\}$
$\mathcal{B}$	二分类学习算法[复杂度：对训练为$\mathcal{F}_{\mathcal{B}}(m, d)$；对（每个实例）测试为$\mathcal{F}'_{\mathcal{B}}(d)$]
$\mathcal{M}$	多分类学习算法[复杂度：对训练为$\mathcal{F}_{\mathcal{M}}(m, d, q)$；对（每个实例）测试为$\mathcal{F}'_{\mathcal{M}}(d, q)$]

2）关键挑战

很明显，如果每个样本都被限制为只有一个标签，那么传统的监督学习可以看作是多标签学习的退化版本。然而，多标签学习的普遍性不可避免地使相应的学习任务更加难以解决。实际上，从多标签数据中学习的关键挑战在于输出空间的巨大，即标签集的数量随着类标签数量的增加呈指数增长。例如，对于有20个类标签的标签空间（$q = 20$），可能的标签集数量将超过100万个(即2^{20})。

为了应对指数规模的输出空间的挑战，有必要利用标签之间的相关性（或依赖性）来促进学习过程[95，106]。例如，如果我们知道一张图片上有热带雨林和足球的标签，那么它被标注为巴西的可能性就会很高；如果一份文件与政治有关，它就不太可能被贴上娱乐的标签。因此，有效地利用标签相关信息被认为是多标签学习技术成功的关键。根据学习技术考虑到的相关性的程度，现有的相关性利用策略大致可以分为三类[106]：

- 一阶策略：多标签学习的任务是以标签对标签的方式进行处理，从而忽略了与其他标签的共存，如将多标签学习问题分解为一系列独立的二分类问题（每个标签成为一个）[5，16，108]。一阶策略的突出优点在于概念简单、效率高。另一方面，由于忽略了标签相关性，所得到的方法的有效性可能不是最优的。
- 二阶策略：考虑标签之间的成对关系来处理多标签学习任务，如相关标签与不相关标签之间的排序[27，30，107]，或任意一对标签之间的相互作用[33，67，97，114]等等。由于标签关联在一定程度上被二阶策略所利用，所得到的方法具有良好的泛化性能。然而，有些真实应用的标签相关性超越了二阶假设。
- 高阶策略：考虑标签之间的高阶关系来处理多标签学习任务，如将所有其他标签影响到每个标签[13，34，47，103]，或处理标签随机子集之间的连接[71，72，94]等等。显然，高阶策略比一阶和二阶策略具有更强的关联建模能力，而另一方面，高阶策略的计算要求更高，可伸缩性更差。

在第三节中，我们将详细介绍一些采用不同策略的多标签学习算法，以便更好地展示每种策略的优缺点。

3）阈值标定

如第2-A1）节所述，多标签学习中常见的做法是将某个实值函数$f(\cdot,\cdot)$作为学习模型返回[95]。在这种情况下，为了确定未知实例x的正确标签集（即$h(x)$），应该根据阈值函数输出$t(x)$对每个标签上的实值输出$f(x,y)$进行标定。

一般来说，阈值标定可以通过两种策略来完成，一种是将$t(\cdot)$设为常数函数，另一种是通过训练实例来生成$t(\cdot)$[44]。对于第一个策略，当$f(x,y)$在$\mathbb{R}$中取值时，一个直接的选择是使用0作为标定常数[5]。另一个情况是，当$f(x,y)$表示y是x的合适标签的后验概率时，常用的校准常数是0.5[16]。此外，当测试集中所有未知的实例都可用时，可以通过最小化训练集与测试集在某些多标签指标上的差异，尤其是标签基数，来设置标定常数[72]。

对于第二种策略，将使用stacking式过程来确定阈值函数[27，69，107]。一种流行的选择是假设$t(\cdot)$为线性模型，即$t(x)=\langle w^*,f^*(x)\rangle+b^*$，其中$f^*(x)=(f(x,y_1),\ldots,f(x,y_q))^T\in\mathbb{R}^q$是一个$q$维stacking向量，它将学习系统对每个标签的实值输出存储在相应位置上。具体来说，为了求出q维权值向量w^*和偏置b^*，在训练集$\mathcal{D}$的基础上，求解如下线性最小二乘问题：

$$\min_{\{w^*,b^*\}}\sum_{i=1}^{m}\left(\langle \boldsymbol{w}^*,\boldsymbol{f}^*\left(\boldsymbol{x}_i\right)\rangle+b^*-s\left(\boldsymbol{x}_i\right)\right)^2 \tag{1}$$

其中，$s(x_i)=\arg\min_{a\in\mathbb{R}}(|\{y_j|y_j\in Y_i,f(x_i,y_j)\leq a\}|+|\{y_k|y_k\in\overline{Y}_i,f(x_i,y_k)\geq a\}|)$表示stacking模型的目标输出，该模型以分类误差最小对每个训练实例将$\mathcal{Y}$分成相关和不相关的标签。

上述阈值标定策略均为通用技术，可作为任意返回实值函数$f(\cdot,\cdot)$的多标签学习算法的后处理步骤。因此，也存在一些特定于学习算法的自适应阈值标定技术[30，94]，将在第3节中作为其固有的组成部分介绍。由$f(\cdot,\cdot)$诱导$h(\cdot)$的等价机制不是利用阈值函数$t(\cdot)$，而是用$t':\mathcal{X}\rightarrow\{1,2,\ldots q\}$为每个例子指定相关标签的数量，使得$h(X)=\{y|rank_f(X,y)\leq t'(X)\}$[44，82]。其中，当$\mathcal{Y}$中的所有类标签根据$f(x,\cdot)$按降序排列时，$rank_f(x,y)$返回$y$的排序。

B.评价指标

1）简要分类

在传统的监督学习中，使用精度、F-平均、ROC曲线下面积（AUC）等常规指标来评价学习系统的泛化性能。然而，多标签学习中的性能评估比传统的单标签问题复杂得多，因为每个例子都可以同时关联多个标签。因此，提出了许多针对多标签学习的评价指标，这些指标一般可分为基于实例的指标[33，34，75]和基于标签的指标[94]。

根据表1中的符号，设$\mathcal{S}=\{(x_i,Y_i)|1\leq i\leq p)\}$为测试集，$h(\cdot)$为学习完成的多标签分类器。基于实例指标的运作是通过评估学习系统分别在每一个测试样本上的性能，然后返回整个测试集的平均值。不同于上述基于实例指标，基于标签指标的运作通过评估学习系统在每个单独类别标签的性能，然后返回在所有类标签的marco/micro-平均值。

注意，对于$h(\cdot)$，学习系统的泛化性能是从分类的角度度量的。但是，无论是基于实例的还是基于标签的指标，对于大多数多标签学习系统作为一种常见的做法返回的实值函数$f(\cdot,\cdot)$，也可以从排名的角度来衡量泛化性能。图1总结了接下来要介绍的主要多标签评价指标。

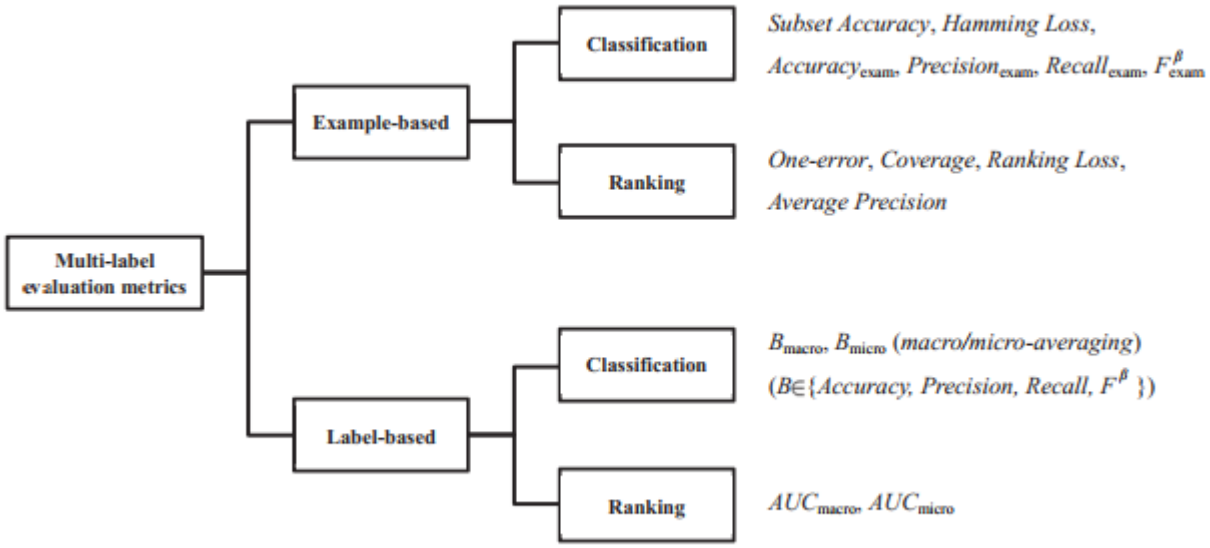

图1 主要多标签评价指标总结

2）基于样本的指标

根据表1中的表示法，基于多标签分类器$h(\cdot)$可以定义6个基于样本的分类度量[33，34，75]：

- 子集精度：

$$\text{subsetacc}(h)=\frac{1}{p}\sum_{i=1}^{p}[h(\boldsymbol{x}_i)=Y_i]$$

子集精度评估正确分类样本的占比，即预测标签集与真实标签集是一样的。直观地，精度可以视为传统的精度指标的多标签推广，它往往是过于严格，特别是当标签空间的大小（即q）很大。

- 汉明损失：

$$\text{hloss}(h)=\frac{1}{p}\sum_{i=1}^{p}|h(\boldsymbol{x}_i)\,\Delta Y_i|$$

其中，Δ表示两个集合之间的对称差。汉明损失评估了误分类的实例-标签对的比例，即丢失了相关的标签或预测了不相关的标签。注意，当$\mathcal{S}$中的每个样本只与一个标签关联时，$hloss_{\mathcal{S}}(h)$将是传统误分类率的$2/q$倍。

- $Accuracy_{exam}$、$Precision_{exam}$、$Recall_{exam}$、F^{β}_{exam}：

$$Accuracy_{\text{exam}}(h)=\frac{1}{p}\sum_{i=1}^{p}\frac{|Y_i\cap h(\boldsymbol{x}_i)|}{|Y_i\bigcup h(\boldsymbol{x}_i)|}$$

$$Precision_{exam}(h)=\frac{1}{p}\sum_{i=1}^{p}\frac{|Y_i\cap h(\boldsymbol{x}_i)|}{|h(\boldsymbol{x}_i)|}$$

$$\mathrm{Recall}_{\mathrm{exam}}(h)=\frac{1}{p}\sum_{i=1}^{p}\frac{|Y_i\cap h(\boldsymbol{x}_i)|}{|Y_i|}$$

$$F_{\mathrm{exam}}^{\beta}(h)=\frac{(1+\beta^2)\cdot Precision_{exam}(h)\cdot Recall_{exam}(h)}{\beta^2\cdot Precision_{exam}(h)+Recall_{exam}(h)}$$

此外，F_{exam}^{β}是带有平衡因子$beta>0$的$Precision_{exam}(h)$和$Recall_{exam}(h)$的综合版本。最常见的选择是$\beta=1$，导出准确率和召回率的调和平均。

当作为中间结果的实值函数$f(\cdot,\cdot)$可用时，还可以定义四个基于实例的排名指标[75]：

- 单误差：

$$\text{one-error}(f)=\frac{1}{p}\sum_{i=1}^{p}\left[\left[\arg\max_{y\in\mathcal{Y}}f(\boldsymbol{x}_i,y)\right]\notin Y_i\right]$$

单误差将评估最优标签不在相关标签集中的样本的比例。

- 覆盖率：

$$\text{coverage}(f)=\frac{1}{p}\sum_{i=1}^{p}\max_{y\in Y_i}\text{rank}_f(\boldsymbol{x}_i,y)-1$$

覆盖率评估了平均需要多少步骤才能向下移动排列好的标签列表，从而覆盖样本的所有相关标签。

- 排序损失：

$$r\,\text{loss}(f)=\frac{1}{p}\sum_{i=1}^{p}\frac{1}{|Y_i||\overline{Y}_i|}\left|\left\{\left(y',y''\right)|f(\boldsymbol{x}_i,y')\leq f(\boldsymbol{x}_i,y''),\left(y',y''\right)\in Y_i\times\overline{Y}_i\right)\right\}|$$

排序损失评估的是逆序标签对的比例，即一个不相关的标签的排名高于一个相关的标签。

- 平均准确率：

$$avgprec(f)=\frac{1}{p}\sum_{i=1}^{p}\frac{1}{|Y_i|}\sum_{y\in Y_i}\frac{|\{y'|\,\text{rank}_f(\boldsymbol{x},y')\leq\text{rank}_f(\boldsymbol{x}_i,y),y'\in Y_i\}|}{\text{rank}_f(\boldsymbol{x}_i,y)}$$

平均准确率评估的是相关标签的平均分数排名高于$y\in Y_i$特定标签。

对于单误差、覆盖率和排序损失，指标值越小，系统的性能越好，对于覆盖率，最优值为$\frac{1}{p}\sum_{i=1}^{p}|Y_i|-1$，对于单误差和排序损失最优值为0。对于其他基于样本的多标签指标，指标值越大，系统的性能越好，最优值为1。

3）基于标签的指标

对于第j类的标签y_j，基于$h(\cdot)$可以定义表征该标签二分类性能的四个基本量：

$$TP_j=|\{x_i|y_j\in Y_i\wedge y_j\in h(x_i),1\leq i\leq p\}|;FP_j=|\{x_i|y_j\notin Y_i\wedge y_j\in h(x_i),1\leq i\leq p\}|$$
$$TN_j=|\{x_i|y_j\notin Y_i\wedge y_j\notin h(x_i),1\leq i\leq p\}|;FN_j=|\{x_i|y_j\in Y_i\wedge y_j\notin h(x_i),1\leq i\leq p\}|$$

换句话说，TP_j, FP_j, TN_j, FN_j代表了关于y_j的真阳性、假阳性、真阴性和假阴性测试样本的数量。根据上述定义，$TP_j + FP_j + TN_j + FN_j = p$自然成立。

基于上述四个量，大部分的二分类指标都可以得到相应的推广。令$B(TP_j, FP_j, TN_j, FN_j)$代表了特定的二分类指标（$B \in \{Accuracy, Precision, Recall, F^\beta\}$（注释4）），基于标签的分类指标可以得到以下模式[94]：

注释4：例如，$Accuracy(TP_j, FP_j, TN_j, FN_j) = \frac{TP_j+TN_j}{TP_j+FP_j+TN_j+FN_j}$，$Precision(TP_j, FP_j, TN_j, FN_j) = \frac{TP_j}{TP_j+FP_j}$，$Recall(TP_j, FP_j, TN_j, FN_j) = \frac{TP_j}{TP_j+FN_j}$，$F^\beta(TP_j, FP_j, TN_j, FN_j) = \frac{(1+\beta^2)\cdot TP_j}{(1+\beta^2)\cdot TP_j+\beta^2\cdot FN_j+FP_j}$。

- Macro平均：

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

- Micro平均：

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

从概念上讲，macro平均和micro平均假设标签和样本的权重相等。不难看出$Accuracy_{macro}(h) = Accuracy_{micro}(h)$和$Accuracy_{micro}(h) + hloss(h) = 1$。注意，macro/micro平均版本（$B_{macro}/B_{micro}$）与第2节-B2小节中基于样本的版本不同。

当中间实值函数$f(\cdot,\cdot)$可用时，可以推导出一个基于标签的排序指标，即macro平均AUC：

$$AUC_{\text{macro}} = \frac{1}{q}\sum_{j=1}^{q} AUC_j = \frac{1}{q}\sum_{j=1}^{q} \frac{\left|\left\{\left(\boldsymbol{x}', \boldsymbol{x}''\right) | f\left(\boldsymbol{x}', y_j\right) \geq f\left(\boldsymbol{x}'', y_j\right), \left(\boldsymbol{x}', \boldsymbol{x}''\right) \in \mathcal{Z}_j \times \overline{\mathcal{Z}}_j\right\}\right|}{\left|\mathcal{Z}_j\right|\left|\overline{Z}_j\right|} \quad (2)$$

其中，$\mathcal{Z}_j = \{x_i | y_j \in Y_i, 1 \leq i \leq p\}$/($\overline{\mathcal{Z}}_j = \{x_i | y_j \notin Y_i, 1 \leq i \leq p\}$)分别对应测试实例带有/不带有标签$y_j$的集合。式（2）的第二行是AUC与Wilcoxon-Mann-Whitney统计量[39]的密切关系。相应地，micro平均AUC也可以推导为：

$$AUC_{\text{micro}} = \frac{\left|\left\{\left(\boldsymbol{x}', \boldsymbol{x}'', y', y''\right) | f\left(\boldsymbol{x}', y'\right) \geq f\left(\boldsymbol{x}'', y''\right), \left(\boldsymbol{x}', y'\right) \in \mathcal{S}^+, \left(\boldsymbol{x}'', y''\right) \in \mathcal{S}^-\right\}\right|}{\left|\mathcal{S}^+\right|\left|\mathcal{S}^-\right|}$$

其中，$\mathcal{S}^+ = \{(x_i, y) | y \in Y_i, 1 \leq i \leq p\}$/ $\mathcal{S}^- = \{(x_i, y) | y \notin Y_i, 1 \leq i \leq p\}$)分别对应于一组相关的/不相关的实例标签对。

对于上述基于标签的多标签指标，指标值越大，系统的性能越好，最优值为1。

4）理论结果

基于指标定义，可以明显看出，现有的多标签指标从多个方面考虑性能，因此具有不同的性质。如第3节所示，大多数多标签学习算法实际上是通过显式或隐式地优化一个特定的指标，从训练样本中学习的。因此，基于公平和诚实的评价，多标签学习算法的性能应该在广泛的度量范围内进行测试，而不是只在优化一个指标。具体来说，最近的理论研究表明，如果从汉明损失的角度来评估分类器，那么以最大化子集精度为目标的分类器将表现得相当差，反之亦然[22，23]。

由于多标签指标通常是非凸和不连续的，因此在实践中，大多数学习算法都是通过优化（凸的）替代多标签指标来实现的[65，66]。近年来，对多标签学习的一致性[32]进行了研究，即学习分类器的期望损失是否随着训练集大小的增大而收敛到贝叶斯损失。具体来说，多标签学习一致性的充要条件基于给定的替代损失函数，这是直观的，而且可以非正式地表示为，对于$\mathcal{X} \times 2^{\mathcal{Y}}$上的一个固定的分布，分类器集合生成的最优替代损失必须下降到分类器集合生成的最优的原始多标签损失。

通过对排序损失的研究，揭示了在标签对上定义的成对凸替代损失没有一个与排序损失一致，最近的一些多标签方法[40]甚至对于确定性多标签学习[32]也是不一致的。（注释5）有趣的是，与这个负面的结果相反，在一致的多标签学习中，一个互补的正面结果被报道为排序损失最小化[21]。通过对二部排序问题的约简[55]，证明了单标签上定义的简单单变量凸替代损失（指数或logistic）与排序损失是一致的，具有明确的界限和收敛速度。

注释5：其中，确定性多标签学习对应于更简单的学习情况，对于任意实例$x \in \mathcal{X}$，存在一个标签子集$Y \subseteq \mathcal{Y}$，使得在给定x的情况下观察Y的后验概率大于0.5，即$\mathbb{P}(Y|x) > 0.5$。

3.学习算法

A.简要分类

算法开发一直是机器学习研究的核心问题，多标签学习也不例外。在过去的十年中，人们提出了大量的算法来学习多标签数据。考虑到在有限的空间内遍历所有现有的算法是不可行的，本文选择了8种具有代表性的多标签学习算法。在此，所选算法的代表性具有条件：a）广泛性：每种算法都具有涵盖多种算法设计策略的独特特征；b）原始影响：大多数算法会导致沿着其研究方向的一系列后续或相关方法；c）有利影响：每种算法在多标签学习领域都是被高度引用的。（注释6）

注释6：根据谷歌学术统计（截止2013年1月），八种算法的每篇论文至少被引用90次，平均被引用200多次。

当我们试图使选择不那么偏向于上述标准时，我们应该注意到，要详细介绍的八种算法并不排除其他方法的重要性。此外，为了保持符号的一致性和数学上的严谨性，我们选择了在常用符号下对每个算法进行重新措辞和表示。本文采用一种简单的多标签学习算法分类[95]：

问题转换方法：这类算法通过将多标签学习问题转换为其他已确定的学习场景来解决多标签学习问题。代表算法包括一阶方法二元关联[5]和高阶方法分类器链[72]，它们将多标签学习的任务转换成二分类任务，二阶方法校准标签的排序方法[30]，它们将多标签学习的任务转换为标签的排序任务，还有高阶方法随机k标签集[94]，它将多标签学习的任务转换成多分类的任务。

算法自适应方法：这类算法通过采用流行的学习技术直接处理多标签数据来解决多标签学习问题。代表性的算法有一阶方法ML-kNN[108]自适应延迟学习技术，一阶方法ML-DT[16]自适应决策树技术，二阶方法排序支持向量机[27]自适应核技术，二阶方法CML[33]自适应信息论技术。

简而言之，问题转换方法的核心思想是将数据拟合到算法中，而算法自适应方法的核心思想是将算法拟合到数据中。图2总结了上述算法，将在本节的其余部分中详细介绍。

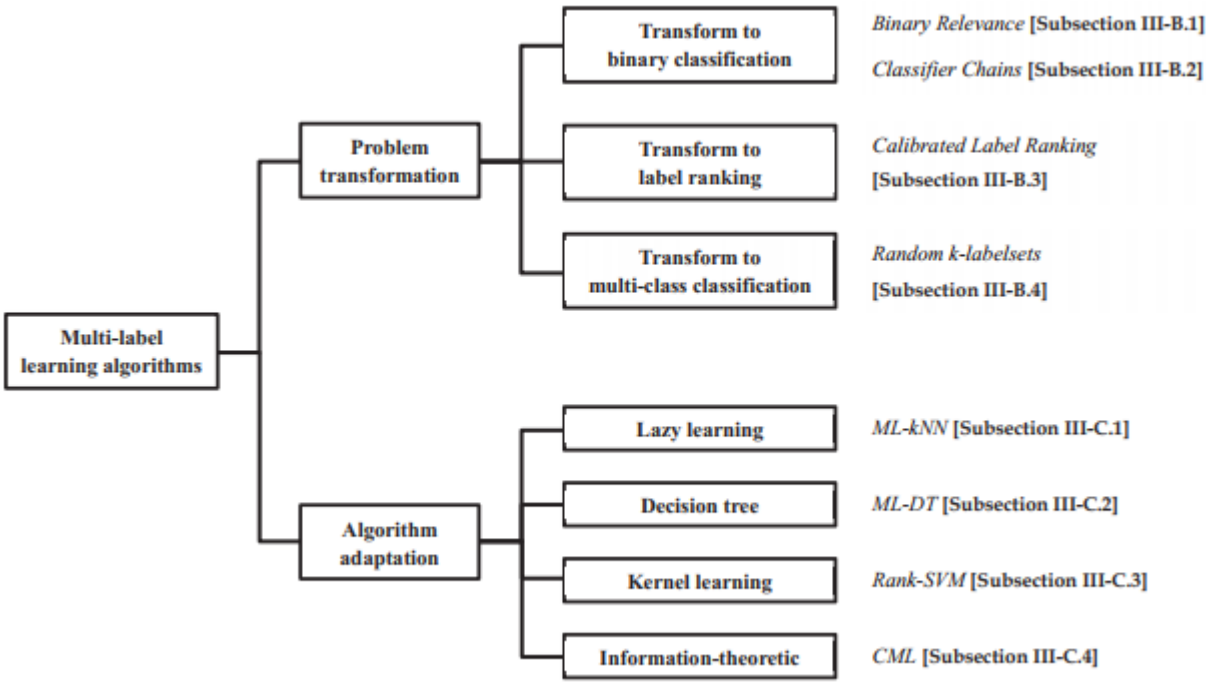

图2 具有代表性的多标签学习算法的分类的综述。

B.问题转换方法

1）二元关联

该算法的基本思想是将多标签学习问题分解为q个独立的二分类问题，其中每个二分类问题对应于标签空间中可能存在的一个标签[5]。

根据表1中的表示法，对于第j类标签y_j，通过考虑每个训练样本与y_j的相关性，构造相应的二分类训练集：

$$\begin{aligned} &\mathcal{D}_j = \{(\boldsymbol{x}_i, \phi(Y_i, y_j)) \,|1 \le i \le m\} \\ &\text{其中 } \phi(Y_i, y_j) = \begin{cases} +1, & \text{当 } y_j \in Y_i \\ -1, & \text{其它} \end{cases} \end{aligned} \tag{3}$$

然后利用一些二分类算法$\mathcal{B}$来诱导一个二分类器$g_j : \mathcal{X} \to \mathbb{R}$，即$g_j \leftarrow \mathcal{B}(\mathcal{D}_j)$。因此，对于任何多标签训练样本$(x_i, Y_i)$，实例$x_i$都将参与$q$个二分类器的学习过程。对于相关标签$y_j \in Y_i$，$x_i$被认为是诱导$g_j(\cdot)$的一个正实例；另一方面，对于不相关的标签$y_k \notin \overline{Y}_i$，$x_i$被视为一个负实例。上述训练策略在[5]中称为交叉训练。

对于未知的实例x，二元关联通过查询每个单独的二分类器上的标签相关性，然后结合相关的标签，预测其关联的标签集Y：

$$Y = \{y_j | g_j(x) > 0, 1 \le j \le q\} \tag{4}$$

注意，当所有二分类器都产生负输出时，预测的标签集Y将为空。为避免产生空预测，可以应用T准则：

$$Y = \{y_j | g_j(x) > 0, 1 \le j \le q\} \bigcup \left\{ y_{j^*} | j^* = \arg\max_{1 \le j \le q} g_j(x) \right\} \tag{5}$$

简而言之，当没有一个二分类器产生正预测时，T准则通过包含输出最大（最小负）的类标签来补充式（4）。除了T准则外，在[5]中还可以找到一些基于每个二分类器输出的标签集预测规则。

评论：二元关联的伪代码如图3所示。它是一种一阶方法，分别为每个标签构建分类器，并为并行实现提供了自然的机会。二元关联最显著的优势在于其处理多标签数据的极其直接的方式（步骤1-4），这已经被用作许多最先进的多标签学习技术的构建块[20，34，72，106]。另一方面，二元关联完全忽略了标签之间的潜在相关性，当q较大，标签密度（即$LDen(\mathcal{D})$）较低时，每个标签的二值分类器可能会出现类不平衡的问题。如图3所示，二元关联的训练复杂度为$\mathcal{O}(q\cdot\mathcal{F}_{\mathcal{B}}(m,d))$，测试复杂度为$\mathcal{O}(q\cdot\mathcal{F}'_{\mathcal{B}}(d))$。（注释7）

注释7：在本文中，计算复杂度主要考虑三个因素，这三个因素对于所有的学习算法来说都是很常见的，即：m（训练样本数）、d（维数）、q（可能的类标签数）。此外，对于问题转换方法中嵌入的二分类/多类学习算法$\mathcal{B}/\mathcal{M}$，我们将其训练复杂度分别表示为$\mathcal{F}_{\mathcal{B}}(m,d)/\mathcal{F}_{\mathcal{M}}(m,d,q)$，其（每个实例）的测试复杂度表示为$\mathcal{F}'_{\mathcal{B}}(d)/\mathcal{F}'_{\mathcal{M}}(d,q)$。本文所报道的计算复杂度结果均为最坏情况界。

Y=BinaryRelevance($\mathcal{D}$, $\mathcal{B}$, $\boldsymbol{x}$)
1. **for** $j=1$ to q **do**
2. Construct the binary training set $\mathcal{D}_j$ according to Eq.(3);
3. $g_j \leftarrow \mathcal{B}(\mathcal{D}_j)$;
4. **endfor**
5. Return Y according to Eq.(5);

图3 二元关联的伪代码。

2）分类器链

该算法的基本思想是将多标签学习问题转化为一个二分类问题链，其中链中后续的二分类器是建立在前面二分类器预测的基础上的[72，73]。

对于可能的类标签$\{y_1,y_2,\ldots,y_q\}$，令$\tau:\{1,\ldots,q\}\rightarrow\{1,\ldots,q\}$是一个置换函数用于指定它们的一个顺序，即$y_{\tau(1)}\succ y_{\tau(2)}\succ\ldots\succ y_{\tau(q)}$。对于在顺序列表中的第$j$个标签$y_{\tau(j)}(1\leq j\leq q)$，相应的二元训练集由附加的每个实例关联这些标签$y_{\tau(j)}$构造：

$$\begin{aligned}&\mathcal{D}_{\tau(j)}=\left\{\left(\left[\boldsymbol{x}_i,\mathrm{pre}^i_{\tau(j)}\right],\phi\left(Y_i,y_{\tau(j)}\right)\right)|1\leq i\leq m\right\}\\&\text{其中}\quad\mathrm{pre}^i_{\tau(j)}=\left(\phi\left(Y_i,y_{\tau(1)}\right),\cdots,\phi\left(Y_i,y_{\tau(j-1)}\right)\right)^{\top}\end{aligned}\tag{6}$$

其中，$[x_i,pre^i_{\tau(j)}]$连接向量x_i和$pre^i_{\tau(j)}$，$pre^i_{\tau(j)}$表示在x_i上按之前的$y_{\tau(j)}$这些标签的二元赋值（具体地说，$pre^i_{\tau(1)}=\emptyset$）（注释8）。之后，一些二元学习算法$\mathcal{B}$是利用诱导二分类器$g_{\tau(j)}:\mathcal{X}\times\{-1+1\}^{j-1}\rightarrow\mathbb{R}$，即$g_{\tau(j)}\leftarrow\mathcal{B}(\mathcal{D}_{\tau(j)})$。换句话说，$g_{\tau(j)}(\cdot)$决定$y_{\tau(j)}$是否为相关的标签。

注释8：在分类器链中[72，73]，二元赋值由0和1表示。为了保持符号的一致性，本文将二元赋值用-1和$+1$表示。

对于未知的实例x，通过迭代遍历分类器链来预测其关联的标签集Y。令$\lambda^x_{\tau(j)}\in\{-1+1\}$代表$x$预测$y_{\tau(j)}$的二元赋值，递归地推导出如下：

$$\begin{aligned}\lambda^x_{\tau(1)}&=\mathrm{sign}\left[g_{\tau(1)}(\boldsymbol{x})\right]\\\lambda^x_{\tau(j)}&=\mathrm{sign}\left[g_{\tau(j)}\left(\left[\boldsymbol{x},\lambda^x_{\tau(1)},\cdots,\lambda^x_{\tau(j-1)}\right]\right)\right]\quad(2\leq j\leq q)\end{aligned}\tag{7}$$

其中，$sign[\cdot]$是符号函数。因此，预测的标签集对应于：

$$Y=\left\{y_{\tau(j)}|\lambda^x_{\tau(j)}=+1,1\leq j\leq q\right\}\tag{8}$$

很明显，分类器链的得到如上所述，其有效性在很大程度上是受指定的排序τ的影响。为了说明排序的效果，一个分类器链的集成可以由标签空间n随机排列构建，即$\tau^{(1)}, \tau^{(2)}, \cdots, \tau^{(n)}$。对于每个排列$\tau^{(r)}(1 \leq r \leq n)$，不是通过在原始训练集$\mathcal{D}$上直接应用$\tau^{(r)}$诱导一个分类器链，而是通过不放回抽样$\mathcal{D}$（$|\mathcal{D}^{(r)}| = 0.67 \cdot |\mathcal{D}|$）[72]或放回抽样（$|\mathcal{D}^{(r)}| = |\mathcal{D}|$）[73]修改后的训练集$\mathcal{D}^{(r)}$。

评论：分类器链的伪代码如图4所示。它是一种高阶方法，以随机的方式考虑标签之间的相关性。与二元关联[5]相比，分类器链具有利用标签关联的优点，但由于其链的特性，使其失去了并行实现的机会。在训练阶段，分类器链增加实例空间来自真实标签的额外特性（即式（6）的$pre^{i}_{\tau(j)}$）。另一种可能性是，当$\mathcal{B}$返回的模型（如朴素贝叶斯）能够产生后验概率时，不保留额外的二值特征，而是将其设置为分类器的概率输出[20，105]。如图4所示，分类器链的训练计算复杂度为$\mathcal{O}(q \cdot \mathcal{F}_{\mathcal{B}}(m, d+q))$，测试计算复杂度为$\mathcal{O}(q \cdot \mathcal{F}'_{\mathcal{B}}(d+q))$。

Y=ClassifierChains($\mathcal{D}$, $\mathcal{B}$, τ, $\boldsymbol{x}$)
1. **for** $j = 1$ to q **do**
2. Construct the chaining binary training set $\mathcal{D}_{\tau(j)}$ according to Eq.(6);
3. $g_{\tau(j)} \leftarrow \mathcal{B}(\mathcal{D}_{\tau(j)})$;
4. **endfor**
5. Return Y according to Eq.(8) (in conjunction with Eq.(7));

图4 分类器链的伪代码。

3）标定标签排序

该算法的基本思想是将多标签学习问题转化为标签排序问题，其中标签之间的排序采用成对比较技术来实现[30]。

对于q个可能的类标签$\{y_1, y_2, \cdots, y_q\}$，总共$q(q-1)/2$个二分类器可以通过成对比较生成，一对为每个标签对$(y_j, y_k)(1 \leq j < k \leq q)$。具体地说，对每一个标签对$(y_j, y_k)$，首先通过考虑每个训练样本对$y_j$和$y_k$的相对相关性构造成对比较对应的二元训练集：

$$\begin{aligned} \mathcal{D}_{jk} &= \{(\boldsymbol{x}_i, \psi(Y_i, y_j, y_k)) \,|\, \phi(Y_i, y_j) \neq \phi(Y_i, y_k), 1 \leq i \leq m\} \\ \text{其中 } \psi(Y_i, y_j, y_k) &= \begin{cases} +1, & \text{当 } \phi(Y_i, y_j) = +1 \text{ 和 } \phi(Y_i, y_k) = -1 \\ -1, & \text{当 } \phi(Y_i, y_j) = -1 \text{ 和 } \phi(Y_i, y_k) = +1 \end{cases} \end{aligned} \tag{9}$$

换句话说，只有与y_j和y_k有明显关联的实例才会包含在$\mathcal{D}_{jk}$中。然后，利用一些二元学习算法$\mathcal{B}$来诱导一个二分类器$g_{jk}: \mathcal{X} \to \mathbb{R}$，即$g_{jk} \leftarrow \mathcal{B}(\mathcal{D}_{jk})$。因此，对于任何多标签训练样本$(x_i, Y_i)$，实例$x_i$都会参与到$|Y_i||\overline{Y}_i|$个二分类器的学习过程中。对于任何实例$x \in \mathcal{X}$，当$g_{jk}(x) > 0$，则学习系统将投票给$y_j$而不投给$y_k$。

对于未知的实例x，经过校准的标签排序首先将其提供给$q(q-1)/2$个训练好的二分类器，以获得每个可能的类标签的总投票：

$$\zeta(\boldsymbol{x}, y_j) = \sum_{k=1}^{j-1} [g_{kj}(\boldsymbol{x}) \leq 0] + \sum_{k=j+1}^{q} [g_{jk}(\boldsymbol{x}) > 0] \quad (1 \leq j \leq q) \tag{10}$$

基于上述定义，不难验证$\sum_{j=1}^{q} \zeta(x, y_j) = q(q-1)/2$。其中，$\mathcal{Y}$中的标签可以根据它们各自的投票进行排序。

然后，进一步指定阈值函数，将排序后的标签列表划分为相关的和不相关的标签集。为了在成对比较框架下实现这一点，标定后的标签排序在每个多标签训练样本(x_i, Y_i)中加入一个虚拟标签y_V。从概念上讲，虚拟标签是x_i的相关和不相关标签之间的一个人工分界点[6]。也就是说，我们认为y_V的排序低于$y_j \in Y_i$，而高于$y_k \in \overline{Y}_i$。

除了原始的$q(q-1)/2$个二分类器外，还将引入q个辅助二分类器，每个新标签对(y_j, y_V)一个。与式（9）相似，可以构造(y_j, y_V)对应的二元训练集如下：

$$\mathcal{D}_{j\mathrm{V}}=\{(\boldsymbol{x}_i,\varphi(Y_i,y_j,y_{\mathrm{V}}))\,|1\le i\le m\}$$
$$\text{其中 }\varphi(Y_i,y_j,y_{\mathrm{V}})=\begin{cases}+1, & \text{当 } y_j\in Y_i\\ -1, & \text{其它}\end{cases}\tag{11}$$

在此基础上，利用二元学习算法$\mathcal{B}$，推导出与虚拟标签对应的二分类器$g_{jV}:\mathcal{X}\to\mathbb{R}$，即$g_{jV}\leftarrow\mathcal{B}(\mathcal{D}_{jV})$。之后，公式（10）中指定的总投票将使用新引入的分类器进行更新：

$$\zeta^*(\boldsymbol{x},y_j)=\zeta(\boldsymbol{x},y_j)+[g_{j\mathrm{V}}(\boldsymbol{x})>0]\quad(1\le j\le q)\tag{12}$$

此外，虚拟标签的总投票可以计算为：

$$\zeta^*(\boldsymbol{x},y_{\mathrm{V}})=\sum_{j=1}^{q}[g_{j\mathrm{V}}(\boldsymbol{x})\le 0]\tag{13}$$

因此，对未知实例x对应的预测标签集为：

$$Y=\{y_j|\zeta^*(\boldsymbol{x},y_j)>\zeta^*(\boldsymbol{x},y_{\mathrm{V}}),1\le j\le q\}\tag{14}$$

通过将式（11）与式（3）进行比较，可以看出标定标签排序所使用的训练集D_{jV}与二元关联[5]所使用的训练集D_j是相同的。因此，标定后的标签排序可以看作是成对比较的一个强化版本，将常规的$q(q-1)/2$个二分类器与具有二元相关性的q个二分类器进行扩充，促进学习[30]。

评论：标定标签排序的伪代码如图5所示。它是一种二阶方法，为任何一对类标签构建分类器。与之前介绍的算法[5，72]所采用一对其余的方式构造二分类器相比，标定后的标签排序采用一对一的方式构造二分类器（虚拟标签除外），具有减轻类不平衡问题的负面影响的优点。另一方面，由标定标签排序生成的二分类器的数量依据类别标签的数量（即q）从线性扩展到二次规模。改进标定标签排序主要关注减少分类器的二次的规模，通过在测试阶段查询精确修剪[59]或近似修剪[60，61]。利用底层二元学习算法$\mathcal{B}$的特性，如感知器的对偶表示[58]，可以在训练阶段更有效地诱导分类器[57]。如图5所示，标定标签排序的训练计算复杂度为$\mathcal{O}(q^2\cdot\mathcal{F}_{\mathcal{B}}(m,d))$，测试计算复杂度为$\mathcal{O}(q^2\cdot\mathcal{F}'_{\mathcal{B}}(d))$。

Y=CalibratedLabelRanking($\mathcal{D}$, $\mathcal{B}$, $\boldsymbol{x}$)
1. **for** $j=1$ to $q-1$ **do**
2. **for** $k=j+1$ to q **do**
3. Construct the binary training set $\mathcal{D}_{jk}$ according to Eq.(9);
4. $g_{jk}\leftarrow\mathcal{B}(\mathcal{D}_{jk})$;
5. **endfor**
6. **endfor**
7. **for** $j=1$ to q **do**
8. Construct the binary training set $\mathcal{D}_{j\mathrm{V}}$ according to Eq.(11);
9. $g_{j\mathrm{V}}\leftarrow\mathcal{B}(\mathcal{D}_{j\mathrm{V}})$;
10. **endfor**
11. Return Y according to Eq.(14) (in conjunction with Eqs.(10)-(13));

图5 校准标签排序的伪代码。

4）随机k标签集

该算法的基本思想是将多标记学习问题转换为一个集成的多分类问题，集成中的每个组件学习器都以$\mathcal{Y}$的一个随机子集为目标，在这个子集上使用标签幂集（LP）技术诱导出一个多类分类器[92，94]。

LP是一种将多标签学习问题转化为多类别（单标签）分类问题的直接方法。令$\sigma_{\mathcal{Y}}:2^{\mathcal{Y}}\to\mathbb{N}$是从$\mathcal{Y}$的幂集映射到自然数的单射函数，$\sigma_{\mathcal{Y}}^{-1}$是对应的反函数。在训练阶段，LP首先将原多标签训练集$\mathcal{D}$转换为如下的多类训练集，将$\mathcal{D}$中出现的每一个不同的标签集作为一个新的类处理：

$$\mathcal{D}_{\mathcal{Y}}^{\dagger}=\left\{\left(\boldsymbol{x}_i,\sigma_{\mathcal{Y}}\left(Y_i\right)\right)|1\leq i\leq m\right\} \tag{15}$$

其中$\mathcal{D}_{\mathcal{Y}}^{\dagger}$涵盖的新类集合对应于:

$$\Gamma\left(\mathcal{D}_{\mathcal{Y}}^{\dagger}\right)=\left\{\sigma_{\mathcal{Y}}\left(Y_i\right)|1\leq i\leq m\right\} \tag{16}$$

显然，$|\Gamma(\mathcal{D}_{\mathcal{Y}}^{\dagger})|\leq\min(m,2^{|\mathcal{Y}|})$。然后，一些多类学习算法$\mathcal{M}$利用诱导的多分类器 $g_{\mathcal{Y}}^{\dagger}:\mathcal{X}\to\Gamma(\mathcal{D}_{\mathcal{Y}}^{\dagger})$，即 $g_{\mathcal{Y}}^{\dagger}\leftarrow\mathcal{M}(\mathcal{D}_{\mathcal{Y}}^{\dagger})$。因此，对于任何多标签训练样本$(x_i,Y_i)$，实例$x_i$将重新赋值新的单标签映射$\sigma_{\mathcal{Y}}(Y_i)$，然后参与多分类诱导。

对于未知的实例x，LP首先查询多类分类器的预测，然后将其映射回$\mathcal{Y}$的幂集，从而预测其关联的标签集Y：

$$Y=\sigma_{\mathcal{Y}}^{-1}\left(g_{\mathcal{Y}}^{\dagger}(\boldsymbol{x})\right) \tag{17}$$

遗憾的是，LP在实际可行性方面存在两大局限性：a）不完全性：如式（16）和（17）所示，LP仅局限于预测训练集中出现的标签集，即不能推广到$\{Y_i|1\leq i\leq m\}$以外；b）效率低：当$\mathcal{Y}$很大时，可能会有太多的新映射类$\Gamma(\mathcal{D}_{\mathcal{Y}}^{\dagger})$，导致训练$g_{\mathcal{Y}}^{\dagger}(\cdot)$的过高复杂性，而且对于一些新的映射类有非常少的训练样本。

为了保持LP的简单性，同时克服其两个主要缺点，随机k标签集将集成学习[24，112]与LP相结合，以从多标签数据中学习。关键策略是仅在随机k标签集（$\mathcal{Y}$中大小为k的子集）上调用LP来保证计算效率，然后集成多个LP分类器来实现预测的完整性。

令$\mathcal{Y}^k$代表$\mathcal{Y}$中所有可能的k标签集的集合，其中第l个k标签集表示为$\mathcal{Y}^k(l)$，即$\mathcal{Y}^k(l)\subseteq\mathcal{Y},|\mathcal{Y}^k(l)|=k,1\leq l\leq\binom{q}{k}$。与式（15）类似，将原标签空间$\mathcal{Y}$压缩为$\mathcal{Y}^k(l)$，也可以构造一个多类训练集：

$$\mathcal{D}_{y^*(l)}^{\dagger}=\left\{\left(x_i,\sigma_{y^*(l)}\left(Y_i\cap\mathcal{Y}^k(l)\right)\right)|1\leq i\leq m\right\} \tag{18}$$

其中，$\mathcal{D}_{\mathcal{Y}^k(l)}^{\dagger}$所覆盖的新类集合对应：

$$\Gamma\left(\mathcal{D}_{y^k(l)}^{\dagger}\right)=\left\{\sigma_{y^*(l)}\left(Y_i\cap\mathcal{Y}^k(l)\right)|1\leq i\leq m\right\}$$

然后利用多类学习算法$\mathcal{M}$来诱导多类分类器$g_{y^k(i)}^{\dagger}:\mathcal{X}\to\Gamma\left(\mathcal{D}_{y^k(l)}^{\dagger}\right)$，即$g_{\mathcal{Y}^k(l)}^{\dagger}\leftarrow\mathcal{M}\left(\mathcal{D}_{\mathcal{Y}^k(l)}^{\dagger}\right)$。

为了创建一个包含n个组件分类器的集成，随机k标签集调用n个随机k标签集$\mathcal{Y}^k(l_r)(1\leq r\leq n)$上的LP，每个LP调用一个多类分类器$g_{\mathcal{Y}^k(l_r)}^{\dagger}(\cdot)$。对于未知的实例$x$，每个类标签计算以下两个量：

$$\begin{aligned}\tau\left(\boldsymbol{x},y_j\right)&=\sum_{r=1}^{n}\left[y_j\in\mathcal{Y}^k\left(l_r\right)\right]\quad(1\leq j\leq q)\\\mu\left(\boldsymbol{x},y_j\right)&=\sum_{r=1}^{n}\left[y_j\in\sigma_{y^k(l_r)}^{-1}\left(g_{y^k(l_r)}^{\dagger}(\boldsymbol{x})\right)\right]\quad(1\leq j\leq q)\end{aligned} \tag{19}$$

其中，$\tau(x,y_j)$计算y_j可以从集成中收到的最大数量的投票，而$\mu(x,y_j)$计算y_j从集成中收到的实际投票数量。因此，预测的标签集对应于：

$$Y = \{y_j | \mu(\boldsymbol{x}, y_j) / \tau(\boldsymbol{x}, y_j) > 0.5, 1 \leq j \leq q\} \tag{20}$$

换句话说，当实际票数超过最大票数的一半时，就认为y_j是相关的。对于n个k标签集创建的集合，每个标签上的最大投票数平均为nk/q。随机k标签集的经验设置是$k = 3$和$n = 2q$[92，94]。

评论：随机k标签集的伪代码如图6所示。这是一种高阶方法，其中标签相关的程度由k标签集的大小控制。除了使用k标签集，另一种改进LP的方法是对$\mathcal{D}$中出现小于预先指定的计数阈值的不同标签集进行修剪[71]。虽然随机k标签集嵌入集成学习作为其固有的部分来修正LP的主要缺点，但是集成学习可以作为一种元学习策略，通过包含同质[72，76]或异构[74，83]组件的多标签学习器来促进多标签学习。如图6所示，随机k标签集的训练计算复杂度为$\mathcal{O}(n \cdot \mathcal{F}_{\mathcal{M}}(m, d, 2^k))$，测试计算复杂度为$\mathcal{O}(n \cdot \mathcal{F}'_{\mathcal{M}}(d, 2^k))$。

Y=Random$k$-Labelsets($\mathcal{D}$, $\mathcal{M}$, k, n, $\boldsymbol{x}$)

1. **for** $r = 1$ to n **do**
2. Randomly choose a k-labelset $\mathcal{Y}^k(l_r) \subseteq \mathcal{Y}$ with $|\mathcal{Y}^k(l_r)| = k$;
3. Construct the multi-class training set $\mathcal{D}^{\dagger}_{\mathcal{Y}^k(l_r)}$ according to Eq.(18);
4. $g^{\dagger}_{\mathcal{Y}^k(l_r)} \leftarrow \mathcal{M}\left(\mathcal{D}^{\dagger}_{\mathcal{Y}^k(l_r)}\right)$;
5. **endfor**
6. Return Y according to Eq.(20) (in conjunction with Eq.(19));

图6 随机k标签集的伪代码。