

[关闭]

@zhuanxu 2018-01-15 08:29 字数 1511 阅读 79

# 记:多标签分类问题

机器学习 多标签分类

最近遇到给个标签问题，就是给一个object打个多个标签，网上查了很多资料。发现百度没搜索出什么，后来是到知网上找到一些靠谱的资料，然后在Google一下。现在总结下多标签问题。

多标签方法大致可以分为两类，分别是问题转换和算法改造。

先描述下问题：

多标记学习问题可以描述如下：设样本的特征属性 $X = \{x_i: i = 1 \dots m\}$ ，有限标记集合 $Y = \{\lambda_j: j = 1 \dots q\}$ ，给定学习样本集 $S = \{(x_i, L_i): i = 1 \dots k\}$ ，其中 $x_i \in X, L_i \subseteq Y$ 。要求构造分类器  $h$ ，能够对未知样本集 $T = \{(x_{k+i}, ?): i = 1 \dots p\}$ 进行标记。

先介绍问题转换方法.

## 问题转换方法

第一个大类是基于标记转换方法。

第一个是Binary Relevance (BR)。

根据标签我们将数据重新组成正负样本，针对每个类别标签，我们分别训练基分类器，整体复杂度 $q \times O(C)$ ，其中  $O(C)$  为基础分类算法的复杂度，因此，BR 算法针对标记数量  $q$  比较小的情况下适用。但是在很场景中，标记是有树状的层次的关联的。对于这种情况，BR 就没有考虑到这些标记之间的关联性。

Instances(E)	Labels(Y)		Labels	Positive	Negative
E <sub>1</sub>	Y <sub>1</sub> = {y <sub>2</sub> , y <sub>3</sub> }	➡	y <sub>1</sub>	E <sub>2</sub> , E <sub>4</sub>	E <sub>1</sub> , E <sub>3</sub>
E <sub>2</sub>	Y <sub>2</sub> = {y <sub>1</sub> }		y <sub>2</sub>	E <sub>1</sub>	
E <sub>3</sub>	Y <sub>3</sub> = {y <sub>3</sub> , y <sub>4</sub> }		y <sub>3</sub>	E <sub>1</sub> , E <sub>3</sub> , E <sub>4</sub>	E <sub>2</sub>
E <sub>4</sub>	Y <sub>4</sub> = {y <sub>1</sub> , y <sub>3</sub> }		y <sub>4</sub>	E <sub>3</sub>	

第二个是Classifier Chain(CC)。

针对BR中标签关联性的问题，CC中它将这些基分类器  $C_j, j = 1 \dots q$  串联起来形成一条链，前一个基分类器的输出作为下一个基分类器的输入。

第二大类是基于样本实例转换方法

第一个是创新新的标记(Label-Powerset)。

Instances(E)	Labels(Y)		Instances(E)	Labels(Y)
E <sub>1</sub>	Y <sub>1</sub> = {y <sub>2</sub> , y <sub>3</sub> }	➡	E <sub>1</sub>	y <sub>2,3</sub>
E <sub>2</sub>	Y <sub>2</sub> = {y <sub>1</sub> }		E <sub>2</sub>	y <sub>1</sub>
E <sub>3</sub>	Y <sub>3</sub> = {y <sub>3</sub> , y <sub>4</sub> }		E <sub>3</sub>	y <sub>3,4</sub>
E <sub>4</sub>	Y <sub>4</sub> = {y <sub>1</sub> , y <sub>3</sub> }		E <sub>4</sub>	y <sub>1,3</sub>

这样做的代价是标记的数量就会增加，并且一些标记只有很少的实例，但是 LP 的优点是考虑到了标记之间的关联性。

第二个是分解多标记

Instances(E)	Labels(Y)		Instances(E)	Labels(Y)
E <sub>1</sub>	Y <sub>1</sub> = {y <sub>2</sub> , y <sub>3</sub> }	➡	E <sub>1</sub>	y <sub>2</sub>
E <sub>2</sub>	Y <sub>2</sub> = {y <sub>1</sub> }		E <sub>1</sub>	y <sub>3</sub>
E <sub>3</sub>	Y <sub>3</sub> = {y <sub>3</sub> , y <sub>4</sub> }		E <sub>2</sub>	y <sub>1</sub>
E <sub>4</sub>	Y <sub>4</sub> = {y <sub>1</sub> , y <sub>3</sub> }		E <sub>3</sub>	y <sub>3</sub>
			E <sub>3</sub>	y <sub>4</sub>
			E <sub>4</sub>	y <sub>1</sub>
			E <sub>4</sub>	y <sub>3</sub>

上面图中的意思是我们可以将训练数据多次使用，叫做cross-training，即我们将上图中 E1 既当做训练 y2 类别是正样本，也当做训练 y3 样本时候的正样本，感觉跟Binary Relevance (BR)算法是一个意思。

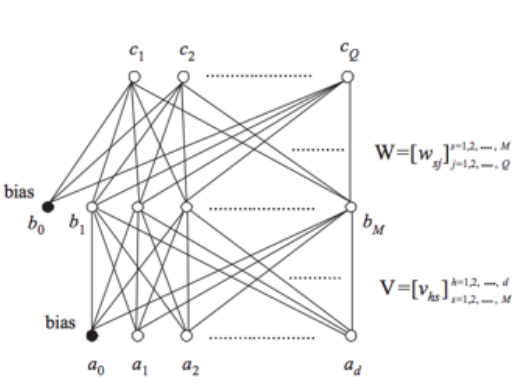
算法改造方法

算法改造算法针对特殊的算法改造而来。  
主要介绍两个，可以具体参考 <http://scikit.ml/api/classify.html#adapted-algorithms> 。

神经网络

此处介绍下论文 [Multi-Label Neural Networks with Applications to Functional Genomics and Text Categorization](#)，一种神经网络算法。

其实就是简单的深度网络：



但是需要注意的是我们的 loss 函数的选取，假设我们选择

$$E_i = \sum_{j=1}^Q (c_j^i - d_j^i)^2$$

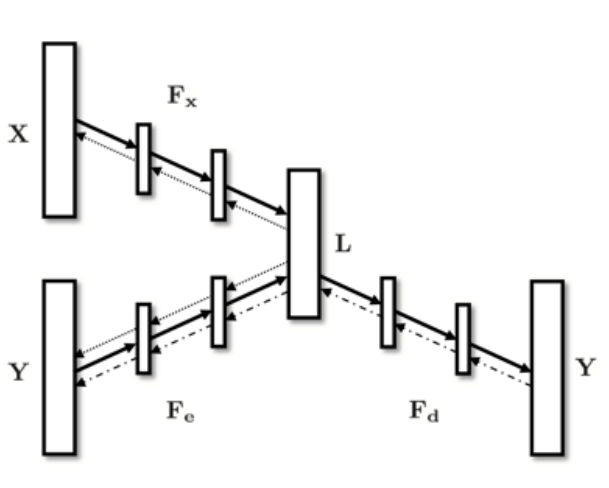
那相当于只是考虑了单个标签值，0 or 1，没有考虑不同标签之间相关性，所以我们将loss改为如下：

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

上面k是有标签的下标，而l是没有标签的下标，我们考量了有标签的值其意义大于没有标签的值。

最后我们在介绍一篇新出的神经网络的模型，论文[Learning Deep Latent Spaces for Multi-Label Classification](#)

其模型如下：



其中 Fx, Fe, Fd 分别是3个 dnn，分别代表 特征提取，标签encode，隐向量 decode，而 loss 函数有两部分组成：

$$\Theta = \min_{F_x, F_e, F_d} \Phi(F_x, F_e) + \alpha \Gamma(F_e, F_d)$$

其中 embedding loss 为：

$$\Phi(F_x, F_e) = \text{Tr}(C_1^T C_1) + \lambda \text{Tr}(C_2^T C_2 + C_3^T C_3),$$

where

$$\begin{aligned} C_1 &= F_x(X) - F_e(Y) \\ C_2 &= F_x(X) F_x(X)^T - I \\ C_3 &= F_e(Y) F_e(Y)^T - I. \end{aligned}$$

output loss 为：

$$\begin{aligned} \Gamma(F_e, F_d) &= \sum_{i=1}^N E_i \\ E_i &= \frac{1}{|Y_i^1| |Y_i^0|} \sum_{(p,q) \in Y_i^1 \times Y_i^0} \exp(-(F_d(F_e(x_i)))^q - F_d(F_e(x_i))^p), \end{aligned}$$

可以看到这个跟[Multi-Label Neural Networks with Applications to Functional Genomics and Text Categorization](#)中的loss函数是一样的。

如果对这篇论文还有不理解的，非常幸运的是网上有论文的实现，见[C2AE-Multilabel-Classification](#).

## 总结

本文对多标签问题简单做了个介绍，想起现在图片分类，视频内容识别等场景好多都是多标签问题，有时间再继续深入了解的。

你的鼓励是我继续写下去的动力，期待我们共同进步。