Multi-source domain adaptation

ZZL

2017 年 10 月 14 日

一 Introduction

域适应问题是机器学习中一个常见的重要的问题，在这方面也已经有了很多工作，但有文章已经指出，域迁移的效果和源域与目标域之间分布的差异有很大关系，当分布的差异较大时候，甚至会出现负迁移的情况。对于一个未知的目标域，我们考虑基于多个源域的迁移学习，这样可以使得我们的模型更加鲁棒。

在多源域迁移的工作中，大致可以分为两个大类，一类 Feature representation approaches，一类是 Combination of prelearned classifiers。这篇文章主要介绍第二类的相关经典方法。

由于阅读的论文比较老，本篇文章只介绍其中的我个人认为的经典内容，不会对全文进行介绍。

二 Cross-domain Representation-learning Framework with Combination of Class-separate and Domain-merge Objectives

这篇文章的思想在于，扩大不同类别之间的差异，减小不同数据分布之间的差异。很多文章的工作是基于减小数据分布之间的差异，这篇文章考虑了不同类别之间的差异，是这篇文章的主要贡献点。这篇文章借助了很多 represenation learning 的思想。这一块的工作可以说是表示学习与迁移学习的交叉。

本文介绍这篇文章的两个方法，一是基于特征选择的方法，二是基于特征重建的方法。

2.1 Cross-domain Feature Selection

皮尔森相关系数：

$$r = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}}.$$

皮尔森相关系数是衡量两组变量之间的线性相关系数，可以看作是两组变量的协方差除以两组变量的标准差的乘积。分母的标准差可以看作对协方差做归一化，因为协方差的大小跟数据的值有直接关系。

这个方法有借助 the theory of the Correlation Feature Selection（CFS）。CFS 的核心思想是好的特征集是特征和标签是高度相关和其他特征低相关（结合去相关的预处理来理解这里）。

$$CFS = \max_{S_k}\left[\frac{r_{cf_1} + r_{cf_2} + \cdots + r_{cf_k}}{\sqrt{k + 2(r_{f_1f_2} + \cdots + r_{f_if_j} + \cdots r_{f_kf_1})}}\right]$$

r_{cf_i}表示类和样本之间的相关性，$r_{f_if_i}$表示样本和样本之间的相关性。

这个算法分为两步，第一步是分别进行 class-separate and domain-separate evaluations，第二步是利用 CFS 将 c-s 与 d-s 结合起来。

$$r_{cf/df} = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i^{c/d} - \overline{Y^{c/d}})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i^{c/d} - \overline{Y^{c/d}})^2}}.$$

这里Y^c和Y^d分别表示类和域的标签，“/”表示 or。r_{cf}、r_{df}表示特征和类别、域之间的相关性。我们希望r_{cf}具有高相关性，r_{df}具有低相关性。由此目标再结合 CFS，我们的优化目标是 $Merit_{S_k} = \frac{k\overline{r_{cf}} - \alpha * k\overline{r_{df}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$，α是权重，k 是样本的数量。

2.2 Cross-domain Feature Reconstruction

这里考虑线性的特征重建，但又跟线性自编码器不相同，自编码器的数学模型可以表示为W_2W_1X，X 为样本集，W_1是编码矩阵，W_2是解码矩阵。而这篇文章中采用的模型是 $J(\phi) = \phi^\top Q_{max}\phi$，$\phi$是转换矩阵。Q 可以理解为对 X 的预处理后的数据，在预处理中，我们添加了我们需要的信息。

这个算法借鉴了 Fisher Discriminant Analysis Theory，FDA 的主要思想是 maximize the class-separate degree（最大化类之间的距离），对于域之间的差异来说，我们只需要把目标从最大变为最小即可。

这里介绍两个与类有关的公式：$S_b = \sum_{i=1,2} n_i(\mu_i - \mu)(\mu_i - \mu)^\top$

$S_w = \sum_{i=1,2}(\sum_{j=1}^{n_i}(x_j^i - \mu_i)(x_j^i - \mu_i)^\top)$。这里 i 表示类的编号，$n_i$表示类 i 中有样本的个数，u 表示全体样本的均值，x_j^i表示第 i 个类中的第 j 个样本。S_b、S_w分别表示类之间的距离，类中各个样本之间的距离，我们需要类之间的距离大，类中的数据联系紧密，这也是 kmeans 聚类算法的思想。

下面是两个跟域有关的公式：$S_d^U = (\mu^{D_1} - \mu^{D_2})(\mu^{D_1} - \mu^{D_2})^\top$

$S_d^L = (\mu_1^{D_1} - \mu_1^{D_2})(\mu_1^{D_1} - \mu_1^{D_2})^\top + (\mu_2^{D_1} - \mu_2^{D_2})(\mu_2^{D_1} - \mu_2^{D_2})^\top$。这里$u^{D_i}$表示第 i 个域中数据的均值，$u_i^{D_i}$表示第 i 个域中第 i 个类的数据的均值。这两个公式分别表示的是域之间分布的差异和考虑类别的情况下域之间分布的差异。这篇文章结合这两个公式来表示域之间的分布的差异$S_d = S_d^U + (1 + \min(n_{tr}^{D_1}/n_{te}^{D_1} + n_{tr}^{D_2}/n_{te}^{D_2}))S_d^L$，$n_{tr}^{D_i}$、$n_{te}^{D_i}$表示域$D_i$中有标签和无标签的样本的数量。如果是多个源域，比如 n 个域，那这里S_d为（n-1）*n/2 个域对之间差异的和。

接下来结合类差异与域差异，优化目标为$J(\phi) = \frac{\phi^\top S_b \phi}{\phi^\top (S_w + \alpha S_d)\phi}$。因为$S_w$、$S_d$的目标是最小化，那这里取其倒数即可。

2.3 思考

关于 2.2 中的重建，目前没有理解重建的意义，但这两个算法给了我们很好的启示，一是思考方向多了类距离，二是度量函数的构建，三是两者的组合。在综述中把这篇文章划入了联合模型得类别，感觉这也是个分类错误。

三 Boosting for transfer learning with multiple sources

这篇文章提供了两个基于 boosting 得算法框架，典型得一篇联合模型得文章，很有借鉴意义。

3.1 Adaboost

Adaboost 算法是将多个弱分类器，组合成强分类器。Adaboost 是集成学习得一种，在各种数据挖掘、人工智能比赛中，冠军大多数都采用集成学习得思路。

Adaboost 原理如下：

（1）初始化训练数据（每个样本）的权值分布：如果有 N 个样本，则每一个训练的样本点最开始时都被赋予相同的权重：1/N。

（2）训练弱分类器。具体训练过程中，如果某个样本已经被准确地分类，那么在构造下一个训练集中，它的权重就被降低；相反，如果某个样本点没有被准确地分类，那么它的权重就得到提高。同时，得到弱分类器对应的话语权。然后，更新权值后的样本集被用于训练下一个分类器，整个训练过程如此迭代地进行下去。

（3）将各个训练得到的弱分类器组合成强分类器。各个弱分类器的训练过程结束后，分类误差率小的弱分类器的话语权较大，其在最终的分类函数中起着较大的决定作用，而分类误差率大的弱分类器的话语权较小，其在最终的分类函数中起着较小的决定作用。换言之，误差率低的弱分类器在最终分类器中占的比例较大，反之较小。

总得来说，Adaboost 在训练过程中，给误分类样本更高得权重，在预测过程

中，给予正确率高的弱分类器更高得权重。

算法流程如下：

输入：训练数据集$T=\{(x_1,y_1),(x_2,y_2),\dots(x_N,y_N)\}$,其中$x_i \in X \subseteq R^n, y_i \in Y=\{-1,+1\}$;弱学习算法。
输出：最终分类器G(x)。
（1）初始化训练数据的权值分布

$$D_1=(w_{11},\dots,w_{1i},\dots,w_{1N}), w_{1i}=\frac{1}{N}, i=1,2,\dots,N$$

注：第一次训练弱分类器时各个样本的权值是相等的。
（2）对m=1,2，...,M　**注：这里是个循环**
（a）使用具有权值分布D_m的训练数据集学习，得到基本分类器

$$G_m: X \to \{-1,+1\}$$

（b）计算$G_m(x)$在训练集上的分类误差率

$$e_m=P(G_m(x_i)\neq y_i)=\sum_{i=1}^{n} w_{mi} I(G_m(x_i)\neq y_i)$$

注：$I(G_m(x_i)\neq y_i)$：不等函数I值为1.相等函数值为0。
（c）计算$G_m(x)$的系数

$$\alpha_m=\frac{1}{2} log \frac{1-e_m}{e_m}$$

这里的对数是自然对数。注：显然α_m是e_m的调单减函数，这里就解释了为什么对于没有正确分类的数据要加大权值。
（d）更新训练数据集的权值分布

$$D_{m+1}=(w_{m+1,1},\dots,w_{m+1,i},\dots,w_{m+1,N})$$

$$w_{m+1,i}=\frac{w_{mi}}{Z_m} exp(-\alpha_m y_i G_m(x_i)) \qquad i=1,2,\dots,N$$

这里，Z_m是规范化因子

$$Z_m=\sum_{i=1}^{N} w_{mi} exp(-\alpha_m y_i G_m(x_i))$$

它使D_{m+1}成为一个概率分布。
注：自己比较Z_m与$w_{m+1,i}$的表达式，会发现这里的Z_m就是在对$w_{m+1,i}$进行归一化工作。
（3）构建基本分类器的线性组合

$$f(x)=\sum_{m=1}^{M} \alpha_m G_m(x)$$

得到最终分类器

$$G(x)=sign(f(x))=sign(\sum_{m=1}^{M} \alpha_m G_m(x))$$

参数解释

1、$G_m(x)$的系数

$$\alpha_m = \frac{1}{2} log \frac{1-e_m}{e_m}$$

α_m表示$G_m(x)$在最终分类器中的重要性。由α_m的表达式可知，当$e_m \leqslant \frac{1}{2}$时，$\alpha_m \geqslant 0$,并且$\alpha_m$随着$e_m$的减小而增大，所以分类误差越小的基本分类器在最终分类器中的作用越大。

2、计算基本分类器$G_m(x)$在加权训练数据集上的分类误差率：

$$e_m = P(G_m(x_i) \neq y_i) = \sum_{G_m(x_i) \neq y_i} w_{mi}$$

,这里，w_{mi}表示第m轮中第i个实例的权值，$\sum_{i=1}^{N} = 1$（因为权值利用Z_m进行了归一化）。这表明，$G_m(x)$在加权的训练数据集上的分类误差是被$G_m(x)$误分类杨蓓的权值之和，由此可以看出数据权值分布D_m与基本分类器$G_m(x)$的分类误差率的关系。

3.2 TrAdaBoost with multiple sources

这个算法是基于多域得集成迁移学习，算法流程如下：

Algorithm 1: *MultiSourceTrAdaBoost*

Input: Source training data $D_{S_1}, \cdots, D_{S_N}$, target training data D_T, and the maximum number of iterations M

Output: Target classifier function $\hat{f}_T : \mathcal{X} \to \mathcal{Y}$

Set $\alpha_S \doteq \frac{1}{2} \ln \left(1 + \sqrt{2 \ln \frac{n_S}{M}}\right)$, where $n_S \doteq \sum_k n_{S_k}$

Initialize the weight vector $(\mathbf{w}^{S_1}, \cdots, \mathbf{w}^{S_N}, \mathbf{w}^T)$, where $\mathbf{w}^{S_k} \doteq (w_1^{S_k}, \cdots, w_{n_{S_k}}^{S_k})$, and $\mathbf{w}^T \doteq (w_1^T, \cdots, w_{n_T}^T)$ to the desired distribution

for $t \leftarrow 1$ **to** M **do**

  Empty the set of candidate weak classifiers, $\mathcal{F} \leftarrow \emptyset$

  Normalize to 1 the weight vector $(\mathbf{w}^{S_1}, \cdots, \mathbf{w}^{S_N}, \mathbf{w}^T)$

  for $k \leftarrow 1$ **to** N **do**

    Find the candidate weak classifier $h_t^k : \mathcal{X} \to \mathcal{Y}$ that minimizes the classification error over the combined set $D_{S_k} \bigcup D_T$, weighted according to $(\mathbf{w}^{S_k}, \mathbf{w}^T)$

    Compute the error of h_t^k on D_T:

$$\epsilon_t^k \doteq \sum_j \frac{w_j^T [y_j^T \neq h_t^k(\mathbf{x}_j^T)]}{\sum_i w_i^T}$$

    $\mathcal{F} \leftarrow \mathcal{F} \cup (h_t^k, \epsilon_t^k)$

  Find the weak classifier $h_t : \mathcal{X} \to \mathcal{Y}$ such that

$$(h_t, \epsilon_t) \doteq \arg \min_{(h,\epsilon) \in \mathcal{F}} \epsilon$$

9 Set $\alpha_t \doteq \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$, where $\epsilon_t < 1/2$

10 Update the weight vector

$$w_i^{S_k} \leftarrow w_i^{S_k} e^{-\alpha_S |h_t(\mathbf{x}_i^{S_k}) - y_i^{S_k}|}$$
$$w_i^{T} \leftarrow w_i^{T} e^{\alpha_t |h_t(\mathbf{x}_i^{T}) - y_i^{T}|}$$

return $\hat{f}_T(\mathbf{x}) \doteq \text{sign}\left(\sum_t \alpha_t h_t(\mathbf{x})\right)$

这个算法在每次迭代中，都为 N 个（源域，目标域）组合训练出一个模型，再从 N 个模型中选取分类误差最小的模型，作为此轮迭代的模型，最终一共有 M 个模型。关于α_S的赋值，是一个定植，作者并没有给出相应解释。这里有个有意思的地方是权重更新，对于源域中的数据，如果分类错误，则认为这个数据跟目标域数据差异较大，因此给予他更小的权重，对于目标域中数据的更新则跟传统 adboost 算法一样，错误分类的数据给予更高的权重。

3.3 Boosting for transferring source tasks

这个算法分为两部分，第一部分对 N 个源域进行单独训练，迭代 M 次（注意，上一个算法在训练过程中，训练集为（源域，目标域），该算法第一部分训练集只包括源域），在迭代过程中，只要模型的正确率大于我们设定的域值，我们就将它放入我们此部分的模型集合 set 中。第二部分，对目标域的有标签数据集进行 M 次迭代，每次迭代过程遍历 set 中的模型，如果误差率大于 50%，即将此模型从 set 中取出，最终联合 set 中的多个模型对目标域进行分类。

具体算法如下：

Algorithm 2: Phase-I of *TaskTrAdaBoost*

Input: Source training data $D_{S_1}, \cdots, D_{S_N}$, the maximum number of iterations M, and the regularizing threshold γ

Output: Set of candidate weak classifiers $\mathcal{H}$

Empty the set of candidate weak classifiers, $\mathcal{H} \leftarrow \emptyset$

for $k \leftarrow 1$ **to** N **do**

  Initialize the weight vector $\mathbf{w}^{S_k} \doteq (w_1^{S_k}, \cdots, w_{n_{S_k}}^{S_k})$, to the desired distribution

  for $t \leftarrow 1$ **to** M **do**

    Normalize to 1 the weight vector $\mathbf{w}^{S_k}$

    Find the candidate weak classifier $h_t^k : \mathcal{X} \rightarrow \mathcal{Y}$ that minimizes the classification error over the set D_{S_k}, weighted according to $\mathbf{w}^{S_k}$

    Compute the error $\epsilon \leftarrow \sum_j w_j^{S_k} [y_j^{S_k} \neq h_t^k(\mathbf{x}_j^{S_k})]$

    $\alpha \leftarrow \frac{1}{2} \ln \frac{1-\epsilon}{\epsilon}$, where $\epsilon < 1/2$

    if $\alpha > \gamma$ **then**

      $\mathcal{H} \leftarrow \mathcal{H} \cup h_t^k$

    Update the weights $w_i^{S_k} \leftarrow w_i^{S_k} e^{-\alpha y_i^{S_k} h_t^k(\mathbf{x}_i^{S_k})}$

Algorithm 3: Phase-II of *TaskTrAdaBoost*

Input: Target training data D_T, the set of candidate weak classifiers $\mathcal{H}$, and the maximum number of iterations M

Output: Target classifier function $\hat{f}_T : \mathcal{X} \to \mathcal{Y}$

Initialize the weight vector $\mathbf{w}^T \doteq (w_1^T, \cdots, w_{n_T}^T)$, to the desired distribution

for $t \leftarrow 1$ **to** M **do**

  Normalize to 1 the weight vector $\mathbf{w}^T$

  Empty the current weak classifier set $\mathcal{F} \leftarrow \emptyset$

  foreach $h \in \mathcal{H}$ **do**

    Compute the error of h on D_T

$$\epsilon \leftarrow \sum_j w_j^T [y_j^T \neq h(\mathbf{x}_j^T)] \tag{1}$$

    if $\epsilon > 1/2$ **then**

      $h \leftarrow -h$

      Update ϵ via (1)

    $\mathcal{F} \leftarrow \mathcal{F} \cup (h, \epsilon)$

  Find the weak classifier $h_t : \mathcal{X} \to \mathcal{Y}$ such that

$$(h_t, \epsilon_t) \doteq \arg \min_{(h,\epsilon)\in\mathcal{F}} \epsilon$$

  $\mathcal{H} \leftarrow \mathcal{H} \setminus h_t$

  Set $\alpha_t \doteq \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$

  Update the weights $w_i^T \leftarrow w_i^T e^{-\alpha_t y_i^T h_t^k(\mathbf{x}_i^T)}$

return $\hat{f}_T(\mathbf{x}) \doteq \text{sign}\left(\sum_t \alpha_t h_t(\mathbf{x})\right)$

参考文献

[1] http://blog.csdn.net/gyqjn/article/details/45501185

[2] Tu W, Sun S. Cross-domain representation-learning framework with combination of class-separate and domain-merge objectives[C]//Proceedings of the 1st International Workshop on Cross Domain Knowledge Discovery in Web and Social Network Mining. ACM, 2012: 18-25.

[3] Yao Y, Doretto G. Boosting for transfer learning with multiple sources[C]//Computer vision and pattern recognition (CVPR), 2010 IEEE conference on. IEEE, 2010: 1855-1862.