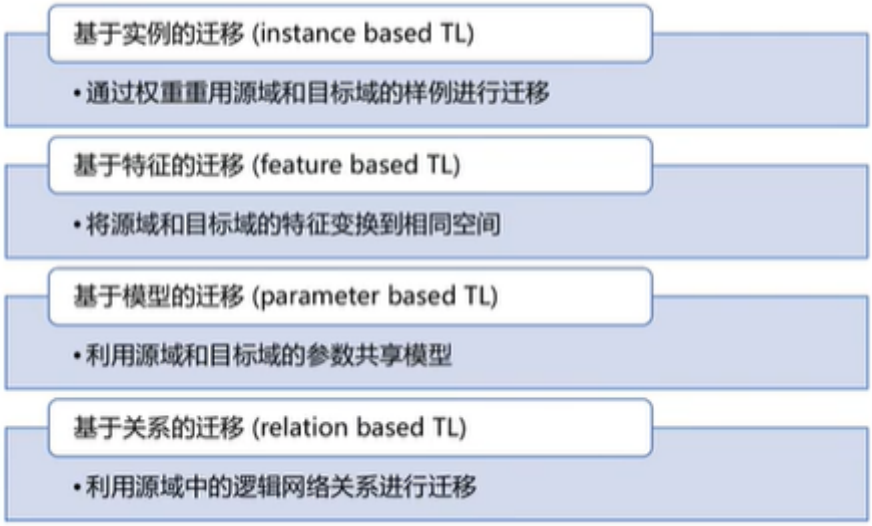
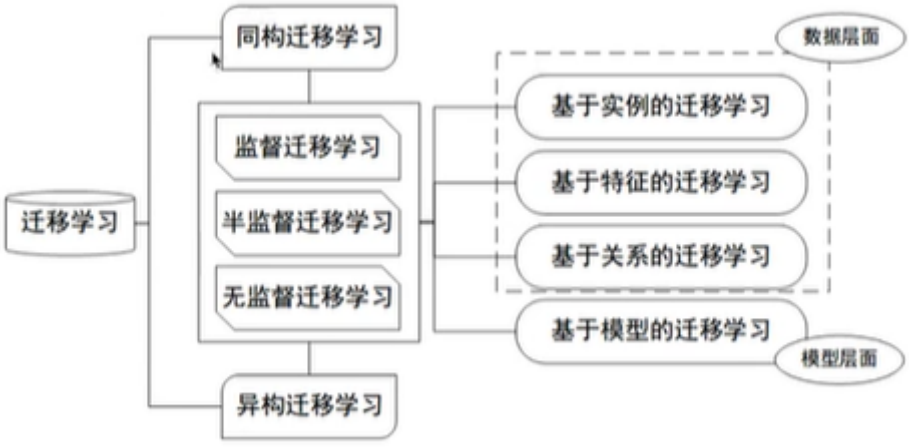
迁移学习

- 通过减小源域(辅助领域)到目标域的**分布差异**，进行**知识迁移**，从而实现数据标定。





形式化

形式化

- 条件：给定一个源域 $\mathcal{D}_S$ 和源域上的学习任务 $\mathcal{T}_S$，目标域 $\mathcal{D}_T$ 和目标域上的学习任务 $\mathcal{T}_T$
- 目标：利用 $\mathcal{D}_S$ 和 $\mathcal{T}_S$ 学习在目标域上的预测函数 $f(\cdot)$。
- 限制条件：$\mathcal{D}_S \neq \mathcal{D}_T$ 或 $\mathcal{T}_S \neq \mathcal{T}_T$

领域自适应问题

描述

Domain Adaptation (DA); cross-domain learning; 同构迁移学习

问题定义：有标签的源域和无标签的目标域共享相同的特征和类别，但是特征分布不同，如何利用源域标定目标域

$$\mathcal{D}_S \neq \mathcal{D}_T : P_S(X) \neq P_T(X)$$

特征的分布不一样，维度一样

按照目标域有无标签

- 目标域全部有标签：supervised DA
- 目标域有一些标签：semi-supervised DA
- 目标域全没有标签：unsupervised DA

解决

基本假设

- 数据分布角度：源域和目标域的**概率分布相似**
  - **最小化**概率分布距离
- 特征选择角度：源域和目标域共享着**某些特征**
  - **选择**出这部分公共特征
- 特征变换角度：源域和目标域共享**某些子空间**
  - 把两个域**变换**到相同的子空间

边缘分布适配 (Marginal distribution adaptation)

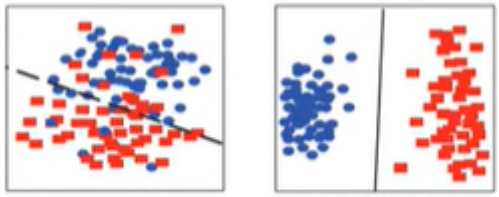
- 假设：$P(\mathbf{X}_s) \neq P(\mathbf{X}_t)$

条件分布适配 (Conditional distribution adaptation)

- 假设：$P(y_s|\mathbf{X}_s) \neq P(y_t|\mathbf{X}_t)$

联合分布适配 (Joint distribution adaptation)

- 假设：$P(\mathbf{X}_s, y_s) \neq P(\mathbf{X}_t, y_t)$



目标域数据(1)
优先考虑边缘分布

目标域数据(2)
优先考虑条件分布

边缘分布反映整体分布

条件分布表示细致的形状

边缘分布

迁移成分分析 (Transfer Component Analysis, TCA) [Pan, TNN-11]

- 优化目标：

$$\min_{\varphi} \quad \text{Dist}(\varphi(\mathbf{X}_S), \varphi(\mathbf{X}_T)) + \lambda\Omega(\varphi)$$
$$\text{s.t.} \quad \text{constraints on } \varphi(\mathbf{X}_S) \text{ and } \varphi(\mathbf{X}_T)$$

- 最大均值差异(Maximum Mean Discrepancy, MMD)

$$\text{Dist}(P(X_S), P(X_T)) = \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \Phi(x_{S_i}) - \frac{1}{n_T} \sum_{j=1}^{n_T} \Phi(x_{T_j}) \right\|_{\mathcal{H}}$$

令距离最小就完事了

这个“距离”是最大均值差异

扩展：

迁移成分分析 (TCA)方法的一些扩展

- Adapting Component Analysis (ACA) [Dorri, ICDM-12] $maximize \ \frac{tr(HK_XHL_\Phi)}{tr(HL_MHL_\Phi)}$
  - 最小化MMD，同时维持迁移过程中目标域的结构
- Domain Transfer Multiple Kernel Learning (DTMKL) [Duan, PAMI-12]
  - 多核MMD
- Deep Domain Confusion (DDC) [Tzeng, arXiv-14]
  - 把MMD加入到神经网络中
- Deep Adaptation Networks (DAN) [Long, ICML-15]
  - 把MKK-MMD加入到神经网络中
- Distribution-Matching Embedding (DME) [Baktashmotlagh, JMLR-16]
  - 先计算变换矩阵，再进行映射
- Central Moment Discrepancy (CMD) [Zellinger, ICLR-17]
  - 不只是一阶的MMD，推广到了k阶

$$k = \sum_{m=1}^{M} d_m k_m$$

条件分布适配

比较少

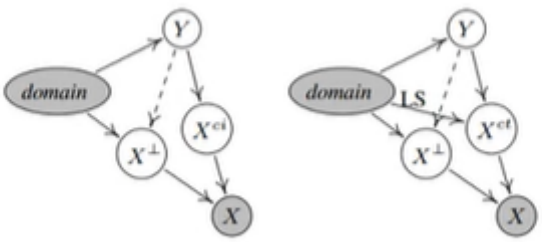
Domain Adaptation of Conditional Probability Models via Feature Subsetting [Satpal, PKDD-07]

- 条件随机场+分布适配
- 优化目标：

$$\text{argmax}_{\mathbf{w},S} \sum_{(\mathbf{x},\mathbf{y}) \in D} \sum_{k \in S} w_k f_k(\mathbf{x}, \mathbf{y}) - \log z_{\mathbf{w}}(\mathbf{x})$$
$$\text{such that} \quad \text{dist}(\mathcal{D}, \mathcal{D}' | S, D, D') \leq \epsilon.$$

Conditional Transferrable Components (CTC) [Gong, ICML-15]

- 定义条件转移成分，对其进行建模



联合分布适配

两个都适配

问题：怎么获得条件分布？啥：

联合分布适配 (Joint Distribution Adaptation, JDA) [Long, ICCV-13]

- 直接继承于TCA，但是加入了条件分布适配
- 优化目标：

$$D(\mathcal{D}_s, \mathcal{D}_t) \approx D(P(\mathbf{x}_s), P(\mathbf{x}_t)) + D(P(y_s|\mathbf{x}_s), P(y_t|\mathbf{x}_t))$$

- 问题：如何获得估计条件分布？
  - 充分统计量：用类条件概率近似条件概率
  - 用一个弱分类器生成目标域的初始软标签
- 最终优化形式

$$\min_{\mathbf{A}^{\mathrm{T}}\mathbf{K}\mathbf{H}\mathbf{K}^{\mathrm{T}}\mathbf{A}=\mathbf{I}} \sum_{c=0}^{C} \mathrm{tr}\left(\mathbf{A}^{\mathrm{T}}\mathbf{K}\mathbf{M}_c\mathbf{K}^{\mathrm{T}}\mathbf{A}\right) + \lambda\|\mathbf{A}\|_F^2$$

- **联合分布适配的结果普遍优于比单独适配边缘或条件分布**

扩展：

联合分布适配 (2)

- 联合分布适配(JDA)方法的一些扩展
  - Adaptation Regularization (ARTL) [Long, TKDE-14]
    - 分类器学习+联合分布适配
  - Visual Domain Adaptation (VDA) [Tahmoresnezhad, KIS-17]
    - 加入类内距、类间距
  - Joint Geometrical and Statistical Alignment (JGSA) [Zhang, CVPR-17]
    - 加入类内距、类间距、标签适配
  - [Hsu, TIP-16]：加入结构不变性控制
  - [Hsu, AVSS-15]：目标域选择
  - Joint Adaptation Networks (JAN) [Long, ICML-17]
    - 提出JMMD度量，在深度网络中进行联合分布适配

JGSA是当时公开数据中最好的

联合分布适配的问题：

平衡分布适配 (Balanced Distribution Adaptation, BDA) [Wang, ICDM-2017]

- 仅仅适配条件分布和边缘分布就够了吗？
  - 联合分布适配的问题：两种分布同等重要
  - 真实环境：两种分布**不一定**同等重要
- 加入**平衡因子**动态衡量两种分布的重要性

$$D(\mathcal{D}_s, \mathcal{D}_t) \approx (1-\mu) D(P(\mathbf{x}_s), P(\mathbf{x}_t)) + \mu D(P(y_s|\mathbf{x}_s), P(y_t|\mathbf{x}_t)) \qquad \mu \in [0,1]$$

平衡因子

  - 当 $\mu \to 0$，表示边缘分布更占优，应该优先适配
  - 当 $\mu \to 1$，表示条件分布更占优，应该优先适配
- 最终表示形式

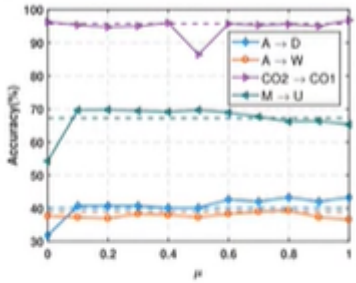
$$\min \ \mathrm{tr}\left(\mathbf{A}^\top \mathbf{X}\left((1-\mu)\mathbf{M}_0 + \mu \sum_{c=1}^{C} \mathbf{M}_c\right)\mathbf{X}^\top \mathbf{A}\right) + \lambda \|\mathbf{A}\|_F^2$$
$$\text{s.t. } \mathbf{A}^\top \mathbf{X}\mathbf{H}\mathbf{X}^\top \mathbf{A} = \mathbf{I}, \quad 0 \le \mu \le 1$$

两个方面的权重

μ是超参数？可训练参数？

平衡分布适配 (BDA)：平衡因子的重要性

- 对于不同的任务，边缘分布和条件分布并不是同等重要，因此，BDA方法可以**有效衡量**这两个分布的权重，从而达到最好的结果

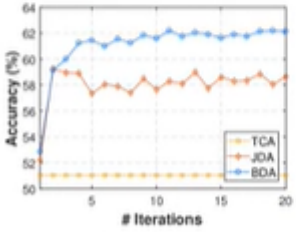


平衡分布适配 (BDA)：平衡因子的求解与估计

- 目前尚无精确的估计方法; 我们采用A-distance来进行估计
  - 求解源域和目标域整体的A-distance
  - 对目标域聚类，计算源域和目标域每个类的A-distance
  - 计算上述两个距离的比值，则为平衡因子

总结:

- **概率分布适配：总结**
  - 方法
    - 基础：大多数方法基于MMD距离进行优化求解
    - 分别进行边缘 / 条件 / 联合概率适配
    - **效果**：平衡 (BDA) > 联合 (JDA) > 边缘 (TCA) > 条件
  - 使用
    - 数据整体差异性大 (相似度较低)，边缘分布更重要
    - 数据整体差异性小 (协方差漂移)，条件分布更重要
  - 最新成果
    - **深度学习+分布适配**往往有更好的效果 (DDC、DAN、JAN)



BDA、JDA、TCA精度比较

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
AlexNet (Krizhevsky et al., 2012)	61.6±0.5	95.4±0.3	99.0±0.2	63.8±0.5	51.1±0.6	49.8±0.4	70.1
TCA (Pan et al., 2011)	61.0±0.0	93.2±0.0	95.2±0.0	60.8±0.0	51.6±0.0	50.9±0.0	68.8
GFK (Gong et al., 2012)	60.4±0.0	95.6±0.0	95.0±0.0	60.6±0.0	52.4±0.0	48.1±0.0	68.7
DDC (Tzeng et al., 2014)	61.8±0.4	95.0±0.5	98.5±0.4	64.4±0.3	52.1±0.6	52.2±0.4	70.6
DAN (Long et al., 2015)	68.5±0.5	96.0±0.3	99.0±0.3	67.0±0.4	54.0±0.5	53.1±0.5	72.9
RTN (Long et al., 2016)	73.3±0.3	**96.8**±0.2	**99.6**±0.1	71.0±0.2	50.5±0.3	51.0±0.1	73.7
RevGrad (Ganin & Lempitsky, 2015)	73.0±0.5	96.4±0.3	99.2±0.3	72.3±0.3	53.4±0.4	51.2±0.5	74.3
JAN (ours)	74.9±0.3	96.6±0.2	99.5±0.2	71.8±0.2	**58.3**±0.3	55.0±0.4	76.0
JAN-A (ours)	**75.2**±0.4	96.6±0.2	**99.6**±0.1	**72.8**±0.3	57.5±0.2	**56.3**±0.2	**76.3**

DDC、DAN、JAN与其他方法结果比较

深度学习+ 相对要好

特征选择法

- 从源域和目标域中选择提取共享的特征，建立统一模型
- Structural Correspondence Learning (STL) [Blitzer, ECML-06]
  - 寻找Pivot feature，将源域和目标域进行对齐

SCL

找出轴特征并进行对齐

扩展：

特征选择法其他扩展

- Joint feature selection and subspace learning [Gu, IJCAI-11]
  - 特征选择/变换+子空间学习
  - 优化目标：$\min_{\mathbf{A}} \ \|\mathbf{A}\|_{2,1} + \mu \mathrm{tr}(\mathbf{A}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{A})$ s.t. $\mathbf{A}^T\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{A} = \mathbf{I}$,
- Transfer Joint Matching (TJM) [Long, CVPR-14]
  - MMD分布适配+源域样本选择
  - 优化目标：$\min_{\mathbf{A}^T\mathbf{K}\mathbf{H}\mathbf{K}^T\mathbf{A}=\mathbf{I}} \mathrm{tr}\left(\mathbf{A}^T\mathbf{K}\mathbf{M}\mathbf{K}^T\mathbf{A}\right)+\lambda\left(\|\mathbf{A}_s\|_{2,1}+\|\mathbf{A}_t\|_F^2\right)$
- Feature Selection and Structure Preservation (FSSL) [Li, IJCAI-16]
  - 特征选择+信息不变性
  - 优化目标：

$$\min_{\mathbf{P},\mathbf{Z},\mathbf{E}} \|\mathbf{P}\|_{2,1} + \frac{\lambda}{2}\mathrm{tr}(\mathbf{P}^\top\mathbf{X}\mathbf{L}\mathbf{X}^\top\mathbf{P}) + \frac{\beta}{2}\|\mathbf{Z}\|_{\mathrm{F}}^2 + \gamma\|\mathbf{E}\|_1$$
$$s.t.\ \mathbf{P}^\top\mathbf{X} = \mathbf{P}^\top\mathbf{X}_s\mathbf{Z} + \mathbf{E}, \mathbf{P}^\top\mathbf{X}\mathbf{D}\mathbf{X}^\top\mathbf{P} = \mathbf{I},$$

加了很多项

或者把分类器也

特征选择法：总结

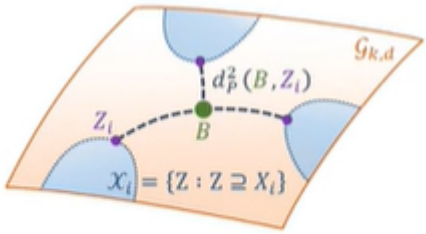
- 从源域和目标域中选择提取共享的特征，建立统一模型
- 通常与分布适配进行结合
- 选择特征通常利用稀疏矩阵

这部分研究一般和别的方法结合

子空间学习法

- 将源域和目标域变换到相同的子空间，然后建立统一的模型
  - 统计特征变换 (Statistical Feature Transformation)
    - 将源域和目标域的一些统计特征进行变换对齐
  - 流形学习 (Manifold Learning)
    - 在流形空间中进行子空间变换

统计特征变换



流形学习

⚠流形

子空间对齐法 (Subspace Alignment, SA) [Fernando, ICCV-13]

- 直接寻求一个线性变换，把source变换到target空间中
- 优化目标： $F(M) = ||X_S M - X_T||_F^2$

$$M^* = argmin_M(F(M))$$

- 直接获得线性变换的闭式解：

$$F(M) = ||X_S' X_S M - X_S' X_T||_F^2 = ||M - X_S' X_T||_F^2.$$

子空间分布对齐法 (Subspace Distribution Alignment, SDA) [Sun, BMVC-15]

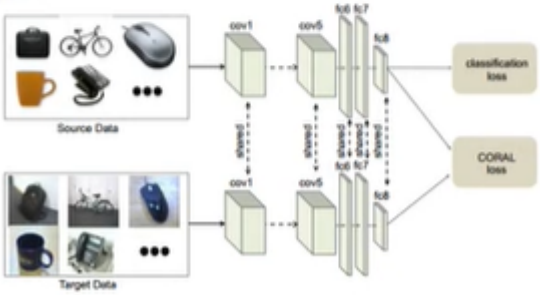
- 子空间对齐+概率分布适配

$$M_s = S_s T_{TS} A_{TS} S_t^T = S_s (S_s^T S_t)(E_s^{\frac{-1}{2}} E_t^{\frac{1}{2}}) S_t^T$$

空间对齐法：方法简洁，计算高效

- 关联对齐法 (CORrelation Alignment, CORAL) [Sun, AAAI-15]
  - 最小化源域和目标域的二阶统计特征
  - 优化目标：
    $$\min_A \|C_{\hat{S}} - C_T\|_F^2 = \min_A \|A^\top C_S A - C_T\|_F^2$$
  - 形式简单，求解高效
- 深度关联对齐 (Deep-CORAL) [Sun, ECCV-16]
  - 在深度网络中加入CORAL
  - CORAL loss: $\ell_{CORAL} = \frac{1}{4d^2}\|C_S - C_T\|_F^2$

$$C_S = \frac{1}{n_S - 1}(D_S^\top D_S - \frac{1}{n_S}(\mathbf{1}^\top D_S)^\top(\mathbf{1}^\top D_S))$$
$$C_T = \frac{1}{n_T - 1}(D_T^\top D_T - \frac{1}{n_T}(\mathbf{1}^\top D_T)^\top(\mathbf{1}^\top D_T))$$



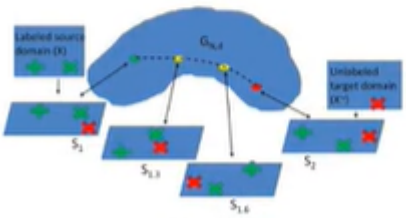
流形

在空间上把domain抽象成两个点，画一条最短距离（测地线）

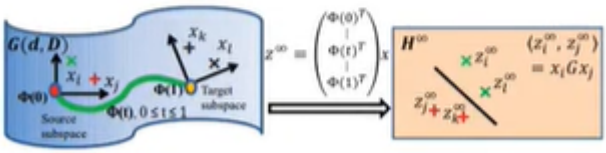
取有限的点或无穷的点（积分

- 采样测地线流方法 (Sample Geodesic Flow, SGF) [Gopalan, ICCV-11]
  - 把领域自适应的问题看成一个增量式“行走”问题
  - 从源域走到目标域就完成了一个自适应过程
  - 在流形空间中采样有限个点，构建一个测地线流
- 测地线流式核方法 (Geodesic Flow Kernel, GFK) [Gong, CVPR-12]
  - 继承了SGF方法，采样无穷个点
  - 转化成Grassmann流形中的核学习，构建了GFK
  - 优化目标：

$$\langle z_i^\infty, z_j^\infty \rangle = \int_0^1 (\Phi(t)^\top x_i)^\top (\Phi(t)^\top x_j)\, dt = x_i^\top G x_j$$



SGF方法



GFK方法

别的方法：

- 域不变映射 (Domain-Invariant Projection, DIP) [Baktashmotlagh, CVPR-13]
  - 直接度量分布距离是不好的：原始空间特征扭曲
  - 仅作流形子空间学习：无法刻画分布距离
  - 解决方案：流形映射+分布度量
- 统计流形法 (Statistical Manifold) [Baktashmotlagh, CVPR-14]
  - 在统计流形(黎曼流形)上进行分布度量
  - 用Fisher-Rao distance (Hellinger distance)进行度量

$$\min_{\alpha} \frac{1}{\sum_{i=1}^{n_s}\alpha_i}\sum_{i=1}^{n_s}\alpha_i\left(\sqrt{\hat{T}(x_i^s)}-\sqrt{1-\hat{T}(x_i^s)}\right)^2 + \frac{1}{n_t}\sum_{i=1}^{n_t}\left(\sqrt{\hat{T}(x_i^t)}-\sqrt{1-\hat{T}(x_i^t)}\right)^2$$

总结

子空间学习法：总结

- 主要包括统计特征对齐和流形学习方法两大类
- 和分布适配结合效果更好
- 趋势：与神经网络结合

最新（2017）研究成果

与深度学习进行结合

- Deep Adaptation Networks (DAN) [Long, ICML-15]
  - 深度网络+MMD距离最小化
- Joint Adaptation Networks (JAN) [Long, ICML-17]
  - 深度网络+联合分布距离最小化
- Simultaneous feature and task transfer [Tzeng, ICCV-15]
  - 特征和任务同时进行迁移
- Deep Hashing Network (DHN) [CVPR-17]
  - 在深度网络中同时学习域适应和深度Hash特征
- Label Efficient Learning of Transferable Representations across Domains and Tasks [Luo, NIPS-17]
  - 在深度网络中进行任务迁移

- 与对抗学习进行结合
  - Domain-adversarial neural network [Ganin, JMLR-16]
    - 深度网络中加入对抗
  - Adversarial Discriminative Domain Adaptation (ADDA) [Tzeng, arXiv-17]
    - 对抗+判别
- 开放世界领域自适应
  - Open set domain adaptation [Busto, ICCV-17]
    - 当源域和目标域只共享一部分类别时如何迁移？
- 与张量 (Tensor)表示相结合
  - When DA Meets tensor representation [Lu, ICCV-17]
    - 用tensor的思想来做领域自适应
- 与增量学习结合
  - Learning to Transfer (L2T) [Wei, arXiv-17]
    - 提取已有的迁移学习经验，应用于新任务

没了