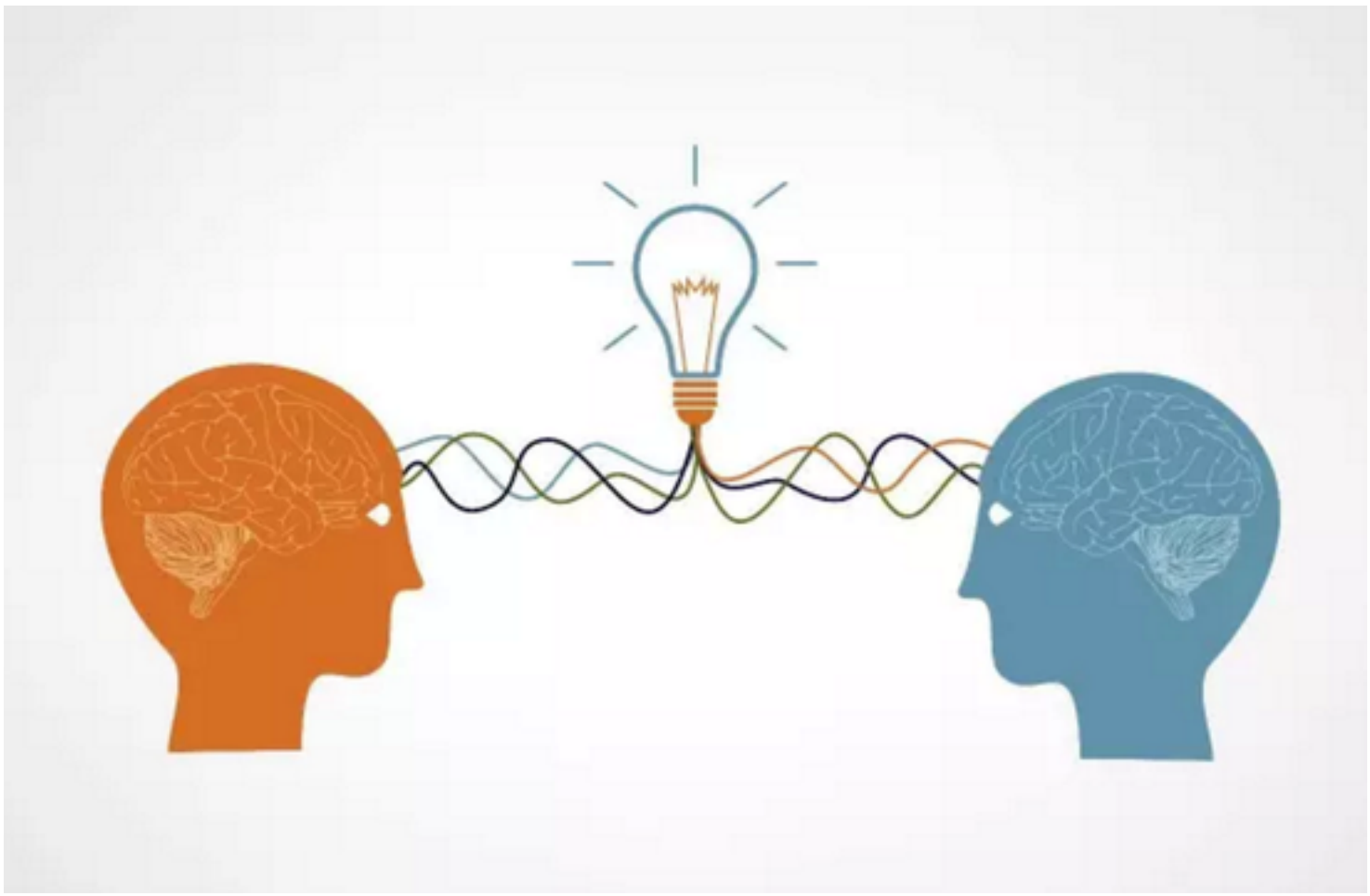
迁移学习：MMD和再生核希尔伯特空间

4 个月前



写在前面：

以后我会把论文导读的部分写到知乎，某些源码实现的部分依然放在CSDN，上面还有几篇文章就不一一搬运了，有需要的同学可以参考：我的CSDN。本人非专业大神，如有不当的地方或者有疑问，不用私信请在评论区留言，我会及时修改。

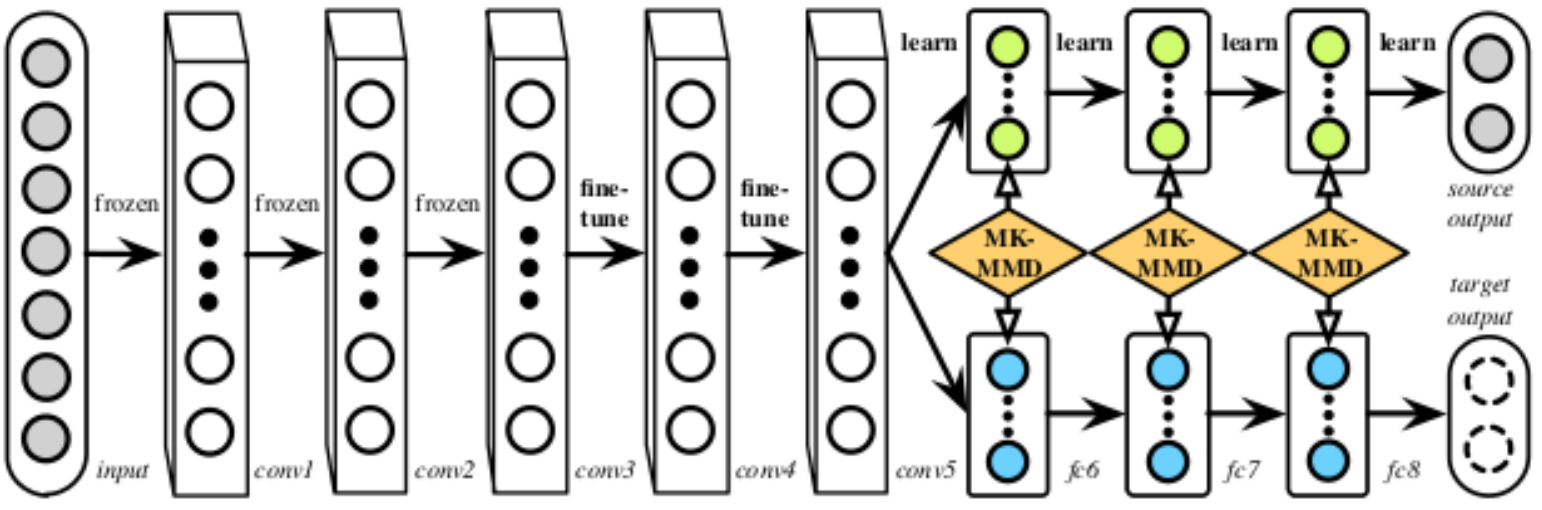
内容还是以迁移学习为主，当然其他方面的论文也会涉及

一.引入

今天主要说说迁移学习当中source domain—>target domain的差异性度量，传统学习方法有一个假设：training sample和test sample都是从同一个分布抽样得到，即训练集和测试集是iid.（独立同分布）的，这个假设使得离线的学习方法得以运行。在迁移学习环境下training sample和test sample分别取样自分布p和q，两个样本不同但相关，现在我们要通过测试样本改善模型性能。求解这个问题的思路比较多，这里只列一个目前流行的

思路：

利用深度神经网络的特征变换能力，来做特征空间的transformation，直到变换后的特征分布相匹配，这个过程可以是source domain一直变换直到匹配target domain，也可以是source domain和target domain一起变换直到匹配（例如下图）



引入了那么多，终于扯到匹配了，而我们重点讲的也是如何匹配的问题。匹配的意义就是度量两个分布之间的差异，有人第一时间会想到Kullback-Leibler Divergence，事实上还有很多度量的方法，详见wiki：[Statistical distance]

Examples [edit]

Some important statistical distances include the following:

- f-divergence: includes
  - Kullback–Leibler divergence
  - Hellinger distance
  - Total variation distance (sometimes just called "the" statistical distance)
- Rényi's divergence
- Jensen–Shannon divergence
- Lévy–Prokhorov metric
- Bhattacharyya distance
- Wasserstein metric: also known as the Kantorovich metric, or earth mover's distance
- The Kolmogorov–Smirnov statistic represents a distance between two probability distributions defined on a single real variable
- The **maximum mean discrepancy** which is defined in terms of the kernel embedding of distributions

上面列了一大堆方法，其中f-divergence包含一类用一个function：$Df(P||Q)$来度量的方法。类似KL divergence的方法虽然经典，但其实并不适用于在线学习模型，mini batch所包含的样本毕竟是有限的，在少量样本下KL divergence并不准确。直到2006年，陆续有论文开始使用一种Maximum Mean Discrepancy的度量方法。

二. Maximum Mean Discrepancy

$$\mathrm{MMD}\left[\mathcal{F}, p, q\right] := \sup_{f \in \mathcal{F}} \left(\mathbf{E}_p[f(x)] - \mathbf{E}_q[f(y)]\right)$$

$$\mathrm{MMD}\left[\mathcal{F}, X, Y\right] := \sup_{f \in \mathcal{F}} \left(\frac{1}{m}\sum_{i=1}^{m} f(x_i) - \frac{1}{n}\sum_{i=1}^{n} f(y_i)\right)$$

MMD的书面表达就是source sample：x和target sample：y经过函数

随机投影后，期望值

和

的差值上确界。其中F指将特征空间映射到实数集R所有函数f的集合。

我们可以这样看这个公式：如果p=q，很显然x与y在特征空间下的分布是一致的，我们无论将x与y做怎样的随机投影，期望都会相同，于是在p=q情况下，MMD=0;但如果pq，而F又是足够rich的，那MMD的值就会永远取不到0而且会和随机投影有关，具体等于多少取决于让其差异最大的那个投影。不过当F太过rich时，MMD很容易取到无穷，所以我们需要对F有一个约束。

接下来就是如何计算了。我们需要构建一个函数的空间，并假设是空间当中的一个点，那么F又可以表示成该空间中的一块小区域（因为F是有约束的，所以它不能取遍整个空间）。为了计算MMD构建起来的这个空间，就是RKHS：reproducing kernel Hilbert space。经证明当F是RKHS中的单位球时，是最佳的。（单位球就是到原点模1的空间，在二维坐标系也就是单位圆）

三. Reproducing Kernel Hilbert Space

Reproducing Kernel Hilbert Space是一个比较理论化的东西，无奈被数学包装得比较华丽的论文，才能称得上好的论文，所以近几年迁移学习方向的论文没有一篇是不提到RKHS的。

对函数空间的要求是完备的内积空间，这样的空间又称Hilbert Space（这种理论化的东西，我也不太能理解，所以就一笔带过了）。那为什么前面要加上Reproducing Kernel呢？因为它有一个性质，它可以用空间内的点积表示

的映射，也即

$$f(x) = \langle f, \phi(x) \rangle_{\mathcal{H}}$$

接下来就来说说这个性质有什么用，这里的表示一个

的映射，

取决于x的值，你会发现这个性质将函数

的值拆分成了两部分，函数以及自变量x，它让我们更好的将抽离出来做最大化（上面已经说过）。然后在进一步，我们将分布引入，用

来替代

，你会发现，

表示成了和

的点积！事实上这个

是有名字的，他叫Kernel embedding of distributions，详见wiki：[Kernel embedding of distributions]

用表示

，表示

，有如下的推导

$$\begin{aligned}
\mathrm{MMD}[\mathcal{F}, p, q] &= \sup_{\|f\|_{\mathcal{H}} \leq 1} \mathbf{E}_p\left[f(x)\right] - \mathbf{E}_q\left[f(y)\right] \\
&= \sup_{\|f\|_{\mathcal{H}} \leq 1} \mathbf{E}_p\left[\langle \phi(x), f \rangle_{\mathcal{H}}\right] - \mathbf{E}_q\left[\langle \phi(y), f \rangle_{\mathcal{H}}\right] \\
&= \sup_{\|f\|_{\mathcal{H}} \leq 1} \langle \mu_p - \mu_q, f \rangle_{\mathcal{H}} = \|\mu_p - \mu_q\|_{\mathcal{H}} .
\end{aligned}$$

现在MMD已经可以用RKHS中两个点的距离表示了，把上式两边平方

$$\begin{aligned}\mathrm{MMD}^2\left[\mathcal{F}, p, q\right] &:= \left\langle \mu_p - \mu_q, \mu_p - \mu_q \right\rangle_{\mathcal{H}} \\ &= \left\langle \mu_p, \mu_p \right\rangle_{\mathcal{H}} + \left\langle \mu_q, \mu_q \right\rangle_{\mathcal{H}} - 2\left\langle \mu_p, \mu_q \right\rangle_{\mathcal{H}} \\ &= \mathbf{E}_p \left\langle \phi(x), \phi(x') \right\rangle_{\mathcal{H}} + \mathbf{E}_q \left\langle \phi(y), \phi(y') \right\rangle_{\mathcal{H}} \\ &\quad - 2\mathbf{E}_{p,q} \left\langle \phi(x), \phi(y) \right\rangle_{\mathcal{H}},\end{aligned}$$

式中的点积可以用核函数k（x，x‘）来计算，因为RKHS常常是高维甚至是无限维的空间，对应的核一般选择表示无穷维的高斯核

$$k(x, x') = \exp(-\|x - x'\|^2/(2\,\sigma^2)).$$

放在实际当中，一个batch的MMD约束计算如下：

$$\begin{aligned}\mathrm{MMD}^2[\mathcal{F}, X, Y] = &\frac{1}{m(m-1)} \sum_{i \neq j}^{m} k(x_i, x_j) \\ &+ \frac{1}{n(n-1)} \sum_{i \neq j}^{n} k(y_i, y_j) - \frac{2}{mn} \sum_{i,j=1}^{m,n} k(x_i, y_j).\end{aligned}$$

现在再来解释为什么是“Reproducing Kernel”就比较容易了，我们使用这个空间，就是为了构造空间当中表示分布p和q的Kernel embedding of distributions，然后用kernel function计算这两个点之间的距离。

最后提醒一下，MMD的计算复杂度是

四.论文

我贴几篇使用MMD比较早的，被引用多的论文，里面有详细的介绍，方便借鉴和学习。

1.Borgwardt, Karsten M., Gretton, Arthur, Rasch, Malte J.,Kriegel, Hans-Peter, Schölkopf, Bernhard, and Smola, Alexander J. Integrating structured biological data by kernel maximum mean discrepancy. In ISMB, pp. 49–57, 2006.

2.Huang, Jiayuan, Smola, Alexander J., Gretton, Arthur, Borgwardt, Karsten M., and Schölkopf, Bernhard. Correcting sample selection bias by unlabeled data. In NIPS, pp. 601–608, 2006.

3.N. Quadrianto, J. Petterson, and A. J. Smola. Distribution matching for transduction. In Proceedings of NIPS, 2009.

4.A. Gretton, K. Borgwardt, M. Rasch, B. Scḧolkopf, and A. Smola. A kernel method for the two sample problem. Technical Report 157, MPI for Biological Cybernetics, 2008

另外，写的过程中意外的发现了一篇好文，把链接附上，优秀的文章还是需要去挖掘的：

MMD： maximum mean discrepancy