基于模型的迁移学习

田鸿龙

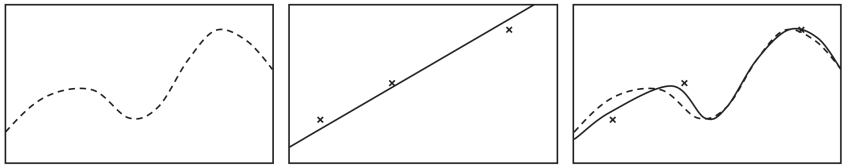
LAMDA, Nanjing University

November 25, 2020

目录

基于模型的迁移学习

核心思想：在模型层次上原任务和目标任务共享部分通用知识。



(a) Source model (the dash line)

(b) Target model (the solid line) only with limited target data (the crosses)

(c) The target model (the solid line) trans-ferred with the source model (the dash line) as a prior.

基于模型的迁移学习和多任务学习

相同点

- 都是通过模型共享知识
- 往往可以通过调整多任务学习算法得到对应的迁移学习算法

不同点

- 多任务学习是同时训练的，而迁移学习具有顺序性
- 多任务学习要求算法在多个任务上表现良好，而迁移学习只关注**目标任务**
- 多任务学习无法达到单个任务的最优，而迁移学习要求在**目标任务**上达到最优

分类

- 基于共享模型成分的迁移学习
- 基于正则化的迁移学习

目录

目录

统计学习中的回归问题定义

- 对于每个输入 $\mathbf{x}$，对应一个输出 y，通过数据集 $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^T$，$\mathbf{y} = [y_1, y_2, \ldots, y_N]^T$ 学习一个映射函数。
- y 往往是含有噪声的，也就是说我们的数据集并不完全准确，定义潜变量 z 为 $\mathbf{x}$ 对应的输出，我们的目标其实是学习映射函数 $f: \mathbf{x} \to z$
- 根据数据集 $\mathbf{X}$ 和 $\mathbf{y}$ 得不到 $f: \mathbf{x} \to z$ 的有效信息，我们假定 $p(y \mid z) = N(z, \beta^{-1})$
- 其中 β 表示精度，它的倒数 β^{-1} 表示方差，β 可以被学习得到。

高斯过程回归

回归就是给一堆已知的 $\mathbf{x}$ 和 y，然后当拿出一个新 $\mathbf{x}$ 的时候，能够预测出对应的 y。
高斯过程是一种贝叶斯方法，能够预测出新的 y 的分布来。
高斯过程的出发点就是，如果两个 $\mathbf{x}$ 比较近，那么对应的 y 一定是比较接近。给出的新 $\mathbf{x}$，就看这个新的 $\mathbf{x}$ 与之前给出的一堆 $\mathbf{x}$ 有多近，从而知道新的 y 与之前的一堆 y 有多近，从而预测新 y 的值。
基于上面的出发点，高斯过程回归要解决的问题如下：

- 如何度量 $\mathbf{x}$ 之间有多近
- $\mathbf{x}$ 的“近”又如何反应到 y 有多近

高斯过程回归 (cont.)

高斯先验

$$p(\mathbf{z} \mid \mathbf{X}, \theta) = N(\mathbf{0}, \mathbf{K})$$

- 高斯先验只和数据集中的 $\mathbf{X}$ 有关
- $\mathbf{K}$ 是 $\mathbf{X}$ 的相关系数（协方差矩阵），在这里相关系数用 Kernel 表示（假设有 N 个样本，则 Kernel 是 $N \times N$ 的矩阵，和传统意义上的协方差矩阵不同，见下页 Slide）
- 可以理解为，在算法没有“看到” y 时，根据 $\mathbf{X}$ 对 y 的相关性的假设
- 因为算法没有“看到” y，高斯分布的均值设为 0

高斯过程回归 (cont.)

Kernel

$$K(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\ell^2}\right)$$

- 上述（书中）使用的 Kernel 是高斯核，实际上不限于此
- Kernel 用来表示两个向量 $\mathbf{x}$ 和 $\mathbf{x}'$ 的某些关系，高斯核往往用来描述两个向量有多“近”
- Kernel 实际上可以看作向量 $\mathbf{x}$ 在某个映射下的协方差矩阵

高斯过程回归 (cont.)

$$p(\mathbf{y}) = \int p(\mathbf{y} \mid \mathbf{z}) p(\mathbf{z}) \mathrm{d}\mathbf{z} = N(\mathbf{y} \mid \mathbf{0}, \mathbf{C})$$

$$C(\mathbf{x}_n, \mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m) + \beta^{-1} \delta_{nm}$$

用 $\mathbf{y_N}$ 表示原来的数据集（训练集），用 $\mathbf{y_{N+1}}$ 表示训练集和测试集的并集（不是一般性，假设测试集只有一个样本）

$$p(y_{N+1}) = \frac{p(\mathbf{y}_{N+1})}{p(\mathbf{y}_N)}$$

高斯过程回归 (cont.)

$$p\left(\mathbf{y}_{N+1}\right)=\mathcal{N}\left(\mathbf{y}_{N+1} \mid \mathbf{0}, \mathbf{C}_{N+1}\right)$$

其中

$$\mathbf{C}_{N+1}=\begin{pmatrix} \mathbf{C}_N & \mathbf{k} \\ \mathbf{k}^{\mathrm{T}} & c \end{pmatrix}$$

$\mathbf{k}$ has elements $k(\mathbf{x}_n, \mathbf{x}_{N+1})$ and $n=1, \cdots, N$, $c=k\left(\mathbf{x}_{N+1}, \mathbf{x}_{N+1}\right)+\beta^{-1}$

最后

$$p(y_{N+1} \mid \mathbf{y}_N)=N(m\left(\mathbf{x}_{N+1}\right), \sigma^2\left(\mathbf{x}_{N+1}\right))$$

其中

$$m\left(\mathbf{x}_{N+1}\right)=\mathbf{k}^{\mathrm{T}} \mathbf{C}_N^{-1} \mathbf{t}$$
$$\sigma^2\left(\mathbf{x}_{N+1}\right)=c-\mathbf{k}^{\mathrm{T}} \mathbf{C}_N^{-1} \mathbf{k}$$

多任务下的高斯过程

设多个任务对应的任务集为 $\{(\mathbf{X}_m, \mathbf{y}_m)\}$，联合概率分布 $\mathbf{y} = \left(\mathbf{y}_1^T, \ldots, \mathbf{y}_M^T\right)^T$ 的概率为

$$p(\mathbf{y} \mid \mathbf{X}, \theta) = \prod_{m=1}^{M} p\left(\mathbf{y}_m \mid \mathbf{X}_m, \theta\right)$$

参考文献