

Transfer Learning with Active Queries from Source Domain

Optimizing Without Relaxation

优化式修改

$$\min MMD(\hat{S}, T) + MMD(\hat{S}_L \cup \hat{Q} \cup T_L, T_U \cup \hat{S}_U) + \lambda \alpha \beta |g_S u|$$

问题：选择的size越大，uncertainty项（到超平面距离）的累加和就越多，与最小化函数的目标相矛盾

优化式修改

$$\min MMD(\hat{S}, T) + MMD(\hat{S}_L \cup \hat{Q} \cup T_L, T_U \cup \hat{S}_U) + \lambda \alpha \beta |g_S u|$$

问题：选择的size越大，uncertainty项（到超平面距离）的累加和就越多，与最小化函数的目标相矛盾

$$\min MMD(\hat{S}, T) + MMD(\hat{S}_L \cup \hat{Q} \cup T_L, T_U \cup \hat{S}_U) + \frac{\lambda}{\sum \alpha} \alpha \beta |g_S u|$$

尝试：在uncertainty项上除以被选择的样本个数，即对到超平面的距离求平均，能否消除这种冲突

优化式修改

$$\min MMD(\hat{S}, T) + MMD(\hat{S}_L \cup \hat{Q} \cup T_L, T_U \cup \hat{S}_U) + \lambda \alpha \beta |g_S u|$$

问题：选择的size越大，uncertainty项（到超平面距离）的累加和就越多，与最小化函数的目标相矛盾

$$\min MMD(\hat{S}, T) + MMD(\hat{S}_L \cup \hat{Q} \cup T_L, T_U \cup \hat{S}_U) + \frac{\lambda}{\sum \alpha} \alpha \beta |g_S u|$$

尝试：在uncertainty项上除以被选择的样本个数，即对到超平面的距离求平均，能否消除这种冲突

结果：仍然不理想，可能的原因是，如size=1时选到了最小的超平面距离，那么当size=2时，无论新选进来的样本到超平面有多近，都会使平均值变高。

POSS方法修改

$$s.o_1 = \begin{cases} +\infty, & |s| < k \text{ or } |s| > 2k \\ f(s), & otherwise \end{cases}, \quad s.o_2 = |s|.$$

问题：之前仿照KDD一篇文章的方法，把所有 $|s| < k$ 的值都设置成了无穷。这样的设置虽然取得了不错的效果，但是和帕累托优化的思路有些不符。

$$s.o_1 = \begin{cases} +\infty, & s = \{0\}^n, \text{ or } |s| \geq 2k \\ f(s), & otherwise \end{cases}, \quad s.o_2 = |s|.$$

POSS原文中给出的限制如上。

$$\min MMD(\hat{S}, T) + MMD(\hat{S}_L \cup \hat{Q} \cup T_L, T_U \cup \hat{S}_U) + \lambda \alpha \beta |g_S u|$$

思考：我们优化式中会导致一直返回较小size选择的本质原因是：优化式求得的值在不同的size情况下是不可比的。即，只有size相同的情况下，优化式求出值的相对大小才有比较意义。

POSS方法修改

Algorithm 2 POSS

Input: all variables $V = \{X_1, \dots, X_n\}$, a given criterion f and an integer parameter $k \in [1, n]$

Parameter: the number of iterations T and an isolation function $I : \{0, 1\}^n \rightarrow \mathbb{R}$

Output: a subset of V with at most k variables

Process:

```
1: Let  $s = \{0\}^n$  and  $P = \{s\}$ .
2: Let  $t = 0$ .
3: while  $t < T$  do
4:   Select  $s$  from  $P$  uniformly at random.
5:   Generate  $s'$  from  $s$  by flipping each bit of  $s$  with probability  $1/n$ .
6:   if  $\nexists z \in P$  such that  $I(z) = I(s')$  and  $((z.o_1 < s'.o_1 \wedge z.o_2 \leq s'.o_2)$  or  $(z.o_1 \leq s'.o_1 \wedge$ 
       $z.o_2 < s'.o_2))$  then
7:      $Q = \{z \in P \mid I(z) = I(s') \wedge s'.o_1 \leq z.o_1 \wedge s'.o_2 \leq z.o_2\}$ .
8:      $P = (P \setminus Q) \cup \{s'\}$ .
9:   end if
10:   $t = t + 1$ .
11: end while
12: return  $\arg \min_{s \in P, |s| \leq k} f(s)$ 
```

之前忽略的细节：隔离函数的存在就是为了对应这种不可比的情况。我们的隔离函数可以设置成 $I(s)=\text{sizeof}(s)$ ，只有当size相同时才进入比较。

POSS方法修改

Algorithm 2 POSS

Input: all variables $V = \{X_1, \dots, X_n\}$, a given criterion f and an integer parameter $k \in [1, n]$

Parameter: the number of iterations T and an isolation function $I : \{0, 1\}^n \rightarrow \mathbb{R}$

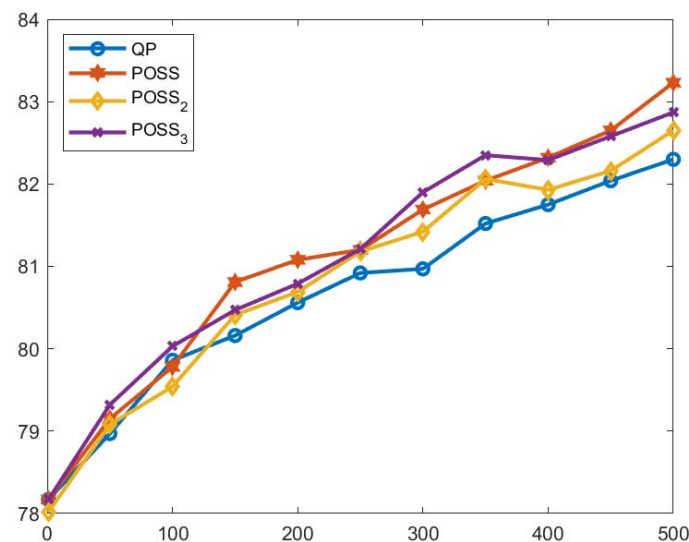
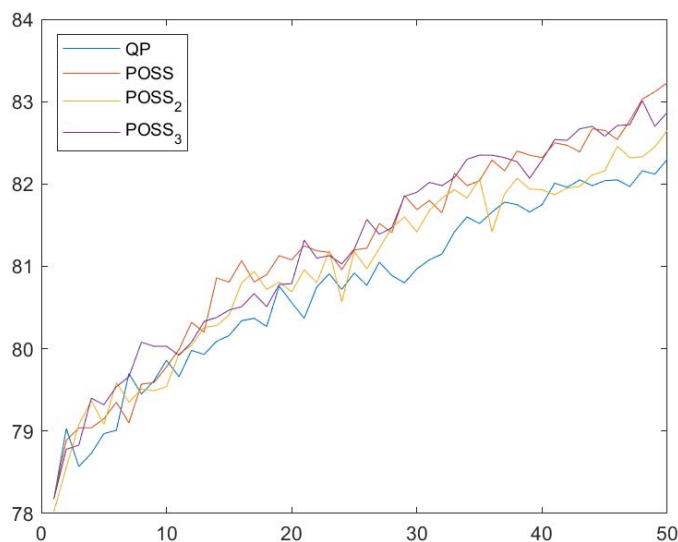
Output: a subset of V with at most k variables

Process:

```
1: Let  $s = \{0\}^n$  and  $P = \{s\}$ .
2: Let  $t = 0$ .
3: while  $t < T$  do
4:   Select  $s$  from  $P$  uniformly at random.
5:   Generate  $s'$  from  $s$  by flipping each bit of  $s$  with probability  $1/n$ .
6:   if  $\nexists z \in P$  such that  $I(z) = I(s')$  and  $((z.o_1 < s'.o_1 \wedge z.o_2 \leq s'.o_2)$  or  $(z.o_1 \leq s'.o_1 \wedge$ 
       $z.o_2 < s'.o_2))$  then
7:      $Q = \{z \in P \mid I(z) = I(s') \wedge s'.o_1 \leq z.o_1 \wedge s'.o_2 \leq z.o_2\}$ .
8:      $P = (P \setminus Q) \cup \{s'\}$ .
9:   end if
10:   $t = t + 1$ .
11: end while
12: return  $\arg \min_{s \in P, |s| \leq k} f(s)$ 
```

这样的设置相当于把删除条件设置为： $z.o_1 \leq s'.o_1$ and $z.o_2 \leq s'.o_2$ 。

POSS方法修改

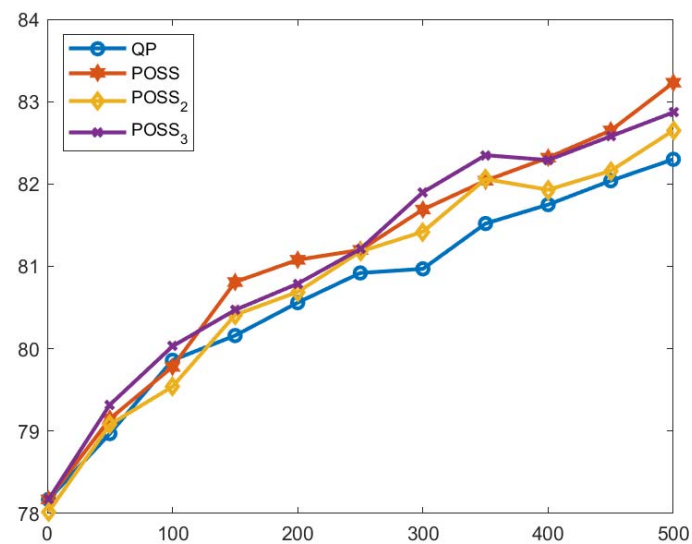
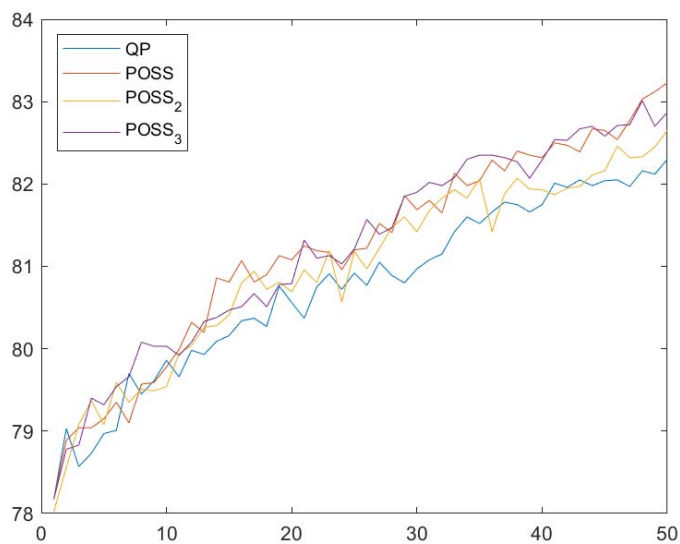


修改后的实验结果：一般☹（其实也没有很一般）

和之前KDD文章中POSS方法跑出来的结果基本类似，没有太大区别。

好处：1.给目标优化式使用POSS优化时只返回较少的size这一问题一个合理的解释；2.相比KDD文章中给出的方法，更加利用了POSS优化的特性。

POSS方法修改



没有取得更好效果的可能原因：POSS中待进行翻转的候选集变大，由原来1-2个候选变成了20个候选，被选中进行翻转的概率变成了 $1/20$ ，和size=10相关的候选项（size接近10）被翻转的相对较少。