

1、优化空间与算法显著性比较

比较准则：计算各方法的覆盖率较之于random baseline的提升百分比

people = 3000, k_favor = 5, eps = 4

n%	no noisy*	our method	noisy baseline	random baseline
candidate=1000	80	22	0.4	0
candidate=800	94	19	6	0
candidate=600	134	21	-2	0
candidate=400	200	34	2	0
candidate=200	267	46	14	0

people = 3000, k_favor = 3, eps = 4

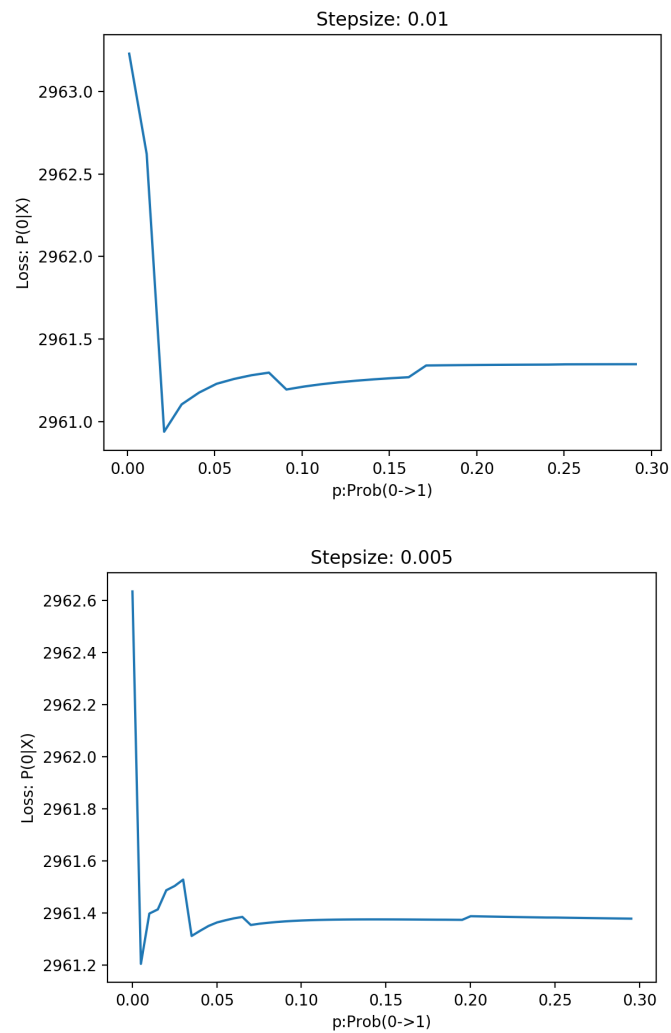
n%	no noisy*	our method	noisy baseline	random baseline
candidate=1000	74	17	-8	0
candidate=800	89	12	-10	0
candidate=600	130	23	-1	0
candidate=400	196	24	-7	0
candidate=200	282	47	-1	0

*: no noisy 方法用的是 $p=1e-5$ ($p=0$ 梯度无法计算，取较小数保证计算可进行)， $q=1$ ，之后仍采用梯度下降的优化方法；较之于利用真实数据随机替换的策略，可以获得更高的覆盖率，可能原因是利用真实数据随机替换的策略需要更多的迭代次数才能收敛到一个比较好的解。

总体而言，随着优化空间的增大，目前算法较之于baseline的优势也在提升，但是和最优结果之间的差距也在逐渐增大。

2、utility function在大数据集上的表现

在服务器上开20个进程并行计算，需要1小时左右完成网格搜索。utility function选择为各地点后验概率向量和先验是否有人0、1向量的RMSE；



选取两图中的最优点 $p=0.005$, $p=0.02$ 以及两个极端值 $p=0.001$, $p=0.3$ 进行训练。

n%	no noisy	our method	noisy baseline	random baseline
p=0.005	809	529	497	474
p=0.02	809	528	474	480
p=0.0001	809	460	464	458
p=0.2	809	510	450	471

可见，两种步长下所选取的最优值点在最终的覆盖率上都要优于其他在utility function上表现次优的点，说明utility function是可取的。