扩样数据的极大熵算法

2018 年 12 月 24 日

1 介绍

这篇文章主要复现文章 [3] 的结果, 利用极大熵算法求解联合概率分布，再依据边界分布再对每一个统计量分别采样，从何合成高维数据。

2 算法介绍

对于在有限集 I 上的联合概率密度函数 p，这里仅考虑具有对数分布的联合概率密度函数

$$\boldsymbol{p} = \pi \prod_{r=1}^{c} \lambda_r^{a_{ri}} \tag{1}$$

其中 λ_r 为单个统计量概率函数，且满足如下约束等式，

$$a_r \boldsymbol{p} = h_r, \qquad r = 1, 2, \cdots, c \tag{2}$$

其中由于对数分布的函数 $\boldsymbol{p}$，得

$$r_i = \begin{cases} 1, & \text{for } T = x \\ 0, & \text{else} \end{cases}$$

引入熵 $I(\boldsymbol{p}||\boldsymbol{\pi}) = \boldsymbol{p} \log \frac{\boldsymbol{p}}{\boldsymbol{\pi}}$, 其中 π 是另一个概率密度函数，而熵 $I(\boldsymbol{p}||\boldsymbol{\pi})$ 则度量了两个概率密度函数的相似程度，类似于几何距离，而极大熵

$$I(\boldsymbol{q}||\boldsymbol{\pi}) = \min_{p \in \mathcal{E}} I(\boldsymbol{p}||\boldsymbol{\pi}) \tag{3}$$

称之为 $\boldsymbol{\pi}$ 在概率空间 $\mathcal{E}$ 上的 $I-$ 投影，且 $\boldsymbol{p} = \boldsymbol{q}^*$, 其具体的几何性质在文章 [1] 中有具体的解释。主要利用极大熵 $I(\boldsymbol{q}||\boldsymbol{\pi})$ 的正交性，可以依据定理，构造迭代算法 [2] 如下，

Theorem 1. 考虑如下概率分布序列 $\{\boldsymbol{p}^{(n)};\quad n=0,1,2,\dots\}$，定义如下

$$\boldsymbol{p}^{(0)}=\boldsymbol{\pi}$$
$$\boldsymbol{p}^{(n+1)}=\boldsymbol{p}^{(n)}\prod_{r=1}^{c}\left(\frac{h_r}{h_r^{(n)}}\right)^{a_{ri}},\qquad n=0,1,2,\cdots$$

其中 $h_r^{(n)}=a_{ri}\boldsymbol{p}^{(n)}$ 为当在函数 $\boldsymbol{p}$ 下的约束等式的值。可以证明序列 $\boldsymbol{p}^{(n)}$ 一致收敛到 $\boldsymbol{q}$，使得 $I(\boldsymbol{q}||\boldsymbol{\pi})=\min_{p\in\mathcal{E}}I(\boldsymbol{p}||\boldsymbol{\pi})$。

于是对于统计量空间 $\mathcal{A}=\{(A)_1,(A)_2,\cdots,(A)_q\}$ 以及每个统计量的样本空间 $\mathcal{R}(\boldsymbol{A}_i)=\{a_1^{(i)},a_2^{(i)},\cdots,a_{k_i}^{(i)}\}$ 以及全样本空间 $\boldsymbol{S}=\times_{i=1}^{q}\mathcal{R}(\boldsymbol{A}_i)$，考虑高维统计量 $T=((A)_1,(A)_2,\cdots,(A)_q)$，并令 $X=(A_i|A_i\in C,C\subset\mathcal{A})$ 为约束变量空间，称为模式。如 $X=(A_1)$ 时，为 $\boldsymbol{p}$ 在单变量 A_1 下的变量空间；如 $X=(A_1,A_2)$ 时，为 $\boldsymbol{p}$ 在双变量 A_1,A_2 下的变量空间。而 $S_x=\times_{A_i\in C}\mathcal{R}(A_i)$ 为约束样本空间。对于给定采样目录数据，得到后验概率 $\tilde{\boldsymbol{p}}(T=x|D)$，于是给出利用极大熵拟合后验概率求解联合概率分布的算法如下。

Algorithm 1: 极大熵求解联合概率分布算法

```
Result: A joint probability distribution
Initialize p;
while p is not convergence do
    for X_i ∈ X do
        for x_{i,j} ∈ S_{X_i} and p̃(T = x_{i,j}|D) ∈ p̃ do
            compute p(T = x_{i,j});
            λ_{i,j} ← λ_{i,j} p̃(T=x_{i,j}|D) / p(T=x_{i,j});
        end
    end
end
```

利用极大熵逼近约束概率条件解空间的正交性，可以将子变量空间的分布进行分别计算，得到一个高维的约束在后续的计算中进行利用，例如可以将三个 1 维边间概率分布进行极大熵拟合得到一个 3 维的边界概率约束。因而具有分布式计算的特性。特别地，如果两组模式 X_1,X_2 为互为独立变量且 $X_1\cup X_2=X$，那么可以分别求得 $\boldsymbol{p}_1^*,\boldsymbol{p}_2^*$ 使得 $\boldsymbol{p}^*=\boldsymbol{p}_1^*\boldsymbol{p}_2^*$，但真实世界中人口数据统计变量很难做到完全独立，所以这一点应用的场景不多。除此之外，如果对于约束条件中较多存在 $\boldsymbol{p}(T=x)=0$ 可以将其作为根节点，

再将其子节点的约束分别计算最终整合成一个更高维的联合概率分布。这些在原论文中都有提到。

最后对于求得的概率分布函数再依据其边际函数的分布，进行采样得到高维数据，算法如下

Algorithm 2: 合成数据采样算法

Initialize $T = \emptyset$;
for $\boldsymbol{A}_i \in \boldsymbol{A}$ **do**
  for $a_j^{(i)} \in \mathcal{R}(\boldsymbol{A}_i)$ **do**
    compute the conditional probability $\boldsymbol{p}^*(a_j^{(i)}|T)$;
  end
  $T(\boldsymbol{A}_i) \leftarrow \text{Sample}(\mathcal{R}(\boldsymbol{A}_i, \{\boldsymbol{p}^*(a_j^{(i)}|T)\}))$;
end

3 数值算例

接下来利用人口数据去除缺省值，并分别按 5%,10% 进行抽样来算法的验证，主要验证算法 1，取 income,age,id,gender,econActivity 共 5 个类别，各个统计量的样本个数为 13,17,11,2,13。为了评价算法结果的近似程度，文章给出类两个度量，非别是

$$\text{BIC}_{\mathcal{X}} = -2\log \mathcal{L}_{\mathcal{X}} + \mathcal{N} \cdot \log |D|,$$

其中

$$\log \mathcal{L}_{\mathcal{X}} = \sum_{T \in D} \log \boldsymbol{p}^*(T) = |D| \Bigg(\sum_{X_i \in \mathcal{X}} \sum_{x_{i,j} \in S_{X_i}} \tilde{\boldsymbol{p}}(T = x_{i,j}|D) \cdot \log u_{i,j} \Bigg),$$

$\mathcal{N}$ 为参数个数，$|D|$ 为数据的类别数，即统计量个数，以及 Kullback-Leibler(KL) divergence

$$h(\alpha, \beta) = \alpha \log \frac{\alpha}{\beta} + (1 - \alpha) \log \frac{1 - \alpha}{1 - \beta}.$$

3.1 算例 1.

在算例 1. 中，对总样本的 5% 进行抽样，取各个统计量的边际密度函数利用算法 1，再分别求得联合分布的边际密度函数如下，最终得到结果如

图 1所示，并且由于对数概率函数的假设以及本身各个统计量边际分布之间并没有矛盾或者冲突，例如给定两个变量的边际分布，又给定改两个变量的二维边际分布，可能存在约束本身不符合概率乘法公式，使得考虑各个情况下的最佳逼近。因此在这个例子下算法 1仅需要 1 步就得到收敛解，

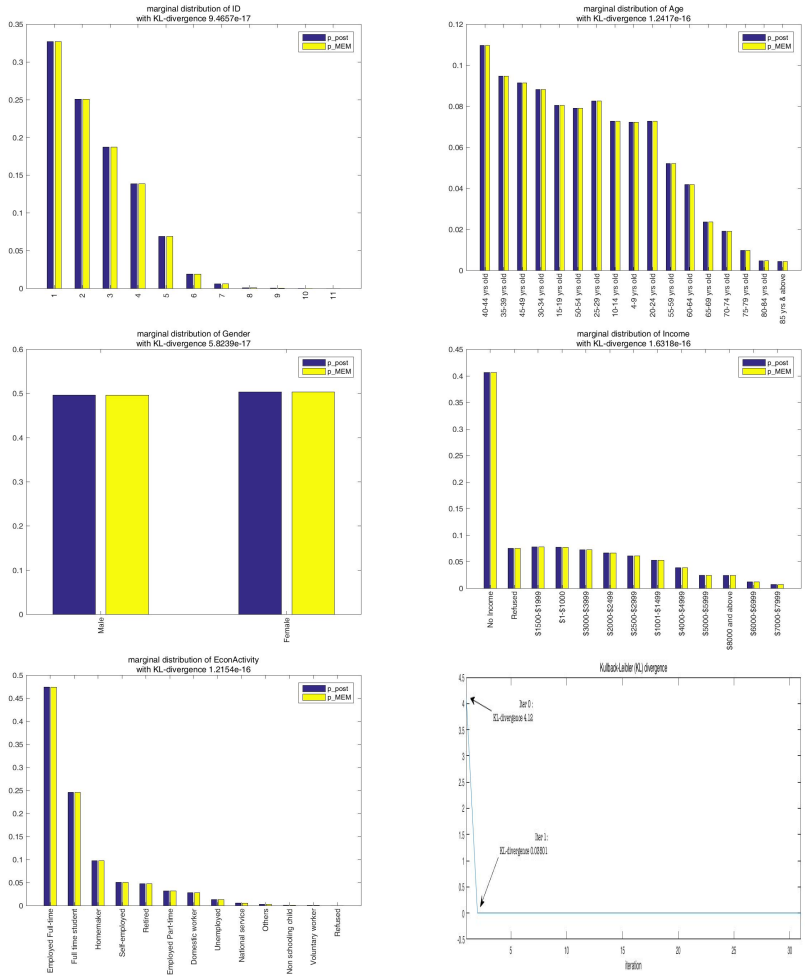

图 1: 抽样边际密度和极大熵拟合联合概率分布的边际分布对比以及 KL-divergence 迭代情况

而对比原本的全样本边际密度分布可得算法的拟合结果如下，

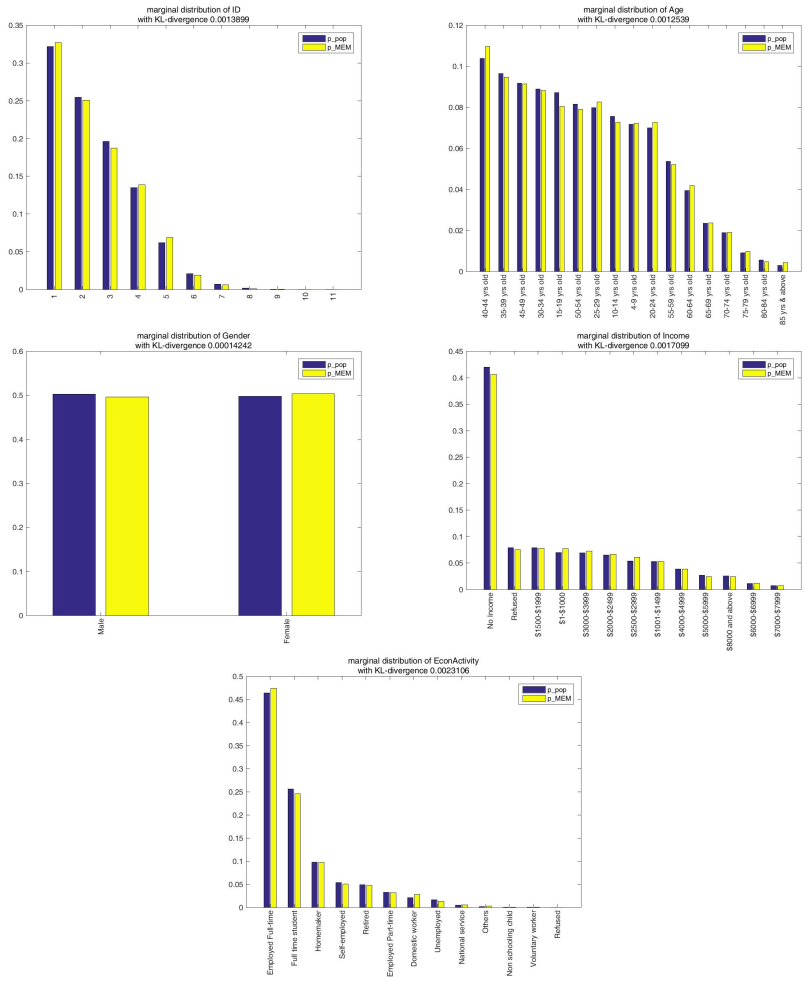

图 2: 全样本边际密度和极大熵拟合联合概率分布的边际分布对比

3.2 算例 2.

在算例 2. 中，采用对总样本 10% 进行抽样，同样可得全样本的边际分布和抽样边际分布逼近的联合概率的边际分布对比，如图 3所示。可以发现每个统计量的分布 KL-divergence 均比起 5% 的抽样结果并没有改进反而更大了。这是因为算法本身及其依赖抽样的正确性，而增大抽样概率能得到更好的对于全样本的描述，有理由相信，更大的抽样概率更能代表整体，因而也可以更好的对联合分布的拟合进行扩样，但是实际情况因人而异。

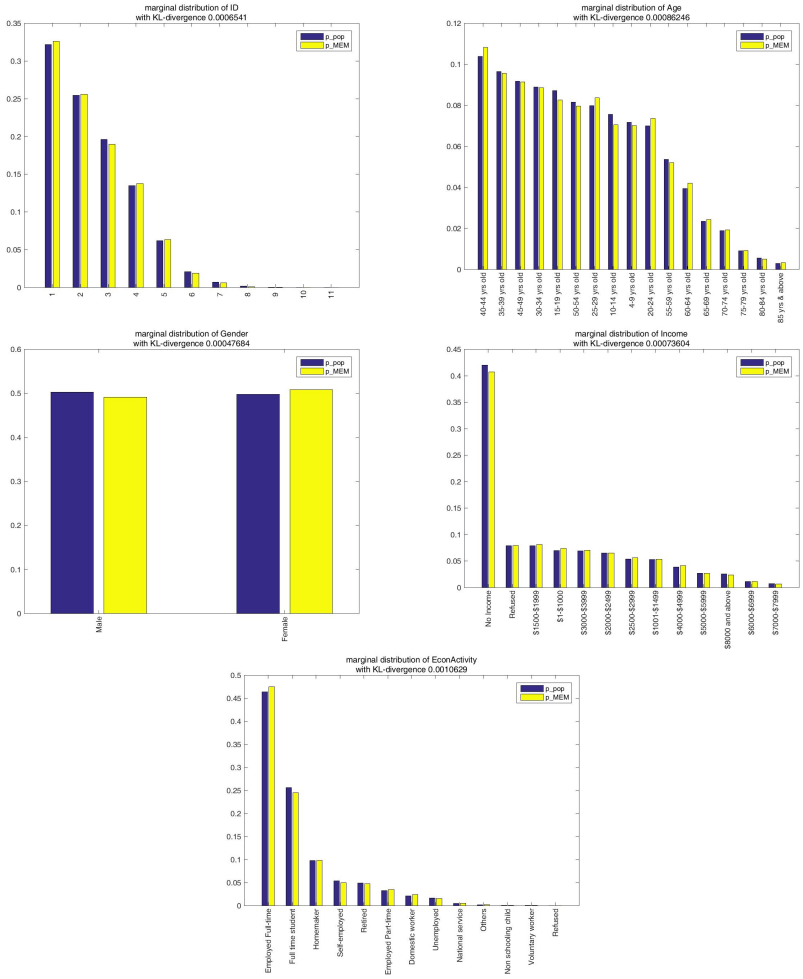

图 3: 全样本边际密度和极大熵拟合联合概率分布的边际分布对比

3.3 算例 3.

在算例 3. 中，将利用算例 1. 中得到联合概率分布中，对二维边际分布进行部分的随机采样，采取如下的统计量模式 {'ID','Age'}, {'Gender', 'Income'},{'Income','EconActivity'}，分别总共有 11*17,2*13,13*13 种组合。而部分随机采样个数采取边际分布的最小维度，和最大维度，得到均得到不错的近似结果，这里仅展示最小维度的情况如图 5。同时又将这两种方法和全部统计量的边际分布迭代的收敛情况，作对比，显然由于信息的缺失，收敛的速度会降低一些，而且由于抽样有随机性，并不能较好的覆盖尽可能多的统计变量的信息，因而部分随机抽样的迭代效果并没有随抽样个数的多少呈现较强的相关性，这些可以从图 6可以看出。并且约束个数的不同，计算量也不尽相同，算法 1所花费的时间也不同，如下表所示

表 1: 不同约束个数下的迭代时长花费

约束种类	模式个数	约束参数的个数	30 次迭代时间 (s)
全变量边际分布	5	56 (11/11;17/17;2/2;13/13;13/13)	0.206591
2D 最小维度抽样边际	3	26 (11/187;2/26;13/169)	0.112209
2D 最大维度抽样边际	3	43 (17/187;13/26;13/169)	0.119276

3.4 算例 4.

在算例 4. 中，将考虑人口数据中的家庭特征"H1_HHID"，将不同的"H1_HHID" 依据所述成员的个数多少进行分类，得到不同家庭成员数的家庭个数，并依据家庭"H1_HHID" 不同的个数，抽样其中 1% 的家庭，其人口数据占总体人口数据的 8.33%，而具体分布如图 6所示。

由于有一些家庭成员较大的家庭，总共的个数都在 1 个，这些家庭的抽样情况要么是 0%，要么是 100%。这部分的信息是极易缺失的，而在计算 KL-divergence 时，因为需要对于 0 作除法，故都将这类的 KL-divergence 记为 0。除此之外，考虑在所有已有抽样的家庭类别中，抽样情况最差的家庭类别，如下图所示，抽样得到的概率分布与原有家庭类别下的概率分布有很大的差异。可以看出只有一个成员的家庭抽样的分布具有与原分布最大

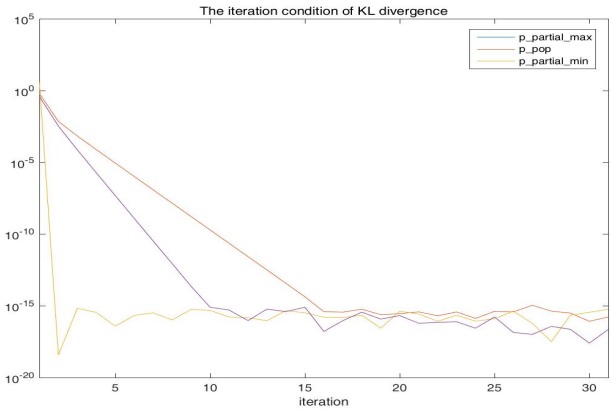

图 4: 不同约束下的 KL-divergence 迭代情况

的 KL-divergence，具体情况可以从图 8看出，

最后，在对家庭类别分别进行抽样分布的拟合的同时，依据抽样的权重拟合全样本的概率得到分布情况如图 9所示。事实上，这么作的目的是将原有抽样算法考虑家庭类别的因素，希望在得到全样本概率分布的同时也保留对家庭分布的保留。可以看到基于家庭类别的抽样拟合方式比直接依据特征拟合的效果要更差一些，但总体保留了一定的样本分布信息。

3.5 算例 5

在算例 5. 中，对于同样具有家庭特征”H1_HHID” 的数据的数据进行验证，也是同样的将不同的”H1_HHID” 依据所述成员的个数多少进行分类，得到不同家庭成员数的家庭个数，但这里人为的将一部分成员特别多（8 个及以上，占总家庭百分比数不足 5% 的家庭归为一类）并依据家庭”H1_HHID” 不同的个数，抽样其中 5% 的家庭，其人口数据占总体人口数据的 18.53%，而具体分布如图 10所示。

而得到的边际概率分布如下图所示，由此这次除家庭编号’H1_HHID’外，共有 8 个特征（’H2_DwellingType’，’H5_VehAvailable’，’Pax_ID’，’P1_Age’，’P5_Employ’，’P6_Occup’，’Area_name’，’P2_Gender’）。且每一个特征的值也很多，分别为 14，2，11，17，11，11，33，2 个，所以在迭代次数设定为 5 次的情况下（在第二次迭代就已经达到收敛解），所花费的时间也有约 16 分钟。要注意的是，在”P6_Occup” 这一特征中，只在”P5_Employ”

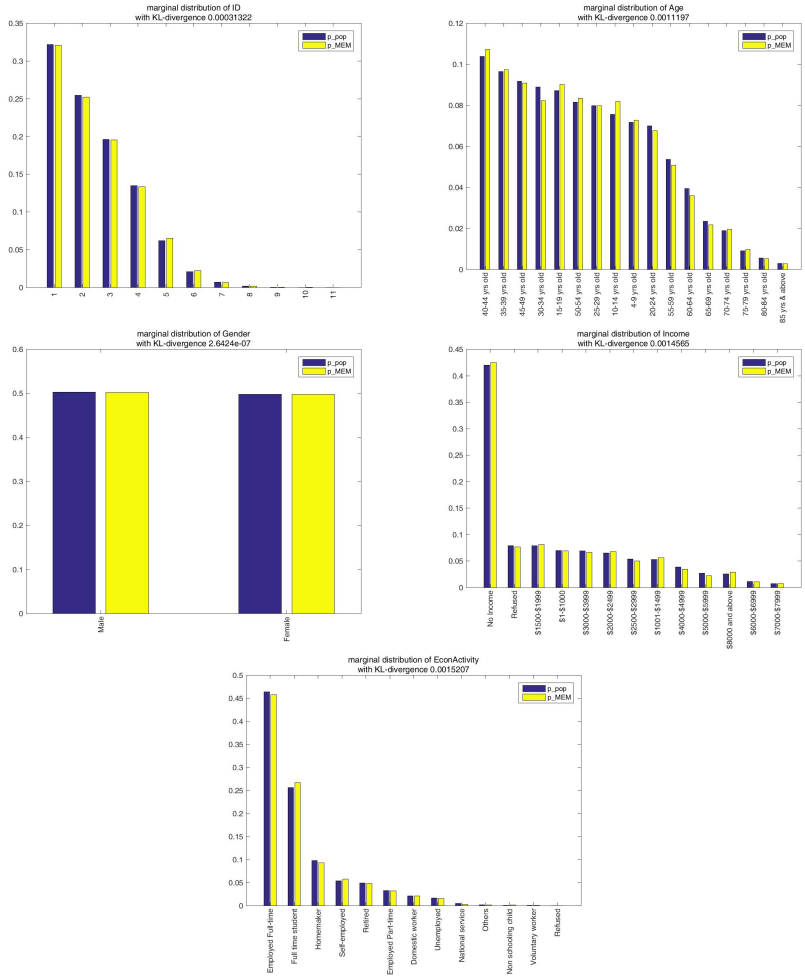

图 5: 全样本边际密度和部分抽样的极大熵拟合联合概率分布的边际分布对比

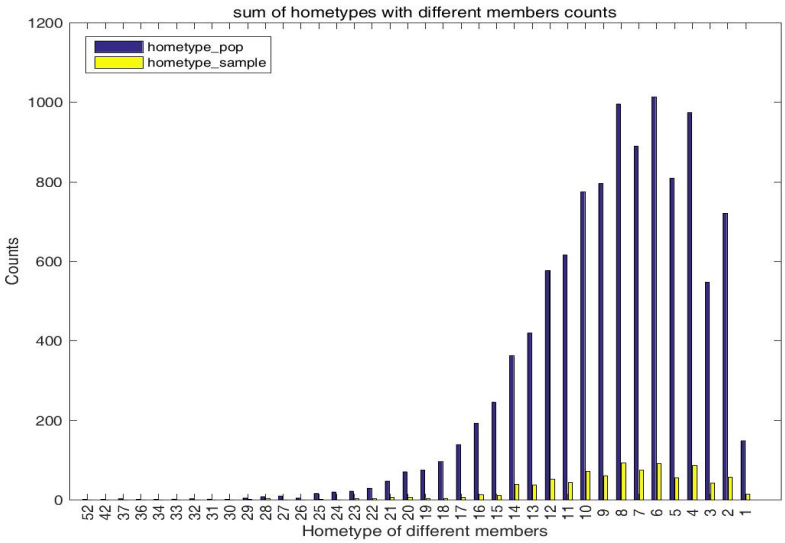

图 6: 依据家庭成员将家庭进行分类得到的分布情况

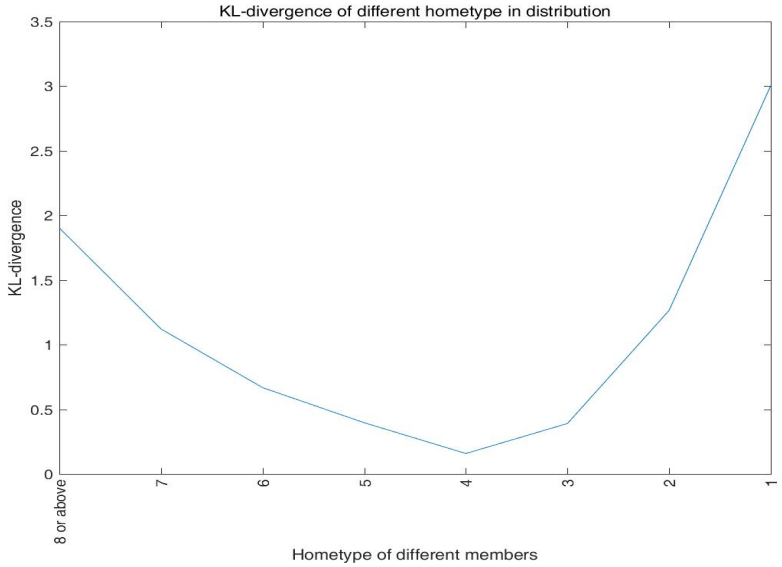

图 7: 不同家庭类型抽样所得分布的 KL-divergence

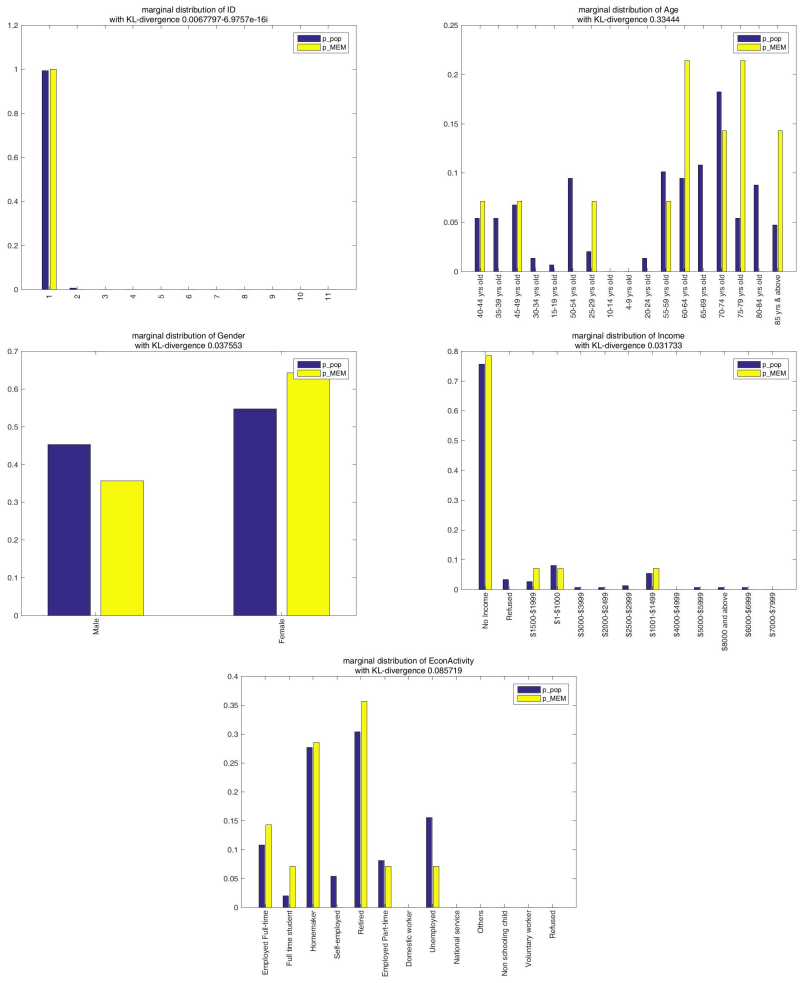

图 8: 家庭人口数为 1 的抽样分布与样本分布

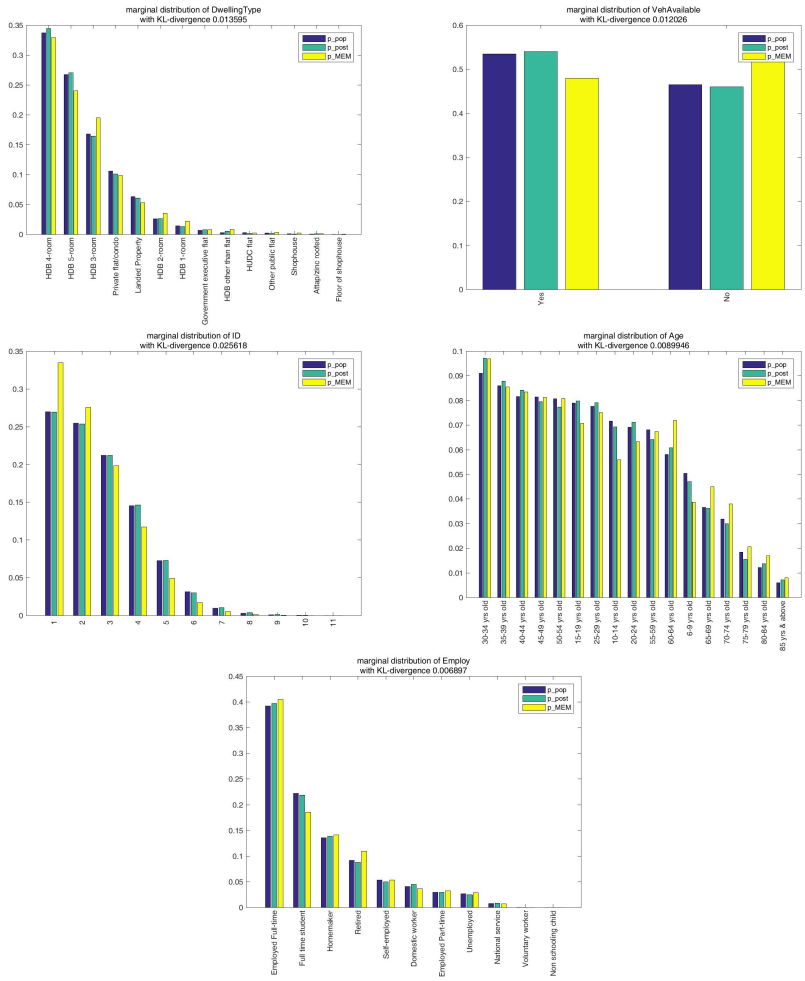

图 9: 家庭人口数为 1 的抽样分布与样本分布

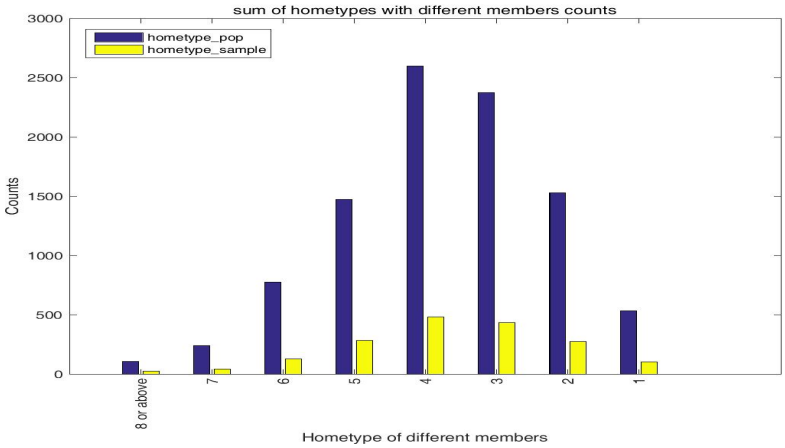

图 10: 依据家庭成员将家庭进行分类得到的分布情况

中取”Employed Full-time”,”Self-employed”, ”Employed Part-time” 时才有可取的变量值，其余为空值。于是在计算时仅考虑”P6_Occup” 非空的情况，在这一情况下已经统计，于是可以将”P6_Occup” 作为”P5_Employed” 的子特征，在扩样时，依据生成的”P5_Employed” 值，进一步抽样。这一关系可由下图看出。

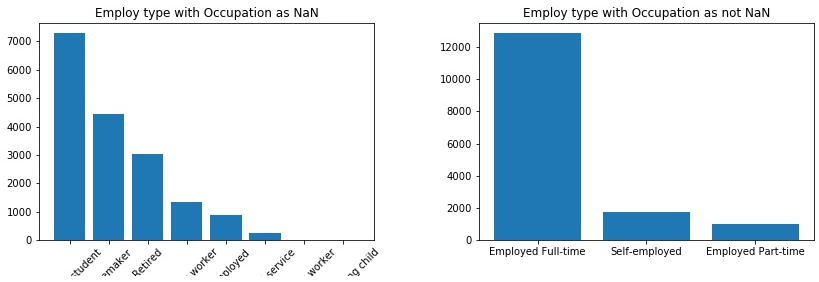

图 11: ”P6_Occup” 的取值与”P5_Employ” 的关系

于是最终的扩样结果如下，

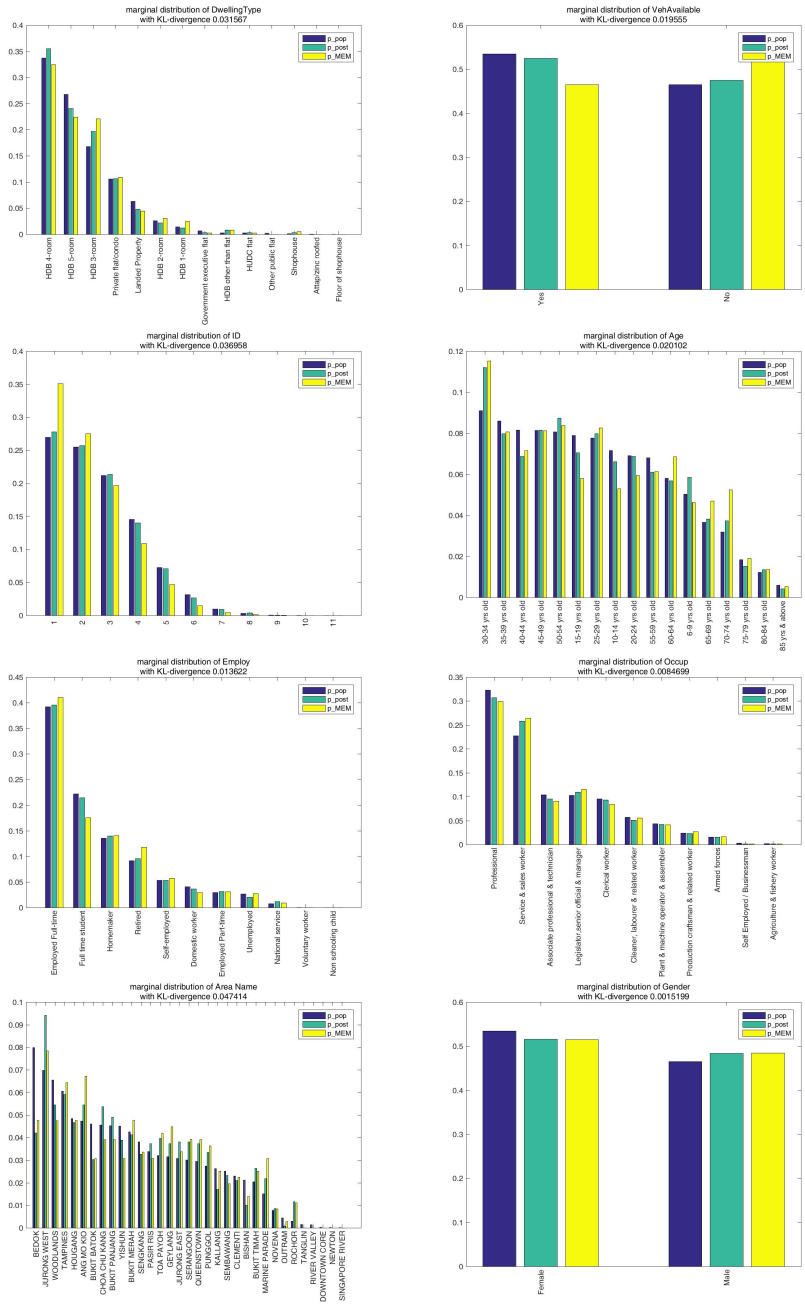

图 12: 基于家庭的扩样与直接扩样与全样本的边际分布

4 总结

文章中数值算例里给出的约束是二维边际分布的一部分，比如对于性别和收入两个统计量，各 2,13 个样本，一共有 26 种组合，但只给出 10 个组合的概率。像这样对于不同的统计量的组合给出不全的概率分布，又恰好可以覆盖全部变量空间（不然仅仅只是部分变量空间的概率）。并由此提出了一些基于给定约束模式的加速算法。

参考文献

[1] I. Csiszar. I-Divergence Geometry of Probability Distributions and Minimization Problems. *Annals of Statistics*, 11(1):141–156, 1983.

[2] J.N. Darroch Ratcliff and D. Generalized Iterative Scaling for Log-Linear Models. *The Annals of Mathematical Statistics*, 1(1):1–2, 1930.

[3] Hao Wu, Yue Ning, Prithwish Chakraborty, Jilles Vreeken, Nikolaj Tatti, and Naren Ramakrishnan. Generating Realistic Synthetic Population Datasets. 2016.

附录 1. 关于代码

1.1 目录结构及环境配置

运行所需要的环境配置：Python，安装 numpy 和 pandas，除此之外是 MATLAB2015b(2014b 以上即可，需要对 table 类型的支持)。

目录结构如下，生成的'*_hometype.csv' 文件是根据家庭人数作分类，再统计这样分类的总体和抽样个数；生成的'*_ 特征.csv' 文件按照列是特征值（比如男、女，对于性别）；列名称前两列'population','sample' 是对与总体与抽样总体的个数统计，之后各列两两配对，依据不同家庭类别下的总体与抽样的个数统计，列名分别为'population_hometype 值','sample_hometype 值'（比如 hometype 为 8 的列名分别为'population_8','sample_8'）

```
code  % matlab scipts & python scripts
      BIC_gen.m  % util
      homeMEM.m  % main 1
```

```
4         KL_gen.m  % util
5         mem.m  % util
6         memDemo.m  % main 2
7         sample.py  % sampling from csv file
8
9   data  % file  for demo 2
10        population_sample.csv  % original csv file, renamed from pop_sample2.
11        population_sample_Area_name.csv
12        population_sample_extension.csv
13     % generated extension 100000 size csv file
14        population_sample_H2_DwellingType.csv
15        population_sample_H5_VehAvailable.csv
16        population_sample_hometype.csv  % hometype csv file
17        population_sample_P1_Age.csv
18        population_sample_P2_Gender.csv
19        population_sample_P5_Employ.csv
20        population_sample_P6_Occup.csv
21        population_sample_Pax_ID.csv
22        pop_sample2.csv
23
24       sample  % file  for demo 1
25              population_sample.csv  % original csv file
26              population_sample_Area_name.csv
27              population_sample_extension.csv
28             % generated extension 1000 csv file
29              population_sample_H2_DwellingType.csv
30              population_sample_H5_VehAvailable.csv
31              population_sample_hometype.csv  % hometype csv file
32              population_sample_P1_Age.csv
33              population_sample_P2_Gender.csv
34              population_sample_P5_EconActivity.csv
35              population_sample_P5_Employ.csv
36              population_sample_P6_Occup.csv
```

```
37              population_sample_P8_Income.csv
38              population_sample_Pax_ID.csv
```

1.2 使用样例

demo 流程的如下，现在 Windows 中 CMD 中，将样本数据抽样，并统计各个特征生成至每个 csv 文件，并加上特征的英文后缀，

```
1  $ python sample.py
2  How much PROPORTION will to sample? (default 5% in all homes):
3  integer format in percentage, e.g., 6 for 6%
4
5  you mean 5.0%
6  which demo want to choose (1/2): [1]
7
8  88600 entries of data
9  37044 entries of sampled data
10 home will be seperated in the following types:
11
12 ['21 or above' 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1]
```

再在 matlab 命令行环境中依次输入如下命令，将会看到抽样之后的统计结果，先是不考虑家庭因素的情况，再在考虑了家庭因素的情况，并最终生成扩样后的文件"*_extension.csv"。

```
1  >> homeMEM();  % demo
2  Which demo do you want? 1/2 [1]:
3  1 - for 5 dimension population data;
4  2 - for 8 dimesion population data;
5  1
6  ...
7  MEM result without considering hometype, [ENTER] to continue
8  [ENTER]
9  How much size want to extend?[default None]:
10
11 ...
```

```
MEM result without considering hometype
```

在实际运行情况下，是不需要再次抽样再扩样的，可以将抽样比例换成100%，解决这一问题。

```
% in windows CMD
$ python sample.py 'relate/path/to/*.csv'
% in MATLAB
>> homeMEM('relate/path/to/*.csv');
% generate 'relate/path/to/*_extension.csv'
```

1.3 函数介绍

下面将介绍几个主要的函数

```
function [p,prefix,columns] = homeMEM(varargin)
% INPUT
% vatargin              : 输入'related/path/to/*.csv'样本统计数据文件
% RETURN
% p                                   : 抽样极大熵联合概率分布
% prefix                        : 文件路径去除'.csv'的前缀
% columns                       : 统计特征的元胞字符串数组
%                                                   : 生成扩样的数据，由

function [p,p_pop,category,patterns,varSub,p_post] = memDemo(varargin)
% INPUT
% varargin              : 输入为类似于(3+2*i)的值，其中i是第i个hometype的ind
%                                                   : 不输入时为demo函数
% OUTPUT
% p                                   : 第i个hometype的联合概率分布，或抽样
% p_pop                         : 总体联合概率分布的一维边际分布集合元胞
% category      : 某一特征的具体变量值，如男、女之于性别
% patterns              : 模式标签的集合矩阵，第3维（共5维）的模式矩阵为第三
% varSub                        : 下标的集合元胞，1维边际下的下标为[[1;2;3;4
% p_post                        : 后验概率的集合元胞，在该patterns下的条件概

```

```
22 function [p,out_p_cond, kl] = mem(p_post, patterns, nVar, varSub, params)
23 % p_post:   the posterior probability of constraint, a (1,nPatterns) cell,
24 %           each cell with size(1,nVar)
25 % patterns: with 1 or 0 to express the constraint pattern, with size
26 %           (nPatterns, nFeature)
27 % nVar:     # of  variables in each constraint, with size(1,nPatterns)
28 % varSub:   the sub of each p_post, a (1,nPatterns) cell, each cell with
29 %           size(nVar, ndim) - varSub{i}(j,:) --> p_post{i}(j)
30 % params:
31 %     params.iterMAX:       max iteration, default 5
32 %     params.graph:         make a graph or not, default not
33 %     params.outpatterns:   return out_p_cond for outpatterns and outVarSub
34 %                           will not return out_p_cond if missing
35 %     params.outVarSub:
```