Case 1. Test KCI (the conditional version) on X;Z|Y

In this case, we test the conditional independence of X and Z conditional on Y. The datasets are generated by Gaussian distributions, exponential, uniform distribution and the mixture of these distributions. For each distribution, we generate 20 datasets and report the resulting p-values and the mean of p-values.

For these datasets, X should be independent on Z conditional on Y, so the p-value is expected to be larger than 0.01. It works as expected if we use the Gaussian process to determine the kernel width for z.

However, when we do not use the Gaussian process to determine the kernel width for z (set IF_GP=0 in the file 'CInd_test_new_withGP.m', Line 19.), but manually set the kernel width to 0.05, 0.1 or 0.2 (pars.width=0.05, pars.width=0.1, pars.width=0.2), the resulting p-values are much smaller that 0.01 (p10, p11, p12) sometimes.

So, here since I don't understand the KCI algorithm well enough, I'm wondering whether these results are expected from our KCI algorithm or not?

- Dataset with Gaussian distribution

```
rng(10,'philox');
n=20;
% dataset with Gaussian distribution
p10Gauss=zeros(n,1);
p11Gauss=zeros(n,1);
p12Gauss=zeros(n,1);
for i=1:n
    X = randn(300,1);
    X_prime = randn(300,1);
    Y = X+0.5*randn(300,1);
    Z = Y+0.5*randn(300,1);
    pars.width = 0.05;
    [p_val stat]=indtest_new(X,Z,Y,pars);
    p10Gauss(i,1) = p_val;
    pars.width = 0.1;
    [p_val stat]=indtest_new(X,Z,Y,pars);
    p11Gauss(i,1) = p_val;
    pars.width = 0.2;
    [p_val stat]=indtest_new(X,Z,Y,pars);
    p12Gauss(i,1) = p_val;
end
p10Gauss
```

```
p10Gauss = 20×1
    0.0012
    0.0000
    0.0005
    0.0004
    0.0005
    0.0002
    0.0018
    0.0004
    0.0006
```

```
    0.0007
      ⋮
```

```
mean(p10Gauss)
```

```
ans = 6.5768e-04
```

```
p11Gauss
```

```
p11Gauss = 20×1
    0.0108
    0.0004
    0.2689
    0.0299
    0.0160
    0.0044
    0.0686
    0.0517
    0.0213
    0.2086
      ⋮
```

```
mean(p11Gauss)
```

```
ans = 0.0607
```

```
p12Gauss
```

```
p12Gauss = 20×1
    0.0655
    0.0081
    0.7303
    0.1625
    0.1510
    0.3610
    0.2581
    0.6663
    0.4649
    0.6695
      ⋮
```

```
mean(p12Gauss)
```

```
ans = 0.2891
```

- Dataset with exponential distribution

```
% dataset with exponential distribution
p10Expo=zeros(n,1);
p11Expo=zeros(n,1);
p12Expo=zeros(n,1);
for i=1:n
    X = exprnd(1, 300, 1);
    X_prime = exprnd(1, 300, 1);
    Y = X + 0.5 * exprnd(1, 300, 1);
    Z = Y + 0.5 * exprnd(1, 300, 1);
```

```
    pars.width = 0.05;
    [p_val stat]=indtest_new(X,Z,Y,pars);
    p10Expo(i,1) = p_val;
    pars.width = 0.1;
    [p_val stat]=indtest_new(X,Z,Y,pars);
    p11Expo(i,1) = p_val;
    pars.width = 0.2;
    [p_val stat]=indtest_new(X,Z,Y,pars);
    p12Expo(i,1) = p_val;
end
p10Expo
```

```
p10Expo = 20×1
    0.0013
    0.0123
    0.0047
    0.0067
    0.0048
    0.0050
    0.1117
    0.0021
    0.0004
    0.0036
      ⋮
```

```
mean(p10Expo)
```

```
ans = 0.0134
```

```
p11Expo
```

```
p11Expo = 20×1
    0.2518
    0.1881
    0.0368
    0.3910
    0.0755
    0.1124
    0.3520
    0.0380
    0.0094
    0.3422
      ⋮
```

```
mean(p11Expo)
```

```
ans = 0.1690
```

```
p12Expo
```

```
p12Expo = 20×1
    0.3489
    0.2350
    0.5695
    0.6430
    0.2457
    0.4047
    0.2028
```

```
    0.6086
    0.0875
    0.7738
      ⋮
```

```
mean(p12Expo)
```

```
ans = 0.3794
```

- Dataset with uniform distribution

```
% dataset with uniform distribution
p10Unif=zeros(n,1);
p11Unif=zeros(n,1);
p12Unif=zeros(n,1);
for i=1:n
    X = rand(300, 1);
    X_prime = rand(300,1);
    Y = X + 0.5 * rand(300, 1);
    Z = Y + 0.5 * rand(300, 1);
    pars.width = 0.05;
    [p_val stat]=indtest_new(X,Z,Y,pars);
    p10Unif(i,1) = p_val;
    pars.width = 0.1;
    [p_val stat]=indtest_new(X,Z,Y,pars);
    p11Unif(i,1) = p_val;
    pars.width = 0.2;
    [p_val stat]=indtest_new(X,Z,Y,pars);
    p12Unif(i,1) = p_val;
end
p10Unif
```

```
p10Unif = 20×1
10^-3 ×
    0.2686
    0.7056
    0.0971
    0.0485
    0.2928
    0.0809
    0.0669
    0.1101
    0.0999
    0.2259
      ⋮
```

```
mean(p10Unif)
```

```
ans = 2.4611e-04
```

```
p11Unif
```

```
p11Unif = 20×1
```

```
    0.0415
    0.0296
    0.0122
    0.0258
    0.0537
    0.0096
    0.0155
    0.1439
    0.0352
    0.0076
      ⋮
```

```
mean(p11Unif)
```

```
ans = 0.0617
```

```
p12Unif
```

```
p12Unif = 20×1
    0.4975
    0.2396
    0.2782
    0.3412
    0.1917
    0.4636
    0.2024
    0.6632
    0.1174
    0.5811
      ⋮
```

```
mean(p12Unif)
```

```
ans = 0.4141
```

- Dataset with mixed distribution

```
% dataset with mixed distribution
p10Mix=zeros(n,1);
p11Mix=zeros(n,1);
p12Mix=zeros(n,1);
for i=1:n
    X = rand(300, 1);
    X_prime = randn(300,1);
    Y = X + 0.5 * exprnd(1, 300, 1);
    Z = Y + 0.5 * randn(300, 1);
    pars.width = 0.05;
    [p_val stat]=indtest_new(X,Z,Y,pars);
    p10Mix(i,1) = p_val;
    pars.width = 0.1;
    [p_val stat]=indtest_new(X,Z,Y,pars);
    p11Mix(i,1) = p_val;
    pars.width = 0.2;
    [p_val stat]=indtest_new(X,Z,Y,pars);
    p12Mix(i,1) = p_val;
```

```
end
p10Mix
```

```
p10Mix = 20×1
    0.0006
    0.0025
    0.0039
    0.0034
    0.0011
    0.0013
    0.0025
    0.0008
    0.0010
    0.0004
      ⋮
```

```
mean(p10Mix)
```

```
ans = 0.0021
```

```
p11Mix
```

```
p11Mix = 20×1
    0.0425
    0.0087
    0.1729
    0.0761
    0.0294
    0.1598
    0.0245
    0.0230
    0.0473
    0.0632
      ⋮
```

```
mean(p11Mix)
```

```
ans = 0.0687
```

```
p12Mix
```

```
p12Mix = 20×1
    0.1433
    0.0733
    0.6245
    0.4001
    0.3892
    0.3446
    0.2446
    0.2414
    0.4544
    0.1408
      ⋮
```

```
mean(p12Mix)
```

```
ans = 0.3326
```

Case 2. Test KCI (the unconditional version) on X;Z

For the following data with mixed distribution, X should be dependent on Z, so the p-value is expected to be much smaller than 0.01.

However, when the kernel width is manually set to 0.05 or 0.1, p-values (p20, p21, respectively) are larger than 0.01; while for a bit larger kernel width (pars.width=0.2) , the resulting p-value (p22) looks more reasonable.

While for datasets generated by a single distribution, like Gaussian, exponential or uniform distributions, p-values are smaller than 0.01 as expected.

So again, I'm wondering whether these results are expected from our KCI algorithm or not?

- Dataset with mixed distribution

```
% dataset with mixed distribution
p20Mix=zeros(n,1);
p21Mix=zeros(n,1);
p22Mix=zeros(n,1);
for i=1:n
    X = rand(300, 1);
    X_prime = randn(300,1);
    Y = X + 0.5 * exprnd(1, 300, 1);
    Z = Y + 0.5 * randn(300, 1);
    pars.width = 0.05;
    [p_val stat]=indtest_new(X,Z,[],pars);
    p20Mix(i,1) = p_val;
    pars.width = 0.1;
    [p_val stat]=indtest_new(X,Z,[],pars);
    p21Mix(i,1) = p_val;
    pars.width = 0.2;
    [p_val stat]=indtest_new(X,Z,[],pars);
    p22Mix(i,1) = p_val;
end
p20Mix
```

```
p20Mix = 20×1
    0.2400
    0.0090
    0.0330
    0.0310
    0.1880
    0.0030
    0.5220
    0.0830
         0
    0.0080
     ⋮
```

```
mean(p20Mix)
```

```
ans = 0.1534
```

```
p21Mix
```

```
p21Mix = 20×1
    0.0130
         0
    0.0010
         0
    0.0180
    0.0020
    0.0950
    0.0100
         0
         0
      ⋮
```

```
mean(p21Mix)
```

```
ans = 0.0288
```

```
p22Mix
```

```
p22Mix = 20×1
         0
         0
         0
         0
         0
    0.0010
    0.0030
         0
         0
         0
      ⋮
```

```
mean(p22Mix)
```

```
ans = 0.0037
```

- Dataset with Gaussian distribution

```
% dataset with Gaussian distribution
p20Gauss=zeros(n,1);
p21Gauss=zeros(n,1);
p22Gauss=zeros(n,1);
for i=1:n
    X = randn(300,1);
    X_prime = randn(300,1);
    Y = X+0.5*randn(300,1);
    Z = Y+0.5*randn(300,1);
    pars.width = 0.05;
    [p_val stat]=indtest_new(X,Z,[],pars);
    p20Gauss(i,1) = p_val;
    pars.width = 0.1;
    [p_val stat]=indtest_new(X,Z,[],pars);
```

```
    p21Gauss(i,1) = p_val;
    pars.width = 0.2;
    [p_val stat]=indtest_new(X,Z,[],pars);
    p22Gauss(i,1) = p_val;
end
p20Gauss
```

```
p20Gauss = 20×1
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     ⋮
```

```
mean(p20Gauss)
```

```
ans = 0
```

```
p21Gauss
```

```
p21Gauss = 20×1
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     ⋮
```

```
mean(p21Gauss)
```

```
ans = 0
```

```
p22Gauss
```

```
p22Gauss = 20×1
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     ⋮
```

```
mean(p22Gauss)
```

```
ans = 0
```

- Dataset with exponential distribution

```
% dataset with exponential distribution
p20Expo=zeros(n,1);
p21Expo=zeros(n,1);
p22Expo=zeros(n,1);
for i=1:n
    X = exprnd(1, 300, 1);
    X_prime = exprnd(1, 300, 1);
    Y = X + 0.5 * exprnd(1, 300, 1);
    Z = Y + 0.5 * exprnd(1, 300, 1);
    pars.width = 0.05;
    [p_val stat]=indtest_new(X,Z,[],pars);
    p20Expo(i,1) = p_val;
    pars.width = 0.1;
    [p_val stat]=indtest_new(X,Z,[],pars);
    p21Expo(i,1) = p_val;
    pars.width = 0.2;
    [p_val stat]=indtest_new(X,Z,[],pars);
    p22Expo(i,1) = p_val;
end
p20Expo
```

```
p20Expo = 20×1
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     ⋮
```

```
mean(p20Expo)
```

```
ans = 0
```

```
p21Expo
```

```
p21Expo = 20×1
     0
     0
     0
     0
     0
     0
     0
     0
```

```
    0
    0
    ⋮
```

```
mean(p21Expo)
```

```
ans = 0
```

```
p22Expo
```

```
p22Expo = 20×1
    0
    0
    0
    0
    0
    0
    0
    0
    0
    0
    ⋮
```

```
mean(p22Expo)
```

```
ans = 0
```

- Dataset with uniform distribution

```
% dataset with uniform distribution
p20Unif=zeros(n,1);
p21Unif=zeros(n,1);
p22Unif=zeros(n,1);
for i=1:n
    X = rand(300, 1);
    X_prime = rand(300,1);
    Y = X + 0.5 * rand(300, 1);
    Z = Y + 0.5 * rand(300, 1);
    pars.width = 0.05;
    [p_val stat]=indtest_new(X,Z,[],pars);
    p20Unif(i,1) = p_val;
    pars.width = 0.1;
    [p_val stat]=indtest_new(X,Z,[],pars);
    p21Unif(i,1) = p_val;
    pars.width = 0.2;
    [p_val stat]=indtest_new(X,Z,[],pars);
    p22Unif(i,1) = p_val;
end
p20Unif
```

```
p20Unif = 20×1
    0
    0
    0
```

```
     0
     0
     0
     0
     0
     0
     0
     ⋮
```

```
mean(p20Unif)
```

```
ans = 0
```

```
p21Unif
```

```
p21Unif = 20×1
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     ⋮
```

```
mean(p21Unif)
```

```
ans = 0
```

```
p22Unif
```

```
p22Unif = 20×1
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     ⋮
```

```
mean(p22Unif)
```

```
ans = 0
```