Laplace Approximation

The following section sets up the equations required to compute the exact and the Laplace approximations of the conditional probabilities

```
SetOptions[$FrontEndSession, EvaluationCompletionAction → "ShowTiming"]
(* Exact probability *)
```

$$\texttt{p[y\_, Inp\_, k\_, V\_] :=}\ e^{\frac{2\,V\,y\,(\mathrm{Inp}^2\,k^2-\alpha^2)}{(-\mathrm{Inp}\,k+\alpha)^2}}\ e^{-\mathrm{Log}\left[1+\frac{(\alpha-\mathrm{Inp}\,k)\,y}{\mathrm{Inp}\,k\,O_{tot}}\right]}\ e^{\frac{4\,\mathrm{Inp}\,k\,O_{tot}\,\alpha\,V}{(\alpha-\mathrm{Inp}\,k)^2}\,\mathrm{Log}\left[1+\frac{(\alpha-\mathrm{Inp}\,k)\,y}{\mathrm{Inp}\,k\,O_{tot}}\right]};$$

```
(* I checked that this probability is the correct solution to FP equation *)
(* define f(x) and find max O_max. f(x) is the
  exponent in the conditional probability as V → infinity *)
```

$$\texttt{f[y\_, Inp\_, k\_] :=}\ \frac{4\,\mathrm{Inp}\,k\,O_{tot}\,\alpha}{(\alpha-\mathrm{Inp}\,k)^2}\,\mathrm{Log}\left[1+\frac{(\alpha-\mathrm{Inp}\,k)\,y}{\mathrm{Inp}\,k\,O_{tot}}\right]-\frac{2\,y\,(\mathrm{Inp}\,k+\alpha)}{\alpha-\mathrm{Inp}\,k}$$

```
(*Solve[D[f[y,Inp,k],y]==0,y]; *)
(* the following with the linear correction works really well *)
```

$$\texttt{(*fDoublePrimeyMax[Inp\_,k\_]:=}-\frac{2\ (\alpha+\mathrm{Inp}\ k)}{\mathrm{Inp}\ \ *k*\ \ O_{tot}}-\ \mathrm{Inp}*k;\ \texttt{*)}$$

$$\texttt{fDoublePrimeyMax[Inp\_, k\_] :=}\ -\frac{2}{O_{tot}}-\frac{\mathrm{Inp}\,k}{\alpha\,O_{tot}}-\frac{\alpha}{\mathrm{Inp}\,k\,O_{tot}};$$

$$\texttt{(*fDoublePrimeyMax[Inp\_,k\_]:=}-\frac{2\ (\alpha+\mathrm{Inp}\ k)}{\mathrm{Inp}\ \ *k*\ \ O_{tot}};\texttt{*)}$$

$$\texttt{fMax[Inp\_, k\_] :=}\ \frac{-2\,\mathrm{Inp}\,k}{\alpha-\mathrm{Inp}\,k}+\frac{(4\,\mathrm{Inp}\,k\,\alpha\,O_{tot})}{(\alpha-\mathrm{Inp}\,k)^2}\,\mathrm{Log}\left[\frac{2\,\alpha}{\alpha-\mathrm{Inp}\,k}\right];$$

$$\texttt{Omax[Inp\_, k\_] :=}\ \frac{\mathrm{Inp}\,k\,O_{tot}}{\mathrm{Inp}\,k+\alpha};$$

```
(* location of the mean of the conditional probability *)
(* Define the laplace approximation of the exact probability *)
(* works okay for boundary near 1 *)
```

$$\texttt{(*pLaplace[y\_,Inp\_,k\_,V\_]:=}$$
$$\mathrm{Exp}\left[V\left(\frac{1}{2/(3*\mathrm{Inp})}\texttt{fDoublePrimeyMax[Inp,k]}\ \ (y-\texttt{Omax[Inp,k]})^2\right)\right]\texttt{*)}$$

$$\texttt{pLaplace[y\_, Inp\_, k\_, V\_] := Exp}\left[V\left(\frac{1}{2}\ \texttt{fDoublePrimeyMax[Inp, k]}\ \left(y-\texttt{Omax[Inp, k]}\right)^2\right)\right]$$

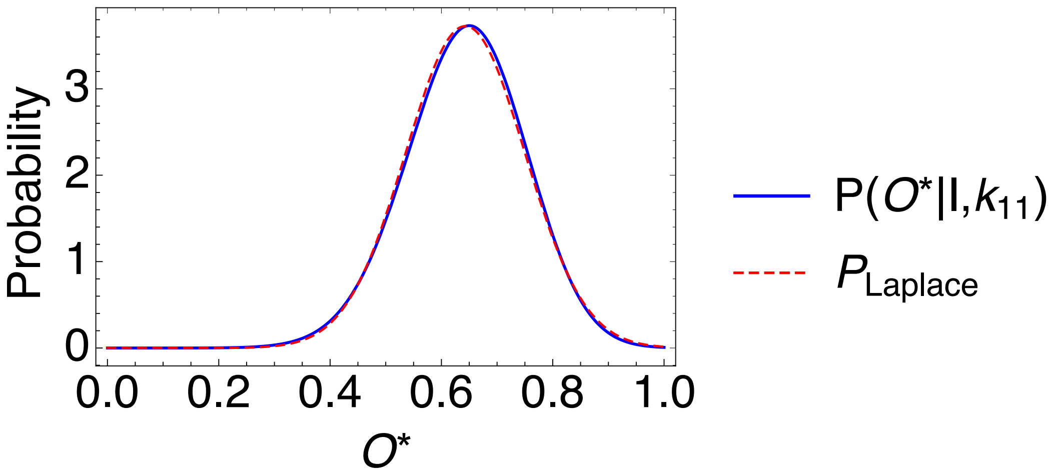
```
pLaplace[y, x, k, V]
(* Some sample parameter choices *)
```

O_{tot} = 1; α = 1;

```
k = 2.001;
Inp = 0.9;
V = 20;

(* the following are the normalizations for one input *)
NormPLap = NIntegrate[pLaplace[y, Inp, k, V], {y, 0, 1}];
NormPCond = NIntegrate[p[y, Inp, k, V], {y, 0, 1}];
(* Recreate probability comparison *)
```

```
Plot[{p[y, Inp, k, V]/NormPCond, pLaplace[y, Inp, k, V]/NormPLap}, {y, 0.0, 1},
 PlotRange → All, PlotLegends → {"P(O*|I,k11)", "PLaplace" },
 PlotStyle → {{Thickness[0.005], Blue}, {Red, Dashed, Thickness[0.004]},
   {Thickness[0.005], Green}}, TicksStyle → Directive["Label", 30],
 Ticks → {Automatic}, LabelStyle → {FontFamily → "Helvetica", FontSize → 24},
 FrameLabel → {"O*", "Probability "}, Frame → True, FrameStyle → Directive[Black]]
```

$$e^{\frac{1}{2} V \left(-\frac{2}{O_{tot}}-\frac{k\,x}{\alpha\,O_{tot}}-\frac{\alpha}{k\,x\,O_{tot}}\right)\left(y-\frac{k\,x\,O_{tot}}{k\,x+\alpha}\right)^2}$$




```
(* Reproduce corrected conditional probability in java. *)
NIntegrate[pLaplace[y, 0.505, k, V], {y, 0, 1}]
```

0.274273

```
pLaplace[0.505, 0.505, 3, 20]
```

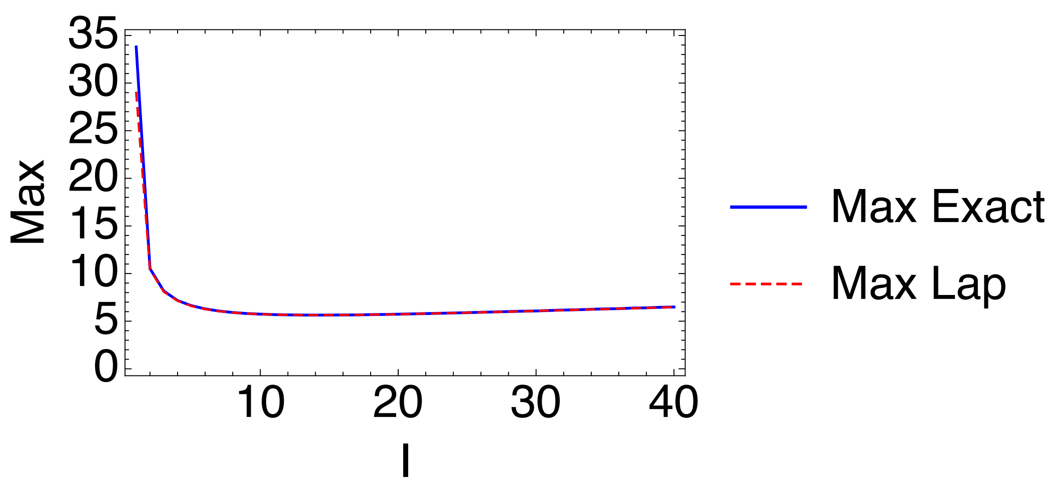
```
pLaplace[0.055, 0.055, 3, 20]/NIntegrate[pLaplace[y, 0.055, k, V], {y, 0, 1}]
```

2.16923

Plot Probability Maxima

```
(* This shows that the maxima disagrees a lot near the boundary y =
 0 unless include the linear correction
   in the variance or fDoublePrimeyMax = -k*input *)
```

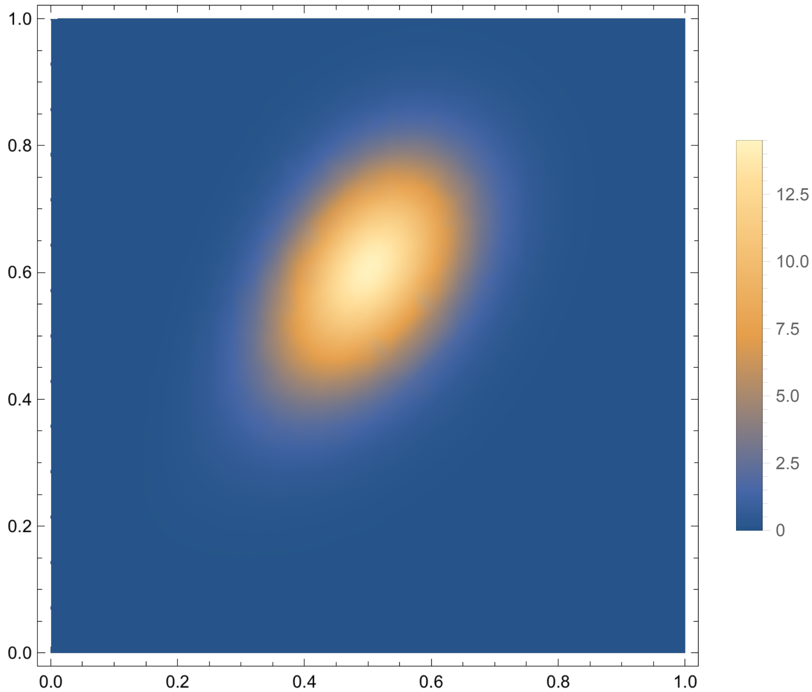
```
maxPy[k_, Inp_] := Max[Function[y, p[y, Inp, k, V] / NIntegrate[p[yp, Inp, k, V], {yp, 0, 1}]] /@
    Range[0.005, 0.995, 0.01]];
maxPyLap[k_, Inp_] := Max[Function[y,
      pLaplace[y, Inp, k, V] / NIntegrate[pLaplace[yp, Inp, k, V], {yp, 0, 1}]] /@ Range[0.005, 0.995, 0.01]];
maxPyList[k_] := Function[Inp, maxPy[k, Inp]] /@ Range[0.005, 0.995, 0.025];
maxPyLapList[k_] := Function[Inp, maxPyLap[k, Inp]] /@ Range[0.005, 0.995, 0.025];

ListLinePlot[{maxPyList[3], maxPyLapList[3]},
 PlotLegends → {"Max Exact", "Max Lap" },
 PlotStyle → {{Thickness[0.005], Blue}, {Red, Dashed, Thickness[0.004]}},
 TicksStyle → Directive["Label", 30], Ticks → {Automatic},
 LabelStyle → {FontFamily → "Helvetica", FontSize → 24}, FrameLabel → {"I", "Max"},
 Frame → True, FrameStyle → Directive[Black], PlotRange → All]
```

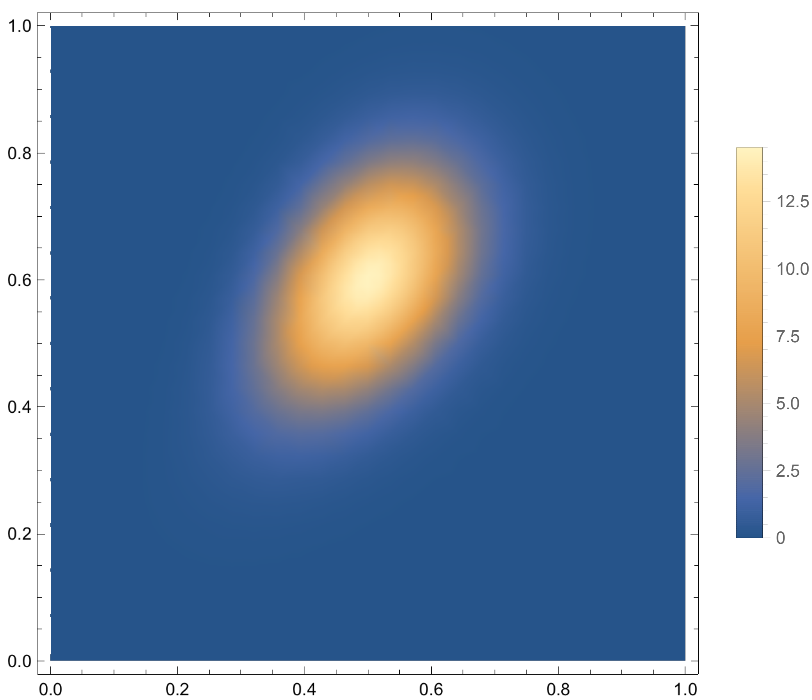


Get Output Probability

```
pInput[Inp_, mu_, sigma_] := Exp[-(Inp - mu)^2/(2 sigma^2)] * 1/(2 Pi sigma^2)^(1/2)
(* This probability is normalized *)
dx = 0.1;
pLaplaceNormalized[y_, Inp_, k_, V_] :=
 Exp[V (1/2 fDoublePrimeyMax[Inp, k] (y - Omax[Inp, k])^2)]
  (NIntegrate[pLaplace[yp, Inp, k, V], {yp, 0.0001, 1}])^-1
pOutputLap[y_, k_, V_] := Sum[pLaplaceNormalized[y, Inp, k, V] *
   pInput[Inp, 0.5, 0.1] * dx, {Inp, 0.0001, 1, dx}]
pOutputLap1[y_, k_, V_] := Sum[pLaplace[y, Inp, k, V] * pInput[Inp, 0.5, 0.1],
  {Inp, 0.0001, 1, dx}]
(* Sum[pOutputLap[y,k,V]*0.1,{y,0,1,0.1}] *)
 pNormalized[y_, Inp_, k_, V_] :=
 p[y, Inp, k, V] (NIntegrate[p[yp, Inp, k, V], {yp, 0.0001, 1}])^-1
pOutput[y_, k_, V_] :=
 Sum[pNormalized[y, Inp, k, V] * pInput[Inp, 0.5, 0.1] * dx, {Inp, 0.0001, 1, dx}]
pOutput1[y_, k_, V_] :=
 Sum[p[y, Inp, k, V] * pInput[Inp, 0.5, 0.1] * dx, {Inp, 0.0001, 1, dx}]
Sum[pNormalized[y, 0.5, 1.5, 100] * 0.1, {y, 0, 1, 0.1}]

pJoint[y_, Inp_, k_, V_] := pNormalized[y, Inp, k, V] * pInput[Inp, 0.5, 0.1]
pJointLaplace[y_, Inp_, k_, V_] :=
 pLaplaceNormalized[y, Inp, k, V] * pInput[Inp, 0.5, 0.1]
(* check to see if joint probabilities look similar *)
DensityPlot[{pJoint[y, x, k, V]}, {x, 0, 1},
 {y, 0, 1}, PlotRange → All, PlotLegends -> Automatic]
DensityPlot[{pJointLaplace[y, x, k, V]}, {x, 0, 1},
 {y, 0, 1}, PlotRange → All, PlotLegends -> Automatic]
```

```
0.998724
```

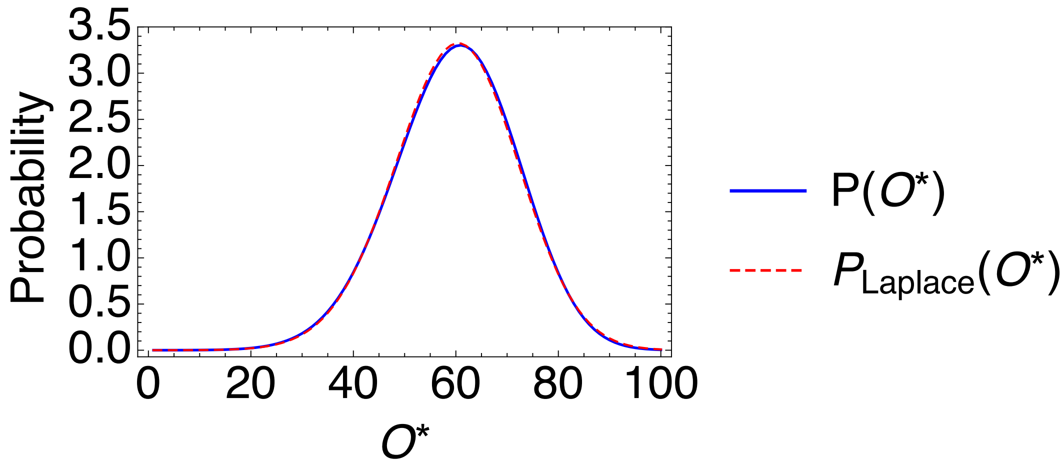




Check that out probabilities match

```
lapList = Function[x, pOutputLap[x, k, V]] /@ Range[0.005, 0.995, 0.01];
exactList = Function[x, pOutput[x, k, V]] /@ Range[0.005, 0.995, 0.01];
```

```
ListLinePlot[{exactList, lapList}, PlotLegends → {"P(O*)", "P_Laplace(O*)" },
 PlotStyle → {{Thickness[0.005], Blue}, {Red, Dashed, Thickness[0.004]}},
 TicksStyle → Directive["Label", 30], Ticks → {Automatic},
 LabelStyle → {FontFamily → "Helvetica", FontSize → 24},
 FrameLabel → {"O*", "Probability "}, Frame → True,
 FrameStyle → Directive[Black], PlotRange → All]
```



Compute Mutual Information

```
(* Use java program to compute these integrals *)
```

```
MI[k_] := Sum[pJoint[y, x, k, V] * Log[pJoint[y, x, k, V] / (pOutput[y, k, V] pInput[x, 0.5, 0.1])] * 0.1 * 0.1,
  {x, 0.01, 1, 0.1}, {y, 0.01, 1, 0.1}]
MILap[k_] := Sum[pJointLaplace[y, x, k, V] *
    Log[pJointLaplace[y, x, k, V] / (pOutputLap[y, k, V] pInput[x, 0.5, 0.1])] * 0.1 * 0.1,
  {y, 0.01, 1, 0.1}, {x, 0.01, 1, 0.1}]

MI[3]
MILap[3]
```

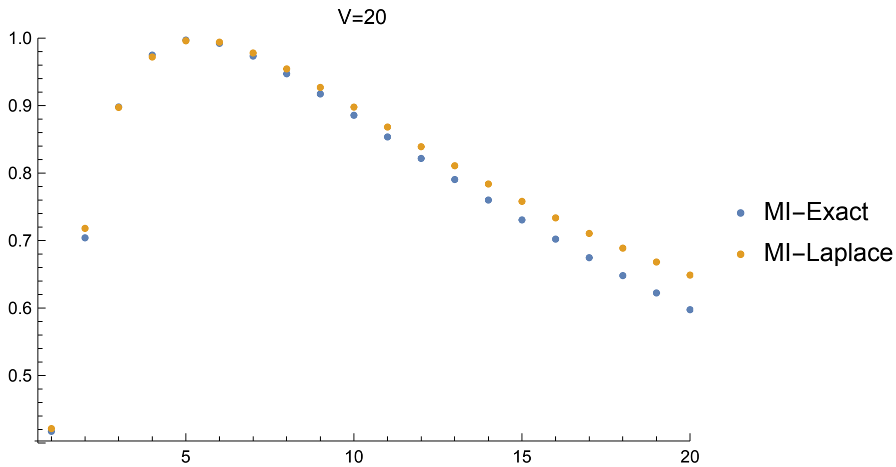
```
0.0977908

0.0981962
```

```
V = 20;
k = 1.5;
MILoop = 0;
For[y = 10^-6, y < 1, y += 0.1,
 For[
  x = 10^-6, x < 1, x += 0.1,
  (*Print[y," - ",x];*)
  If[pJoint[y, x, k, V] > 0,
   MILoop += pJoint[y, x, k, V] * Log[pJoint[y, x, k, V] / (pOutput[y, k, V] pInput[x, 0.5, 0.1])] * 0.1 * 0.1;
   (*Print[MILoop]*)
  ]
 ]
]
MILoop
```

```
0.294445
```

```
MIList = Function[k, MI[k]] /@ Range[0.1, 20, 0.5];
(*MILapList = Function[k,MILap[k]]/@Range[0.1,50,0.5];*)
MILapList = Function[k, MILap[k]] /@ Range[0.1, 20, 0.5];
(* ListPlot[{MIList,MILapList}] *)
(*ListPlot[{MIList,MILapList},PlotRange→All]*)
```

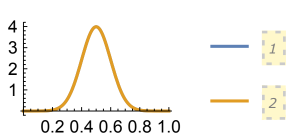
```
ListPlot[{MIList * 10, MILapList * 10}, PlotRange → {{0.5, 20}, {0.4, 1}},
 PlotLegends → {"MI-Exact", "MI-Laplace"}, PlotLabel → "V=20"]
```



Under Development

```
(* Compute MI

  pOutputLap[y_,k_,V_]:=
 NIntegrate[pLaplace[y,Inp,k,V]*pInput[Inp,0.5,0.1],{Inp,0,1}]
   pJointApprox[y_,Inp_,k_,V_]:=pLaplace[y,Inp,k,V]*pInput[Inp,0.5,0.1]

    pOutput[y_,k_,V_]:=NIntegrate[p[y,Inp,k,V]*pInput[Inp,0.5,0.1],{Inp,0,1}]
     pJoint[y_,Inp_,k_,V_]:=p[y,Inp,k,V]*pInput[Inp,0.5,0.1]

MI[k_]:=Sum[Sum[pJoint[y,x,k]*Log[pJoint[y,x,k,V]/(pOutput[y,k,V]pInput[x,0.5,0.01])],
         {x,0.01,1,0.05}],{y,0.01,1, 0.05}]
       MI[0.2]
$Aborted

       MIList2 = Function[k,MI[k]]/@Range[0.5,5,0.5];


pIn = ProbabilityDistribution[pInput[x,0.5,0.1],{x,0,1},Method→"Normalize"]

   Plot[{PDF[pIn,x], pInput[x,0.5,0.1]/nIn},{x,0,1},PlotRange→All,PlotLegends→Automatic]
```

ProbabilityDistribution$\left[3.9894250911642475\grave{}\ e^{-49.99999999999999\grave{}\ (-0.5\grave{}+x)^2},\{x,0,1\}\right]$



```
nIn = NIntegrate[pInput[x,0.5,0.1],{x,0,1}]
0.25066268375731465`
Manipulate[
      nOutLap = Sum[pOutputLap[y,k,V]*0.005,{y,0.0,1,0.005}];
      nOut = Sum[pOutput[y,k,V]*0.005,{y,0,1,0.005}];
      Plot[{pOutput[y,k,V]/nOut, pOutputLap[y,k,V]/nOutLap},{y,0.0,1},PlotRange→All],{V,1,1000}]

Sum[pOutputLap[y,k,V]*0.01/nOutLap,{y,0,1,0.01}]
1.0000000017787507`
V=5
5
      Normalize via lists
dx=dy=0.1;
pNormalizedList[y_,Inp_,k_,V_]:=
 p[y,Inp,k,V](Total[Function[yp,p[yp,Inp,k,V]*dy]/@Range[0,1,dy]])^-1
   (*Sum[pNormalizedList[y,Inp,k,V]*dy,{y,0,1,dy}]*) (* Output → 1 *)
   pOutput[y_,k_,V_]:=
  Sum[pNormalizedList[y,Inp,k,V]*pInput[Inp,0.5,0.1]*dx,{Inp,10^-8,1,dx}]
    (* Sum[pOutput[y,k,V]*dy,{y,0,1,dy}] Output → 0.9999991426275873 *)
    pJoint[y_,Inp_,k_,V_]:=pNormalizedList[y,Inp,k,V]*pInput[Inp,0.5,0.1]
(* Output probability as list *)
(*Column[Function[y,pOutput[y,0.2,5]]/@Range[0.005,0.995,0.1]]*)
(*Plot[pOutput[y,0.4,10],{y,0,1}]*)
*)
```